



---

Junos<sup>®</sup> OS

# Security Configuration Guide

Release

11.1



---

Published: 2011-05-19

Juniper Networks, Inc.  
1194 North Mathilda Avenue  
Sunnyvale, California 94089  
USA  
408-745-2000  
www.juniper.net

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986–1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., copyright © 1996, 1997, Maker Communications, Inc.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

*Junos OS Security Configuration Guide*  
Release 11.1  
Copyright © 2011, Juniper Networks, Inc.  
All rights reserved.

Revision History  
February 2011—Revision 01

The information in this document is current as of the date listed in the revision history.

#### YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. The Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

#### SOFTWARE LICENSE

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions. Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details. For complete product documentation, please see the Juniper Networks website at [www.juniper.net/techpubs](http://www.juniper.net/techpubs).

## END USER LICENSE AGREEMENT

**READ THIS END USER LICENSE AGREEMENT (“AGREEMENT”) BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE.** BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. **The Parties.** The parties to this Agreement are (i) Juniper Networks, Inc. (if the Customer’s principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer’s principal office is located outside the Americas) (such applicable entity being referred to herein as “Juniper”), and (ii) the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software (“Customer”) (collectively, the “Parties”).

2. **The Software.** In this Agreement, “Software” means the program modules and features of the Juniper or Juniper-supplied software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. “Software” also includes updates, upgrades and new releases of such software. “Embedded Software” means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the equipment.

3. **License Grant.** Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

a. Customer shall use Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.

b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.

c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer’s use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer’s use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

d. For any trial copy of the Software, Customer’s right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.

e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer’s enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any ‘locked’ or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the

Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

5. **Audit.** Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. **Confidentiality.** The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. **Ownership.** Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. **Warranty, Limitation of Liability, Disclaimer of Warranty.** The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. **Termination.** Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. **Taxes.** All license fees payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

11. **Export.** Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. **Commercial Computer Software.** The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14 (ALT III) as applicable.

13. **Interface Information.** To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. **Third Party Software.** Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at <http://www.gnu.org/licenses/gpl.html>, and a copy of the LGPL at <http://www.gnu.org/licenses/lgpl.html>.

15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentes confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattache, soient rédigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).



# Abbreviated Table of Contents

	About This Guide .....	xli
<b>Part 1</b>	<b>Introduction to Junos OS</b>	
Chapter 1	Introducing Junos OS for SRX Series Services Gateways .....	3
Chapter 2	Understanding IPv6 Flow-Based Processing .....	47
Chapter 3	Introducing Junos OS for J Series Services Routers .....	93
<b>Part 2</b>	<b>Security Zones and Interfaces</b>	
Chapter 4	Security Zones and Interfaces .....	111
Chapter 5	Address Books and Address Sets .....	133
<b>Part 3</b>	<b>Security Policies</b>	
Chapter 6	Security Policies .....	145
Chapter 7	Security Policy Schedulers .....	181
Chapter 8	Security Policy Applications .....	187
<b>Part 4</b>	<b>Application Layer Gateways</b>	
Chapter 9	ALGs .....	217
Chapter 10	H.323 ALGs .....	223
Chapter 11	ALG for IKE and ESP .....	259
Chapter 12	SIP ALGs .....	269
Chapter 13	SCCP ALGs .....	325
Chapter 14	MGCP ALGs .....	347
Chapter 15	RPC ALGs .....	379
<b>Part 5</b>	<b>User Authentication</b>	
Chapter 16	Firewall User Authentication .....	389
Chapter 17	Infranet Authentication .....	421
<b>Part 6</b>	<b>Virtual Private Networks</b>	
Chapter 18	Internet Protocol Security .....	451
Chapter 19	Public Key Cryptography for Certificates .....	569
Chapter 20	Dynamic VPNs .....	597
Chapter 21	Group VPNs .....	655

<b>Part 7</b>	<b>Intrusion Detection and Prevention</b>	
Chapter 22	IDP Policies .....	701
Chapter 23	Application-Level Distributed Denial of Service .....	763
Chapter 24	IDP Signature Database .....	777
Chapter 25	IDP Application Identification .....	795
Chapter 26	IDP SSL Inspection .....	809
Chapter 27	IDP Class of Service Action .....	817
Chapter 28	IDP Performance and Capacity Tuning .....	825
Chapter 29	IDP Logging .....	827
<b>Part 8</b>	<b>Unified Threat Management</b>	
Chapter 30	Unified Threat Management Overview .....	843
Chapter 31	Antispam Filtering .....	851
Chapter 32	Full Antivirus Protection .....	869
Chapter 33	Express Antivirus Protection .....	929
Chapter 34	Sophos Antivirus Protection .....	951
Chapter 35	Content Filtering .....	969
Chapter 36	Web Filtering .....	985
<b>Part 9</b>	<b>Attack Detection and Prevention</b>	
Chapter 37	Attack Detection and Prevention .....	1017
Chapter 38	Reconnaissance Deterrence .....	1019
Chapter 39	Suspicious Packet Attributes .....	1051
Chapter 40	Denial-of-Service Attacks .....	1065
<b>Part 10</b>	<b>Application Identification</b>	
Chapter 41	Junos OS Application Identification .....	1103
Chapter 42	AppTrack Application Tracking .....	1129
<b>Part 11</b>	<b>Chassis Cluster</b>	
Chapter 43	Chassis Cluster .....	1137
<b>Part 12</b>	<b>Network Address Translation</b>	
Chapter 44	Network Address Translation .....	1335
<b>Part 13</b>	<b>GPRS</b>	
Chapter 45	General Packet Radio Service .....	1433
<b>Part 14</b>	<b>Index</b>	
	Index .....	1467



# Table of Contents

	<b>About This Guide</b> .....	<b>xli</b>
	J Series and SRX Series Documentation and Release Notes .....	xli
	Objectives .....	xlii
	Audience .....	xlii
	Supported Routing Platforms .....	xlii
	Document Conventions .....	xlii
	Documentation Feedback .....	xliv
	Requesting Technical Support .....	xliv
	Self-Help Online Tools and Resources .....	xliv
	Opening a Case with JTAC .....	xlvi
<b>Part 1</b>	<b>Introduction to Junos OS</b>	
<b>Chapter 1</b>	<b>Introducing Junos OS for SRX Series Services Gateways</b> .....	<b>3</b>
	SRX Series Services Gateways Processing Overview .....	3
	Understanding Flow-Based Processing .....	4
	Zones and Policies .....	5
	Flows and Sessions .....	5
	Understanding Packet-Based Processing .....	5
	Stateless Firewall Filters .....	6
	Class-of-Service Features .....	6
	Screens .....	6
	Sessions for SRX Series Services Gateways .....	7
	Session Characteristics for SRX Series Services Gateways .....	7
	Understanding Session Characteristics for SRX Series Services Gateways .....	7
	Example: Controlling Session Termination for SRX Series Services Gateways .....	8
	Example: Disabling TCP Packet Security Checks for SRX Series Services Gateways .....	9
	Example: Setting the Maximum Segment Size for All TCP Sessions for SRX Series Services Gateways .....	10
	Monitoring Sessions for SRX Series Services Gateways .....	11
	Understanding How to Obtain Session Information for SRX Series Services Gateways .....	12
	Displaying Global Session Parameters for All SRX Series Services Gateways .....	12
	Displaying a Summary of Sessions for SRX Series Services Gateways .....	13
	Displaying Session and Flow Information About Sessions for SRX Series Services Gateways .....	14

Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways . . . . .	14
Using Filters to Display Session and Flow Information for SRX Series Services Gateways . . . . .	15
Information Provided in Session Log Entries for SRX Series Services Gateways . . . . .	16
Clearing Sessions for SRX Series Services Gateways . . . . .	20
Terminating Sessions for SRX Series Services Gateways . . . . .	20
Terminating a Specific Session for SRX Series Services Gateways . . . . .	20
Using Filters to Specify the Sessions to Be Terminated for SRX Series Services Gateways . . . . .	20
Debugging for SRX Series Services Gateways . . . . .	21
Data Path Debugging for SRX Series Services Gateways . . . . .	21
Understanding Data Path Debugging for SRX Series Services Gateways . . . . .	21
Debugging the Data Path (CLI Procedure) . . . . .	22
Security Debugging for SRX Series Services Gateways . . . . .	22
Understanding Security Debugging Using Trace Options . . . . .	22
Setting Security Trace Options (CLI Procedure) . . . . .	23
Displaying Output for Security Trace Options . . . . .	24
Flow Debugging for SRX Series Services Gateways . . . . .	24
Understanding Flow Debugging Using Trace Options . . . . .	24
Setting Flow Debugging Trace Options (CLI Procedure) . . . . .	24
Understanding SRX Series Services Gateways Central Point Architecture . . . . .	25
Load Distribution in Combo Mode . . . . .	25
Sharing Processing Power and Memory in Combo Mode . . . . .	26
Expanding Session Capacity by Device . . . . .	26
Expanding Session Capacity on an SRX3400 or SRX3600 Device . . . . .	27
Expanding Session Capacity on an SRX5800 Device . . . . .	27
Reverting to Default Session Capacity on an SRX5800 Device . . . . .	28
Verifying the Current Session Capacity . . . . .	28
SRX5600 and SRX5800 Services Gateways Processing Overview . . . . .	28
Understanding First-Packet Processing . . . . .	30
Understanding Fast-Path Processing . . . . .	31
Understanding the Data Path for Unicast Sessions . . . . .	32
Session Lookup and Packet Match Criteria . . . . .	32
Understanding Session Creation: First-Packet Processing . . . . .	32
Understanding Fast-Path Processing . . . . .	35
Understanding Packet Processing . . . . .	38
Understanding Services Processing Units . . . . .	39
Understanding Scheduler Characteristics . . . . .	39
Understanding Network Processor Bundling . . . . .	39
Network Processor Bundling Limitations . . . . .	40
SRX1400, SRX3400, and SRX3600 Services Gateways Processing Overview . . . . .	41
Components Involved in Setting up a Session . . . . .	41
Understanding the Data Path for Unicast Sessions . . . . .	42
Session Lookup and Packet Match Criteria . . . . .	42
Understanding Session Creation: First Packet Processing . . . . .	42
Understanding Fast-Path Processing . . . . .	44

	SRX210 Services Gateway Processing Overview . . . . .	44
	Understanding Flow Processing and Session Management . . . . .	45
	Understanding First-Packet Processing . . . . .	45
	Understanding Session Creation . . . . .	45
	Understanding Fast-Path Processing . . . . .	46
<b>Chapter 2</b>	<b>Understanding IPv6 Flow-Based Processing . . . . .</b>	<b>47</b>
	Understanding IP Version 6 (IPv6) . . . . .	48
	About the IPv6 Address Space, Addressing, and Address Types . . . . .	48
	About IPv6 Address Types and How Junos OS for SRX Series Services Gateway and J-series Devices Use Them . . . . .	49
	About the IPv6 Address Format . . . . .	50
	The IPv6 Packet Header and SRX Series and J-series Devices Overview . . . . .	51
	About the IPv6 Basic Packet Header . . . . .	52
	Understanding IPv6 Packet Header Extensions . . . . .	54
	About IPv6 Packet Header Verification Performed by the Flow Module for SRX Series and J-series Devices . . . . .	55
	IPv6 Advanced Flow . . . . .	55
	Understanding IPv6 Dual-Stack Lite . . . . .	57
	Understanding How SRX Series and J-series Devices Handle ICMPv6 Packets . . . . .	59
	Understanding Path MTU Messages for IPv6 Packets . . . . .	61
	Understanding How SRX Series and J-series Devices Handle Packet Fragmentation for IPv6 Flows . . . . .	62
	Understanding Sessions for IPv6 Flows . . . . .	63
	Understanding SRX5600 and SRX5800 Architecture and Flow Processing . . . . .	63
	Enabling Flow-Based Processing for IPv6 Traffic . . . . .	66
	Using Filters to Display IPv6 Session and Flow Information for SRX Series Services Gateways . . . . .	67
	IPv6 NAT . . . . .	71
	IPv6 NAT Overview . . . . .	71
	Source NAT Translations Supported by IPv6 NAT . . . . .	72
	Destination NAT Mappings Supported by IPv6 NAT . . . . .	72
	Static NAT Mappings Supported by IPv6 NAT . . . . .	72
	IPv6 NAT PT Overview . . . . .	73
	IPv6 NAT-PT Communication Overview . . . . .	74
	Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Default Destination Address Prefix Static Mapping . . . . .	75
	Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Static Destination Address One-to-One Mapping . . . . .	78
	Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Default Destination Address Prefix Static Mapping . . . . .	82

	Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Static Destination Address One-to-One Mapping . . . . .	85
	IPv6 ALGs . . . . .	88
	IPv6 DNS ALG for Routing, NAT, and NAT-PT . . . . .	88
	IPv6 DNS ALG Traffic in NAT mode . . . . .	88
	IPv6 DNS ALG Traffic in NAT-PT mode . . . . .	89
	IPv6 FTP ALG for Routing . . . . .	90
	EPRT mode . . . . .	90
	EPSV mode . . . . .	90
	Understanding IPV6 ALG support for ICMP . . . . .	91
	ICMP Error Messages . . . . .	91
	ICMP ALG Functionality . . . . .	91
<b>Chapter 3</b>	<b>Introducing Junos OS for J Series Services Routers . . . . .</b>	<b>93</b>
	Understanding Stateful and Stateless Data Processing for J Series Services Routers . . . . .	93
	Zones and Policies . . . . .	95
	Flows and Sessions . . . . .	95
	Understanding Packet-Based Processing . . . . .	96
	Stateless Firewall Filters . . . . .	96
	Class-of-Service Features . . . . .	97
	Session Characteristics for J Series Services Routers . . . . .	97
	Understanding Session Characteristics for J Series Services Routers . . . . .	97
	Example: Controlling Session Termination for J Series Services Routers . . . . .	98
	Example: Disabling TCP Packet Security Checks for J Series Services Routers . . . . .	101
	Example: Accommodating End-to-End TCP Communication for J Series Services Routers . . . . .	102
	Understanding the Data Path for J Series Services Routers . . . . .	104
	Understanding the Forwarding Processing . . . . .	105
	Understanding the Session-Based Processing . . . . .	105
	Session Lookup . . . . .	105
	First-Packet Path Processing . . . . .	105
	Fast-Path Processing . . . . .	106
	Understanding Forwarding Features . . . . .	107

<b>Part 2</b>	<b>Security Zones and Interfaces</b>	
<b>Chapter 4</b>	<b>Security Zones and Interfaces</b>	<b>111</b>
	Security Zones and Interfaces Overview	111
	Understanding Security Zone Interfaces	112
	Understanding Interface Ports	112
	Security Zones	112
	Understanding Functional Zones	113
	Understanding Security Zones	113
	Example: Creating Security Zones	114
	Host Inbound Traffic	116
	Understanding How to Control Inbound Traffic Based on Traffic Types	116
	Supported System Services for Host Inbound Traffic	117
	Example: Controlling Inbound Traffic Based on Traffic Types	118
	Protocols	121
	Stream Control Transmission Protocol Overview	121
	Configuration Overview	122
	Understanding How to Control Inbound Traffic Based on Protocols	122
	Example: Controlling Inbound Traffic Based on Protocols	123
	TCP-Reset Parameters	125
	Understanding How to Identify Duplicate Sessions Using the TCP-Reset Parameter	125
	Example: Configuring the TCP-Reset Parameter	126
	DNS	127
	DNS Overview	127
	DNS Components	127
	DNS Server Caching	127
	Forwarders	127
	Example: Configuring the TTL Value for DNS Server Caching	128
	Example: Configuring a Forwarder for a DNS server	129
	DNSSEC Overview	129
	Example: Configuring DNSSEC	129
	Example: Configuring Keys for DNSSEC	130
	Example: Configuring Secure Domains and Trusted Keys for DNSSEC	130
<b>Chapter 5</b>	<b>Address Books and Address Sets</b>	<b>133</b>
	Security Policy Address Books and Address Sets Overview	133
	Understanding Address Books	134
	Understanding Address Sets	135
	Limitations of Addresses and Address Sets	138
	Example: Configuring Address Books	139
	Verifying Address Book Configuration	141
<b>Part 3</b>	<b>Security Policies</b>	
<b>Chapter 6</b>	<b>Security Policies</b>	<b>145</b>
	Security Policies Overview	145
	Understanding Security Policy Rules	148
	Understanding Wildcard Addresses	150
	Understanding Security Policy Elements	151

	Security Policies Configuration Overview . . . . .	151
	Configuring Policies Using the Firewall Wizard . . . . .	152
	Example: Configuring a Security Policy to Permit or Deny All Traffic . . . . .	152
	Example: Configuring a Security Policy to Permit or Deny Selected Traffic . . . . .	156
	Example: Configuring a Security Policy to Permit or Deny Wildcard Address Traffic . . . . .	161
	Application Firewall Overview . . . . .	164
	Understanding Application Firewall Rule Sets . . . . .	165
	Configuring an Application Firewall Within a Security Policy . . . . .	166
	Application Firewall Support in Chassis Cluster . . . . .	166
	Example: Configuring Application Firewall Rule Sets Within Security Policy (CLI) . . . . .	167
	Understanding Security Policy Ordering . . . . .	173
	Example: Reordering the Policies . . . . .	175
	Troubleshooting Security Policies . . . . .	176
	Checking a Security Policy Commit Failure . . . . .	176
	Verifying a Security Policy Commit . . . . .	176
	Debugging Policy Lookup . . . . .	177
	Monitoring Policy Statistics . . . . .	177
	Matching Security Policies . . . . .	178
<b>Chapter 7</b>	<b>Security Policy Schedulers . . . . .</b>	<b>181</b>
	Security Policy Schedulers Overview . . . . .	181
	Example: Configuring Schedulers . . . . .	182
	Verifying Scheduled Policies . . . . .	184
<b>Chapter 8</b>	<b>Security Policy Applications . . . . .</b>	<b>187</b>
	Security Policy Applications Overview . . . . .	187
	Policy Application Sets Overview . . . . .	188
	Example: Configuring Applications and Application Sets . . . . .	189
	Custom Policy Applications . . . . .	190
	Understanding Custom Policy Applications . . . . .	190
	Custom Application Mappings . . . . .	190
	Example: Adding and Modifying Custom Policy Applications . . . . .	191
	Example: Defining a Custom ICMP Application . . . . .	192
	Policy Application Timeouts . . . . .	194
	Understanding Policy Application Timeout Configuration and Lookup . . . . .	194
	Understanding Policy Application Timeouts Contingencies . . . . .	195
	Example: Setting a Policy Application Timeout . . . . .	197
	Understanding the ICMP Predefined Policy Application . . . . .	198
	Default Behaviour of ICMP Unreachable Errors . . . . .	202
	Understanding Internet-Related Predefined Policy Applications . . . . .	202
	Understanding Microsoft Predefined Policy Applications . . . . .	204
	Understanding Dynamic Routing Protocols Predefined Policy Applications . . . . .	205
	Understanding Streaming Video Predefined Policy Applications . . . . .	206
	Understanding Sun RPC Predefined Policy Applications . . . . .	206
	Understanding Security and Tunnel Predefined Policy Applications . . . . .	207
	Understanding IP-Related Predefined Policy Applications . . . . .	208
	Understanding Instant Messaging Predefined Policy Applications . . . . .	209
	Understanding Management Predefined Policy Applications . . . . .	209

	Understanding Mail Predefined Policy Applications . . . . .	211
	Understanding UNIX Predefined Policy Applications . . . . .	211
	Understanding Miscellaneous Predefined Policy Applications . . . . .	212
<b>Part 4</b>	<b>Application Layer Gateways</b>	
<b>Chapter 9</b>	<b>ALGs . . . . .</b>	<b>217</b>
	ALG Overview . . . . .	217
	Understanding ALG Types . . . . .	218
	Understanding VoIP DSCP Rewrite Rules . . . . .	220
	Example: Configuring VoIP DSCP Rewrite Rules . . . . .	220
<b>Chapter 10</b>	<b>H.323 ALGs . . . . .</b>	<b>223</b>
	Understanding H.323 ALGs . . . . .	223
	Understanding the Avaya H.323 ALG . . . . .	225
	Avaya H.323 ALG-Specific Features . . . . .	225
	Call Flow Details in the Avaya H.323 ALG . . . . .	225
	H.323 ALG Configuration Overview . . . . .	227
	H.323 ALG Endpoint Registration Timeouts . . . . .	227
	Understanding H.323 ALG Endpoint Registration Timeouts . . . . .	227
	Example: Setting H.323 ALG Endpoint Registration Timeouts . . . . .	228
	H.323 ALG Media Source Port Ranges . . . . .	229
	Understanding H.323 ALG Media Source Port Ranges . . . . .	229
	Example: Setting H.323 ALG Media Source Port Ranges . . . . .	229
	H.323 ALG DoS Attack Protection . . . . .	230
	Understanding H.323 ALG DoS Attack Protection . . . . .	230
	Example: Configuring H.323 ALG DoS Attack Protection . . . . .	231
	H.323 ALG Unknown Message Types . . . . .	232
	Understanding H.323 ALG Unknown Message Types . . . . .	232
	Example: Allowing Unknown H.323 ALG Message Types . . . . .	233
	Example: Passing H.323 ALG Traffic to a Gatekeeper in the Private Zone . . . . .	234
	Example: Passing H.323 ALG Traffic to a Gatekeeper in the External Zone . . . . .	239
	Example: Using NAT with the H.323 ALG to Enable Incoming Calls . . . . .	246
	Example: Using NAT with the H.323 ALG to Enable Outgoing Calls . . . . .	252
<b>Chapter 11</b>	<b>ALG for IKE and ESP . . . . .</b>	<b>259</b>
	Understanding ALG for IKE and ESP . . . . .	259
	Understanding ALG for IKE and ESP Operation . . . . .	260
	Example: Configuring the IKE and ESP ALG . . . . .	261
	Example: Enabling IKE and ESP ALG and Setting Timeouts . . . . .	266
<b>Chapter 12</b>	<b>SIP ALGs . . . . .</b>	<b>269</b>
	Understanding SIP ALGs . . . . .	269
	SIP ALG Operation . . . . .	270
	SDP Session Descriptions . . . . .	271

Pinhole Creation . . . . .	272
Understanding SIP ALG Request Methods . . . . .	274
SIP ALG Configuration Overview . . . . .	275
SIP ALG Call Duration and Timeouts . . . . .	275
Understanding SIP ALG Call Duration and Timeouts . . . . .	275
Example: Setting SIP ALG Call Duration and Timeouts . . . . .	276
SIP ALG DoS Attack Protection . . . . .	278
Understanding SIP ALG DoS Attack Protection . . . . .	278
Example: Configuring SIP ALG DoS Attack Protection . . . . .	278
SIP ALG Unknown Message Types . . . . .	279
Understanding SIP ALG Unknown Message Types . . . . .	280
Example: Allowing Unknown SIP ALG Message Types . . . . .	280
SIP ALG Hold Resources . . . . .	281
Understanding SIP ALG Hold Resources . . . . .	281
Retaining SIP ALG Hold Resources (J-Web Procedure) . . . . .	282
Retaining SIP ALG Hold Resources (CLI Procedure) . . . . .	282
SIP ALGs and NAT . . . . .	282
Understanding SIP ALGs and NAT . . . . .	283
Outgoing Calls . . . . .	284
Incoming Calls . . . . .	284
Forwarded Calls . . . . .	285
Call Termination . . . . .	285
Call Re-INVITE Messages . . . . .	285
Call Session Timers . . . . .	285
Call Cancellation . . . . .	285
Forking . . . . .	286
SIP Messages . . . . .	286
SIP Headers . . . . .	286
SIP Body . . . . .	288
SIP NAT Scenario . . . . .	288
Classes of SIP Responses . . . . .	290
Understanding Incoming SIP ALG Call Support Using the SIP Registrar and NAT . . . . .	292
Example: Configuring Interface Source NAT for Incoming SIP Calls . . . . .	293
Example: Configuring a Source NAT Pool for Incoming SIP Calls . . . . .	298
Example: Configuring Static NAT for Incoming SIP Calls . . . . .	304
Example: Configuring the SIP Proxy in the Private Zone and NAT in the Public Zone . . . . .	309
Example: Configuring a Three-Zone SIP ALG and NAT Scenario . . . . .	314
Verifying SIP ALG Configurations . . . . .	321
Verifying SIP ALGs . . . . .	321
Verifying SIP ALG Calls . . . . .	321
Verifying SIP ALG Call Details . . . . .	322
Verifying SIP ALG Counters . . . . .	322
Verifying the Rate of SIP ALG Messages . . . . .	323



<b>Chapter 13</b>	<b>SCCP ALGs</b> . . . . .	<b>325</b>
	Understanding SCCP ALGs . . . . .	325
	SCCP Security . . . . .	326
	SCCP Components . . . . .	327
	SCCP Client . . . . .	327
	CallManager . . . . .	327
	Cluster . . . . .	327
	SCCP Transactions . . . . .	327
	Client Initialization . . . . .	328
	Client Registration . . . . .	328
	Call Setup . . . . .	328
	Media Setup . . . . .	328
	SCCP Control Messages and RTP Flow . . . . .	328
	SCCP Messages . . . . .	329
	SCCP ALG Configuration Overview . . . . .	330
	SCCP ALG Inactive Media Timeout . . . . .	331
	Understanding SCCP ALG Inactive Media Timeouts . . . . .	331
	Example: Setting SCCP ALG Inactive Media Timeouts . . . . .	331
	SCCP ALG Unknown Message Types . . . . .	332
	Understanding SCCP ALG Unknown Message Types . . . . .	332
	Example: Allowing Unknown SCCP ALG Message Types . . . . .	333
	SCCP ALG DoS Attack Protection . . . . .	334
	Understanding SCCP ALG DoS Attack Protection . . . . .	334
	Example: Configuring SCCP ALG DoS Attack Protection . . . . .	335
	Example: Configuring the SCCP ALG Call Manager or TFTP Server in the Private Zone . . . . .	336
	Verifying SCCP ALG Configurations . . . . .	342
	Verifying SCCP ALGs . . . . .	342
	Verifying SCCP Calls . . . . .	343
	Verifying SCCP Call Details . . . . .	343
	Verifying SCCP Counters . . . . .	344
<b>Chapter 14</b>	<b>MGCP ALGs</b> . . . . .	<b>347</b>
	Understanding MGCP ALGs . . . . .	347
	MGCP Security . . . . .	348
	Entities in MGCP . . . . .	348
	Endpoint . . . . .	348
	Connection . . . . .	349
	Call . . . . .	349
	Call Agent . . . . .	349
	Commands . . . . .	350
	Response Codes . . . . .	352
	MGCP ALG Configuration Overview . . . . .	353
	MGCP ALG Call Duration and Timeouts . . . . .	353
	Understanding MGCP ALG Call Duration and Timeouts . . . . .	353
	Example: Setting MGCP ALG Call Duration . . . . .	354
	Example: Setting MGCP ALG Inactive Media Timeout . . . . .	355

	Example: Setting MGCP ALG Transaction Timeout . . . . .	357
	MGCP ALG DoS Attack Protection . . . . .	358
	Understanding MGCP ALG DoS Attack Protection . . . . .	358
	Example: Configuring MGCP ALG DoS Attack Protection . . . . .	358
	MGCP ALG Unknown Message Types . . . . .	359
	Understanding MGCP ALG Unknown Message Types . . . . .	359
	Example: Allowing Unknown MGCP ALG Message Types . . . . .	360
	Example: Configuring Media Gateways in Subscriber Homes Using MGCP ALGs . . . . .	361
	Example: Configuring Three-Zone ISP-Hosted Service Using MGCP ALG and NAT . . . . .	368
<b>Chapter 15</b>	<b>RPC ALGs . . . . .</b>	<b>379</b>
	Understanding RPC ALGs . . . . .	379
	Sun RPC ALGs . . . . .	380
	Understanding Sun RPC ALGs . . . . .	380
	Enabling Sun RPC ALGs (J-Web Procedure) . . . . .	381
	Enabling Sun RPC ALGs (CLI Procedure) . . . . .	381
	Sun RPC Services and Applications . . . . .	381
	Understanding Sun RPC Services . . . . .	382
	Customizing Sun RPC Applications (CLI Procedure) . . . . .	382
	Microsoft RPC ALGs . . . . .	383
	Understanding Microsoft RPC ALGs . . . . .	383
	Enabling Microsoft RPC ALGs (J-Web Procedure) . . . . .	384
	Enabling Microsoft RPC ALGs (CLI Procedure) . . . . .	384
	Microsoft RPC Services and Applications . . . . .	385
	Understanding Microsoft RPC Services . . . . .	385
	Customizing Microsoft RPC Applications (CLI Procedure) . . . . .	385
	Verifying the Microsoft RPC ALG Tables . . . . .	386
<b>Part 5</b>	<b>User Authentication</b>	
<b>Chapter 16</b>	<b>Firewall User Authentication . . . . .</b>	<b>389</b>
	Firewall User Authentication Overview . . . . .	389
	Pass-Through Authentication . . . . .	390
	Understanding Pass-Through Authentication . . . . .	390
	Example: Configuring Pass-Through Authentication . . . . .	391
	Web Authentication . . . . .	397
	Understanding Web Authentication . . . . .	397
	Example: Configuring Web Authentication . . . . .	399
	External Authentication . . . . .	405
	Understanding External Authentication Servers . . . . .	406
	Understanding SecurID User Authentication . . . . .	406
	Example: Configuring RADIUS and LDAP User Authentication . . . . .	407
	Example: Configuring SecurID User Authentication . . . . .	411
	Example: Deleting the SecurID Node Secret File . . . . .	414
	Client Groups for Firewall Authentication . . . . .	415
	Understanding Client Groups for Firewall Authentication . . . . .	415
	Example: Configuring Local Users for Client Groups . . . . .	416

	Firewall Authentication Banner Customization . . . . .	418
	Understanding Firewall Authentication Banner Customization . . . . .	418
	Example: Customizing a Firewall Authentication Banner . . . . .	418
<b>Chapter 17</b>	<b>Infranet Authentication . . . . .</b>	<b>421</b>
	UAC and Junos OS . . . . .	421
	Understanding UAC in a Junos OS Environment . . . . .	421
	Enabling UAC in a Junos OS Environment (CLI Procedure) . . . . .	423
	Junos OS Enforcer and Infranet Controller Communications . . . . .	424
	Understanding Communications Between the Junos OS Enforcer and the Infranet Controller . . . . .	424
	Configuring Communications Between the Junos OS Enforcer and the Infranet Controller (CLI Procedure) . . . . .	424
	Junos OS Enforcer Policy Enforcement . . . . .	426
	Understanding Junos OS Enforcer Policy Enforcement . . . . .	427
	Testing Junos OS Enforcer Policy Access Decisions Using Test-Only Mode (CLI Procedure) . . . . .	428
	Verifying Junos OS Enforcer Policy Enforcement . . . . .	429
	Displaying Infranet Controller Authentication Table Entries from the Junos OS Enforcer . . . . .	429
	Displaying Infranet Controller Resource Access Policies from the Junos OS Enforcer . . . . .	429
	Junos OS Enforcer and IPsec . . . . .	429
	Understanding Junos OS Enforcer Implementations Using IPsec . . . . .	429
	Example: Configuring the Device as a Junos OS Enforcer Using IPsec (CLI) . . . . .	431
	Junos OS Enforcer and Infranet Agent Endpoint Security . . . . .	437
	Understanding Endpoint Security Using the Infranet Agent with the Junos OS Enforcer . . . . .	437
	Configuring Endpoint Security Using the Infranet Agent with the Junos OS Enforcer . . . . .	438
	Junos OS Enforcer and Captive Portal . . . . .	438
	Understanding the Captive Portal on the Junos OS Enforcer . . . . .	439
	Understanding Captive Portal Configuration on the Junos OS Enforcer . . . . .	440
	Example: Creating a Captive Portal Policy on the Junos OS Enforcer . . . . .	441
	Understanding the Captive Portal Redirect URL Options . . . . .	444
	Example: Configuring a Redirect URL for Captive Portal . . . . .	445
	Junos OS Enforcer and Infranet Controller Cluster Failover . . . . .	447
	Understanding Communications Between Junos OS Enforcer and a Cluster of Infranet Controllers . . . . .	447
	Configuring Junos OS Enforcer Failover Options (CLI Procedure) . . . . .	447
<b>Part 6</b>	<b>Virtual Private Networks</b>	
<b>Chapter 18</b>	<b>Internet Protocol Security . . . . .</b>	<b>451</b>
	VPN Overview . . . . .	451
	IPsec VPN Topologies . . . . .	452
	Comparison of Policy-Based VPNs and Route-Based VPNs . . . . .	452
	Security Associations . . . . .	453

IPsec Key Management . . . . .	454
Manual Key . . . . .	454
AutoKey IKE . . . . .	455
Diffie-Hellman Exchange . . . . .	455
IPsec Security Protocols . . . . .	456
AH Protocol . . . . .	456
ESP Protocol . . . . .	457
IPsec Tunnel Negotiation . . . . .	457
Distributed VPNs in SRX Series Services Gateways . . . . .	458
Understanding IKE and IPsec Packet Processing . . . . .	458
Packet Processing in Tunnel Mode . . . . .	459
IKE Packet Processing . . . . .	461
IPsec Packet Processing . . . . .	464
Understanding Phase 1 of IKE Tunnel Negotiation . . . . .	467
Main Mode . . . . .	467
Aggressive Mode . . . . .	468
Understanding Phase 2 of IKE Tunnel Negotiation . . . . .	468
Proxy IDs . . . . .	469
Perfect Forward Secrecy . . . . .	469
Replay Protection . . . . .	469
Route-Based VPNs . . . . .	470
Understanding Route-Based IPsec VPNs . . . . .	470
Example: Configuring a Route-Based VPN . . . . .	470
Policy-Based VPNs . . . . .	488
Understanding Policy-Based IPsec VPNs . . . . .	488
Example: Configuring a Policy-Based VPN . . . . .	489
Hub-and-Spoke VPNs . . . . .	506
Understanding Hub-and-Spoke VPNs . . . . .	506
Example: Configuring a Hub-and-Spoke VPN . . . . .	507
Configuring IPsec VPN Using the VPN Wizard . . . . .	538
Understanding IPv6 IKE and IPsec Packet Processing . . . . .	538
Packet Processing in IPv6 6in6 Tunnel Mode . . . . .	538
IPv6 IKE Packet Processing . . . . .	539
IPv6 IPsec Packet Processing . . . . .	540
AH Protocol in IPv6 . . . . .	541
ESP Protocol in IPv6 . . . . .	541
Integrity Check Value (ICV) Calculation in IPv6 . . . . .	541
Header Construction in IPv6 Tunnel Mode . . . . .	542
IPv6 IPsec Configuration Overview . . . . .	543
Example: Configuring an IPv6 IPsec Manual VPN . . . . .	543
Example: Configuring an IPv6 AutoKey IKE Policy-Based VPN . . . . .	546
Global SPI and VPN Monitoring Features . . . . .	562
Understanding Global SPI and VPN Monitoring Features . . . . .	562
Example: Configuring Global SPI and VPN Monitoring Features . . . . .	563
Virtual Router Support for Route-Based VPNs . . . . .	563
Example: Configuring an st0 Interface in a Virtual Router . . . . .	564
Understanding Virtual Router Limitations . . . . .	568

<b>Chapter 19</b>	<b>Public Key Cryptography for Certificates</b>	<b>569</b>
	Understanding Certificates and PKI	569
	Certificate Signatures and Verification	570
	Public Key Infrastructure	570
	PKI Management and Implementation	572
	Internet Key Exchange	573
	Digital Certificates Configuration Overview	574
	Enabling Digital Certificates Online: Configuration Overview	574
	Manually Generating Digital Certificates: Configuration Overview	574
	Public-Private Key Pairs	575
	Understanding Public Key Cryptography	575
	Example: Generating a Public-Private Key Pair	576
	CA Profiles	577
	Understanding Certificate Authority Profiles	577
	Example: Configuring a CA Profile	577
	Certificate Enrollment	578
	Understanding Online CA Certificate Enrollment	579
	Enrolling a CA Certificate Online Using SCEP	579
	Example: Enrolling a Local Certificate Online Using SCEP	580
	Example: Using SCEP to Automatically Renew a Local Certificate	581
	Certificate Requests	583
	Understanding Local Certificate Requests	583
	Example: Manually Generating a CSR for the Local Certificate and Sending it to the CA Server	584
	Certificate Loading	585
	Understanding Certificate Loading	585
	Example: Loading CA and Local Certificates Manually	585
	CRLs	587
	Understanding Certificate Revocation Lists	587
	Example: Manually Loading a CRL onto the Device	587
	Example: Verifying Certificate Validity	589
	Example: Configuring a Certificate Authority Profile with CRL Locations	590
	Deleting a Loaded CRL (CLI Procedure)	591
	Self-Signed Certificates	592
	Understanding Self-Signed Certificates	592
	Generating Self-Signed Certificates	592
	Automatically Generating Self-Signed Certificates	593
	Manually Generating Self-Signed Certificates	593
	Using Automatically Generated Self-Signed Certificates (CLI Procedure)	593
	Example: Manually Generating Self-Signed Certificates	594
	Deleting Certificates (CLI Procedure)	595

<b>Chapter 20</b>	<b>Dynamic VPNs</b>	<b>597</b>
	Dynamic VPN Overview	597
	Understanding Remote Client Access to the VPN	599
	Dynamic VPN Configuration	599
	Understanding Dynamic VPN Tunnels	600
	Dynamic VPN Configuration Overview	601
	Example: Configuring Dynamic VPN	603
	Local Authentication and Address Assignment	612
	Understanding Local Authentication and Address Assignment	612
	Example: Configuring Local Authentication and Address Pool	613
	Dynamic VPN Proposal Sets	616
	Group and Shared IKE IDs	617
	Understanding Group and Shared IKE IDs	618
	Group IKE IDs	618
	Shared IKE IDs	619
	Example: Configuring a Group IKE ID for Multiple Users	619
	Example: Configuring Individual IKE IDs for Multiple Users	626
	Junos Pulse Client for Dynamic VPN Access	637
	Understanding Junos Pulse Client	637
	Junos Pulse Client Installation Requirements	637
	Deploying Junos Pulse Client Software	638
	Junos Pulse Interface and Connections	639
	Junos Pulse Interface	639
	Junos Pulse Connection Type	640
	Junos Pulse Connection Status	640
	Junos Pulse Log Files	640
	Managing Junos Pulse Connections	641
	Add a connection	641
	Connect to a Network	642
	Disconnect from an Active Network	643
	View Connection Properties	643
	Edit Connection Properties	643
	Forget Saved Settings	644
	Delete a Connection	644
	Troubleshoot a Junos Pulse Connection Issue	644
	Annotate Log Files	645
	Set Log Level	646
	Save Log Files	646
	View Component Version Information	646
	Access Manager Client-Side Reference	646
	Access Manager Client-Side System Requirements	647
	Access Manager Client-Side Files	647
	Access Manager Client-Side Registry Changes	650
	Access Manager Client-Side Error Messages	650
	Troubleshooting Access Manager Client-Side Problems	654

<b>Chapter 21</b>	<b>Group VPNs</b> . . . . .	<b>655</b>
	Group VPN Overview . . . . .	655
	Group VPNs . . . . .	657
	Understanding the GDOI Protocol . . . . .	657
	Understanding Group Servers and Members . . . . .	658
	Understanding IKE Phase 1 Configuration for Group VPN . . . . .	659
	Understanding IPsec SA Configuration for Group VPN . . . . .	659
	Understanding Dynamic Policies . . . . .	660
	Understanding Antireplay . . . . .	662
	Understanding VPN Group Configuration . . . . .	662
	Group VPN Configuration Overview . . . . .	663
	Example: Configuring Group VPNs . . . . .	663
	Colocation Mode . . . . .	679
	Understanding Colocation Mode . . . . .	679
	Example: Configuring Group VPN with Server-Member Colocation . . . . .	680
	Server-Group Communications . . . . .	689
	Understanding Server-Member Communication . . . . .	689
	Understanding Group Key Operations . . . . .	690
	Group Keys . . . . .	690
	Rekey Messages . . . . .	691
	Member Registration . . . . .	692
	Key Activation . . . . .	692
	Understanding Heartbeat Messages . . . . .	693
	Example: Configuring Server-Member Communication for Unicast Rekey Messages . . . . .	694
	Example: Configuring Server-Member Communication for Multicast Rekey Messages . . . . .	695
	Understanding Group VPN Limitations . . . . .	697
	Understanding Interoperability with Cisco GET VPN . . . . .	698
<b>Part 7</b>	<b>Intrusion Detection and Prevention</b>	
<b>Chapter 22</b>	<b>IDP Policies</b> . . . . .	<b>701</b>
	IDP Policies Overview . . . . .	701
	Example: Enabling IDP in a Security Policy . . . . .	702
	IDP Inline Tap Mode . . . . .	705
	Understanding IDP Inline Tap Mode . . . . .	705
	Example: Configuring IDP Inline Tap Mode . . . . .	706
	IDP Rules and Rulebases . . . . .	707
	Understanding IDP Policy Rules . . . . .	707
	Understanding IDP Rule Match Conditions . . . . .	707
	Understanding IDP Rule Objects . . . . .	708
	Understanding IDP Rule Actions . . . . .	710
	Understanding IDP Rule IP Actions . . . . .	711

Understanding IDP Rule Notifications . . . . .	712
IDP Rulebases . . . . .	713
Understanding IDP Policy Rulebases . . . . .	713
Example: Inserting a Rule in the IDP Rulebase . . . . .	714
Example: Deactivating and Activating Rules in an IDP Rulebase . . . . .	714
Understanding IDP Application-Level DDoS Rulebases . . . . .	715
IDP IPS Rulebase . . . . .	716
Understanding IDP IPS Rulebases . . . . .	716
Example: Defining Rules for an IDP IPS Rulebase . . . . .	717
IDP Exempt Rulebase . . . . .	721
Understanding IDP Exempt Rulebases . . . . .	721
Example: Defining Rules for an IDP Exempt Rulebase . . . . .	722
IDP Terminal Rules . . . . .	724
Understanding IDP Terminal Rules . . . . .	724
Example: Setting Terminal Rules in Rulebases . . . . .	725
IDP DSCP Rules . . . . .	727
Understanding DSCP Rules in IDP Policies . . . . .	727
Example: Configuring DSCP Rules in an IDP Policy . . . . .	727
IDP Applications and Application Sets . . . . .	730
Understanding IDP Application Sets . . . . .	730
Example: Configuring IDP Applications and Services . . . . .	731
Example: Configuring IDP Applications Sets . . . . .	733
IDP Attacks and Attack Objects . . . . .	735
Understanding Custom Attack Objects . . . . .	736
Attack Name . . . . .	736
Severity . . . . .	736
Service and Application Bindings . . . . .	736
Protocol and Port Bindings . . . . .	740
Time Bindings . . . . .	742
Attack Properties (Signature Attacks) . . . . .	743
Attack Properties (Protocol Anomaly Attacks) . . . . .	748
Attack Properties (Compound or Chain Attacks) . . . . .	749
IDP Protocol Decoders . . . . .	752
Understanding IDP Protocol Decoders . . . . .	752
Example: Configuring IDP Protocol Decoders . . . . .	753
Understanding Multiple IDP Detector Support . . . . .	754
IDP Signature-Based Attacks . . . . .	754
Understanding IDP Signature-Based Attacks . . . . .	754
Example: Configuring IDP Signature-Based Attacks . . . . .	755
IDP Protocol Anomaly-Based Attacks . . . . .	758
Understanding IDP Protocol Anomaly-Based Attacks . . . . .	758
Example: Configuring IDP Protocol Anomaly-Based Attacks . . . . .	759
Listing IDP Test Conditions for a Specific Protocol . . . . .	761
<b>Chapter 23 Application-Level Distributed Denial of Service . . . . .</b>	<b>763</b>
IDP Application-Level DDoS Attack Overview . . . . .	763
IDP Application-Level DDoS Protection Overview . . . . .	763
Understanding the Application-Level DDoS Module . . . . .	764
Understanding the Application-Level DDoS Definition . . . . .	765



	Understanding the Application-Level DDoS Rule . . . . .	766
	Understanding Application-Level DDoS IP-Action . . . . .	767
	Understanding Application-Level DDoS Session Action . . . . .	768
	Example: Enabling IDP Protection Against Application-Level DDoS Attacks . . .	768
	Understanding Application-level DDoS Statistics Reporting . . . . .	772
	Example: Configuring Application-Level DDoS Statistics Reporting . . . . .	775
<b>Chapter 24</b>	<b>IDP Signature Database . . . . .</b>	<b>777</b>
	Understanding the IDP Signature Database . . . . .	777
	Predefined IDP Policy Templates . . . . .	778
	Understanding Predefined IDP Policy Templates . . . . .	778
	Downloading and Using Predefined IDP Policy Templates (CLI Procedure) . . . . .	780
	IDP Signature Databases . . . . .	781
	Understanding Predefined IDP Attack Objects and Object Groups . . . . .	781
	Predefined Attack Objects . . . . .	781
	Predefined Attack Object Groups . . . . .	782
	Understanding the IDP Signature Database Version . . . . .	783
	Updating the IDP Signature Database Overview . . . . .	783
	Updating the IDP Signature Database Manually Overview . . . . .	784
	Example: Updating the IDP Signature Database Manually . . . . .	785
	Example: Updating the Signature Database Automatically . . . . .	788
	Example: Adding a Detector Sensor Configuration (J-Web) . . . . .	789
	Verifying the Signature Database . . . . .	790
	Verifying the IDP Policy Compilation and Load Status . . . . .	790
	Verifying the IDP Signature Database Version . . . . .	792
<b>Chapter 25</b>	<b>IDP Application Identification . . . . .</b>	<b>795</b>
	IDP Application Identification . . . . .	795
	Understanding IDP Application Identification . . . . .	795
	Understanding IDP Service and Application Bindings by Attack Objects . . .	796
	Example: Configuring IDP Policies for Application Identification . . . . .	798
	Disabling Application Identification for an IDP Policy (CLI Procedure) . . . .	799
	IDP Application Identification for Nested Applications . . . . .	800
	Understanding IDP Application Identification for Nested Applications . . . .	800
	Activating IDP Application Identification for Nested Applications (CLI Procedure) . . . . .	800
	Example: Adding IDP Application Information to Attack Logging for Nested Applications (CLI) . . . . .	800
	IDP Application System Cache . . . . .	801
	Understanding the IDP Application System Cache . . . . .	801
	Understanding IDP Application System Cache Information for Nested Application Identification . . . . .	802
	Deactivating IDP Application System Cache Information for Nested Application Identification (CLI Procedure) . . . . .	802
	Verifying IDP Application System Cache Statistics . . . . .	803

	IDP Memory and Session Limits . . . . .	804
	Understanding Memory and Session Limit Settings for IDP Application Identification . . . . .	804
	Example: Setting Memory and Session Limits for IDP Application Identification Services . . . . .	805
	Verifying IDP Counters for Application Identification Processes . . . . .	806
<b>Chapter 26</b>	<b>IDP SSL Inspection . . . . .</b>	<b>809</b>
	IDP SSL Overview . . . . .	809
	Supported IDP SSL Ciphers . . . . .	810
	Understanding IDP Internet Key Exchange . . . . .	811
	Understanding IDP SSL Server Key Management and Policy Configuration . . . . .	812
	Displaying IDP SSL Keys and Associated Servers . . . . .	812
	Adding IDP SSL Keys and Associated Servers . . . . .	813
	Deleting IDP SSL Keys and Associated Servers . . . . .	813
	Configuring an IDP SSL Inspection (CLI Procedure) . . . . .	814
<b>Chapter 27</b>	<b>IDP Class of Service Action . . . . .</b>	<b>817</b>
	IDP Class of Service Action Overview . . . . .	817
	Example: Applying the CoS Action in an IDP Policy . . . . .	818
<b>Chapter 28</b>	<b>IDP Performance and Capacity Tuning . . . . .</b>	<b>825</b>
	Performance and Capacity Tuning for IDP Overview . . . . .	825
	Configuring Session Capacity for IDP (CLI Procedure) . . . . .	826
<b>Chapter 29</b>	<b>IDP Logging . . . . .</b>	<b>827</b>
	Understanding IDP Logging . . . . .	827
	IDP Application-Level DDoS Logging . . . . .	828
	Understanding Application-Level DDoS Logging . . . . .	828
	Enabling Attack and IP-Action Logging (CLI Procedure) . . . . .	830
	IDP Log Suppression Attributes . . . . .	830
	Understanding IDP Log Suppression Attributes . . . . .	830
	Example: Configuring IDP Log Suppression Attributes . . . . .	831
	Security Packet Capture . . . . .	832
	Understanding Security Packet Capture . . . . .	832
	Example: Configuring Security Packet Capture . . . . .	833
	Example: Configuring Packet Capture for Datapath Debugging . . . . .	835
	Verifying Security Packet Capture . . . . .	838
	Understanding IDP Log Information Usage on the Infranet Controller . . . . .	838
	Message Filtering to the Infranet Controller . . . . .	838
	Configuring Infranet Controller Logging . . . . .	839

<b>Part 8</b>	<b>Unified Threat Management</b>	
<b>Chapter 30</b>	<b>Unified Threat Management Overview</b>	<b>843</b>
	Unified Threat Management Overview	843
	Understanding UTM Custom Objects	844
	UTM Licensing	845
	Understanding UTM Licensing	845
	Updating UTM Licenses (CLI Procedure)	846
	WELF Logging for UTM Features	846
	Understanding WELF Logging for UTM Features	846
	Example: Configuring WELF Logging for UTM Features	847
<b>Chapter 31</b>	<b>Antispam Filtering</b>	<b>851</b>
	Antispam Filtering Overview	851
	Server-Based Spam Filtering	851
	Understanding Server-Based Antispam Filtering	852
	Server-Based Antispam Filtering Configuration Overview	853
	Example: Configuring Server-Based Antispam Filtering	853
	Local List Spam Filtering	859
	Understanding Local List Antispam Filtering	859
	Local List Antispam Filtering Configuration Overview	860
	Example: Configuring Local List Antispam Filtering	861
	Understanding Spam Message Handling	867
	Blocking Detected Spam	867
	Tagging Detected Spam	867
<b>Chapter 32</b>	<b>Full Antivirus Protection</b>	<b>869</b>
	Full Antivirus Protection Overview	869
	Full Antivirus Scanner Pattern Database	870
	Understanding Full Antivirus Pattern Updates	870
	Full Antivirus Pattern Update Configuration Overview	871
	Example: Configuring the Full Antivirus Pattern Update Server	872
	Example: Automatically Updating Full Antivirus Patterns (J-Web)	873
	Example: Automatically Updating Full Antivirus Patterns	874
	Manually Updating, Reloading, and Deleting Full Antivirus Patterns (CLI Procedure)	875
	Full Antivirus File Scanning	875
	Understanding the Full Antivirus Internal Scan Engine	875
	Global, Profile-Based, and Policy-Based Full Antivirus Scan Settings	876
	Understanding Full Antivirus Scan Level Settings	876
	Example: Configuring Full Antivirus Scan Settings at Different Levels	877
	Full Antivirus Scan Modes	879
	Understanding Full Antivirus Scan Mode Support	879
	Example: Configuring Full Antivirus File Extension Scanning	880
	Configuring Full Antivirus File Extension Scanning (CLI Procedure)	881
	Full Antivirus Intelligent Prescreening	881
	Understanding Full Antivirus Intelligent Prescreening	881
	Example: Configuring Full Antivirus Intelligent Prescreening	882

Full Antivirus Content Size Limits . . . . .	883
Understanding Full Antivirus Content Size Limits . . . . .	883
Configuring Full Antivirus Content Size Limits (CLI Procedure) . . . . .	883
Full Antivirus Decompression Layer Limit . . . . .	884
Understanding Full Antivirus Decompression Layer Limits . . . . .	884
Configuring Full Antivirus Decompression Layer Limits (CLI Procedure) . . . . .	884
Full Antivirus Scanning Timeout . . . . .	885
Understanding Full Antivirus Scanning Timeouts . . . . .	885
Configuring Full Antivirus Scanning Timeouts (CLI Procedure) . . . . .	885
Full Antivirus Scan Session Throttling . . . . .	885
Understanding Full Antivirus Scan Session Throttling . . . . .	886
Configuring Full Antivirus Scan Session Throttling (CLI Procedure) . . . . .	886
Full Antivirus Application Protocol Scanning . . . . .	886
Understanding Full Antivirus Application Protocol Scanning . . . . .	886
HTTP Full Antivirus Scanning . . . . .	887
Understanding HTTP Scanning . . . . .	888
Enabling HTTP Scanning (CLI Procedure) . . . . .	889
Understanding HTTP Tricking . . . . .	889
Configuring HTTP Tricking to Prevent Timeouts During Antivirus Scanning (CLI Procedure) . . . . .	889
Understanding MIME Whitelists . . . . .	890
Example: Configuring MIME Whitelists to Bypass Antivirus Scanning . . . . .	890
Understanding URL Whitelists . . . . .	891
Configuring URL Whitelists to Bypass Antivirus Scanning (CLI Procedure) . . . . .	892
FTP Full Antivirus Scanning . . . . .	892
Understanding FTP Antivirus Scanning . . . . .	892
Enabling FTP Antivirus Scanning (CLI Procedure) . . . . .	893
SMTP Full Antivirus Scanning . . . . .	893
Understanding SMTP Antivirus Scanning . . . . .	893
Enabling SMTP Antivirus Scanning (CLI Procedure) . . . . .	895
POP3 Full Antivirus Scanning . . . . .	895
Understanding POP3 Antivirus Scanning . . . . .	895
Enabling POP3 Antivirus Scanning (CLI Procedure) . . . . .	896
IMAP Full Antivirus Scanning . . . . .	897
Understanding IMAP Antivirus Scanning . . . . .	897
Enabling IMAP Antivirus Scanning (CLI Procedure) . . . . .	899
Full Antivirus Scan Results and Notification Options . . . . .	899
Understanding Full Antivirus Scan Result Handling . . . . .	899
Protocol-Only Virus-Detected Notifications . . . . .	899
Understanding Protocol-Only Virus-Detected Notifications . . . . .	900
Configuring Protocol-Only Virus-Detected Notifications (CLI Procedure) . . . . .	900
E-Mail Virus-Detected Notifications . . . . .	900
Understanding E-Mail Virus-Detected Notifications . . . . .	900
Configuring E-Mail Virus-Detected Notifications (CLI Procedure) . . . . .	901

Custom Message Virus-Detected Notifications . . . . .	901
Understanding Custom Message Virus-Detected Notifications . . . . .	901
Configuring Custom Message Virus-Detected Notifications (CLI Procedure) . . . . .	902
Full Antivirus Scanning Fallback Options . . . . .	902
Understanding Antivirus Scanning Fallback Options . . . . .	902
Example: Configuring Antivirus Scanning Fallback Options . . . . .	903
Full Antivirus Configuration Overview . . . . .	906
Configuring Full Antivirus (J-Web Procedure) . . . . .	907
Configuring Full Antivirus Custom Objects (J-Web Procedure) . . . . .	907
Configuring Full Antivirus Feature Profiles (J-Web Procedure) . . . . .	909
Configuring Full Antivirus UTM Policies (J-Web Procedure) . . . . .	912
Attaching Full Antivirus UTM Policies to Security Policies (J-Web Procedure) . . . . .	913
Example: Configuring Full Antivirus (CLI) . . . . .	914
Example: Configuring Full Antivirus Custom Objects . . . . .	914
Example: Configuring Full Antivirus Feature Profiles . . . . .	917
Example: Configuring Full Antivirus UTM Policies . . . . .	922
Example: Attaching Full Antivirus UTM Policies to Security Policies . . . . .	923
Monitoring Antivirus Sessions and Scan Results . . . . .	924
Monitoring Antivirus Scan Engine Status . . . . .	924
Monitoring Antivirus Session Status . . . . .	925
Monitoring Antivirus Scan Results . . . . .	925
<b>Chapter 33 Express Antivirus Protection . . . . .</b>	<b>929</b>
Express Antivirus Protection Overview . . . . .	929
Express Antivirus Packet-Based Scanning Versus File-Based Scanning . . . . .	929
Express Antivirus Expanded MIME Decoding Support . . . . .	930
Express Antivirus Scan Result Handling . . . . .	930
Express Antivirus Intelligent Prescreening . . . . .	930
Express Antivirus Limitations . . . . .	930
Express Antivirus Scanner Pattern Database . . . . .	931
Understanding Express Antivirus Scanner Pattern Updates . . . . .	931
Example: Automatically Updating Express Antivirus Patterns (J-Web) . . . . .	932
Example: Automatically Updating Express Antivirus Patterns . . . . .	933
Manually Updating, Reloading, and Deleting Express Antivirus Patterns (CLI Procedure) . . . . .	934
Express Antivirus Configuration Overview . . . . .	934
Configuring Express Antivirus (J-Web Procedure) . . . . .	935
Configuring Express Antivirus Custom Objects (J-Web Procedure) . . . . .	935
Configuring Express Antivirus Feature Profiles (J-Web Procedure) . . . . .	937
Configuring Express Antivirus UTM Policies (J-Web Procedure) . . . . .	939
Attaching Express Antivirus UTM Policies to Security Policies (J-Web Procedure) . . . . .	940
Example: Configuring Express Antivirus (CLI) . . . . .	941
Example: Configuring Express Antivirus Custom Objects . . . . .	941
Example: Configuring Express Antivirus Feature Profiles . . . . .	943
Example: Configuring Express Antivirus UTM Policies . . . . .	948
Example: Attaching Express Antivirus UTM Policies to Security Policies . . . . .	949

<b>Chapter 34</b>	<b>Sophos Antivirus Protection</b> . . . . .	<b>951</b>
	Sophos Antivirus Protection Overview . . . . .	951
	Sophos Antivirus Features . . . . .	952
	Comparison of Sophos Antivirus to Kaspersky Antivirus . . . . .	953
	Understanding Sophos Antivirus Data File Update . . . . .	953
	Managing Sophos Antivirus Data Files . . . . .	954
	Sophos Antivirus Configuration Overview . . . . .	955
	Example: Configuring Sophos Antivirus Custom Objects . . . . .	955
	Example: Configuring Sophos Antivirus Feature Profile . . . . .	959
	Example: Configuring Sophos Antivirus UTM Policies . . . . .	965
	Example: Configuring Sophos Antivirus Firewall Security Policies . . . . .	966
<b>Chapter 35</b>	<b>Content Filtering</b> . . . . .	<b>969</b>
	Content Filtering Overview . . . . .	969
	Content Filtering Protocol Support . . . . .	970
	Understanding Content Filtering Protocol Support . . . . .	970
	HTTP Support . . . . .	970
	FTP Support . . . . .	971
	E-Mail Support . . . . .	971
	Specifying Content Filtering Protocols (CLI Procedure) . . . . .	971
	Example: Configuring Content Filtering . . . . .	972
	Content Filtering Configuration Overview . . . . .	972
	Example: Configuring Content Filtering Custom Objects . . . . .	973
	Example: Configuring Content Filtering Feature Profiles . . . . .	976
	Example: Configuring Content Filtering UTM Policies . . . . .	979
	Example: Attaching Content Filtering UTM Policies to Security Policies . . . . .	980
	Monitoring Content Filtering Configurations . . . . .	982
<b>Chapter 36</b>	<b>Web Filtering</b> . . . . .	<b>985</b>
	Web Filtering Overview . . . . .	985
	Integrated Web Filtering . . . . .	986
	Understanding Integrated Web Filtering . . . . .	986
	Integrated Web Filtering Process . . . . .	987
	Integrated Web Filtering Cache . . . . .	987
	Integrated Web Filtering Profiles . . . . .	988
	Profile Matching Precedence . . . . .	988
	Example: Configuring Integrated Web Filtering . . . . .	989
	Displaying Global SurfControl URL categories . . . . .	996
	Redirect Web Filtering . . . . .	997
	Understanding Redirect Web Filtering . . . . .	997
	Example: Configuring Redirect Web Filtering . . . . .	998
	Local Web Filtering . . . . .	1005
	Understanding Local Web Filtering . . . . .	1005
	User-Defined URL Categories . . . . .	1005
	Local Web Filtering Process . . . . .	1006
	Local Web Filtering Profiles . . . . .	1006
	Profile Matching Precedence . . . . .	1006
	Example: Configuring Local Web Filtering . . . . .	1007
	Monitoring Web Filtering Configurations . . . . .	1012

<b>Part 9</b>	<b>Attack Detection and Prevention</b>	
<b>Chapter 37</b>	<b>Attack Detection and Prevention</b>	<b>1017</b>
	Attack Detection and Prevention Overview	1017
<b>Chapter 38</b>	<b>Reconnaissance Deterrence</b>	<b>1019</b>
	Reconnaissance Deterrence Overview	1019
	IP Address Sweeps	1019
	Understanding IP Address Sweeps	1019
	Example: Blocking IP Address Sweeps	1020
	Port Scanning	1022
	Understanding Port Scanning	1022
	Example: Blocking Port Scans	1023
	Network Reconnaissance Using IP Options	1025
	Understanding Network Reconnaissance Using IP Options	1025
	Uses for IP Packet Header Options	1025
	Screen Options for Detecting IP Options Used for Reconnaissance	1027
	Example: Detecting Packets That Use IP Screen Options for Reconnaissance	1028
	Operating System Probes	1030
	Understanding Operating System Probes	1030
	TCP Headers with SYN and FIN Flags Set	1030
	Understanding TCP Headers with SYN and FIN Flags Set	1030
	Example: Blocking Packets with SYN and FIN Flags Set	1031
	TCP Headers With FIN Flag Set and Without ACK Flag Set	1033
	Understanding TCP Headers With FIN Flag Set and Without ACK Flag Set	1033
	Example: Blocking Packets With FIN Flag Set and Without ACK Flag Set	1034
	TCP Header with No Flags Set	1035
	Understanding TCP Header with No Flags Set	1035
	Example: Blocking Packets with No Flags Set	1036
	Attacker Evasion Techniques	1038
	Understanding Attacker Evasion Techniques	1038
	Fin Scanning	1038
	Understanding FIN Scans	1038
	Thwarting a FIN Scan (CLI Procedure)	1038
	TCP SYN Checking	1039
	Understanding TCP SYN Checking	1039
	Setting TCP SYN Checking (CLI Procedure)	1041
	Setting Strict SYN Checking (CLI Procedure)	1041
	IP Spoofing	1041
	Understanding IP Spoofing	1041
	Example: Blocking IP Spoofing	1042
	IP Source Route Options	1043
	Understanding IP Source Route Options	1043
	Example: Blocking Packets with Either a Loose or a Strict Source Route Option Set	1046
	Example: Detecting Packets with Either a Loose or a Strict Source Route Option Set	1047

<b>Chapter 39</b>	<b>Suspicious Packet Attributes</b> . . . . .	<b>1051</b>
	Suspicious Packet Attributes Overview . . . . .	1051
	ICMP Fragment Protection . . . . .	1051
	Understanding ICMP Fragment Protection . . . . .	1052
	Example: Blocking Fragmented ICMP Packets . . . . .	1052
	Large ICMP Packet Protection . . . . .	1053
	Understanding Large ICMP Packet Protection . . . . .	1053
	Example: Blocking Large ICMP Packets . . . . .	1054
	Bad IP Option Protection . . . . .	1055
	Understanding Bad IP Option Protection . . . . .	1055
	Example: Blocking IP Packets with Incorrectly Formatted Options . . . . .	1056
	Unknown Protocol Protection . . . . .	1057
	Understanding Unknown Protocol Protection . . . . .	1057
	Example: Dropping Packets Using an Unknown Protocol . . . . .	1058
	IP Packet Fragment Protection . . . . .	1059
	Understanding IP Packet Fragment Protection . . . . .	1059
	Example: Dropping Fragmented IP Packets . . . . .	1060
	SYN Fragment Protection . . . . .	1061
	Understanding SYN Fragment Protection . . . . .	1061
	Example: Dropping IP Packets Containing SYN Fragments . . . . .	1062
<b>Chapter 40</b>	<b>Denial-of-Service Attacks</b> . . . . .	<b>1065</b>
	DoS Attack Overview . . . . .	1065
	Firewall DoS Attacks . . . . .	1065
	Firewall DoS Attacks Overview . . . . .	1066
	Session Table Flood Attacks . . . . .	1066
	Understanding Session Table Flood Attacks . . . . .	1066
	Understanding Source-Based Session Limits . . . . .	1067
	Example: Setting Source-Based Session Limits . . . . .	1068
	Understanding Destination-Based Session Limits . . . . .	1070
	Example: Setting Destination-Based Session Limits . . . . .	1071
	SYN-ACK-ACK Proxy Flood Attacks . . . . .	1072
	Understanding SYN-ACK-ACK Proxy Flood Attacks . . . . .	1072
	Example: Protecting Against a SYN-ACK-ACK Proxy Flood Attack . . . . .	1072
	Network DoS Attacks . . . . .	1073
	Network DoS Attacks Overview . . . . .	1073
	SYN Flood Attacks . . . . .	1074
	Understanding SYN Flood Attacks . . . . .	1074
	Example: Enabling SYN Flood Protection . . . . .	1078
	Configuring SYN Flood Protection Options (CLI Procedure) . . . . .	1079
	Example: Enabling SYN Flood Protection for Webservers in the DMZ . . . . .	1079
	SYN Cookie Protection . . . . .	1085
	Understanding SYN Cookie Protection . . . . .	1085
	Example: Enabling SYN Cookie Protection . . . . .	1087
	ICMP Flood Protection . . . . .	1088
	Understanding ICMP Flood Attacks . . . . .	1088
	Example: Enabling ICMP Flood Protection . . . . .	1089



UDP Flood Attacks . . . . .	1090
Understanding UDP Flood Attacks . . . . .	1090
Example: Enabling UDP Flood Protection . . . . .	1091
Land Attacks . . . . .	1092
Understanding Land Attacks . . . . .	1092
Example: Protecting Against a Land Attack . . . . .	1093
OS-Specific DoS Attacks . . . . .	1094
OS-Specific DoS Attacks Overview . . . . .	1094
Ping of Death Attacks . . . . .	1095
Understanding Ping of Death Attacks . . . . .	1095
Example: Protecting Against a Ping of Death Attack . . . . .	1096
Teardrop Attacks . . . . .	1096
Understanding Teardrop Attacks . . . . .	1096
Example: Protecting Against a Teardrop Attack . . . . .	1098
WinNuke Attacks . . . . .	1098
Understanding WinNuke Attacks . . . . .	1098
Example: Protecting Against a WinNuke Attack . . . . .	1100

## Part 10

## Application Identification

### Chapter 41

<b>Junos OS Application Identification . . . . .</b>	<b>1103</b>
Understanding Junos OS Application Identification Services . . . . .	1103
Junos OS Application Identification Application Package . . . . .	1104
Understanding the Junos OS Application Identification Application Package . . . . .	1104
Example: Updating the Junos OS Application Identification Extracted Application Package Automatically . . . . .	1106
Updating the Junos OS Application Identification Extracted Application Package Manually (CLI Procedure) . . . . .	1107
Verifying the Junos OS Application Identification Extracted Application Package . . . . .	1108
Junos OS Application Identification for Nested Applications . . . . .	1109
Understanding Junos OS Application Identification for Nested Applications . . . . .	1109
Activating Junos OS Application Identification for Nested Applications (CLI Procedure) . . . . .	1110
Junos OS Application Identification Custom Application Signature Definitions . . . . .	1110
Understanding Junos OS Application Identification Custom Application Definitions . . . . .	1111
Custom Application Definitions . . . . .	1111
Custom Nested Application Definitions . . . . .	1112
Example: Configuring Junos OS Application Identification Custom Application Definitions . . . . .	1114
Example: Configuring Junos OS Application Identification Custom Nested Application Definitions . . . . .	1117

	Application System Cache . . . . .	1121
	Understanding the Application System Cache . . . . .	1122
	Deactivating Application System Cache Information for Application Identification (CLI Procedure) . . . . .	1122
	Understanding Application System Cache Information for Nested Application Identification . . . . .	1123
	Deactivating Application System Cache Information for Nested Application Identification (CLI Procedure) . . . . .	1123
	Verifying Application System Cache Statistics . . . . .	1124
	Memory and Session Limits . . . . .	1125
	Understanding Memory and Session Limit Settings for Junos OS Application Identification Services . . . . .	1125
	Example: Setting Memory and Session Limits for Junos OS Application Identification Services . . . . .	1126
	Heuristic Detection of Encrypted P2P Applications . . . . .	1127
	Disabling Junos OS Application Identification (CLI Procedure) . . . . .	1127
<b>Chapter 42</b>	<b>AppTrack Application Tracking . . . . .</b>	<b>1129</b>
	Understanding AppTrack . . . . .	1129
	AppTrack Usage . . . . .	1130
	Example: Configuring AppTrack . . . . .	1130
	Example: Verifying AppTrack Operation (CLI) . . . . .	1133
<b>Part 11</b>	<b>Chassis Cluster</b>	
<b>Chapter 43</b>	<b>Chassis Cluster . . . . .</b>	<b>1137</b>
	Chassis Cluster Overview . . . . .	1137
	Understanding Chassis Cluster Formation . . . . .	1138
	Chassis Cluster Redundancy Groups . . . . .	1139
	Understanding Chassis Cluster Redundancy Groups . . . . .	1139
	Chassis Cluster Redundancy Groups 0 Through 128 . . . . .	1140
	Understanding Chassis Cluster Redundancy Group 0: Routing Engines . . . . .	1140
	Understanding Chassis Cluster Redundancy Groups 1 Through 128 . . . . .	1141
	Example: Configuring Chassis Cluster Redundancy Groups . . . . .	1144
	Chassis Cluster Redundancy Group Interface Monitoring . . . . .	1146
	Understanding Chassis Cluster Redundancy Group Interface Monitoring . . . . .	1146
	Example: Configuring Chassis Cluster Interface Monitoring . . . . .	1147
	Chassis Cluster Redundancy Group IP Address Monitoring . . . . .	1148
	Understanding Chassis Cluster Redundancy Group IP Address Monitoring . . . . .	1148
	Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring . . . . .	1151
	Understanding Chassis Cluster Monitoring of Global-Level Objects . . . . .	1154
	Understanding SPU Monitoring . . . . .	1154
	Understanding Flowd Monitoring . . . . .	1154
	Understanding Cold-Sync Monitoring . . . . .	1155

Chassis Cluster Redundancy Group Failover . . . . .	1156
Understanding Chassis Cluster Redundancy Group Failover . . . . .	1156
Understanding Chassis Cluster Redundancy Group Manual Failover . . . . .	1157
Initiating a Chassis Cluster Manual Redundancy Group Failover . . . . .	1158
Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers . . . . .	1160
Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover . . . . .	1161
Chassis Cluster Redundant Ethernet Interfaces . . . . .	1162
Understanding Chassis Cluster Redundant Ethernet Interfaces . . . . .	1162
Example: Configuring Chassis Cluster Redundant Ethernet Interfaces . . . . .	1164
Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups . . . . .	1169
Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups . . . . .	1169
Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups . . . . .	1171
Example: Configuring Chassis Cluster Minimum Links . . . . .	1174
Conditional Route Advertising in a Chassis Cluster . . . . .	1175
Understanding Conditional Route Advertising in a Chassis Cluster . . . . .	1176
Example: Configuring Conditional Route Advertising in a Chassis Cluster . . . . .	1178
Chassis Cluster Control Plane . . . . .	1181
Understanding the Chassis Cluster Control Plane . . . . .	1182
Understanding Chassis Cluster Control Links . . . . .	1183
Example: Configuring Chassis Cluster Control Ports . . . . .	1184
Example: Configuring Chassis Cluster Control Ports for Dual Control Links . . . . .	1186
Understanding Chassis Cluster Dual Control Links . . . . .	1188
Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster . . . . .	1189
Upgrading the Second Routing Engine When Using Chassis Cluster Dual Control Links on SRX5600 and SRX5800 Devices . . . . .	1191
Understanding Chassis Cluster Control Link Heartbeats . . . . .	1192
Understanding Chassis Cluster Control Link Failure and Recovery . . . . .	1193
Example: Configuring Chassis Cluster Control Link Recovery . . . . .	1195
Verifying Chassis Cluster Control Plane Statistics . . . . .	1196
Clearing Chassis Cluster Control Plane Statistics . . . . .	1197
Chassis Cluster Data Plane . . . . .	1197
Understanding the Chassis Cluster Data Plane . . . . .	1197
Understanding Session RTOs . . . . .	1198
Understanding Data Forwarding . . . . .	1198
Understanding Fabric Data Link Failure and Recovery . . . . .	1199
Understanding Chassis Cluster Fabric Links . . . . .	1199
Understanding Chassis Cluster Dual Fabric Links . . . . .	1200
Example: Configuring the Chassis Cluster Fabric . . . . .	1201
Example: Configuring the Chassis Cluster Dual Fabric Links with Matching Slots and Ports . . . . .	1204

Example: Configuring Chassis Cluster Dual Fabric Links with Different Slots and Ports . . . . .	1206
Verifying Chassis Cluster Data Plane Interfaces . . . . .	1208
Verifying Chassis Cluster Data Plane Statistics . . . . .	1209
Clearing Chassis Cluster Data Plane Statistics . . . . .	1210
Consequences of Enabling Chassis Cluster . . . . .	1210
Understanding What Happens When Chassis Cluster Is Enabled . . . . .	1210
Node Interfaces on Active SRX Series Chassis Clusters . . . . .	1211
Node Interfaces on Active J Series Chassis Clusters . . . . .	1219
Management Interface on an Active Chassis Cluster . . . . .	1221
Fabric Interface on an Active Chassis Cluster . . . . .	1222
Control Interface on an Active Chassis Cluster . . . . .	1222
Building a Chassis Cluster . . . . .	1222
Connecting SRX Series Hardware to Create a Chassis Cluster . . . . .	1223
Layer 2 Ethernet Switching Capability in Chassis Cluster Mode . . . . .	1227
Understanding Layer 2 Ethernet Switching Capability in Chassis Cluster on SRX240 and SRX650 Devices . . . . .	1227
Example: Configuring Switch Fabric Interfaces to Enable Switching in Chassis Cluster Mode (CLI) . . . . .	1228
Example: Configuring IRB and VLAN with Members Across Two Nodes (CLI) . . . . .	1229
Example: Configuring Aggregated Ethernet Device with LAG and LACP (CLI) . . . . .	1232
Disabling Switching on SRX100, SRX210, and SRX220 Devices Before Enabling Chassis Clustering . . . . .	1234
SRX Series Chassis Cluster Configuration Overview . . . . .	1235
Connecting J Series Hardware to Create a Chassis Cluster . . . . .	1238
J Series Chassis Cluster Configuration Overview . . . . .	1239
Example: Setting the Chassis Cluster Node ID and Cluster ID . . . . .	1240
Example: Configuring Chassis Cluster Management Interface . . . . .	1242
Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster . . . . .	1245
Verifying a Chassis Cluster Configuration . . . . .	1246
Verifying Chassis Cluster Statistics . . . . .	1247
Clearing Chassis Cluster Statistics . . . . .	1248
Verifying Chassis Cluster Failover Status . . . . .	1249
Clearing Chassis Cluster Failover Status . . . . .	1250
Chassis Cluster Upgrades . . . . .	1250
Upgrading Each Device in a Chassis Cluster Separately . . . . .	1250
Upgrading Both Devices in a Chassis Cluster Using a Low-Impact ISSU . . . . .	1251
Upgrading Both Devices in a Chassis Cluster Using an ISSU . . . . .	1251
Rolling Back Devices in a Chassis Cluster After an ISSU . . . . .	1252
Guarding Against Service Failure in a Chassis Cluster ISSU . . . . .	1252
Enabling an Automatic Chassis Cluster Node Failback After an ISSU . . . . .	1253
Troubleshooting Chassis Cluster ISSU Failures . . . . .	1253
Deciphering Mismatched Control Link Statistics During a Chassis Cluster ISSU . . . . .	1253
Disabling Chassis Cluster . . . . .	1254

Understanding Multicast Routing on a Chassis Cluster . . . . .	1254
Asymmetric Chassis Cluster Deployment . . . . .	1255
Understanding Asymmetric Routing Chassis Cluster Deployment . . . . .	1255
Understanding Failures in the Trust Zone Redundant Ethernet Interface . . . . .	1256
Understanding Failures in the Untrust Zone Interfaces . . . . .	1257
Example: Configuring an Asymmetric Chassis Cluster Pair . . . . .	1257
Active/Passive Chassis Cluster Deployment (J Series Devices) . . . . .	1269
Understanding Active/Passive Chassis Cluster Deployment . . . . .	1269
Example: Configuring an Active/Passive Chassis Cluster Pair (CLI) . . . . .	1270
Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web) . . . . .	1281
Active/Passive Chassis Cluster Deployment (SRX Series Devices) . . . . .	1283
Example: Configuring an SRX Series Services Gateway for the Branch as a Chassis Cluster . . . . .	1283
Example: Configuring an SRX Series Services Gateway for the High-End as a Chassis Cluster . . . . .	1297
Active/Passive Chassis Cluster Deployment with an IPsec Tunnel . . . . .	1312
Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel . . . . .	1312
Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel . . . . .	1314
Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (J-Web) . . . . .	1329

## Part 12

### Chapter 44

## Network Address Translation

<b>Network Address Translation . . . . .</b>	<b>1335</b>
NAT Overview . . . . .	1335
Configuring NAT Using the NAT Wizard . . . . .	1336
Understanding NAT Rule Sets and Rules . . . . .	1336
NAT Rule Sets . . . . .	1336
NAT Rules . . . . .	1337
Rule Processing . . . . .	1338
Static NAT . . . . .	1339
Understanding Static NAT . . . . .	1339
Understanding Static NAT Rules . . . . .	1339
Static NAT Configuration Overview . . . . .	1340
Static NAT Configuration Examples . . . . .	1340
Example: Configuring Static NAT for Single Address Translation . . . . .	1341
Example: Configuring Static NAT for Subnet Translation . . . . .	1345
Destination NAT . . . . .	1350
Understanding Destination NAT . . . . .	1350
Understanding Destination NAT Address Pools . . . . .	1351

Understanding Destination NAT Rules . . . . .	1352
Destination NAT Configuration Overview . . . . .	1353
Destination NAT Configuration Examples . . . . .	1353
Example: Configuring Destination NAT for Single Address Translation . . . . .	1353
Example: Configuring Destination NAT for IP Address and Port Translation . . . . .	1358
Example: Configuring Destination NAT for Subnet Translation . . . . .	1364
Source NAT . . . . .	1368
Understanding Source NAT . . . . .	1369
Source NAT Pools . . . . .	1370
Understanding Source NAT Pools . . . . .	1370
Understanding Source NAT Pools with PAT . . . . .	1371
Understanding Source NAT Pools Without PAT . . . . .	1372
Understanding Source NAT Pools with Address Shifting . . . . .	1372
Understanding Persistent Addresses . . . . .	1373
Understanding Source NAT Rules . . . . .	1373
Source NAT Configuration Overview . . . . .	1374
Source NAT Configuration Examples . . . . .	1374
Example: Configuring Source NAT for Egress Interface Translation . . . . .	1375
Example: Configuring Source NAT for Single Address Translation . . . . .	1378
Example: Configuring Source NAT for Multiple Addresses with PAT . . . . .	1383
Example: Configuring Source NAT for Multiple Addresses without PAT . . . . .	1388
Example: Configuring Source NAT with Address Shifting . . . . .	1393
Example: Configuring Source NAT with Multiple Rules . . . . .	1398
Example: Configuring Source and Destination NAT Translations . . . . .	1405
Disabling Port Randomization for Source NAT (CLI Procedure) . . . . .	1411
Persistent NAT . . . . .	1412
Understanding Persistent NAT . . . . .	1412
Understanding Session Traversal Utilities for NAT (STUN) Protocol . . . . .	1413
Persistent NAT Configuration Overview . . . . .	1414
Example: Configuring Persistent NAT with Source NAT Address Pool (CLI) . . . . .	1415
Example: Configuring Persistent NAT with Interface NAT (CLI) . . . . .	1416
NAT for Multicast Flows . . . . .	1417
Understanding NAT for Multicast Flows . . . . .	1418
Example: Configuring NAT for Multicast Flows . . . . .	1418
Configuring Proxy ARP (CLI Procedure) . . . . .	1427
Verifying NAT Configuration . . . . .	1428

## Part 13

### Chapter 45

## GPRS

<b>General Packet Radio Service . . . . .</b>	<b>1433</b>
GPRS Overview . . . . .	1433
Gp and Gn Interfaces . . . . .	1434
Gi Interface . . . . .	1435

Operational Modes .....	1436
Policy-Based GTP .....	1437
Understanding Policy-Based GTP .....	1437
Example: Enabling GTP Inspection in Policies .....	1438
GTP Inspection Objects .....	1442
Understanding GTP Inspection Objects .....	1442
Example: Creating a GTP Inspection Object .....	1443
GTP Message Filtering .....	1444
Understanding GTP Message Filtering .....	1444
GTP Message-Length Filtering .....	1444
Understanding GTP Message-Length Filtering .....	1444
Example: Setting the GTP Message Lengths .....	1445
GTP Message-Type Filtering .....	1446
Understanding GTP Message-Type Filtering .....	1446
Example: Permitting and Denying GTP Message Types .....	1446
Supported GTP Message Types .....	1447
GTP Message-Rate Limiting .....	1450
Understanding GTP Message-Rate Limiting .....	1450
Example: Limiting the GTP Message Rate .....	1450
GTP Sequence Number Validation .....	1451
Understanding GTP Sequence Number Validation .....	1451
Example: Enabling GTP Sequence Number Validation .....	1452
Understanding GTP IP Fragmentation .....	1453
GTP Information Elements .....	1453
Understanding GTP Information Elements .....	1453
GTP APN Filtering .....	1454
Understanding GTP APN Filtering .....	1454
Example: Setting a GTP APN and a Selection Mode .....	1455
GTP IMSI Prefix Filtering .....	1456
Understanding IMSI Prefix Filtering of GTP Packets .....	1456
Example: Setting a Combined IMSI Prefix and APN Filter .....	1457
GTP R6 Information Elements .....	1458
Understanding R6 Information Elements Removal .....	1458
Example: Removing R6 Information Elements from GTP Messages ..	1458
Supported R6 Information Elements .....	1459
Understanding GGSN Redirection .....	1462
<b>Part 14</b>	
<b>Index</b>	
Index .....	1467





# About This Guide

This preface provides the following guidelines for using the *Junos OS Security Configuration Guide*:

- J Series and SRX Series Documentation and Release Notes on page xli
- Objectives on page xlii
- Audience on page xlii
- Supported Routing Platforms on page xlii
- Document Conventions on page xlii
- Documentation Feedback on page xliv
- Requesting Technical Support on page xliv

## J Series and SRX Series Documentation and Release Notes

---

For a list of related J Series documentation, see <http://www.juniper.net/techpubs/software/junos-jseries/index-main.html> .

For a list of related SRX Series documentation, see <http://www.juniper.net/techpubs/hardware/srx-series-main.html> .

If the information in the latest release notes differs from the information in the documentation, follow the *Junos OS Release Notes*.

To obtain the most current version of all Juniper Networks<sup>®</sup> technical documentation, see the product documentation page on the Juniper Networks website at <http://www.juniper.net/techpubs/>.

Juniper Networks supports a technical book program to publish books by Juniper Networks engineers and subject matter experts with book publishers around the world. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration using the Junos operating system (Junos OS) and Juniper Networks devices. In addition, the Juniper Networks Technical Library, published in conjunction with O'Reilly Media, explores improving network security, reliability, and availability using Junos OS configuration techniques. All the books are for sale at technical bookstores and book outlets around the world. The current list can be viewed at <http://www.juniper.net/books> .

## Objectives

This guide describes how to use and configure key security features on J Series Services Routers and SRX Series Services Gateways running Junos OS. It provides conceptual information, suggested workflows, and examples where applicable.

## Audience

This manual is designed for anyone who installs, sets up, configures, monitors, or administers a J Series Services Router or an SRX Series Services Gateway running Junos OS. The manual is intended for the following audiences:

- Customers with technical knowledge of and experience with networks and network security, the Internet, and Internet routing protocols
- Network administrators who install, configure, and manage Internet routers

## Supported Routing Platforms

This manual describes features supported on J Series Services Routers and SRX Series Services Gateways running Junos OS.

## Document Conventions

Table 1 on page xlii defines the notice icons used in this guide.

Table 1: Notice Icons





Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2 on page xliii defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
<b>Bold text like this</b>	Represents text that you type.	To enter configuration mode, type the <b>configure</b> command:  user@host> <b>configure</b>
Fixed-width text like this	Represents output that appears on the terminal screen.	user@host> <b>show chassis alarms</b>  No alarms currently active
<i>Italic text like this</i>	<ul style="list-style-type: none"> <li>Introduces important new terms.</li> <li>Identifies book names.</li> <li>Identifies RFC and Internet draft titles.</li> </ul>	<ul style="list-style-type: none"> <li>A policy <i>term</i> is a named structure that defines match conditions and actions.</li> <li><i>Junos OS System Basics Configuration Guide</i></li> <li>RFC 1997, <i>BGP Communities Attribute</i></li> </ul>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name:  [edit] root@# <b>set system domain-name</b> <i>domain-name</i>
Text like this	Represents names of configuration statements, commands, files, and directories; interface names; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none"> <li>To configure a stub area, include the <b>stub</b> statement at the [edit protocols ospf area area-id] hierarchy level.</li> <li>The console port is labeled <b>CONSOLE</b>.</li> </ul>
< > (angle brackets)	Enclose optional keywords or variables.	<b>stub &lt;default-metric metric&gt;;</b>
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	<b>broadcast   multicast</b>  <i>(string1   string2   string3)</i>
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	<b>rsvp { # Required for dynamic MPLS only</b>
[ ] (square brackets)	Enclose a variable for which you can substitute one or more values.	<b>community name members [</b> <i>community-ids</i> <b>]</b>
Indentation and braces ( { } )	Identify a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
:(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	

---

### J-Web GUI Conventions

Table 2: Text and Syntax Conventions (*continued*)

Convention	Description	Examples
<b>Bold text like this</b>	Represents J-Web graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> <li>In the Logical Interfaces box, select <b>All Interfaces</b>.</li> <li>To cancel the configuration, click <b>Cancel</b>.</li> </ul>
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web selections.	In the configuration editor hierarchy, select <b>Protocols&gt;Ospf</b> .

## Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to [techpubs-comments@juniper.net](mailto:techpubs-comments@juniper.net), or fill out the documentation feedback form at <https://www.juniper.net/cgi-bin/docbugreport/>. If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version (if applicable)

## Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need postsales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the JTAC User Guide located at <http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf>.
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/>.
- JTAC Hours of Operation —The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

## Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Find product documentation: <http://www.juniper.net/techpubs/>

- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>
- Download the latest versions of software and review release notes:  
<http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications:  
<https://www.juniper.net/alerts/>
- Join and participate in the Juniper Networks Community Forum:  
<http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: <https://tools.juniper.net/SerialNumberEntitlementSearch/>

### Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at <http://www.juniper.net/cm/> .
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, visit us at <http://www.juniper.net/support/requesting-support.html>



## PART 1

# Introduction to Junos OS

- Introducing Junos OS for SRX Series Services Gateways on page 3
- Understanding IPv6 Flow-Based Processing on page 47
- Introducing Junos OS for J Series Services Routers on page 93





## CHAPTER 1

# Introducing Junos OS for SRX Series Services Gateways

- SRX Series Services Gateways Processing Overview on page 3
- Sessions for SRX Series Services Gateways on page 7
- Debugging for SRX Series Services Gateways on page 21
- Understanding SRX Series Services Gateways Central Point Architecture on page 25
- Expanding Session Capacity by Device on page 26
- SRX5600 and SRX5800 Services Gateways Processing Overview on page 28
- SRX1400, SRX3400, and SRX3600 Services Gateways Processing Overview on page 41
- SRX210 Services Gateway Processing Overview on page 44

## SRX Series Services Gateways Processing Overview

---

Junos OS for SRX Series Services Gateways integrates the world-class network security and routing capabilities of Juniper Networks. Junos OS includes a wide range of packet-based filtering, class-of-service (CoS) classifiers, and traffic-shaping features as well as a rich, extensive set of flow-based security features including policies, screens, network address translation (NAT), and other flow-based services.

Traffic that enters and exits services gateway is processed according to features you configure, such as packet filters, security policies, and screens. For example, the software can determine:

- Whether the packet is allowed into the device
- Which firewall screens to apply to the packet
- The route the packet takes to reach its destination
- Which CoS to apply to the packet, if any
- Whether to apply NAT to translate the packet's IP address
- Whether the packet requires an Application Layer Gateway (ALG)

Packets that enter and exit an SRX Series device undergo both packet-based and flow-based processing:

- Flow-based packet processing treats related packets, or a stream of packets, in the same way. Packet treatment depends on characteristics that were established for the first packet of the packet stream, which is referred to as a flow.

For the distributed processing architecture of the services gateway, all flow-based processing occurs on the SPU and sampling is multi-thread aware. Packet sequencing is maintained for the sampled packets.

- Packet-based, or stateless, packet processing treats packets discretely. Each packet is assessed individually for treatment.

For the distributed processing architecture of the services gateway, some packet-based processing, such as traffic shaping, occurs on the NPU. Some packet-based processing, such as application of classifiers to a packet, occurs on the SPU.

This topic includes the following sections:

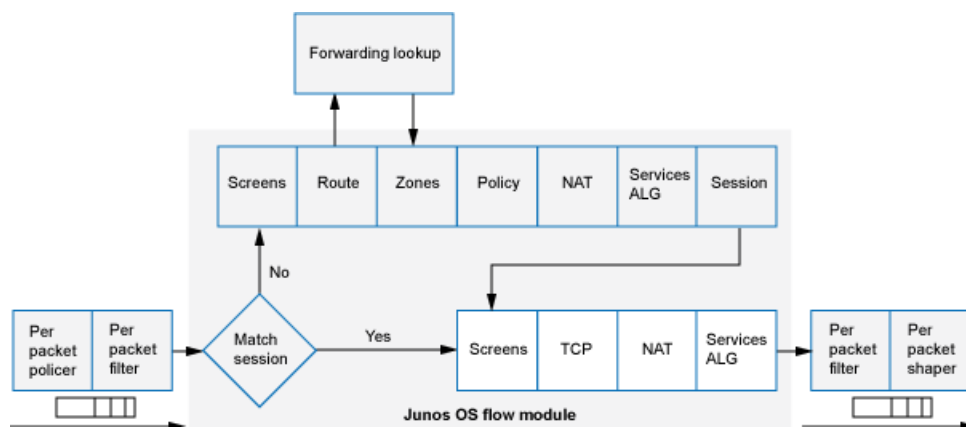
- Understanding Flow-Based Processing on page 4
- Understanding Packet-Based Processing on page 5

## Understanding Flow-Based Processing

A packet undergoes flow-based processing after packet-based filters and some screens have been applied to it. All flow-based processing for a single flow occurs on a single System Processing Unit (SPU). An SPU processes the packets of a flow according to the security features and other services configured for the session.

Figure 1 on page 4 shows a conceptual view of how flow-based traffic processing occurs on services gateway.

Figure 1: Traffic Flow for Flow-Based Processing



A flow is a stream of related packets that meet the same matching criteria and share the same characteristics. Junos OS treats packets belonging to the same flow in the same manner.

Configuration settings that determine the fate of a packet—such as the security policy that applies to it, if it requires an Application Layer Gateway (ALG), if NAT is applied to translate the packet's source and/or destination IP address—are assessed for the first packet of a flow.

To determine if a flow exists for a packet, the NPU attempts to match the packet's information to that of an existing session based on the following match criteria:

- Source address
- Destination address
- Source port
- Destination port
- Protocol
- Unique session token number for a given zone and virtual router

### Zones and Policies

---

The security policy to be used for the first packet of a flow is cached in a flow table for use with the same flow and closely related flows. Security policies are associated with zones. A zone is a collection of interfaces that define a security boundary. A packet's incoming zone, as determined by the interface through which it arrived, and its outgoing zone, as determined by the forwarding lookup, together determine which policy is used for packets of the flow.

### Flows and Sessions

---

Flow-based packet processing, which is stateful, requires the creation of sessions. A session is created for the first packet of a flow for the following purposes:

- To store most of the security measures to be applied to the packets of the flow.
- To cache information about the state of the flow.

For example, logging and counting information for a flow is cached in its session. (Some stateful firewall screens rely on threshold values that pertain to individual sessions or across all sessions.)

- To allocate required resources for the flow for features such as NAT.
- To provide a framework for features such as ALGs and firewall features.

Most packet processing occurs in the context of a flow, including:

- Management of policies, NAT, zones, and most screens.
- Management of ALGs and authentication.

## Understanding Packet-Based Processing

A packet undergoes packet-based processing when it is removed from the queue from its input interface and before it is added to the queue on its output interface.

Packet-based processing applies stateless firewall filters, CoS features, and some screens to discrete packets.

- When a packet arrives at an interface, sanity checks, packet-based filters, some CoS features, and some screens are applied to it.
- Before a packet leaves the device, any packet-based filters, some CoS features, and some screens associated with the interface are applied to the packet.

Filters and CoS features are typically associated with one or more interfaces to influence which packets are allowed to transit the system and to apply special actions to packets as necessary.

The following topics describe the kinds of packet-based features that you can configure and apply to transit traffic.

### Stateless Firewall Filters

---

Also referred to as access control lists (ACLs), stateless firewall filters control access and limit traffic rates. They statically evaluate the contents of packets transiting the device from a source to a destination, or packets originating from or destined for the Routing Engine. A stateless firewall filter evaluates every packet, including fragmented packets.

You can apply a stateless firewall filter to an input or output interface, or to both. A filter contains one or more terms, and each term consists of two components—match conditions and actions. By default, a packet that does not match a firewall filter is discarded.

You can plan and design stateless firewall filters to be used for various purposes—for example, to limit traffic to certain protocols, IP source or destination addresses, or data rates. Stateless firewall filters are executed on the SPU.

### Class-of-Service Features

---

CoS features allow you to classify and shape traffic. CoS features are executed on the SPU.

- Behavior aggregate (BA) classifiers—These classifiers operate on packets as they enter the device. Using behavior aggregate classifiers, the device aggregates different types of traffic into a single forwarding class to receive the same forwarding treatment. BA classifiers allow you to set the forwarding class and loss priority of a packet based on the Differentiated Service (DiffServ) value.
- Traffic shaping—You can shape traffic by assigning service levels with different delay, jitter, and packet loss characteristics to particular applications served by specific traffic flows. Traffic shaping is especially useful for real-time applications, such as voice and video transmission.

### Screens

---

Some screens, such as denial-of-service (DoS) screens, are applied to a packet outside the flow process. They are executed on the Network Processing Unit (NPU).

For details on specific stateless firewall filters and CoS features, see the *Junos OS Routing Protocols and Policies Configuration Guide for Security Devices*, the *Junos OS Class of Service Configuration Guide for Security Devices*, and the *Junos OS CLI Reference*.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Session Characteristics for SRX Series Services Gateways on page 7](#)
- [SRX5600 and SRX5800 Services Gateways Processing Overview on page 28](#)
- [SRX1400, SRX3400, and SRX3600 Services Gateways Processing Overview on page 41](#)
- [SRX210 Services Gateway Processing Overview on page 44](#)

## Sessions for SRX Series Services Gateways

- [Session Characteristics for SRX Series Services Gateways on page 7](#)
- [Monitoring Sessions for SRX Series Services Gateways on page 11](#)
- [Clearing Sessions for SRX Series Services Gateways on page 20](#)

### Session Characteristics for SRX Series Services Gateways

- [Understanding Session Characteristics for SRX Series Services Gateways on page 7](#)
- [Example: Controlling Session Termination for SRX Series Services Gateways on page 8](#)
- [Example: Disabling TCP Packet Security Checks for SRX Series Services Gateways on page 9](#)
- [Example: Setting the Maximum Segment Size for All TCP Sessions for SRX Series Services Gateways on page 10](#)

#### Understanding Session Characteristics for SRX Series Services Gateways

Sessions are created, based on routing and other classification information, to store information and allocate resources for a flow. Sessions have characteristics, some of which you can change, such as when they are terminated. For example, you might want to ensure that a session table is never entirely full to protect against an attacker's attempt to flood the table and thereby prevent legitimate users from starting sessions.

Depending on the protocol and service, a session is programmed with a timeout value. For example, the default timeout for TCP is 1800 seconds. The default timeout for UDP is 60 seconds. When a flow is terminated, it is marked as invalid, and its timeout is reduced to 10 seconds.

If no traffic uses the session before the service timeout, the session is aged out and freed to a common resource pool for reuse. You can affect the life of a session in the following ways:

- You can specify circumstances for terminating sessions by using any of the following methods:
  - Age out sessions based on how full the session table is
  - Set an explicit timeout for aging out TCP sessions

- Configure a TCP session to be invalidated when it receives a TCP RST (reset) message
- You can configure sessions to accommodate other systems as follows:
  - Disable TCP packet security checks
  - Change the maximum segment size

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Juniper Networks Devices Processing Overview on page 3](#)
- [Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12](#)
- [Clearing Sessions for SRX Series Services Gateways on page 20](#)
- [Example: Controlling Session Termination for SRX Series Services Gateways on page 8](#)

**[Example: Controlling Session Termination for SRX Series Services Gateways](#)**

---

This example shows how to terminate sessions for SRX Series devices based on aging out after a certain period of time, or when the number of sessions in the session table is full or reaches a specified percentage. You specify a timeout value or the number of sessions in the session table.

- [Requirements on page 8](#)
- [Overview on page 8](#)
- [Configuration on page 9](#)
- [Verification on page 9](#)

**Requirements**

Before you begin, understand the circumstances for terminating sessions. See “Understanding Session Characteristics for SRX Series Services Gateways” on page 7.

**Overview**

You can control session termination in certain situations—for example, after receiving a TCP FIN Close or receiving an RST message, when encountering ICMP errors for UDP, and when no matching traffic is received before the service timeout. When sessions are terminated, their resources are freed up for use by other sessions.

In this example, you configure the following circumstances to terminate the session:

- A timeout value of 20 seconds.
- An explicit timeout value of 280 seconds, after which the TCP session is removed from the session table.
- Any session that receives a TCP RST (reset) message is invalidated.

**Configuration**

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*

To control session termination for SRX Series devices:

1. Specify an ageout value for the session.
 

```
[edit security flow]
user@host# set aging early-ageout 20
```
2. Configure an aging out value.
 

```
[edit security flow]
user@host# set tcp-session tcp-initial-timeout 280
```
3. Invalidate any session that receives a TCP RST message.
 

```
[edit security flow]
user@host# set tcp-session rst-invalidate-session
```
4. If you are done configuring the device, commit the configuration.
 

```
[edit ]
user@host# commit
```

**Verification**

To verify the configuration is working properly, enter the **show security flow** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Session Characteristics for SRX Series Services Gateways](#) on page 7
- [Example: Disabling TCP Packet Security Checks for SRX Series Services Gateways](#) on page 9
- [Example: Setting the Maximum Segment Size for All TCP Sessions for SRX Series Services Gateways](#) on page 10

**Example: Disabling TCP Packet Security Checks for SRX Series Services Gateways**

This example shows how to disable TCP packet security checks in the device.

- [Requirements](#) on page 9
- [Overview](#) on page 10
- [Configuration](#) on page 10
- [Verification](#) on page 10

**Requirements**

Before you begin, understand the circumstances for disabling TCP packet security checks. See “[Understanding Session Characteristics for SRX Series Services Gateways](#)” on page 7.

### Overview

Junos OS provides a mechanism for disabling security checks on TCP packets to ensure interoperability with hosts and devices with faulty TCP implementations. During no-SYN-check the Junos OS does not look for the TCP SYN packet for session creation. No-sequence check disables TCP sequence checking validation. Also, increases throughput. SYN check and sequence check are enabled by default. The set security flow command disables TCP SYN checks and TCP sequence checks on all TCP sessions thus reduces security. This may be required in scenarios with customers like big transfer files, or with applications that do not correctly work with standards.

### Configuration

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see [Using the CLI Editor in Configuration Mode](#)

To disable TCP packet security checks:

1. Disable the checking of the TCP SYN bit before creating a session.  

```
[edit security flow]  
user@host# set tcp-session no-syn-check
```
2. Disable the checking of sequence numbers in TCP segments during stateful inspection.  

```
[edit security flow]  
user@host# set tcp-session no-sequence-check
```
3. If you are done configuring the device, commit the configuration.  

```
[edit ]  
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security flow** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Controlling Session Termination for SRX Series Services Gateways on page 8](#)
- [Example: Setting the Maximum Segment Size for All TCP Sessions for SRX Series Services Gateways on page 10](#)

### [Example: Setting the Maximum Segment Size for All TCP Sessions for SRX Series Services Gateways](#)

---

This example shows how to set the maximum segment size for all TCP sessions for SRX Series devices.

- [Requirements on page 11](#)
- [Overview on page 11](#)



- Configuration on page 11
- Verification on page 11

### Requirements

Before you begin, understand the circumstances for setting the maximum segment size. See “Understanding Session Characteristics for SRX Series Services Gateways” on page 7.

### Overview

You can terminate all TCP sessions by changing the TCP maximum segment size (TCP-MSS). To diminish the likelihood of fragmentation and to protect against packet loss, you can use the `tcp-mss` statement to specify a lower TCP MSS value. This statement applies to all IPv4 TCP SYN packets traversing all the router’s ingress interfaces whose MSS value is higher than the one you specify.

If the DF bit is set, it will not fragment the packet and Junos OS will send ICMP error type 3 code 4 packet to the application server (Destination Unreachable; Fragmentation Needed and DF set). This ICMP error message contains the correct MTU (as defined in `tcp-mss`) to be used by the application server, which should receive this message and adjust the packet size accordingly. This is specifically required with VPN’s since IPsec has added packet overhead, thus `tcp-mss` has to be lowered appropriately.

### Configuration

#### Step-by-Step Procedure

To configure the maximum segment size for all TCP sessions:

1. Set the TCP maximum segment size for all TCP sessions.

```
[edit security flow]
user@host# set tcp-mss all-tcp mss 1300
```

2. If you are done configuring the device, commit the configuration.

```
[edit ]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security flow** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Example: Controlling Session Termination for SRX Series Services Gateways on page 8
- Example: Disabling TCP Packet Security Checks for SRX Series Services Gateways on page 9

## Monitoring Sessions for SRX Series Services Gateways

- Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12
- Displaying Global Session Parameters for All SRX Series Services Gateways on page 12
- Displaying a Summary of Sessions for SRX Series Services Gateways on page 13

- [Displaying Session and Flow Information About Sessions for SRX Series Services Gateways](#) on page 14
- [Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways](#) on page 14
- [Using Filters to Display Session and Flow Information for SRX Series Services Gateways](#) on page 15
- [Information Provided in Session Log Entries for SRX Series Services Gateways](#) on page 16

### [Understanding How to Obtain Session Information for SRX Series Services Gateways](#)

---

You can obtain information about the sessions and packet flows active on your device, including detailed information about specific sessions. (The SRX Series device also displays information about failed sessions.) You can display this information to observe activity and for debugging purposes. For example, you can use the `show security flow session` command:

- To display a list of incoming and outgoing IP flows, including services
- To show the security attributes associated with a flow, for example, the policies that apply to traffic belonging to that flow
- To display the session timeout value, when the session became active, for how long it has been active, and if there is active traffic on the session

For detailed information about this command, see the *Junos OS CLI Reference*.

Session information can also be logged if a related policy configuration includes the logging option. See “Information Provided in Session Log Entries for SRX Series Services Gateways” on page 16 for details about session information provided in system logs.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Session Characteristics for SRX Series Services Gateways](#) on page 7
- [Clearing Sessions for SRX Series Services Gateways](#) on page 20
- [Displaying Global Session Parameters for All SRX Series Services Gateways](#) on page 12

### [Displaying Global Session Parameters for All SRX Series Services Gateways](#)

---

**Purpose** Obtain information about configured parameters that apply to all flows or sessions.

**Action** To view session information in the CLI, enter the following command:

```
user@host> show security flow
```

**Meaning** The `show security flow` configuration command displays the following information:

For detailed information about this command, see the *Junos OS CLI Reference*.

- **allow-dns-reply**—Identifies if unmatched incoming Domain Name System (DNS) reply packets are allowed.
- **route-change-timeout**—If enabled, displays the session timeout value to be used on a route change to a nonexistent route.
- **tcp-mss**—Shows the current configuration for the TCP maximum segment size value to be used for all TCP packets for network traffic.
- **tcp-session**—Displays all configured parameters that control session parameters.
- **syn-flood-protection-mode**—Displays the SYN Proxy mode.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12
- Displaying a Summary of Sessions for SRX Series Services Gateways on page 13
- Displaying Session and Flow Information About Sessions for SRX Series Services Gateways on page 14
- Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways on page 14
- Using Filters to Display Session and Flow Information for SRX Series Services Gateways on page 15
- Information Provided in Session Log Entries for SRX Series Services Gateways on page 16

#### Displaying a Summary of Sessions for SRX Series Services Gateways

**Purpose** Determine the kinds of sessions on your device, how many of each kind there are—for example, the number of unicast sessions and multicast sessions—the number of failed sessions, the number of sessions that are currently used and the maximum number of sessions that the device supports. This command also displays the details of the sessions that are currently used. For example, valid sessions, pending sessions, invalidated sessions and sessions in other states.

**Action** To view session summary information in the CLI, enter the following CLI command:

```
user@host> show security flow session summary
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12
- Displaying Global Session Parameters for All SRX Series Services Gateways on page 12
- Displaying Session and Flow Information About Sessions for SRX Series Services Gateways on page 14

- [Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways](#) on page 14
- [Using Filters to Display Session and Flow Information for SRX Series Services Gateways](#) on page 15
- [Information Provided in Session Log Entries for SRX Series Services Gateways](#) on page 16

### [Displaying Session and Flow Information About Sessions for SRX Series Services Gateways](#)

---

**Purpose** Display information about all sessions on your device, including the session ID, the virtual system the session belongs to, the Network Address Translation (NAT) source pool (if source NAT is used), the configured timeout value for the session and its standard timeout, and the session start time and how long the session has been active. The display also shows all standard flow information, including the direction of the flow, the source address and port, the destination address and port, the IP protocol, and the interface used for the session.

**Action** To view session flow information in the CLI, enter the following command:

```
user@host> show security flow session
```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding How to Obtain Session Information for SRX Series Services Gateways](#) on page 12
- [Displaying Global Session Parameters for All SRX Series Services Gateways](#) on page 12
- [Displaying a Summary of Sessions for SRX Series Services Gateways](#) on page 13
- [Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways](#) on page 14
- [Using Filters to Display Session and Flow Information for SRX Series Services Gateways](#) on page 15
- [Information Provided in Session Log Entries for SRX Series Services Gateways](#) on page 16

### [Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways](#)

---

**Purpose** When you know the session identifier, you can display all session and flow information for a specific session rather than for all sessions.

**Action** To view information about a specific session in the CLI, enter the following command:

```
user@host> show security flow session session-identifier 40000381
```

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12
  - Displaying Global Session Parameters for All SRX Series Services Gateways on page 12
  - Displaying a Summary of Sessions for SRX Series Services Gateways on page 13
  - Displaying Session and Flow Information About Sessions for SRX Series Services Gateways on page 14
  - Using Filters to Display Session and Flow Information for SRX Series Services Gateways on page 15
  - Information Provided in Session Log Entries for SRX Series Services Gateways on page 16

### [Using Filters to Display Session and Flow Information for SRX Series Services Gateways](#)

**Purpose** You can display flow and session information about one or more sessions by specifying a filter as an argument to the **show security flow session** command. You can use the following filters: application, destination-port, destination-prefix, family, idp, interface, nat, protocol, resource-manager, session-identifier, source-port, source-prefix and tunnel. The device displays the information for each session followed by a line specifying the number of sessions reported on. Here is an example of the command using the source-prefix filter.

**Action** To view information about selected sessions using filters in the CLI, enter the following command:

```
user@host> show security flow session source-prefix 10/8
```

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12
  - Displaying Global Session Parameters for All SRX Series Services Gateways on page 12
  - Displaying a Summary of Sessions for SRX Series Services Gateways on page 13
  - Displaying Session and Flow Information About Sessions for SRX Series Services Gateways on page 14
  - Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways on page 14
  - Information Provided in Session Log Entries for SRX Series Services Gateways on page 16

### Information Provided in Session Log Entries for SRX Series Services Gateways

Session log entries are tied to policy configuration. Each main session event—create, close, and deny—will create a log entry if the controlling policy has enabled logging.

Different fields are logged for session create, session close, and session deny events as shown in Table 3 on page 16, Table 4 on page 17, and Table 5 on page 19. The same field name under each type indicates that the same information is logged, but each table is a full list of all data recorded for that type of session log.

The following table defines the fields displayed in session log entries.

**Table 3: Session Create Log Fields**

Field	Description
<b>source-address</b>	Source IP address of the packet that created the session.
<b>source-port</b>	Source port of the packet that created the session.
<b>destination-address</b>	Destination IP address of the packet that created the session.
<b>destination-port</b>	Destination port of the packet that created the session.
<b>service-name</b>	Application that the packet traversed (for example, "junos-telnet" for Telnet traffic during the session allowed by a policy that permits native Telnet).
<b>nat-source-address</b>	The translated NAT source address if NAT was applied; otherwise, the source address as above.
<b>nat-source-port</b>	The translated NAT source port if NAT was applied; otherwise, the source port as above.
<b>nat-destination-address</b>	The translated NAT destination address if NAT was applied; otherwise, the destination address as above.
<b>nat-destination-port</b>	The translated NAT destination port if NAT was applied; otherwise, the destination port as above.
<b>src-nat-rule-name</b>	The source NAT rule that was applied to the session (if any). If static NAT is also configured and applied to the session and if source address translation takes place, then this field shows the static NAT rule name.*
<b>dst-nat-rule-name</b>	The destination NAT rule that was applied to the session (if any). If static NAT is also configured and applied to the session and if destination address translation takes place, then this field shows the static NAT rule name.*
<b>protocol-id</b>	The protocol ID of the packet that created the session.
<b>policy-name</b>	The name of the policy that permitted the session creation.
<b>session-id-32</b>	The 32-bit session ID.

\* Note that some sessions might have both destination and source NAT applied and the information logged.

Table 4: Session Close Log Fields

Field	Description
<b>reason</b>	The reason the session was closed.
<b>source-address</b>	Source IP address of the packet that created the session.
<b>source-port</b>	Source port of the packet that created the session.
<b>destination-address</b>	Destination IP address of the packet that created the session.
<b>destination-port</b>	Destination port of the packet that created the session.
<b>service-name</b>	Application that the packet traversed (for example, "junos-telnet" for Telnet traffic during the session allowed by a policy that permits native Telnet).
<b>nat-source-address</b>	The translated NAT source address if NAT was applied; otherwise, the source address as above.
<b>nat-source-port</b>	The translated NAT source port if NAT was applied; otherwise, the source port as above.
<b>nat-destination-address</b>	The translated NAT destination address if NAT was applied; otherwise, the destination address as above.
<b>nat-destination-port</b>	The translated NAT destination port if NAT was applied; otherwise, the destination port as above.
<b>src-nat-rule-name</b>	The source NAT rule that was applied to the session (if any). If static NAT is also configured and applied to the session and if source address translation takes place, then this field shows the static NAT rule name.*
<b>dst-nat-rule-name</b>	The destination NAT rule that was applied to the session (if any). If static NAT is also configured and applied to the session and if destination address translation takes place, then this field shows the static NAT rule name.*
<b>protocol-id</b>	The protocol ID of the packet that created the session.
<b>policy-name</b>	The name of the policy that permitted the session creation.
<b>session-id-32</b>	The 32-bit session ID.
<b>packets-from-client</b>	The number of packets sent by the client related to this session.
<b>bytes-from-client</b>	The number of data bytes sent by the client related to this session.
<b>packets-from-server</b>	The number of packets sent by the server related to this session.

Table 4: Session Close Log Fields (*continued*)

Field	Description
bytes-from-server	The number of data bytes sent by the server related to this session.
elapsed-time	The total session elapsed time from permit to close, given in seconds.
unset	<p>During the session creation, you can set the session close reason as <b>unset</b>.</p> <p>The session closes with the reason <b>unset</b> if the session installation on the control point is not successful. The reason for session installation varies, for example, nonavailability of memory for nonmanagement session installation.</p>
TCP RST	RST received from either end.
TCP FIN	FIN received from either end.
response received	Response received for a packet request (for example, ICMP req-reply).
ICMP error	ICMP error received.
aged out	Session aged out was reached.
ALG	ALG errors closed the session (for example, remote access server (RAS) maximum limit reached).
HA	HA message closed the session.
auth	Authentication failed.
IDP	IDP closed the session because of security module (SM) internal error.
synproxy failure	SYN proxy failure closed the session.
synproxy limit	Reason for failure in allocating minor session, need to free original session.
parent closed	Parent session closed.
CLI	Session cleared by a CLI statement.
CP NACK	CP NACK response received.
CP delete	CP ACK deletion closed the session.
policy delete	Corresponding policy marked for deletion.



Table 4: Session Close Log Fields (*continued*)

Field	Description
<b>fwd session</b>	Session closed because of forwarding session deletion.
<b>multicast route change</b>	Session closed because multicast route changed.
<b>first path reroute, session recreated</b>	The first path is rerouted and session is re-created.
<b>source NAT allocation failure</b>	SPU received ACK message from the central point but failed to receive the DIP resource. Therefore this packet is dropped and the session is closed.
<b>other</b>	Session closed because of all other reasons (for example, the pim reg tun needed refreshing).
<b>error create IKE pass-through template</b>	IKE pass-through template creation errors.
<b>IKE pass-through child session ageout</b>	Session is deleted because the IKE pass through template session has no child.
<b>sess timeout on pending state</b>	Pending session closed because time out timer reached the pending state.
<b>unknown</b>	Session closed because of unknown reasons.

*\* Note that some sessions might have both destination and source NAT applied and the information logged.*

Table 5: Session Deny Log Fields

Field	Description
<b>source-address</b>	Source IP address of the packet that attempted to create the session.
<b>source-port</b>	Source port of the packet that attempted to create the session.
<b>destination-address</b>	Destination IP address of the packet that attempted to create the session.
<b>destination-port</b>	Destination port of the packet that attempted to create the session.
<b>service-name</b>	Application that the packet attempted to traverse.
<b>protocol-id</b>	The protocol ID of the packet that attempted to create the session.
<b>icmp-type</b>	The ICMP type if the denied packet was ICMP configured; otherwise, this field will be 0.
<b>policy-name</b>	The name of the policy that denied the session creation.

**Related Documentation** • [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

- [Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12](#)
- [Displaying Global Session Parameters for All SRX Series Services Gateways on page 12](#)
- [Displaying a Summary of Sessions for SRX Series Services Gateways on page 13](#)
- [Displaying Session and Flow Information About Sessions for SRX Series Services Gateways on page 14](#)
- [Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways on page 14](#)
- [Clearing Sessions for SRX Series Services Gateways on page 20](#)
- [Using Filters to Display Session and Flow Information for SRX Series Services Gateways on page 15](#)

## Clearing Sessions for SRX Series Services Gateways

You can use the **clear** command to terminate sessions. You can clear all sessions, including sessions of a particular application type, sessions that use a specific destination port, sessions that use a specific interface or port, sessions that use a certain IP protocol, sessions that match a source prefix, and resource manager sessions.

- [Terminating Sessions for SRX Series Services Gateways on page 20](#)
- [Terminating a Specific Session for SRX Series Services Gateways on page 20](#)
- [Using Filters to Specify the Sessions to Be Terminated for SRX Series Services Gateways on page 20](#)

### Terminating Sessions for SRX Series Services Gateways

---

You can use the following command to terminate all sessions except tunnel and resource manager sessions. The command output shows the number of sessions cleared. Be aware that this command terminates the management session through which the clear command is issued.

```
user@host> clear security flow session all
```

### Terminating a Specific Session for SRX Series Services Gateways

---

You can use the following command to terminate the session whose session ID you specify.

```
user@host> clear security flow session session-identifier 40000381
```

### Using Filters to Specify the Sessions to Be Terminated for SRX Series Services Gateways

---

You can terminate one or more sessions based on the filter parameter you specify for the **clear** command. The following example uses the protocol as a filter.

```
user@host> clear security flow session protocol 89
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

- Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12

## Debugging for SRX Series Services Gateways

---

- Data Path Debugging for SRX Series Services Gateways on page 21
- Security Debugging for SRX Series Services Gateways on page 22
- Flow Debugging for SRX Series Services Gateways on page 24

### Data Path Debugging for SRX Series Services Gateways

- Understanding Data Path Debugging for SRX Series Services Gateways on page 21
- Debugging the Data Path (CLI Procedure) on page 22

#### Understanding Data Path Debugging for SRX Series Services Gateways

---

Data path debugging provides tracing and debugging at multiple processing units along the packet-processing path. The packet filter can be executed with minimal impact to the production system.

In data path debugging, a packet goes through multiple Services Processing Units (SPUs). At the same time, several Flexible PIC Concentrator (FPC) I/O cards (IOCs) provide EZchip ingress and egress traffic management. Junos OS supports IOC for filter-based, per-packet counting and logging to record the processing path of a packet. Only the matched packets are traced by the IOC EZchip ingress, EZchip egress, load-balancing thread (LBT), and packet-ordering thread (POT).

The following events are defined in the packet-processing path:

- ezchip ingress
- ezchip egress
- spu.lbt
- spu.pot



#### NOTE:

The packet-filtering behavior for the port and interface options is as follows:

- The packet filter traces both IPv4 and IPv6 traffic if only **port** is specified.
  - The packet filter traces IPv4, IPV6, and non-IP traffic if only **interface** is specified.
- 

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Juniper Networks Devices Processing Overview on page 3
- Understanding Session Characteristics for SRX Series Services Gateways on page 7

- SRX5600 and SRX5800 Services Gateways Processing Overview on page 28
- SRX1400, SRX3400, and SRX3600 Services Gateways Processing Overview on page 41
- SRX210 Services Gateway Processing Overview on page 44
- Debugging the Data Path (CLI Procedure) on page 22

### [Debugging the Data Path \(CLI Procedure\)](#)

---

To configure the device for data path debugging:

1. Specify the following request command to set the data path debugging for the multiple processing units along the packet-processing path:

```
user@host# set security datapath-debug
```

2. Specify the trace options for data path-debug using the following command:

```
user@host# set security datapath-debug traceoptions
```

3. Using the request security packet-filter command, you can set the packet filter to specify the related packets to perform data path-debug action. A maximum of four filters are supported at the same time. For example, the following command sets the first packet-filter:

```
user@host# set security datapath-debug packet-filter name
```

4. Using the request security action-profile command, you can set the action for the packet match for a specified filter. Only the default action profile is supported, which is the trace option for network processor ezchip ingress, ezchip egress, spu.lbt, and spu.pot:

```
user@host# set security datapath-debug packet-filter name action-profile
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Data Path Debugging for SRX Series Devices on page 21](#)
- [SRX5600 and SRX5800 Services Gateways Processing Overview on page 28](#)

## Security Debugging for SRX Series Services Gateways

- [Understanding Security Debugging Using Trace Options on page 22](#)
- [Setting Security Trace Options \(CLI Procedure\) on page 23](#)
- [Displaying Output for Security Trace Options on page 24](#)

### [Understanding Security Debugging Using Trace Options](#)

---

The Junos OS trace function allows applications to write security debugging information to a file. The information that appears in this file is based on criteria you set. You can use this information to analyze security application issues.

The trace function operates in a distributed manner, with each thread writing to its own trace buffer. These trace buffers are then collected at one point, sorted, and written to trace files. Trace messages are delivered using the InterProcess Communications (IPC)

protocol. A trace message has a lower priority than that of control protocol packets such as BGP, OSPF, and IKE, and therefore delivery is not considered to be as reliable.

### Setting Security Trace Options (CLI Procedure)

Use the following configuration statements to configure security trace options in the CLI configuration editor.

- To disable remote tracing, enter the following statement:

```
user@host# set security traceoptions no-remote-trace
```

- To write trace messages to a local file, enter the following statement. The system saves the trace file in the `/var/log/` directory.

```
user@host# set security traceoptions use-local-files
```

- To specify a name for the trace file, enter the following statement. Valid values range from 1 and 1024 characters. The name cannot include spaces, `/`, or `%` characters. The default filename is `security`.

```
user@host#set security traceoptions file filename
```

- To specify the maximum number of trace files that can accumulate, enter the following statement. Valid values range from 2 to 1000. The default value is 3.

```
user@host# set security traceoptions file files 3
```

- To specify the match criteria that you want the system to use when logging information to the file, enter the following statement. Enter a regular expression. Wildcard (`*`) characters are accepted.

```
user@host# set security traceoptions file match *thread
```

- To allow any user to read the trace file, enter the `world-readable` statement. Otherwise, enter the `no-world-readable` statement.

```
user@host#set security traceoptions file world-readable
user@host# set security traceoptions file no-world-readable
```

- To specify the maximum size to which the trace file can grow, enter the following statement. Once the file reaches the specified size, it is compressed and renamed `filename0.gz`, the next file is named `filename1.gz`, and so on. Valid values range from 10240 to 1,073,741,824.

```
user@host#set security traceoptions file size 10240
```

- To turn on trace options and to perform more than one tracing operation, set the following flags.

```
user@host#set security traceoptions flag all
user@host#set security traceoptions flag compilation
user@host# set security traceoptions flag configuration
user@host#set security traceoptions flag routing-socket
```

- To specify the groups that these trace option settings do or do not apply to, enter the following statements:

```
user@host# set security traceoptions apply-groups value
user@host# set security traceoptions apply-groups-except value
```

## Displaying Output for Security Trace Options

---

**Purpose** Display output for security trace options.

**Action** Use the `show security traceoptions` command to display the output of your trace files. For example:

```
user@host # show security traceoptions file usp_trace
user@host # show security traceoptions flag all
user@host # show security traceoptions rate-limit 888
```

The output for this example is as follows:

```
Apr 11 16:06:42 21:13:15.750395:CID-906489336:FPC-01:PIC-01:THREAD_ID-01:PFE:now
update 0x3607edf8df8in 0x3607e8d0
Apr 11 16:06:42 21:13:15.874058:CID-1529687608:FPC-01:PIC-01:THREAD_ID-01:CTRL:Enter
Function[util_ssam_handler]
Apr 11 16:06:42 21:13:15.874485:CID-00:FPC-01:PIC-01:THREAD_ID-01:CTRL:default1: Rate
limit changed to 888
Apr 11 16:06:42 21:13:15.874538:CID-00:FPC-01:PIC-01:THREAD_ID-01:CTRL:default1:
Destination ID set to 1
Apr 11 16:06:42 21:13:15.874651:CID-00:FPC-01:PIC-01:THREAD_ID-01:CTRL:default2: Rate
limit changed to 888
Apr 11 16:06:42 21:13:15.874832:CID-00:FPC-01:PIC-01:THREAD_ID-01:CTRL:default2:
Destination ID set to 1
Apr 11 16:06:42 21:13:15.874942:CID-00:FPC-01:PIC-01:THREAD_ID-01:CTRL:default3: Rate
limit changed to 888
Apr 11 16:06:42 21:13:15.874997:CID-00:FPC-01:PIC-01:THREAD_ID-01:CTRL:default3:
Destination ID set to 1
```

## Flow Debugging for SRX Series Services Gateways

- Understanding Flow Debugging Using Trace Options on page 24
- Setting Flow Debugging Trace Options (CLI Procedure) on page 24

### Understanding Flow Debugging Using Trace Options

---

For flow trace options, you can define a packet filter using combinations of **destination-port**, **destination-prefix**, **interface**, **protocol**, **source-port**, and **source-prefix**. If the security flow trace flag for a certain module is set, the packet matching the specific packet filter triggers flow tracing and writes debugging information to the trace file.

### Setting Flow Debugging Trace Options (CLI Procedure)

---

The following examples display the options you can set by using **security flow traceoptions**.

- To match the imap destination port for the filter1 packet filter, use the following statement:

```
user@host# set security flow traceoptions packet-filter filter1 destination-port imap
```

- To set the 1.2.3.4 destination IPv4 prefix address for the filter1 packet filter, use the following statement:

```
user@host# set security flow traceoptions packet-filter filter1 destination-prefix 1.2.3.4
```

- To set the fxp0 logical interface for the filter1 packet filter, use the following statement:

```
user@host# set security flow traceoptions packet-filter filter1 interface fxp0
```

- To match the TCP IP protocol for the filter1 packet filter, use the following statement:

```
user@host# set security flow traceoptions packet-filter filter1 protocol tcp
```

- To match the HTTP source port for the filter1 packet filter, use the following statement:

```
user@host# set security flow traceoptions packet-filter filter1 source-port http
```

- To set the 5.6.7.8 IPv4 prefix address for the filter1 packet filter, use the following statement:

```
user@host# set security flow traceoptions packet-filter filter1 source-prefix 5.6.7.8
```

## Understanding SRX Series Services Gateways Central Point Architecture

The central point in the architecture has two basic flow functionalities: load balancing and traffic identification (global session matching). The central point forwards a packet to its Services Processing Unit (SPU) upon session matching, or distributes traffic to an SPU for security processing if the packet does not match any existing session.

On some SRX Series devices, an entire SPU cannot be dedicated for central point functionality, but a certain percentage of the SPU is automatically allocated for central point functionality and the rest is allocated for normal flow processing. When an SPU performs the function of central point as well as normal flow processing, it is said to be in combination, or *combo*, mode.

The percentage of SPU dedicated to the central point functionality depends on the number of SPUs in the device. Based on the number of SPUs, there are three modes available on the SRX Series devices— small central point, medium central point, and large central point.

In small central point mode, a small percentage of an SPU is dedicated to central point functionality and the rest is dedicated to the normal flow processing. In medium central point mode, an SPU is almost equally shared for central point functionality and normal flow processing. In large central point mode, an entire SPU is dedicated to central point functionality. In combo mode, the central point and SPU share the same load-balancing thread (LBT) and packet-ordering thread (POT) infrastructure.

This topic includes the following sections:

- Load Distribution in Combo Mode on page 25
- Sharing Processing Power and Memory in Combo Mode on page 26

### Load Distribution in Combo Mode

The central point maintains SPU mapping table (for load distribution) that lists live SPUs with the logic SPU IDs mapped to the physical Trivial Network Protocol (TNP) addresses mapping. In combo mode, the SPU that hosts the central point is included in the table. The load distribution algorithm is adjusted based on session capacity and processing power to avoid overloading of sessions.

## Sharing Processing Power and Memory in Combo Mode

The CPU processing power in a combo-mode SPU is shared based on the platform and the number of SPUs in the system. Similarly, the CPU memory is also shared between the central point and SPU.

An SPU has multiple cores (CPUs) for networking processing. In "small" SPU combo mode, CPU functionality takes a small portion of the cores, whereas "medium" SPU combo-mode requires a larger portion of cores. The processing power for central point functionalities and flow processing is shared, based on the number of Services Processing Cards (SPC), as shown in Table 6 on page 26.



**NOTE:** An SPC contains one or more SPUs that perform flow processing and central point functionality.

**Table 6: Combo Mode Processing**

SRX Series device	Central point mode with 1 SPC	Central point mode with 2 or More than 2 SPCs
SRX1400	Small	Medium
SRX3400	Small	Medium
SRX3600	Small	Medium
SRX3400 (expanded performance and capacity license)	Small	Large
SRX3600 (expanded performance and capacity license)	Small	Large
SRX5600	Medium	Large
SRX5800	Medium	Large

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12](#)
- [Understanding Session Characteristics for SRX Series Services Gateways on page 7](#)

## Expanding Session Capacity by Device

To take advantage of the processing potential of a fully loaded SRX3400, SRX3600, or SRX5800 device, you can expand the maximum number of concurrent sessions for these devices.



Table 7 on page 27 shows the maximum number of concurrent sessions allowed on these devices by default and with expanded capacity.

**Table 7: Maximum Central Point Session Increases**

SRX Series Devices	Maximum Concurrent Sessions on a Fully Loaded System	
	Default	With Expanded Capacity
SRX3400	2.25 million	3 million
SRX3600	2.25 million	6 million
SRX5600	9 million	Expansion not available
SRX5800	12.5 million	14.0 million

The method used for expanding session capacity depends on the device:

- Central point session license installation and validation on an SRX3400 or SRX3600 device
- CLI optimization option on an SRX5800 device

The following sections provide information about session limit expansion on SRX3400, SRX3600, and SRX5800 devices.

- Expanding Session Capacity on an SRX3400 or SRX3600 Device on page 27
- Expanding Session Capacity on an SRX5800 Device on page 27
- Reverting to Default Session Capacity on an SRX5800 Device on page 28
- Verifying the Current Session Capacity on page 28

### Expanding Session Capacity on an SRX3400 or SRX3600 Device

Expanding session capacity on an SRX3400 or SRX3600 device requires validation of a central point session license on the device.

1. Obtain the central point session license key and install the license on the device. For license installation details, see the *Junos OS Administration Guide for Security Devices*.
2. Reboot the device to implement the expanded session capacity.

### Expanding Session Capacity on an SRX5800 Device

Expanding session capacity on an SRX5800 device requires a CLI configuration change.

1. Enter the following command at the CLI configuration prompt:
 

```
user@host# edit security forwarding-process application-services
maximize-cp-sessions
```
2. Reboot the device to implement the expanded session capacity.

Using the central point sessions optimization technique precludes other optimization methods, disables advanced GTP processing, and reduces routing capacity to 100K prefixes.

### Reverting to Default Session Capacity on an SRX5800 Device

Reverting to the default session capacity on an SRX5800 device requires a CLI configuration change.

1. Enter the following command at the CLI configuration prompt to reestablish the default session capacity value:

```
user@host# set security gprs gtp enable
```

2. Reboot the device to implement the new value.

### Verifying the Current Session Capacity

**Purpose** The central point session summary includes the maximum sessions setting for the device. From this value you can determine if the session capacity has been modified as you expected.

**Action** To verify the current setting of the central point session capacity, enter the following CLI command.

```
user@host> show security flow cp-session summary
```

```
Valid sessions: 114323  
Pending sessions: 11  
Invalidated sessions: 32551  
Sessions in other states: 0  
Total sessions: 146885  
Maximum sessions: 14000000  
Maximum inet6 sessions: 7000000
```

**Meaning** The **Maximum sessions** value reflects the current session capacity on your device. A value of 14000000 means that the SRX5800 device is configured for the expanded number of central point sessions.

---

## SRX5600 and SRX5800 Services Gateways Processing Overview

Junos OS is a distributed parallel processing high throughput and high performance system. The distributed parallel processing architecture of the services gateways includes multiple processors to manage sessions and run security and other services processing. This architecture provides greater flexibility and allows for high throughput and fast performance.



**NOTE:** In SRX1400, SRX3400, SRX3600, SRX5600, and SRX5800 devices, IKE negotiations involving NAT traversal do not work if the IKE peer is behind a NAT device that will change the source IP address of the IKE packets during the negotiation. For example, if the NAT device is configured with DIP, it changes the source IP because the IKE protocol switches the UDP port from 500 to 4500.

The SRX5600 and SRX5800 Services Gateways include I/O cards (IOC) and Services Processing Cards (SPCs) that each contain processing units that process a packet as it traverses the device. A Network Processing Unit (NPU) runs on an IOC. An IOC has one or more NPUs. One or more Services Processing Units (SPUs) run on an SPC.

These processing units have different responsibilities. All flow-based services for a packet are executed on a single SPU. Otherwise, however, the lines are not clearly divided in regard to the kinds of services that run on these processors. (For details on flow-based processing, see “Juniper Networks Devices Processing Overview” on page 3.)

For example:

- An NPU processes packets discretely. It performs sanity checks and applies some screens that are configured for the interface, such as denial-of-service (DoS) screens, to the packet.
- An SPU manages the session for the packet flow and applies security features and other services to the packet. It also applies packet-based stateless firewall filters, classifiers, and traffic shapers to the packet.
- The system uses one processor as a central point to take care of arbitration and allocation of resources and distribute sessions in an intelligent way. The central point assigns an SPU to be used for a particular session when the first packet of its flow is processed.

These discrete, cooperating parts of the system, including the central point, each store the information identifying whether a session exists for a stream of packets and the information against which a packet is matched to determine if it belongs to an existing session.

This architecture allows the device to distribute processing of all sessions across multiple SPUs. It also allows an NPU to determine if a session exists for a packet, to check the packet, and to apply screens to it. How a packet is handled depends on whether it is the first packet of a flow.

The following sections describe the SRX5600 and SRX5800 processing architecture:

- Understanding First-Packet Processing on page 30
- Understanding Fast-Path Processing on page 31
- Understanding the Data Path for Unicast Sessions on page 32
- Understanding Packet Processing on page 38
- Understanding Services Processing Units on page 39

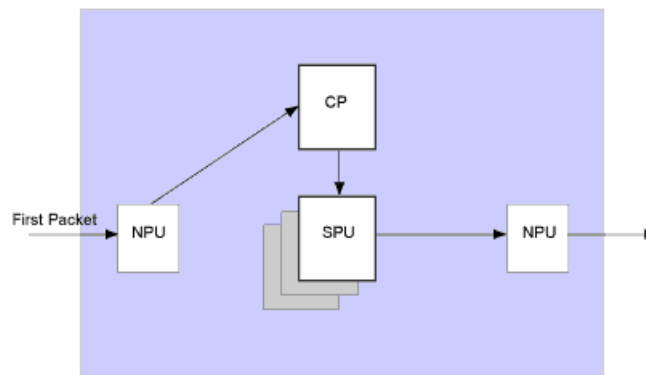
- Understanding Scheduler Characteristics on page 39
- Understanding Network Processor Bundling on page 39

## Understanding First-Packet Processing

If the packet matches an existing flow, processing for the packet is assessed in the context of its flow state. The SPU maintains the state for each session, and the settings are then applied to the rest of the packets in the flow. If the packet does not match an existing flow, it is used to create a flow state and a session is allocated for it.

Figure 2 on page 30 illustrates the path the first packet of a flow takes as it enters the device: the NPU determines that no session exists for the packet, and the NPU sends the packet to the central point; the central point selects the SPU to set up the session for the packet and process it, and it sends the packet to that SPU. The SPU processes the packet and sends it to the NPU for transmission from the device. (This high-level description does not address application of features to a packet.)

**Figure 2: First-Packet Processing**



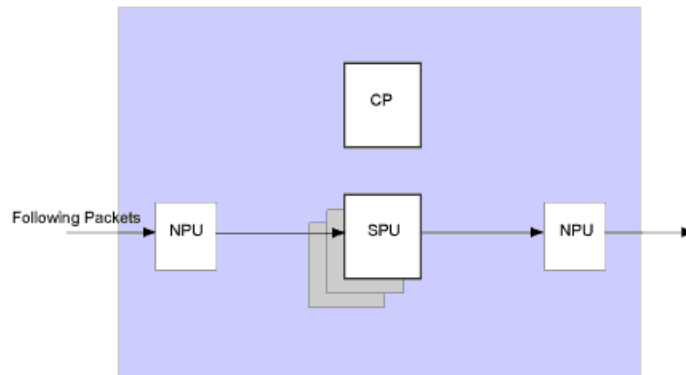
For details on session creation for the first packet in a flow, see “Understanding Session Creation: First-Packet Processing” on page 32.

After the first packet of a flow has traversed the system and a session has been established for it, it undergoes fast-path processing.

Subsequent packets of the flow also undergo fast-path processing; in this case, after each packet enters the session and the NPU finds a match for it in its session table, the NPU forwards the packet to the SPU that manages its session.

Figure 3 illustrates fast-path processing. This is the path a packet takes when a flow has already been established for its related packets. (It is also the path that the first packet of a flow takes after the session for the flow that the packet initiated has been set up.) After the packet enters the device, the NPU finds a match for the packet in its session table, and it forwards the packet to the SPU that manages the packet’s session. Note that the packet bypasses interaction with the central point.

Figure 3: Fast-Path Processing



This section explains how a session is created and the process a packet undergoes as it transits the device.

### Understanding Fast-Path Processing

Here is an overview of the main components involved in setting up a session for a packet and processing the packets both discretely and as part of a flow as they transit the SRX5600 and SRX5800 devices:

- Network Processing Units (NPUs)—NPUs reside on I/O cards. They handle packet sanity checking and application of some screens. NPUs maintain session tables that they use to determine if a session exists for an incoming packet or for reverse traffic.

The NPU session table contains an entry for a session if the session is established on an SPU for a packet that had previously entered the device via the interface and was processed by this NPU. The SPU installs the session in the NPU table when it creates the session.

An NPU determines if a session exists for a packet by checking the packet information against its session table. If the packet matches an existing session, the NPU sends the packet and the metadata for it to the SPU. If there is no session, the NPU sends the packet to the central point for SPU assignment.

- Services Processing Units (SPUs)—The main processors of the SRX5600 and SRX5800 devices reside on Services Processing Cards (SPCs). They establish and manage traffic flows and perform most of the packet processing on a packet as it transits the device. Each SPU maintains a hash table for fast session lookup. The SPU applies stateless firewall filters, classifiers, and traffic shapers to traffic. An SPU performs all flow-based processing for a packet and most packet-based processing. Each multicore SPU processes packets independently with minimum interaction among SPUs on the same or different SPC. All packets that belong to the same flow are processed by the same SPU.

The SPU maintains a session table with entries for all sessions that it established and whose packets it processes. When an SPU receives a packet from an NPU, it checks its session table to ensure that the packet belongs to it. It also checks its session table when it receives a packet from the central point and a message to establish a session for that packet to verify that there is not an existing session for the packet.

- Central point—The SRX Series device uses one processor as a central point to take care of arbitration and allocation of resources and distribute sessions in an intelligent way to avoid multiple SPUs from wrongly handling the same flow.

The central point's main function is to delegate session processing to one of the SPUs. If the session has not yet been established, the central point selects an SPU to establish the session for the flow, based on load- balancing criteria. If the session already exists, the central point forwards packets for that flow to the SPU hosting it. It also redirects packets to the correct SPU in the event that the NPU fails to do so.

The central point maintains a global session table with information about the owner SPU of a particular session. It functions as a central repository and resource manager for the whole system.

- Routing Engine (RE)—The Routing Engine runs the control plane.

## Understanding the Data Path for Unicast Sessions

This topic describes the process of establishing a session for packets belonging to a flow that transits the device.

To illustrate session establishment and the packet “walk” including the points at which services are applied to the packets of a flow, this example uses the simple case of a unicast session.

This packet “walk” brings together the packet-based processing and flow-based processing that the Junos OS performs on the packet.

### Session Lookup and Packet Match Criteria

---

To determine if a packet belongs to an existing flow, the device attempts to match the packet's information to that of an existing session based on the following six match criteria:

- Source address
- Destination address
- Source port
- Destination port
- Protocol
- Unique token from a given zone and virtual router

### Understanding Session Creation: First-Packet Processing

---

This topic explains how a session is set up to process the packets composing a flow. To illustrate the process, this topic uses an example with a source “a” and a destination “b”. The direction from source to destination for the packets of the flow is referred to as (a ->b). The direction from destination to source is referred to as (b->a).

#### ***Step 1. A Packet Arrives at an Interface on the Device and the NPU Processes It.***

This topic describes how a packet is handled when it arrives at an SRX Series device ingress IOC.

1. The packet arrives at the device's IOC and is processed by the NPU on the card.
2. The NPU performs basic sanity checks on the packet and applies some screens configured for the interface to the packet.
3. The NPU checks its session table for an existing session for the packet. (It checks the packet's tuple against those of packets for existing sessions in its session table.)
  - a. If no existent session is found, the NPU forwards the packet to the central point.
  - b. If a session match is found, the session has already been created on an SPU that was assigned to it, so the NPU forwards the packet to the SPU for processing along with the session ID. (See "Understanding Fast-Path Processing" on page 35.)

**Example:** Packet (a ->b) arrives at NPUI. NPUI performs sanity checks and applies DoS screens to the packet. NPU checks its session table for a tuple match and no existing session is found. NPUI forwards the packet to the central point for assignment to an SPU.

***Step 2. The Central Point Creates a Session with a "Pending" State.***

The central point maintains a global session table that includes entries for all sessions that exist across all SPUs on the device. It participates in session creation and delegates and arbitrates session resources allocation.

This process entails the following parts:

1. The central point checks its session table and gate table to determine if a session or a gate exists for the packet it receives from the NPU. (An NPU has forwarded a packet to the central point because its table indicates there is no session for it. The central point verifies this information before allocating an SPU for the session.)
2. If there is no entry that matches the packet in either table, the central point creates a pending wing for the session and selects an SPU to be used for the session, based on its load-balancing algorithm.
3. The central point forwards the first packet of the flow to the selected SPU in a message telling it to set up a session locally to be used for the packet flow.

**Example:** The central point creates pending wing (a ->b) for the session. It selects SPUI to be used for it. It sends SPUI the (a->b) packet along with a message to create a session for it.

***Step 3. The SPU Sets Up the Session.***

Each SPU, too, has a session table, which contains information about its sessions. When the SPU receives a message from the central point to set up a session, it checks its session table to ensure that a session does not already exist for the packet.

1. If there is no existing session for the packet, the SPU sets up the session locally.
2. The SPU sends a message to the central point telling it to install the session.



**NOTE:** During first-packet processing, if NAT is enabled, the SPU allocates IP address resources for NAT. In this case, the first-packet processing for the session is suspended until the NAT allocation process is completed.

The SPU adds to the queue any additional packets for the flow that it might receive until the session has been installed.

**Example:** SPU1 creates the session for (a->b) and sends a message back to the central point telling it to install the pending session.

***Step 4. The Central Point Installs the Session.***

The central point receives the install message from the SPU.

1. It sets the state for the session's pending wing to active.
2. It installs the reverse wing for the session as an active wing.
3. It sends an ACK (acknowledge) message to the SPU, indicating that the session is installed.

**Example:** The central point receives a message from SPU1 to install the session for (a->b). It sets the session state for (a->b) wing to active. It installs the reverse wing (b->a) for the session and makes it active; this allows for delivery of packets from the reverse direction of the flow: destination (b) to be delivered to the source (a).

***Step 5. The SPU Sets Up the Session on the Ingress and Egress NPUs.***

NPUs maintain information about a session for packet forwarding and delivery. Session information is set up on the egress and ingress NPUs (which sometimes are the same) so that packets can be sent directly to the SPU that manages their flows and not to the central point for redirection.

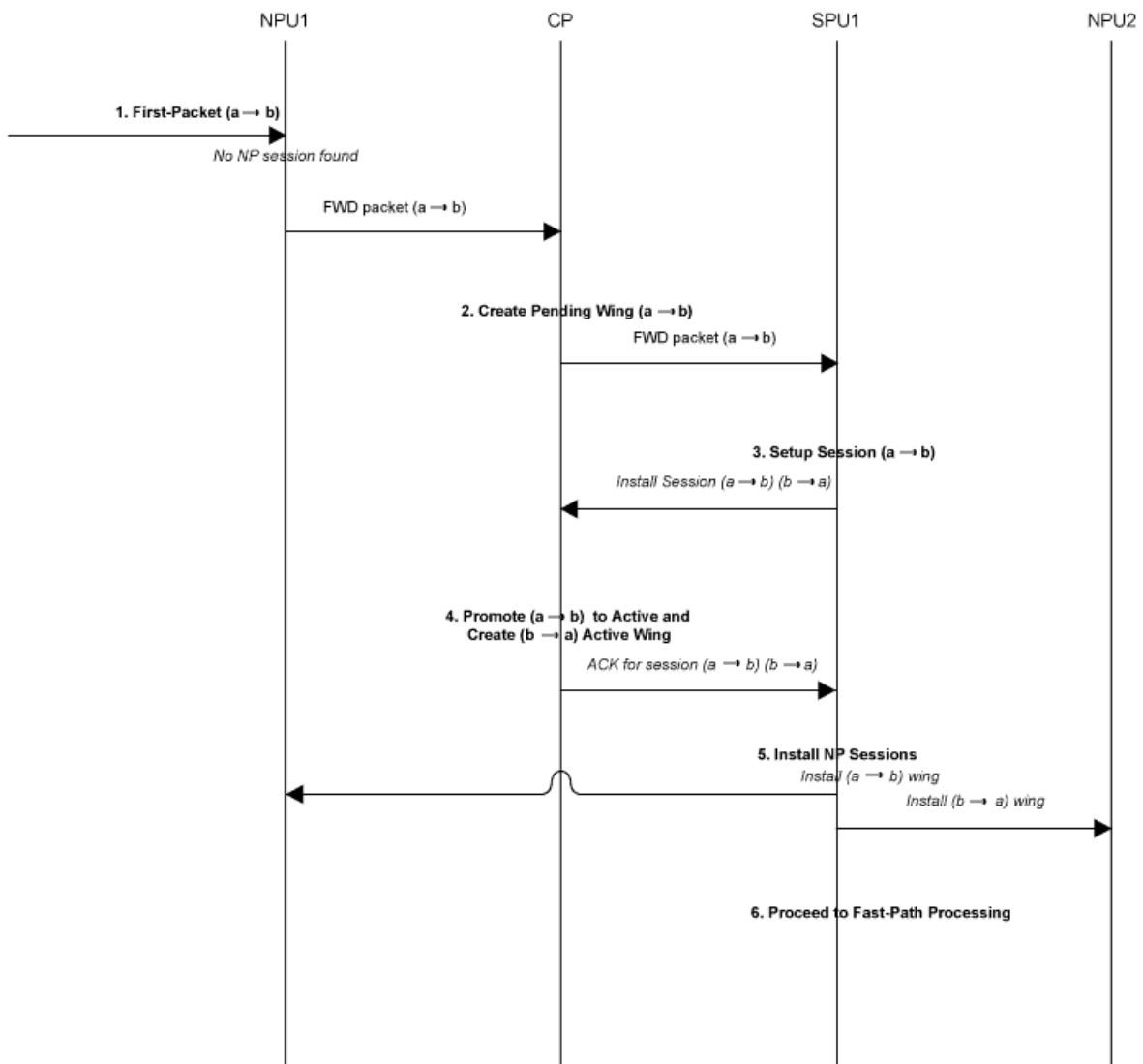
***Step 6. Fast-Path Processing Takes Place.***

For the remainder of the steps entailed in packet processing, proceed to Step 1 in "Understanding Fast-Path Processing" on page 35.

Figure 4 on page 35 illustrates the first part of the process the first packet of a flow undergoes after it reaches the device. At this point a session is set up to process the packet and the rest of the packets belonging to its flow. Subsequently, it and the rest of the packets of flow undergo fast-path processing.



Figure 4: Session Creation: First-Packet Processing



### Understanding Fast-Path Processing

All packets undergo fast-path processing. However, if a session exists for a packet, the packet undergoes fast-path processing and bypasses the first-packet process. When there is already a session for the packet's flow, the packet does not transit the central point.

Here is how fast-path processing works: NPUs at the egress and ingress interfaces contain session tables that include the identification of the SPU that manages a packet's flow. Because the NPUs have this session information, all traffic for the flow, including reverse traffic, is sent directly to that SPU for processing.

To illustrate the fast-path process, this topic uses an example with a source "a" and a destination "b". The direction from source to destination for the packets of the flow is referred to as (a->b). The direction from destination to source is referred to as (b->a).

**Step 1. A Packet Arrives at the Device and the NPU Processes It.**

This topic describes how a packet is handled when it arrives at a service gateway's IOC.

1. The packet arrives at the device's IOC and is processed by the NPU on the card.  
The NPU performs sanity checks and applies some screens, such as denial-of-service (DoS) screens, to the packet.
2. The NPU identifies an entry for an existing session in its session table that the packet matches.
3. The NPU forwards the packet along with metadata from its session table, including the session ID and packet tuple information, to the SPU that manages the session for the flow, applies stateless firewall filters and CoS features to its packets, and handles the packet's flow processing and application of security and other features.

**Example:** Packet (a ->b) arrives at NPU1. NPU1 performs sanity checks on the packet, applies DoS screens to it, and checks its session table for a tuple match. It finds a match and that a session exists for the packet on SPU1. NPU1 forwards the packet to SPU1 for processing.

**Step 2. The SPU for the Session Processes the Packet.**

Most of a packet's processing occurs on the SPU to which its session is assigned. The packet is processed for packet-based features such as stateless firewall filters, traffic shapers, and classifiers, if applicable. Configured flow-based security and related services such as firewall features, NAT, ALGs, and so forth, are applied to the packet. (For information on how security services are determined for a session, see "Juniper Networks Devices Processing Overview" on page 3.)

1. Before it processes the packet, the SPU checks its session table to verify that the packet belongs to one of its sessions.
2. The SPU processes the packet for applicable features and services.

**Example:** SPU1 receives packet (a->b) from NPU1. It checks its session table to verify that the packet belongs to one of its sessions. Then it processes packet (a->b) according to input filters and CoS features that apply to its input interface. The SPU applies the security features and services that are configured for the packet's flow to it, based on its zone and policies. If any are configured, it applies output filters, traffic shapers and additional screens to the packet.

**Step 3. The SPU Forwards the Packet to the NPU.**

1. The SPU forwards the packet to the NPU.
2. The NPU applies any applicable screens associated with the interface to the packet.

**Example:** SPU1 forwards packet (a->b) to NPU2, and NPU2 applies DoS screens.

**Step 4. The Interface Transmits the Packet From the Device.**

**Example:** The interface transmits packet (a->b) from the device.

***Step 5. A Reverse Traffic Packet Arrives at the Egress Interface and the NPU Processes It.***

This step mirrors Step 1 exactly in reverse. See Step 1 in this topic for details.

**Example:** Packet (b->a) arrives at NPU2. NPU2 checks its session table for a tuple match. It finds a match and that a session exists for the packet on SPU1. NPU2 forwards the packet to SPU1 for processing.

***Step 6. The SPU for the Session Processes the Reverse Traffic Packet.***

This step is the same as Step 2 except that it applies to reverse traffic. See Step 2 in this topic for details.

**Example:** SPU1 receives packet (b->a) from NPU2. It checks its session table to verify that the packet belongs to the session identified by NPU2. Then it applies packet-based features configured for the NPU1's interface to the packet. It processes packet (b->a) according to the security features and other services that are configured for its flow, based on its zone and policies. (See "Juniper Networks Devices Processing Overview" on page 3.)

***Step 7. The SPU Forwards the Reverse Traffic Packet to the NPU.***

This step is the same as Step 3 except that it applies to reverse traffic. See Step 3 in this topic for details.

**Example:** SPU1 forwards packet (b->a) to NPU1. NPU1 processes any screens configured for the interface.

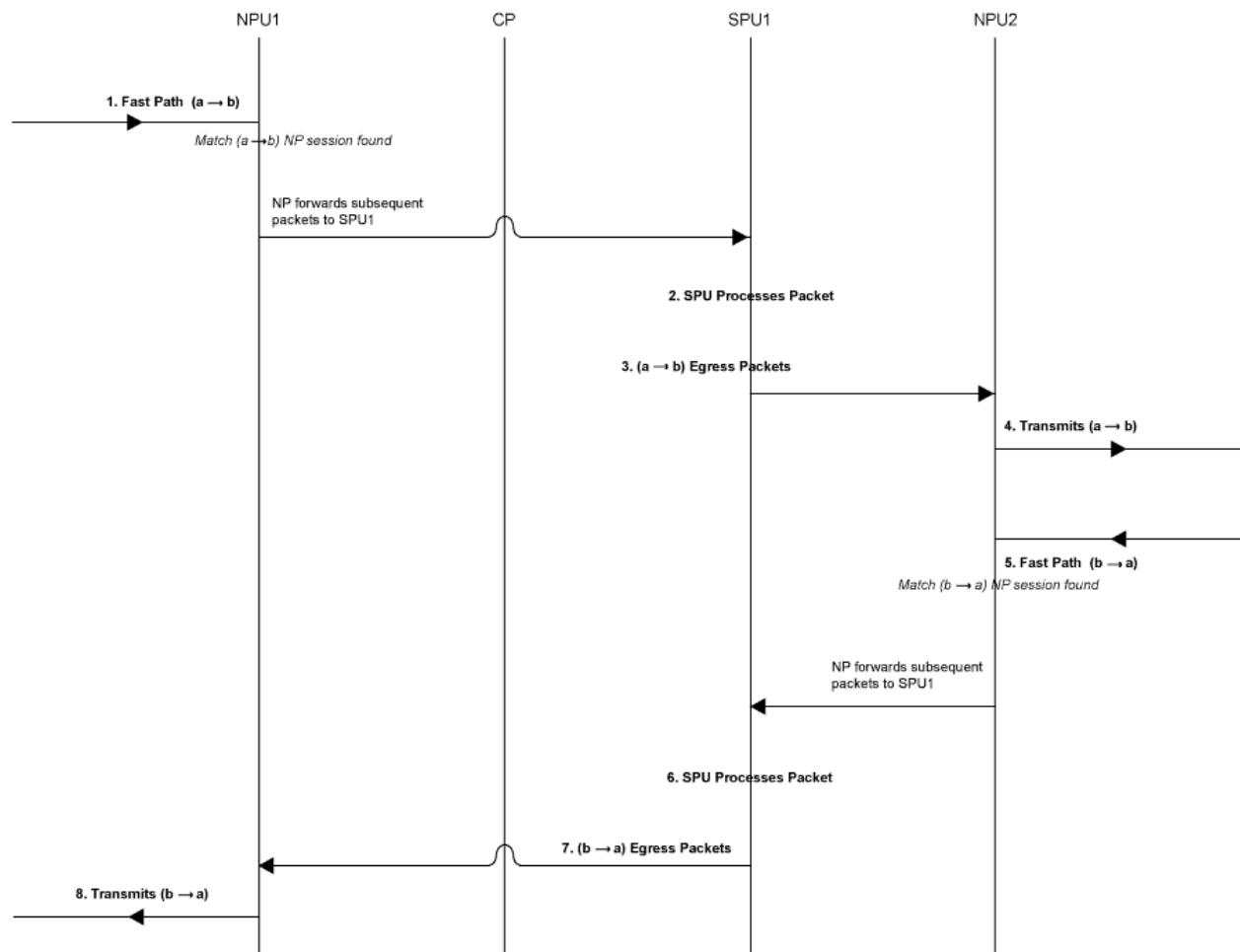
***8. The Interface Transmits the Packet From the Device.***

This step is the same as Step 4 except that it applies to reverse traffic. See Step 4 in this topic for details.

**Example:** The interface transmits packet (b->a) from the device.

Figure 5 on page 38 illustrates the process a packet undergoes when it reaches the device and a session exists for the flow that the packet belongs to.

Figure 5: "Packet Walk" for Fast-Path Processing



## Understanding Packet Processing

This topic explains how a session is set up to process the packets composing a flow:

1. The primary network processor receives the IP packet and analyzes it to obtain several key tuples.
2. The interface is configured to enable distributed flow lookup and the packet is forwarded to a secondary network processor by referring a 5-tuple hash algorithm using the hash value of the primary network processor.
3. The secondary network processor receives the forwarded packet and performs all the tasks for a packet.



**NOTE:** When network processor bundling is enabled, screen settings are programmed into each network processor in the bundle, the settings chosen will affect the screen performance. For example, if network processor bundling is not enabled, and an ICMP screen is set to 100, ICMP packets will be blocked when the processing rate exceeds 100 pps. However, if network processor bundling is enabled and four network processors are bundled together (one primary network processor and three secondary network processors), the ICMP screen would not begin until the ICMP flood exceeds 300 pps, if the ICMP flooding packets are distributed evenly among the three secondary network processors.

## Understanding Services Processing Units

For a given physical interface, the Services Processing Unit (SPU) receives ingress packets from all network processors of the network processor bundle associated to the physical interface. The SPU extracts network processor bundle information from the physical interface and uses the same 5-tuple hash algorithm to map a flow to a network processor index. To determine the network processor, the SPU does a lookup on the network processor index in the network processor bundle. The SPU sends egress packets to the physical interface's local PIC for the outward traffic.



**NOTE:** The network processor and the SPU use the same 5-tuple hash algorithm to get the hash values for the packets.

## Understanding Scheduler Characteristics

For SRX5600 and SRX5800 devices, the IOC supports the following hierarchical scheduler characteristics:

- IFL – The configuration of the network processor bundle is stored in the physical interface data structure. The SRX5600 and SRX5800 devices have a maximum of 48 PICs. The physical interface can use a 48-bit bit-mask to indicate the PIC, or the network processor traffic from this physical interface is distributed in addition to the physical interface's primary network processor. On SRX1400, SRX3400, SRX3600, SRX5600, and SRX5800 devices, the iflset functionality is not supported for aggregated interfaces like *reth*.
- IFD – The logical interface associated to the physical interface of a network processor bundle is passed to all the I/O cards (IOCs) that have a PIC in the network processor bundle.

## Understanding Network Processor Bundling

The network processor bundling feature is available on SRX5600 and SRX5800 Services Gateways. This feature enables distribution of data traffic from one interface to multiple network processors for packet processing. A primary network processor is assigned for an interface that receives the ingress traffic and distributes the packets to several other secondary network processors. A single network processor can act as a primary network

processor or a secondary network processor to multiple interfaces. A single network processor can join only one network processor bundle.

### Network Processor Bundling Limitations

Network processor bundling functionality has the following limitations:

- Network processor bundling allows a total of 16 PICs per bundle and eight different network processor bundles system.
- You need to reboot the device to apply the configuration changes on the bundle.
- Network processor bundling is below the reth interface in the overall architecture. You can choose one or both the interfaces from the network processor bundling to form the reth interface.
- If the IOC is removed from a network processor bundle, the packets forwarded to the PIC on that IOC is lost.
- When the network processor bundle is enabled, the ICMP, UDP and TCP sync flooding thresholds no longer apply to an interface. Packets are distributed to multiple network processors for processing. These thresholds will apply to each network processor in the network processor bundle.
- Network processor bundle is not supported in the Layer 2 mode.
- Due to memory constraints on the EZchip, the number of network processor bundled ports that are supported per PIC is limited. Within the network processor bundle, each port needs to have a global port index. The global port index is calculated using the following formula:  
$$\text{Global\_port\_index} = (\text{global\_pic} * 16) + \text{port\_offset}$$
- Link aggregation groups (LAGs) and redundant Ethernet interface LAGs in chassis cluster implementations can coexist with network processor bundling. However, neither LAGs nor redundant Ethernet interface LAGs can overlap with or share physical links with a network processor bundle.

#### **Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Data Path Debugging for SRX Series Devices on page 21](#)
- [Juniper Networks Devices Processing Overview on page 3](#)
- [Understanding Session Characteristics for SRX Series Services Gateways on page 7](#)
- [Security Policy Schedulers Overview on page 181](#)

---

## SRX1400, SRX3400, and SRX3600 Services Gateways Processing Overview

---

Junos OS for the SRX3400 and SRX3600 Services Gateways integrates the world-class network security and routing capabilities of Juniper networks. Junos OS for these service gateways includes the wide range of security services including policies, screens, network address translation, class-of-service classifiers, and the rich, extensive set of flow-based services that are also supported on the other devices in the services gateways

The distributed parallel processing architecture of the SRX3400 and SRX3600 devices includes multiple processors to manage sessions and run security and other services processing. This architecture provides greater flexibility and allows for high throughput and fast performance.

This topic includes the following information:

- Components Involved in Setting up a Session on page 41
- Understanding the Data Path for Unicast Sessions on page 42
- Session Lookup and Packet Match Criteria on page 42
- Understanding Session Creation: First Packet Processing on page 42
- Understanding Fast-Path Processing on page 44

### Components Involved in Setting up a Session

Here is an overview of the main components involved in setting up a session for a packet and processing the packets as they transit the SRX3400 and SRX3600 devices:

- Services Processing Units (SPUs)—The main processors of the SRX3400 and SRX3600 devices reside on Services Processing Cards (SPCs). They establish and manage traffic flows and perform most of the packet processing on a packet as it transits the device. Each SPU maintains a hash table for fast session lookup. The SPU performs all flow-based processing for a packet, including application of security services, classifiers, and traffic shapers. All packets that belong to the same flow are processed by the same SPU.

The SPU maintains a session table with entries for all sessions that it established and whose packets it processes. When an SPU receives a packet from an NPU, it checks its session table to ensure that the packet belongs to it.

For SRX3400 and SRX3600 devices, one SPU acts in concert performing its regular session management and flow processing functions and acting as a central point in which it arbitrates sessions and allocates resources. When an SPU performs in this manner it is said to be in combo mode.

- Central Point—The central point is used to allocate session management to SPUs based on load balancing criteria. It distributes sessions in an intelligent way to avoid occurrences in which multiple SPUs might wrongly handle the same flow. The central point follows load balancing criteria in allocating sessions to SPUs. If the session exists, the central point forwards packets for that flow to the SPU hosting it. It also redirects packets to the correct SPU in the event that the NPU fails to do so.

For the SRX3400 and SRX3600 devices, one SPU always runs in what is referred to as combo-mode in which it implements both the functionality of the central point and the flow and session management functionality. In combo-mode, the SPU and the central point share the same load-balancing thread (LBT) and packet-ordering thread (POT) infrastructure. For more information, see “Understanding SRX Series Services Gateways Central Point Architecture” on page 25.

- Routing Engine (RE)—The routing engine runs the control plane and manages the Control Plane Processor (CPP).

## Understanding the Data Path for Unicast Sessions

Junos OS for the SRX3400 and SRX3600 Services Gateways is a distributed parallel processing high throughput and high performance system. This topic describes the process of establishing a session for packets belonging to a flow that transits the device.

To illustrate session establishment and the packet “walk” including the points at which services are applied to the packets of a flow, the following example uses the simple case of a unicast session. This packet “walk” brings together the packet-based processing and flow-based processing that the Junos OS performs on the packet.

## Session Lookup and Packet Match Criteria

To determine if a packet belongs to an existing flow, the device attempts to match the packet’s information to that of an existing session based on the following six match criteria:

- Source address
- Destination address
- Source port
- Destination port
- Protocol
- Unique token from a given zone and virtual router

## Understanding Session Creation: First Packet Processing

This topic explains how a session is set up to process the packets composing a flow. To illustrate the process, this topic uses an example with a source “a” and a destination “b”. The direction from source to destination for the packets of the flow is referred to as (a -> b). The direction from destination to source is referred to as (b -> a).

1. A packet arrives at an interface on the device and the IOC processes it.

The IOC dequeues the packet and sends it to the NPU with which it communicates.

2. The NPU receives the packet from the IOC and processes it.
  - The NPU performs basic sanity checks on the packet and applies some screens configured for the interface to the packet.



- If a session match is found, the session has already been created on an SPU that was assigned to it, so the NPU forwards the packet to the SPU for processing along with the session ID.

Example: Packet (a ->b) arrives at NPU1 from IOC1. NPU1 performs sanity checks and applies DoS screens to the packet. NPU1 checks its session table for a tuple match and no existing session is found. NPU1 forwards the packet to the central point on SPU1 for assignment to an SPU.

3. The central point creates a session with a "Pending" state.

The central point maintains a global session table that includes entries for all sessions that exist across all SPUs on the device. It participates in session creation and delegates and arbitrates session resources allocation.

This process entails the following parts:

- a. The central point checks its session table and gate table to determine if a session or a gate exists for the packet it receives from the NPU. (An NPU has forwarded a packet to the central point because its table indicates there is no session for it. The central point verifies this information before allocating an SPU for the session.)
- b. If there is no entry that matches the packet in either table, the central point creates a pending wing for the session and selects an SPU to be used for the session, based on its load-balancing algorithm.
- c. The central point forwards the first packet of the flow to the selected SPU in a message telling it to set up a session locally to be used for the packet flow.

Example: The central point creates pending wing (a ->b) for the session. It selects SPU1 to be used for the session. It sends SPU1 the (a->b) packet along with a message to create a session for it. (It happens to be the case that SPU1 is the SPU that runs in combo mode. Therefore, its session-management and flow-processing services are used for the session.

4. The SPU sets up the session.

Each SPU, too, has a session table, which contains information about its sessions. When the SPU receives a message from the central point to set up a session, it checks its session table to ensure that a session does not already exist for the packet.

- a. If there is no existing session for the packet, the SPU sets up the session locally.
- b. The SPU sends a message to the central point, telling it to install the session.

During first-packet processing, if NAT is enabled, the SPU allocates IP address resources for NAT. In this case, the first-packet processing for the session is suspended until the NAT allocation process is completed.

The SPU adds to the queue any additional packets for the flow that it might receive until the session has been installed.

Example: SPU1 creates the session for (a ->b) and sends a message back to the central point (implemented on the same SPU) telling it to install the pending session.

5. The central point installs the session.

- It sets the state for the session's pending wing to active.
- It installs the reverse wing for the session as an active wing.
- It sends an ACK (acknowledge) message to the SPU, indicating that the session is installed.

Example: The central point receives a message from SPU1 to install the session for (a->b). It sets the session state for (a->b) wing to active. It installs the reverse wing (b->a) for the session and makes it active; this allows for delivery of packets from the reverse direction of the flow: destination (b) to be delivered to the source (a).

6. The SPU sets up the session on the ingress and egress NPUs.

NPUs maintain information about a session for packet forwarding and delivery. Session information is set up on the egress and ingress NPUs (which sometimes are the same) so that packets can be sent directly to the SPU that manages their flows and not to the central point for redirection.

7. Fast-path processing takes place.

For the remainder of the steps entailed in packet processing, proceed to Step 1 in "Understanding Fast-Path Processing".

## Understanding Fast-Path Processing

All packets undergo fast-path processing. However, if a session exists for a packet, the packet undergoes fast-path processing and bypasses the first-packet process. When there is already a session for the packet's flow, the packet does not transit the central point.

Here is how fast-path processing works: NPUs at the egress and ingress interfaces contain session tables that include the identification of the SPU that manages a packet's flow. Because the NPUs have this session information, all traffic for the flow, including reverse traffic, is sent directly to that SPU for processing.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Data Path Debugging for SRX Series Devices](#) on page 21
- [Juniper Networks Devices Processing Overview](#) on page 3
- [Understanding Session Characteristics for SRX Series Services Gateways](#) on page 7

---

## SRX210 Services Gateway Processing Overview

This topic describes the process that the SRX210 Services Gateway undertakes in establishing a session for packets belonging to a flow that transits the device. The flow services of the SRX210 device are single-threaded and non-distributed. Although it differs from the other SRX Series devices in this respect, the same flow model is followed and the same command line interface (CLI) is implemented.

To illustrate session establishment and the packet “walk” including the points at which services are applied to the packets of a flow, the example described in the following sections uses the simple case of a unicast session:

- Understanding Flow Processing and Session Management on page 45
- Understanding First-Packet Processing on page 45
- Understanding Session Creation on page 45
- Understanding Fast-Path Processing on page 46

## Understanding Flow Processing and Session Management

This topic explains how a session is set up to process the packets composing a flow. In the following topic, the SPU refers to the data plane thread of the SRX210 Services Gateway.

At the outset, the data plane thread fetches the packet and performs basic sanity checks on it. Then it processes the packet for stateless filters and CoS classifiers and applies some screens.

## Understanding First-Packet Processing

To determine if a packet belongs to an existing flow, the device attempts to match the packet’s information to that of an existing session based on the following six match criteria:

- Source address
- Destination address
- Source port
- Destination port
- Protocol
- Unique token from a given zone and virtual router

The SPU checks its session table for an existing session for the packet. If no existent session is found, the SPU sets up a session for the flow. If a session match is found, the session has already been created, so the SPU performs fast-path processing on the packet.

## Understanding Session Creation

In setting up the session, the SPU executes the following services for the packet:

- Screens
- Route lookup
- Policy lookup
- Service lookup
- NAT, if required

After a session is set up, it is used for all packets belonging to the flow. Packets of a flow are processed according to the parameters of its session. For the remainder of the steps entailed in packet processing, proceed to Step 1 in “Fast-Path Processing”. All packets undergo fast-path processing.

## Understanding Fast-Path Processing

If a packet matches a session, Junos OS performs fast-path processing as described in the following steps. After a session has been set up for the first packet in a flow, also undergoes fast-path processing. All packets undergo fast-path processing.

1. The SPU applies flow-based security features to the packet.
  - Configured screens are applied.
  - TCP checks are performed.
  - Flow services, such as NAT, ALG, and IPsec are applied, if required.
2. The SPU prepares the packet for forwarding and transmits it.
  - Routing packet filters are applied.
  - Traffic shaping is applied.
  - Traffic prioritizing is applied.
  - Traffic scheduling is applied.
  - The packet is transmitted.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Data Path Debugging for SRX Series Devices on page 21](#)
- [Juniper Networks Devices Processing Overview on page 3](#)
- [Understanding Session Characteristics for SRX Series Services Gateways on page 7](#)

## CHAPTER 2

# Understanding IPv6 Flow-Based Processing

This chapter explains how SRX Series Services Gateway and J-series devices handle flow-based processing for IP version 6 (IPv6) packets. To facilitate understanding of IPv6 flow processing for these devices, this chapter gives an overview of IPv6, including its address space and addressing. It also introduces the architecture for the SRX5600 and SRX5800 devices and uses it as a model to explain IPv6 flow processing. Flow processing is similar on other devices.

A flow is a stream of related packets that meet the same matching criteria and share the same characteristics. Junos OS treats packets belonging to the same flow in the same manner.

This chapter includes the following topics:

- Understanding IP Version 6 (IPv6) on page 48
- About the IPv6 Address Space, Addressing, and Address Types on page 48
- About IPv6 Address Types and How Junos OS for SRX Series Services Gateway and J-series Devices Use Them on page 49
- About the IPv6 Address Format on page 50
- The IPv6 Packet Header and SRX Series and J-series Devices Overview on page 51
- About the IPv6 Basic Packet Header on page 52
- Understanding IPv6 Packet Header Extensions on page 54
- About IPv6 Packet Header Verification Performed by the Flow Module for SRX Series and J-series Devices on page 55
- IPv6 Advanced Flow on page 55
- Understanding IPv6 Dual-Stack Lite on page 57
- Understanding How SRX Series and J-series Devices Handle ICMPv6 Packets on page 59
- Understanding Path MTU Messages for IPv6 Packets on page 61
- Understanding How SRX Series and J-series Devices Handle Packet Fragmentation for IPv6 Flows on page 62
- Understanding Sessions for IPv6 Flows on page 63
- Understanding SRX5600 and SRX5800 Architecture and Flow Processing on page 63

- Enabling Flow-Based Processing for IPv6 Traffic on page 66
- Using Filters to Display IPv6 Session and Flow Information for SRX Series Services Gateways on page 67
- IPv6 NAT on page 71
- IPv6 ALGs on page 88

## Understanding IP Version 6 (IPv6)

---

This topic gives an overview of IP version 6 (IPv6), including its uses and benefits.

The ongoing expansive growth of the Internet and the need to provide IP addresses to accommodate it—to support increasing numbers of new users, computer networks, Internet-enabled devices, and new and improved applications for collaboration and communication—is escalating the emergent use of a new IP protocol. IPv6, with its robust architecture, was designed to satisfy these current and anticipated near future requirements.

IP version 4 (IPv4) is widely used throughout the world today for the Internet, intranets, and private networks. IPv6 builds upon the functionality and structure of IPv4 in the following ways:

- Provides a simplified and enhanced packet header to allow for more efficient routing.
- Improves support for mobile phones and other mobile computing devices.
- Enforces increased, mandatory data security through IPsec (which was originally designed for it).
- Provides more extensive quality-of-service (QoS) support.

### Related Documentation

- About the IPv6 Address Space, Addressing, and Address Types on page 48
- About the IPv6 Address Format on page 50
- About the IPv6 Basic Packet Header on page 52
- Understanding IPv6 Packet Header Extensions on page 54

## About the IPv6 Address Space, Addressing, and Address Types

---

This topic explains IP version 6 (IPv6) addressing and identifies its three types of addresses.

Addressing is the area where most of the differences between IP version 4 (IPv4) and IPv6 exist, but the changes are largely about the ways in which addresses are implemented and used. IPv6 has a vastly larger address space than the impending exhausted IPv4 address space. IPv6 increases the size of the IP address from the 32 bits that compose an IPv4 address to 128 bits. Each extra bit given to an address doubles the size of the address space.

IPv4 has been extended using techniques such as Network Address Translation (NAT), which allows for ranges of private addresses to be represented by a single public address, and temporary address assignment. Although useful, these techniques fall short of the requirements of novel applications and environments such as emerging wireless technologies, always-on environments, and Internet-based consumer appliances.

In addition to the increased address space, IPv6 addresses differ from IPv4 addresses in the following ways:

- Includes a scope field that identifies the type of application that the address pertains to
- Does not support broadcast addresses, but instead uses multicast addresses to broadcast a packet
- Defines a new type of address, called anycast

#### Related Documentation

- About the IPv6 Address Format on page 50
- About IPv6 Address Types and How Junos OS for SRX Series Services Gateway and J-series Devices Use Them on page 49
- About the IPv6 Basic Packet Header on page 52
- Understanding IPv6 Packet Header Extensions on page 54
- Understanding IP Version 6 (IPv6) on page 48

## About IPv6 Address Types and How Junos OS for SRX Series Services Gateway and J-series Devices Use Them

This topic explains the types of IP version 6 (IPv6) addresses that Junos OS for SRX Series and J-series devices support and how they are used.

IP version 6 (IPv6) includes the following types of addresses:

- Unicast

A unicast address specifies an identifier for a single interface to which packets are delivered. Under IPv6, the vast majority of Internet traffic is foreseen to be unicast, and it is for this reason that the largest assigned block of the IPv6 address space is dedicated to unicast addressing. Unicast addresses include all addresses other than loopback, multicast, link-local-unicast, and unspecified.

For SRX Series and J-series devices, the flow module supports the following kinds of IPv6 unicast packets:

- Pass-through unicast traffic, including traffic from and to virtual routers. The device transmits pass-through traffic according to its routing table.
- Host-inbound traffic from and to devices directly connected to SRX Series and J Series interfaces. For example, host-inbound traffic includes logging, routing protocol, and management types of traffic. The flow module sends these unicast packets to the Routing Engine and receives them from it. Traffic is processed by the Routing

Engine instead of by the flow module, based on routing protocols defined for the Routing Engine.

The flow module supports all routing and management protocols that run on the Routing Engine. Some examples are OSPFv3, RIPng, TELNET, and SSH.

- Multicast

A multicast address specifies an identifier for a set of interfaces that typically belong to different nodes. It is identified by a value of 0xFF. IPv6 multicast addresses are distinguished from unicast addresses by the value of the high-order octet of the addresses.

The devices support only host-inbound and host-outbound multicast traffic. Host inbound traffic includes logging, routing protocols, management traffic, and so on.

- Anycast

An anycast address specifies an identifier for a set of interfaces that typically belong to different nodes. A packet with an anycast address is delivered to the nearest node, according to routing protocol rules.

There is no difference between anycast addresses and unicast addresses except for the subnet-router address. For an anycast subnet-router address, the low order bits, typically 64 or more, are zero. Anycast addresses are taken from the unicast address space.

The flow module treats anycast packets in the same way as it handles unicast packets. If an anycast packet is intended for the device, it is treated as host-inbound traffic, and it delivers it to the protocol stack which continues processing it.

**Related  
Documentation**

- About the IPv6 Address Format on page 50
- Understanding IP Version 6 (IPv6) on page 48
- About the IPv6 Basic Packet Header on page 52
- Understanding IPv6 Packet Header Extensions on page 54

## About the IPv6 Address Format

---

This topic explains the format for IP version 6 (IPv6) addresses, including how to compress them, and it gives some examples.

All IPv6 addresses are 128 bits long, written as 8 sections of 16 bits each. They are expressed in hexadecimal representation, so the sections range from 0 to FFFF. Sections are delimited by colons, and leading zeroes in each section may be omitted. If two or more consecutive sections have all zeroes, they can be collapsed to a double colon.

- IPv6 addresses have the following format in which each *xxxx* is a 16-bit hexadecimal value, and each *x* is a 4-bit hexadecimal value.

`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`

- Here is an example of an IPv6 address:



3FFE:0000:0000:0001:0200:F8FF:FE75:50DF

- For an IPv6 address that contains consecutive fields of leading zeros, you can omit the zeros from each section. If you take this approach, you can write the example address in the following way:

3FFE:0:0:1:200:F8FF:FE75:50DF

- For an IPv6 address that includes contiguous sections each of which contain zeros, you can compress the 16-bit groups of zeros to double colons (::) but you can use the double-colon delimiter only once within a single IPv6 address, as shown in the following example:

3FFE::1:200:F8FF:FE75:50DF

#### Related Documentation

- About the IPv6 Address Space, Addressing, and Address Types on page 48
- About IPv6 Address Types and How Junos OS for SRX Series Services Gateway and J-series Devices Use Them on page 49
- About the IPv6 Basic Packet Header on page 52
- Understanding IPv6 Packet Header Extensions on page 54
- Understanding IP Version 6 (IPv6) on page 48

## The IPv6 Packet Header and SRX Series and J-series Devices Overview

This topic identifies the IP version 6 (IPv6) packet header and its extensions and options.

Every IPv6 packet at a minimum has a basic packet header, 40 bytes (320 bits) long. They optionally may have extension headers.

For IPv6 packets, flow processing parses the extension headers and transport layer headers in the following way:

- If the software encounters a TCP, a UDP, an ESP, an AH, or an ICMPv6 header, it parses the header and assumes that the packet payload corresponds to the specified protocol type.
- If the software encounters a hop-by-hop header, a routing and destination header, or a fragment header, it continues to parse the next extension header.
- If it encounters the no-next-header extension header, the software detects that the packet is that of an unknown protocol (protocol equals 0).
- For other extension headers, the software parses the header and identifies the packet as belonging to the protocol indicated by the extension header.

#### Related Documentation

- About the IPv6 Address Space, Addressing, and Address Types on page 48
- About the IPv6 Address Format on page 50
- About the IPv6 Basic Packet Header on page 52
- Understanding IPv6 Packet Header Extensions on page 54

- Understanding IP Version 6 (IPv6) on page 48

## About the IPv6 Basic Packet Header

This topic identifies the IP version 6 (IPv6) basic packet header fields with their bit lengths and uses.

Header Name	Bit Length	Purpose
Version	4	Specifies that IP version 6 is used. The IPv6 version field contains a value of 6 indicating that IPv6 is used, as opposed to 4 for IP version 4.
Traffic Class	8	Allows source nodes or routers to identify different classes (or priorities for quality of service) for IPv6 packets. (This field replaces the IPv4 Type of Service field.)
Flow Label	20	Identifies the flow to which the packet belongs. Packets in a flow share a common purpose, or belong to a common category, as interpreted by external devices such as routers or destination hosts.  <b>NOTE:</b> For IPv6 flow-based packets, Junos OS for SRX Series Services Gateway devices and J-series devices does not use the flow label field.
Payload Length	16	Specifies the length of the IPv6 packet payload, or contents, expressed in octets.

Header Name	Bit Length	Purpose
Next Header	8	<p>Identifies the type of Internet Protocol for the header that immediately follows the IPv6 header. The Next Header field replaces the IPv4 Protocol field. It is an optional field.</p> <p>This protocol can be one of two types:</p> <ul style="list-style-type: none"> <li>• An IPv6 extension header. For example, if the device performs IP security on exchanged packets, the Next Header value is probably 50 (ESP extension header) or 51 (AH extension header). Extension headers are optional.</li> <li>• An upper-layer Protocol Data Unit (PDU). For example, the Next Header value could be 6 (for TCP), 17 (for UDP), or 58 (for ICMPv6).</li> </ul> <p>The flow module processes these headers sequentially within the context of a packet flow.</p> <p>If it encounters one of the following extension headers, the software parses it and regards the packet as a corresponding protocol packet.</p> <ul style="list-style-type: none"> <li>• Internet Control Message Protocol version 6 (ICMPv6)</li> <li>• Transport Control Protocol (TCP)</li> </ul> <p><b>NOTE:</b> The device checks the TCP header length as part of its sanity checks.</p> <ul style="list-style-type: none"> <li>• UDP</li> </ul> <p><b>NOTE:</b> The device checks the UDP length as part of its sanity checks.</p> <ul style="list-style-type: none"> <li>• Enhanced Security Protocol (ESP) or Authentication Header (AH)</li> </ul>
Hop Limit	8	<p>Specifies the maximum number of hops the packet can make after transmission from the host device. When the Hop Limit value is zero, the device drops the packet and generates an error message. (This field is similar to the Time to Live IPv4 field.)</p>
Source IP Address	128	<p>Identifies the host device, or interface on a node, that generated the IPv6 packet.</p>
Destination IP Address	128	<p>Identifies the host device, or interface on a node, to which the IPv6 packet is to be sent.</p> <p><b>NOTE:</b> The destination address may appear twice, the first instance after the hop limit following the source IP address and the second instance after the final extension header.</p>

**Related Documentation**

- Understanding IPv6 Packet Header Extensions on page 54
- About the IPv6 Address Space, Addressing, and Address Types on page 48
- About the IPv6 Address Format on page 50
- Understanding IP Version 6 (IPv6) on page 48

## Understanding IPv6 Packet Header Extensions

This topic defines IP version 6 (IPv6) packet header extensions.

IPv6 extension headers contain supplementary information used by network devices (such as routers, switches, and endpoint hosts) to decide how to direct or process an IPv6 packet. The length of each extension header is an integer multiple of 8 octets. This allows subsequent extension headers to use 8-octet structures.

Any header followed by an extension header contains a Next Header value that identifies the extension header type. Extension headers always follow the basic IPv6 header in order as shown in Table 8 on page 54:



**NOTE:** The destination IP address can appear twice, once after the hop-by-hop header and another after the last extension header.

**Table 8: IPv6 Extension Headers**

Header Name	Purpose
Hop-by-Hop Options	Specifies delivery parameters at each hop on the path to the destination host.  <b>NOTE:</b> A hop-by-hop option can appear only following the IPv6 basic header. If it is used, it should be the first extension header. It cannot appear after another extension header.
Destination Options	Specifies packet delivery parameters for either intermediate destination devices or the final destination host. When a packet uses this header, the Next Header value of the previous header must be 60.
Routing	Defines strict source routing and loose source routing for the packet. (With strict source routing, each intermediate destination device must be a single hop away. With loose source routing, intermediate destination devices can be one or more hops away.) When a packet uses this header, the Next Header value of the previous header must be 43.
Fragment	Specifies how to perform IPv6 fragmentation and reassembly services. When a packet uses this header, the Next Header value of the previous header must be 44.  A source node uses the fragment extension header to tell the destination node the size of the packet that was fragmented so that the destination node can reassemble the packet.
Authentication	Provides authentication, data integrity, and anti-replay protection. When a packet uses this header, the Next Header value of the previous header must be 51.
Encapsulating Security Payload	Provides data confidentiality, data authentication, and anti-replay protection for Encapsulated Security Payload (ESP) packets. When a packet uses this header, the Next Header value of the previous header must be 50.
Destination IP Address	Identifies the host device, or interface on a node, to which the IPv6 packet is to be sent.  <b>NOTE:</b> The destination address may appear twice, the first instance after the hop limit following the source IP address and the second instance after the final extension header.

- Related Documentation**
- About the IPv6 Basic Packet Header on page 52
  - About the IPv6 Address Space, Addressing, and Address Types on page 48
  - About the IPv6 Address Format on page 50
  - Understanding IP Version 6 (IPv6) on page 48

## About IPv6 Packet Header Verification Performed by the Flow Module for SRX Series and J-series Devices

This topic gives an overview of some of the IP version 6 (IPv6) packet header verification that the flow module for SRX Series and J-series devices performs.

To ensure the integrity of an IPv6 packet, the flow module performs the following sanity checks.

For all IPv6 packets, it checks the following parts of the header:

- TCP length
- UDP length
- Hop-by-hop extension to ensure that it follows the basic IPv6 header and does not come after another extension header
- That the IP data length error (IP length—total extension header length is not less than zero (<0))

In addition to these verifications, the software performs other standard checks such as verifying that the correct IP version is specified and that the length of the IP address is correct.

- Related Documentation**
- About the IPv6 Basic Packet Header on page 52
  - Understanding IPv6 Packet Header Extensions on page 54
  - About IPv6 Address Types and How Junos OS for SRX Series Services Gateway and J-series Devices Use Them on page 49
  - About the IPv6 Address Format on page 50
  - Understanding IP Version 6 (IPv6) on page 48

## IPv6 Advanced Flow

IPv6 advanced flow adds IPv6 support for firewall, NAT, NAT-PT, multicast (local link and transit), IPsec, IDP, JSF framework, TCP Proxy and Session manager on SRX Series and J Series devices. MIBs are not used in the IPv6 flow.

In order to avoid the impact on the current IPv4 environment, IPv6 security is used. If IPv6 security is enabled, extended sessions and gates are allocated. The existing address fields and gates are used to store the index of extended sessions or gates. If IPv6 security is disabled, IPv6 security-related resources are not allocated.

New logs are used for IPv6 flow traffic to prevent impact on performance in the existing IPv4 system.

The behavior and implementation of the IPv6 advanced flow are the same as those of IPv4 in most cases.

Some of the differences are explained below:

- **Header Parse** IPv6 advanced flow stops parsing the headers and interprets the packet as the corresponding protocol packet if it encounters the following extension headers:
  - TCP/UDP
  - ESP/AH
  - ICMPv6

IPv6 advanced flow continues parsing headers if it encounters the following extension headers:

- Hop-by-Hop
- Routing and Destination, Fragment

IPv6 advanced flow interprets the packets as an unknown protocol packet if it encounters the extension header **No Next Header**

- **Sanity Checks** IPv6 advanced flow supports the following sanity checks:
  - TCP length
  - UDP length
  - Hop-by-hop
  - IP data length error
  - Layer 3 sanity checks (for example, IP version and IP length)
  - **ICMPv6 Packets** In IPv6 advanced flow, the ICMPv6 packets share the same behavior as normal IPv6 traffic with the following exceptions:
    - Embedded ICMPv6 packet
    - Path MTU message
- **Host Inbound and Outbound Traffic** IPv6 advanced flow supports all route and management protocols running on the Routing Engine (RE), including OSPF v3, RIPng, Telnet, and SSH. Note that no flow label is used in the flow.
- **Tunnel Traffic** IPv6 advanced flow supports the following tunnel types:
  - IPv4 IPIP
  - IPv4 GRE

- IPv4 IPsec
- Dual-stack lite
- **Events and Logs** The following logs are for IPv6-related flow traffic:
  - RT\_FLOW\_IPVX\_SESSION\_DENY
  - RT\_FLOW\_IPVX\_SESSION\_CREATE
  - RT\_FLOW\_IPVX\_SESSION\_CLOSE

The implementations of sessions, gates, ip-actions, processing of multithread, distribution, locking, synchronization, serialization, ordering, packet queuing, asynchronous messaging, IKE traffic issues, sanity check, and queues for IPv6 are similar to IPv4 implementations.

**Related Documentation**

- Understanding IP Version 6 (IPv6) on page 48
- About IPv6 Packet Header Verification Performed by the Flow Module for SRX Series and J-series Devices on page 55

## Understanding IPv6 Dual-Stack Lite

IPv6 dual-stack lite (DS-Lite) is a technology for maintaining connectivity between legacy IPv4 devices and networks for situations in which there is a depleted IPv4 address pool or when a service provider network transitions to IPv6-only deployments.

DS-Lite allows IPv4 customers to continue accessing IPv4 internet content with minimum disruption to their home networks, while enabling IPv6 customers to access IPv6 content.

The DS-Lite deployment model consists of the following components:

- Software initiator (SI) for the DS-Lite home router (SI is not available in Junos release 10.4).
- Software concentrator (SC) for DS-Lite carrier-grade Network Address Translation (NAT).

A softwire is a tunnel-over-IPv6 network. The SI finds the SC address, encapsulates an IPv4 packet, and transmits it across the softwire. The SC receives an IPv4 packet in the IPv6 softwire packet and decapsulates the IPv6 software packet to retrieve the inner IPv4 packet. Multiple SIs can have the same SC as the endpoint of the softwires.

The DS-Lite carrier-grade NAT translates IPv4-to-IPv4 addresses to multiple subscribers through a single global IPv4 address. Overlapping address spaces used by subscribers are disambiguated through the identification of tunnel endpoints.

To configure an SC, the softwire name, the concentrator address, and the softwire type must be set.



**NOTE:** Only IPv4 in IPv6 encapsulation can be used for the software. The IPv4-to-IPv4 NAT with the address overlapped by the existing symmetric source NAT is supported.

DS-Lite supports the address overlapping used by the subscribers. An IPv4 address and application can be mapped to multiple addresses and ports. The cone NAT feature requires an address and application to be mapped to the same address and port. Therefore, DS-Lite cannot work with cone NAT. The issue can be resolved with an additional key for the address mapping of source NAT. That is, you can use the SI IPv6 address along with the IPv4 address and application to find the mapping address and port.

The host-in-bound traffic cannot work with DS-Lite. The traffic initiated from routing engine (RE) also cannot go into the DS-Lite software. You need to use other tunnels with the tunnel interface to carry self traffic over the IPv6 network.

To find out the tunnel information, when the control session created by the DS-Lite hits the gate from the IPv4 side, you need to get the control session for the gate, and find the outgoing interface and the network service provider (NSP) tunnel from the control session.

The data flow of the first packet of one software is as follows:

1. Jexec passes the IPv4-in-IPv6 packet to the flow
2. Flow matches the temporary NSP tunnel with the IPv6 head
3. Flow creates the IPv6 tunnel session
4. Flow decaps the packet
5. Flow creates the IPv4 session for the IPv4 packet with specialized session token for software

The data flow of the other packets of one software is as follows:

1. Jexec passes the IPv4-in-IPv6 packet to the flow
2. Flow matches the tunnel session with the IPv6 head
3. Flow decaps the packet
4. Flow matches or creates the IPv4 session for the IPv4 packet with specialized session token for the software

The session token and 5 tuples are the keys for session matching. The session token must be originated from the software ID in order to distinguish the various IPv4 sessions from the IPv4 source address overlapping among softwares. The general session token always has the highest bit set as 0. The session token for softwares has the highest bit set as 1.

In the Australia platforms, there are two exceptions with respect to the session management to support DS-Lite.



First, The IPv4 incoming wing cannot be installed on CP and NP.

Second, the IPv4 fragments must be reassembled locally by the anchor Services Processing Unit (SPU) rather than being sent to the CP.

The following behaviors are normal for the current tunnel implementation.

1. The temporary NSP tunnels derived from the configuration are created in each SPU.
2. When the first packet for one softwire arrives, the CP randomly chooses the anchor SPU to handle the IPv4-in-IPv6 packet.
3. The tunnel session for the softwire is created by the anchor SPU and also installed on the CP and NP.
4. The anchor SPU decapsulates the packet and creates the IPv4 session for the inner packet.
5. The IPv4 session (the only outgoing wing for the DS-Lite) is installed on the CP and the egress NP.

A mechanism to time out the tunnel session is created for each softwire. A reference counter is added to the tunnel session to count the sessions. When the counter reaches zero, a 30-minute timer is started. If there is no new traffic to create a session during that time, the timer expires and the tunnel session is deleted. When the concentrator configuration is reset, all the tunnel sessions derived from it are deleted. When the tunnel session is deleted, all the sessions referring to it are deleted by session scan.

**Related Documentation**

- [Understanding IP Version 6 \(IPv6\) on page 48](#)
- [Understanding How SRX Series and J-series Devices Handle ICMPv6 Packets on page 59](#)
- [About the IPv6 Basic Packet Header on page 52](#)

## Understanding How SRX Series and J-series Devices Handle ICMPv6 Packets

This topic explains Internet Control Message Protocol (ICMP), ICMP messages, and how Junos OS for SRX Series Services Gateways uses them.

ICMP provides a framework for reporting packet processing errors, for diagnostic purposes, and for implementation-specific functions. ICMP error messages make it possible for one node to inform another node that something has gone wrong during the course of data transfer. When IP version 6 (IPv6) was defined, the differences between IP version 4 (IPv4) and it were significant enough to require a new version of ICMP.

Every ICMPv6 message is preceded by an IPv6 header and zero or more IPv6 extension headers. The ICMPv6 header is identified by a Next Header value of 58 in the immediately preceding header. This is different from the value used to identify ICMP for IPv4. All ICMPv6 error messages have 32 bits of type-specific data to help the packet recipient locate the embedded invoking packet.

Most ICMPv6 packets have the same characteristics and behavior as normal IPv6 packets, and the Junos OS flow module processes them through first path and fast-path processing

in the same way that it does normal IPv6 packets. Table 9 on page 60 shows the ICMPv6 embedded packet types that the flow module handles differently from normal ICMPv6 packets.

For these packets, the flow module uses a tuple that it creates from the embedded ICMPv6 packet to search for a matching session. It continues to process the packet without modifying the maximum transmission unit (MTU) until it finds a matching session, unless it receives an ICMPv6 Packet Too Big message for the interface. In this case, it modifies the MTU size for that interface. If the flow module does not find a matching session or if it cannot obtain a valid IPv6 header from the embedded payload, it drops the packet.



**NOTE:** A Packet Too Big message is the only kind of ICMPv6 packet that will cause the flow module to modify an interface.

**Table 9: ICMPv6 Packets That Junos OS Handles Differently from Other ICMPv6 Packets**

Message	Meaning
01-Destination Unreachable	<p>When a packet cannot be delivered because of a problem with the way it is being sent, it is useful to have a feedback mechanism that can tell the source about the problem, including the reason why delivery of the packet failed. For IPv6, the Destination Unreachable message serves this purpose.</p> <p>Each message includes a code that indicates the nature of the problem that caused the packet delivery to fail. It also includes all or part of the packet that could not be delivered, to help the source device resolve the problem.</p> <p>When the flow module encounters a Destination Unreachable ICMP packet whose embedded packet header data matches the 5-tuple data for a session, the software terminates the session.</p>
02-Packet Too Big	<p>When the flow module receives an ICMPv6 Packet Too Big message intended for it, the flow module sends the packet to the ICMP protocol stack on the Routing Engine to engage the path maximum transmission unit (path MTU) discovery process.</p> <p>If the Packet Too Big message does not pertain to the device but rather is a transit packet, the device attempts to match the embedded 5-tuple data with a session.</p> <ul style="list-style-type: none"> <li>• If a matching session exists, the device delivers it to the source node.</li> <li>• If a matching session does not exist, the device drops the packet</li> </ul> <p><b>NOTE:</b> A Packet Too Big message is the only kind of ICMPv6 packet that will cause the flow module to modify an interface.</p>
03-Time Exceeded	<p>When the flow module receives a packet that cannot be delivered because it has exceeded the hop count specified in the basic header hop-by-hop field, it sends this message to inform the packet's source node that the packet was discarded for this reason.</p>

**Table 9: ICMPv6 Packets That Junos OS Handles Differently from Other ICMPv6 Packets** (*continued*)

Message	Meaning
04-Parameter Problem	When the device finds a problem with a field in the IPv6 header or extension headers that makes it impossible for it to process the packet, the software discards it and sends this ICMPv6 message to the packet's source node, indicating the type and location of the problem.

**Related Documentation**

- Understanding Path MTU Messages for IPv6 Packets on page 61
- Understanding How SRX Series and J-series Devices Handle Packet Fragmentation for IPv6 Flows on page 62
- Understanding IP Version 6 (IPv6) on page 48

## Understanding Path MTU Messages for IPv6 Packets

This topic describes path maximum transmission unit (MTU) and explains how the flow module for SRX Series and J-series devices processes and uses path MTU messages.

Every link has an MTU size that specifies the size of the largest packet the link can transmit. A larger MTU size means that fewer packets are required to transmit a certain amount of data. To achieve the best data transmission performance, IPv6 data packets sent from one node (the source) to another node (the destination) should be the largest possible size that can traverse the path between the nodes. (Larger and fewer packets constrain the cost of packet header processing and routing processes that can affect transmission performance.)

However, for a packet to successfully traverse the path from the source node to the destination node, the MTU size of the source node interface must be no larger than that of the smallest MTU size of all nodes on the path between the source and destination. This value is referred to as the path maximum transmission unit (path MTU). If a packet is larger than a link's MTU size, it is likely that the link will drop it. For IPv6, an intermediate node cannot fragment a packet.

IPv6 defines a standard mechanism called path MTU discovery that a source node can use to learn the path MTU of a path that a packet is likely to traverse. If any of the packets sent on that path are too large to be forwarded by a node along the path, that node discards the packet and returns an ICMPv6 Packet Too Big message. The source node can then adjust the MTU size to be smaller than that of the node that dropped it and sent the ICMPv6 message, and then retransmit the packet. A source node might receive Packet Too Big messages repeatedly until its packet traverses all nodes along the path successfully.

After the path MTU size is determined and the appropriate MTU size is set, an outgoing packet might be routed along a different path with a node whose link MTU size is smaller than the path MTU size determined previously. In this case, the flow module engages the path MTU discovery process again.

When the flow module receives an ICMP Packet Too Big message with a destination address that belongs to it, it:

- Checks to determine if the embedded 5-tuple data of the packet is for a tunnel interface. (That is, it checks to determine if the embedded 5-tuple data matches a tunnel session.) If there is a match, the flow module updates the tunnel interface's MTU size. Then it performs post-fragment processing for the encrypted packets that follow the first packet. Afterward, the flow module delivers the packet to the ICMPv6 stack on the routing engine (RE) for it to continue processing it.
- If the packet is a transit one, the flow module searches for a session that matches the packet's embedded 5-tuple data. If it finds a matching session, it delivers the packet to it. If there is no matching session, it drops the packet.

When the flow module receives a packet, before it transmits it to the egress interface, it checks to determine if the MTU size of the egress interface is greater than the packet length.

- If the MTU size is greater than the packet length, it continues to process the packet.
- If the MTU size is less than the packet length, it drops the packet and sends an ICMPv6 Packet Too Big message to the source node.



**NOTE:** When chassis cluster is configured and the path MTU updates the MTU of the tunnel interface, the flow module does not sync the new MTU to peer nodes. The MTU size might be updated again by a larger packet on a peer node, which has no impact on packet transmission.

#### Related Documentation

- [Understanding How SRX Series and J-series Devices Handle ICMPv6 Packets](#) on page 59
- [About the IPv6 Basic Packet Header](#) on page 52
- [Understanding IP Version 6 \(IPv6\)](#) on page 48

## Understanding How SRX Series and J-series Devices Handle Packet Fragmentation for IPv6 Flows

This topic explains packet fragmentation for IP version 6 (IPv6).

For IPv4 Internet Control Message Protocol (IPv4 ICMP), if a node within the path between a source node and a destination node receives a packet that is larger than its MTU size, it can fragment the packet and transmit the resulting smaller packets. For IPv6, only a source node (the node that sent the packet) can fragment a packet, and this is done to accommodate a path MTU size-adjustment requirement. Nodes along the path of a packet cannot fragment the packet to transmit it.

#### Related Documentation

- [Understanding How SRX Series and J-series Devices Handle ICMPv6 Packets](#) on page 59
- [Understanding Path MTU Messages for IPv6 Packets](#) on page 61
- [Understanding IPv6 Packet Header Extensions](#) on page 54

- Understanding IP Version 6 (IPv6) on page 48

## Understanding Sessions for IPv6 Flows

---

This topic gives an overview of flow-based sessions.

Most packet processing occurs in the context of a flow, including management of policies, zones, and most screens. A session is created for the first packet of a flow for the following purposes:

- To store most of the security measures to be applied to the packets of the flow.
- To cache information about the state of the flow. For example, logging and counting information for a flow is cached in its session. (Also, some stateful firewall screens rely on threshold values that pertain to individual sessions or across all sessions.)
- To allocate resources required for features for the flow.
- To provide a framework for features such as Application Layer Gateways (ALGs).

### Related Documentation

- Understanding SRX5600 and SRX5800 Architecture and Flow Processing on page 63
- Understanding IP Version 6 (IPv6) on page 48

## Understanding SRX5600 and SRX5800 Architecture and Flow Processing

---

This topic introduces the architecture for the SRX5600 and SRX5800 devices and uses it as a model to explain IP version 6 (IPv6) processing. Flow processing is similar on other SRX Series and J-series devices.

High-end SRX Series Services Gateway devices include I/O cards (IOCs) and Services Processing Cards (SPCs) that each contain processing units that process a packet as it traverses the device. These processing units have different responsibilities.

- A Network Processing Unit (NPU) runs on an IOC. An IOC has one or more NPUs. An NPU processes packets discretely and performs basic flow management functions.

When an IPv6 packet arrives at an IOC, the packet flow process begins. The NPU takes the following actions:

- It performs the following IPv6 sanity checks for the packet.
  - For the IPv6 basic header, it performs the following header checks:
    - Version. It verifies that the header specifies IPv6 for the version.
    - Payload length. It checks the payload length to ensure that the combined length of the IPv6 packet and the Layer 2 (L2) header is greater than the L2 frame length.

- Hop limit. It checks to ensure that the hop limit does not specify 0 (zero)
- Address checks. It checks to ensure that the source IP address does not specify ::0 or FF::00 and that the destination IP address does not specify ::0 or ::1.
- It performs IPv6 extension header checks, including the following:
  - Hop-by-hop options. It verifies that this is the first extension header to follow the IPV6 basic header.
  - Routing extension. It verifies that there is only one routing extension header.
  - Destination options. It verifies that no more than two destination options extension headers are included.
  - Fragment. It verifies that there is only one fragment header.



**NOTE:** It treats any other extension header as a Layer 4 (L4) header.

---

- It performs L4 TCP, UDP, and ICMPv6 protocol checks, including the following:
  - UDP. It checks to ensure that UDP packets, other than a first-fragment packet, are at least 8 bytes long.
  - TCP. It checks to ensure that ICMPv6 packets, other than a first-fragment packet, are at least 20 bytes long.
  - ICMPv6. It checks to ensure that ICMPv6 packets, other than a first-fragment packet, are at least 8 bytes long.
- If the packet specifies a TCP or a UDP protocol, it creates a tuple from the packet header data using the following information:
  - Source IP address.
  - Destination IP address.
  - Source port.
  - Destination port.
  - Protocol.
  - Virtual router identifier (VRID). The device looks up the VRID from a VRID table.
- For Internet Control Message Protocol version 6 (ICMPv6) packets, the tuple contains the same information as used for the TCP and the UDP search key, except for the source and destination port fields. The source and destination port fields are replaced with the following information extracted from the ICMPv6 packet:
  - For ICMP error packets: The pattern "0x00010001"
  - For ICMP information packets: The type, or code, field identifier

- For packets with an Authentication Header (AH) or an Encapsulating Security Payload (ESP) header, the search key is the same as that used for the TCP and the UDP tuple, except for the source and destination fields. In this case, the security parameter index (SPI) field value is used instead of the source and destination ports.
- If a session exists for the packet's flow, the NPU sends the packet to the SPU that manages the session.
- If a matching session does not exist,
  - The NPU sends the packet information to the central point (CP), which creates a pending session.
  - The CP selects an SPU to process the packet and create sessions for it.
  - The SPU then sends session creation messages to the CP and the ingress and egress NPUs, directing them to create a session for the packet flow.
- A central point which can run on a dedicated SPU, or share the resources of one if there is only one SPU. A CP takes care of arbitration and allocation of resources, and it distributes sessions in an intelligent way. The CP assigns an SPU to be used for a particular session when the SPU processes the first packet of its flow.
  - Juniper Networks SRX5000 line devices have at least two SPUs. If an SRX5000 line device has only two SPUs, one acts in combination (*combo mode*) serving as both the CP and the SPU.
  - For SRX3000 line devices, the CP and an SPU always run in combo mode.
- One or more SPUs that run on a Services Processing Card (SPC). All flow-based services for a packet are executed on a single SPU, within the context of a session that is set up for the packet flow.

The SPC for SRX5000 line devices has two SPUs. The SPC for SRX3000 line devices has one SPU.

Several SPCs can be installed in a chassis.

Primarily, an SPU performs the following tasks:

- It manages the session and applies security features and other services to the packet.
- It applies packet-based stateless firewall filters, classifiers, and traffic shapers.
- If a session does not already exist for a packet, it sends a request message to the NPU that performed the search for the packet's session, to direct it to add a session for it.

These discrete, cooperating parts of the system store the information identifying whether a session exists for a stream of packets and the information against which a packet is matched to determine if it belongs to an existing session.

**Related Documentation**

- Understanding Sessions for IPv6 Flows on page 63
- Understanding IP Version 6 (IPv6) on page 48

## Enabling Flow-Based Processing for IPv6 Traffic

By default, the SRX Series or J Series device drops IPv6 traffic. To enable processing by security features such as zones, screens, and firewall policies, you must enable flow-based forwarding for IPv6 traffic.

To enable flow-based forwarding for IPv6 traffic, modify the **mode** statement at the [edit **security forwarding-options family inet6**] hierarchy level:

```
security {
  forwarding-options {
    family {
      inet6 {
        mode flow-based;
      }
    }
  }
}
```

The following example shows the CLI commands you use to configure forwarding for IPv6 traffic.

1. Use the **set** command to change the forwarding option mode for IPv6 to flow-based.

```
[edit]
user@host# set security forwarding-options family inet6 mode flow-based
```

2. Use the **show** command to review your configuration.

```
[edit]
user@host# show security forwarding-options

family {
  inet6 {
    mode flow-based;
  }
}
```

3. Check your changes to the configuration before committing.

```
[edit]
user@host# commit check

warning: You have enabled/disabled inet6 flow.
You must reboot the system for your change to take effect.
If you have deployed a cluster, be sure to reboot all nodes.
configuration check succeeds
```

4. Commit the configuration.

```
[edit]
user@host# commit

warning: You have enabled/disabled inet6 flow.
You must reboot the system for your change to take effect.
If you have deployed a cluster, be sure to reboot all nodes.
commit complete
```

5. At an appropriate time, reboot the device.



Table 10 on page 67 summarizes device status upon forwarding option configuration change.

**Table 10: Device Status Upon Configuration Change**

Configuration Change	Commit Warning	Reboot Required	Impact on Existing Traffic Before Reboot	Impact on New Traffic Before Reboot
Drop to flow-based	Yes	Yes	Dropped	Dropped
Drop to packet-based	No	No	Packet-based	Packet-based
Flow-based to packet-based	Yes	Yes	None	Flow sessions created
Flow-based to drop	Yes	Yes	None	Flow sessions created
Packet-based to flow-based	Yes	Yes	Packet-based	Packet-based
Packet-based to drop	No	No	Dropped	Dropped

To process IPv6 traffic, you also need to configure IPv6 addresses for the transit interfaces that receive and forward the traffic. For information on the inet6 protocol family and procedures for configuring IPv6 addresses for interfaces, see the *Junos OS Interfaces Configuration Guide for Security Devices*.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding IP Version 6 \(IPv6\) on page 48](#)
- [Using Filters to Display IPv6 Session and Flow Information for SRX Series Services Gateways on page 67](#)

## Using Filters to Display IPv6 Session and Flow Information for SRX Series Services Gateways

**Purpose** You can display flow and session information about one or more sessions with the **show security flow session** command. IPv6 sessions are included in aggregated statistics.

You can use the following filters with the **show security flow session** command: application, destination-port, destination-prefix, family, idp, interface, nat, protocol, resource-manager, session-identifier, source-port, source-prefix and tunnel.



**NOTE:** Except the session-identifier filter, the output of all the other filters can be viewed in brief, summary and extensive mode. Brief mode is the default mode. The output of the session- identifier filter can be viewed only in the brief mode.

You can use the same filter options with the **clear security flow session** command to terminate sessions.

**Action** The following examples show how to use IPv6-related filters to display summaries and details for IPv6 sessions.

**Filtered summary  
report based on family**

```
root> show security flow session summary family ?
Possible completions:
  inet          Show IPv4 sessions
  inet6        Show IPv6/IPv6-NATPT sessions

root> show security flow session summary family inet6
Flow Sessions on FPC4 PIC1:

Valid sessions: 71
Pending sessions: 0
Invalidated sessions: 56
Sessions in other states: 0
Total sessions: 127

Flow Sessions on FPC5 PIC0:

Valid sessions: 91
Pending sessions: 0
Invalidated sessions: 53
Sessions in other states: 0
Total sessions: 144

Flow Sessions on FPC5 PIC1:

Valid sessions: 91
Pending sessions: 0
Invalidated sessions: 54
Sessions in other states: 0
Total sessions: 145
```

**Filtered detailed report  
based on family**

```

root> show security flow session family ?
Possible completions:
  inet          Show IPv4 sessions
  inet6        Show IPv6/IPv6-NATPT sessions

root> show security flow session family inet6
Flow Sessions on FPC4 PIC1:

Session ID: 170001887, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 4000::100/9 --> 4000::200/27490;icmp6, If: ge-0/0/2.0, Pkts: 1, Bytes: 104

  Out: 4000::200/27490 --> 4000::100/9;icmp6, If: .local..0, Pkts: 1, Bytes: 104
Total sessions: 1

Flow Sessions on FPC5 PIC0:

Session ID: 200001865, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 4000::100/10 --> 4000::200/27490;icmp6, If: ge-0/0/2.0, Pkts: 1, Bytes: 104

  Out: 4000::200/27490 --> 4000::100/10;icmp6, If: .local..0, Pkts: 1, Bytes: 104
Total sessions: 1

Flow Sessions on FPC5 PIC1:

Session ID: 210001865, Policy name: self-traffic-policy/1, Timeout: 4, Valid
  In: 4000::100/11 --> 4000::200/27490;icmp6, If: ge-0/0/2.0, Pkts: 1, Bytes: 104

  Out: 4000::200/27490 --> 4000::100/11;icmp6, If: .local..0, Pkts: 1, Bytes: 104
Total sessions: 1

```

**Filtered brief report  
based on family**

```

root> show security flow session family inet brief
Flow Sessions on FPC4 PIC1:

Session ID: 170067516, Policy name: self-traffic-policy/1, Timeout: 4, Valid
  In: 40.0.0.100/23 --> 40.0.0.1/26637;icmp, If: ge-0/0/2.0, Pkts: 1, Bytes: 84
  Out: 40.0.0.1/26637 --> 40.0.0.100/23;icmp, If: .local..0, Pkts: 1, Bytes: 84
Total sessions: 1

Flow Sessions on FPC5 PIC0:

Session ID: 200066737, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 40.0.0.100/21 --> 40.0.0.1/26637;icmp, If: ge-0/0/2.0, Pkts: 1, Bytes: 84
  Out: 40.0.0.1/26637 --> 40.0.0.100/21;icmp, If: .local..0, Pkts: 1, Bytes: 84
Total sessions: 1

Flow Sessions on FPC5 PIC1:

Session ID: 210066726, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 40.0.0.100/22 --> 40.0.0.1/26637;icmp, If: ge-0/0/2.0, Pkts: 1, Bytes: 84
  Out: 40.0.0.1/26637 --> 40.0.0.100/22;icmp, If: .local..0, Pkts: 1, Bytes: 84
Total sessions: 1

```

**Filtered detailed report  
based on an IPv6  
source-prefix**

```

root> show security flow session source-prefix 4000::100
Flow Sessions on FPC4 PIC1:

Session ID: 170001907, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 4000::100/69 --> 4000::200/27490;icmp6, If: ge-0/0/2.0, Pkts: 1, Bytes: 104

  Out: 4000::200/27490 --> 4000::100/69;icmp6, If: .local..0, Pkts: 1, Bytes: 104
Total sessions: 1

```

Flow Sessions on FPC5 PIC0:

```
Session ID: 200001885, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 4000::100/70 --> 4000::200/27490;icmp6, If: ge-0/0/2.0, Pkts: 1, Bytes: 104
  Out: 4000::200/27490 --> 4000::100/70;icmp6, If: .local..0, Pkts: 1, Bytes: 104
Total sessions: 1
```

Flow Sessions on FPC5 PIC1:

```
Session ID: 210001885, Policy name: self-traffic-policy/1, Timeout: 4, Valid
  In: 4000::100/71 --> 4000::200/27490;icmp6, If: ge-0/0/2.0, Pkts: 1, Bytes: 104
  Out: 4000::200/27490 --> 4000::100/71;icmp6, If: .local..0, Pkts: 1, Bytes: 104
Total sessions: 1
```

**Multiple-filtered  
detailed report based  
on family, protocol and  
source-prefix**

```
root> show security flow session family inet protocol icmp source-prefix 40/8
```

Flow Sessions on FPC4 PIC1:

```
Session ID: 170029413, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 40.0.0.100/50 --> 40.0.0.1/1369;icmp, If: ge-0/0/2.0, Pkts: 1, Bytes: 84
  Out: 40.0.0.1/1369 --> 40.0.0.100/50;icmp, If: .local..0, Pkts: 1, Bytes: 84
Total sessions: 1
```

Flow Sessions on FPC5 PIC0:

```
Session ID: 200029073, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 40.0.0.100/51 --> 40.0.0.1/1369;icmp, If: ge-0/0/2.0, Pkts: 1, Bytes: 84
  Out: 40.0.0.1/1369 --> 40.0.0.100/51;icmp, If: .local..0, Pkts: 1, Bytes: 84
Total sessions: 1
```

Flow Sessions on FPC5 PIC1:

```
Session ID: 210029083, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 40.0.0.100/52 --> 40.0.0.1/1369;icmp, If: ge-0/0/2.0, Pkts: 1, Bytes: 84
  Out: 40.0.0.1/1369 --> 40.0.0.100/52;icmp, If: .local..0, Pkts: 1, Bytes: 84
Total sessions: 1
```

**Clearing all sessions,  
including IPv6 sessions**

```
root> clear security flow session all
```

```
This command may terminate the current session too.
Continue? [yes,no] (no) yes
```

```
0 active sessions cleared
1 active sessions cleared
1 active sessions cleared
1 active sessions cleared
```

**Clearing only IPv6  
sessions**

```
root> clear security flow session family ?
```

```
Possible completions:
```

```
inet          Clear IPv4 sessions
inet6         Clear IPv6/IPv6-NATPT sessions
```

```
root> clear security flow session family inet6
```

```
0 active sessions cleared
1 active sessions cleared
1 active sessions cleared
1 active sessions cleared
```

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Enabling Flow-Based Processing for IPv6 Traffic on page 66](#)
  - [Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12](#)
  - [Displaying a Summary of Sessions for SRX Series Services Gateways on page 13](#)
  - [Displaying Session and Flow Information About Sessions for SRX Series Services Gateways on page 14](#)
  - [Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways on page 14](#)
  - [Information Provided in Session Log Entries for SRX Series Services Gateways on page 16](#)
  - [Clearing Sessions for SRX Series Services Gateways on page 20](#)

## IPv6 NAT

---

- [IPv6 NAT Overview on page 71](#)
- [IPv6 NAT PT Overview on page 73](#)
- [IPv6 NAT-PT Communication Overview on page 74](#)
- [Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Default Destination Address Prefix Static Mapping on page 75](#)
- [Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Static Destination Address One-to-One Mapping on page 78](#)
- [Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Default Destination Address Prefix Static Mapping on page 82](#)
- [Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Static Destination Address One-to-One Mapping on page 85](#)

### IPv6 NAT Overview

IPv6 has a vastly larger address space than the impending exhausted IPv4 address space. IPv4 has been extended using techniques such as Network Address Translation (NAT), which allows for ranges of private addresses to be represented by a single public address, and temporary address assignment. There are a lot of technologies to provide the transition mechanism for the legacy IPv4 host to keep the connection to the Internet. IPv6 NAT provides address translation between IPv4 and IPv6 addressed network devices. It also provides address translation between IPv6 hosts. NAT between IPv6 hosts is done in a similar manner and for similar purposes as IPv4 NAT.

IPv6 NAT in Junos OS provides the following NAT types:

- Source NAT
- Destination NAT
- Static NAT

### Source NAT Translations Supported by IPv6 NAT

---

Source NAT is the translation of the source IP address of a packet leaving the Juniper Networks device. Source NAT is used to allow hosts with private IP addresses to access a public network.

IPv6 NAT in Junos OS supports the following source NAT translations:

- Translation of one IPv6 subnet to another IPv6 subnet without port address translation
- Translation of IPv4 addresses to IPv6 prefix + IPv4 addresses
- Translation of IPv6 hosts to IPv6 hosts with or without port address translation
- Translation of IPv6 hosts to IPv4 hosts with or without port address translation
- Translation of IPv4 hosts to IPv6 hosts with or without port address translation

### Destination NAT Mappings Supported by IPv6 NAT

---

Destination NAT is the translation of the destination IP address of a packet entering the Juniper Networks device. Destination NAT is used to redirect traffic destined to a virtual host (identified by the original destination IP address) to the real host (identified by the translated destination IP address).

IPv6 NAT in Junos OS supports the following destination NAT translations:

- Prefix translation between IPv4 and IPv6 prefix
- Mapping of one IPv6 subnet to another IPv6 subnet
- Mapping of one IPv6 subnet to an IPv6 host
- Mapping of one IPv6 subnet to one IPv4 subnet
- Mapping of one IPv4 subnet to one IPv6 subnet
- Mapping of one IPv6 host (and optional port number) to one special IPv6 host (and optional port number)
- Mapping of one IPv6 host (and optional port number) to one special IPv4 host (and optional port number)
- Mapping of one IPv4 host (and optional port number) to one special IPv6 host (and optional port number)

### Static NAT Mappings Supported by IPv6 NAT

---

Static NAT defines a one-to-one mapping from one IP subnet to another IP subnet. The mapping includes destination IP address translation in one direction and source IP address translation in the reverse direction. From the NAT device, the original destination address is the virtual host IP address while the mapped-to address is the real host IP address.

IPv6 NAT in Junos OS supports the following static NAT translations:

- Translation of one IPv6 subnet to another IPv6 subnet
- Translation of one IPv6 host to another IPv6 host

- Translation of one IPv4 address a.b.c.d to IPv6 address Prefix::a.b.c.d
- Translation of IPv4 hosts to IPv6 hosts
- Translation of IPv6 hosts to IPv4 hosts
- Mapping of one IPv6 prefix to one IPv4 prefix
- Mapping of one IPv4 prefix to one IPv6 prefix

#### Related Documentation

- IPv6 NAT PT Overview on page 73
- IPv6 NAT-PT Communication Overview on page 74
- Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Static Destination Address One-to-One Mapping on page 85
- Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Default Destination Address Prefix Static Mapping on page 82
- Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Static Destination Address One-to-One Mapping on page 78
- Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Default Destination Address Prefix Static Mapping on page 75

## IPv6 NAT PT Overview

IPv6 Network Address Translation- Protocol Translation (NAT-PT) provides address allocation and protocol translation between IPv4 and IPv6 addressed network devices. The translation process is based on the Stateless IP/ICMP Translation (SIIT) method; however, the state and the context of each communication are retained during the session lifetime. IPv6 NAT-PT supports Internet Control Message Protocol (ICMP), TCP, and UDP packets.

IPv6 NAT-PT supports the following types of NAT-PT:

- Traditional NAT-PT— In traditional NAT-PT, the sessions are unidirectional and outbound from the IPv6 network . Traditional NAT-PT allows hosts within an IPv6 network to access hosts in an IPv4 network. There are two variations to traditional NAT-PT: basic NAT-PT and NAPT-PT.

In basic NAT-PT, a block of IPv4 addresses at an IPv4 interface is set aside for translating addresses as IPv6 hosts as they initiate sessions to the IPv4 hosts. The basic NAT-PT translates the source IP address and related fields such as IP, TCP, UDP, and ICMP header checksums for packets outbound from the IPv6 domain . For inbound packets, it translates the the destination IP address and the checksums.

Network Address Port Translation and Protocol Translation (NAPT-PT) can be combined with basic NAT-PT so that a pool of external addresses is used in conjunction with port translation. NAPT-PT allows a set of IPv6 hosts to share a single IPv4 address. NAPT-PT translates the source IP address, source transport identifier, and related fields such as IP, TCP, UDP, and ICMP header checksums, for packets outbound from the IPv6 network. The transport identifier can be a TCP/UDP port or an ICMP query ID.

For inbound packets, it translates the destination IP address, destination transport identifier, and the IP and the transport header checksums.

- **Bidirectional NAT-PT**— In bidirectional NAT-PT, sessions can be initiated from hosts in the IPv4 network as well as the IPv6 network. IPv6 network addresses are bound to IPv4 addresses, either statically or dynamically as connections are established in either direction. The static configuration is similar to static NAT translation. Hosts in IPv4 realm access hosts in the IPv6 realm using DNS for address resolution. A DNS ALG must be employed in conjunction with bidirectional NAT-PT to facilitate name-to-address mapping. Specifically, the DNS ALG must be capable of translating IPv6 addresses in DNS queries and responses into their IPv4 address bindings, and vice versa, as DNS packets traverse between IPv6 and IPv4 realms.



**NOTE:** The SRX Series and the J Series devices partially supports the Bidirectional NAT-PT specification. It supports flow of bidirectional traffic assuming that there are other ways to convey the mapping between the IPv6 address and the dynamically allocated IPv4 address. For example, a local DNS can be configured with the mapped entries for IPv4 nodes to identify the addresses.

**NAT-PT Operation**— The SRX Series and the J Series devices support the Traditional NAT-PT and allows static mapping for the user to communicate from IPv4 to IPv6 . The user needs to statically configure the DNS server with an IPv4 address for the host name and then create a static NAT on the device for the IPv6-only node to communicate from an IPv4-only node to an IPv6-only node based on the DNS.

#### Related Documentation

- IPv6 NAT Overview on page 71
- IPv6 NAT-PT Communication Overview on page 74
- Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Static Destination Address One-to-One Mapping on page 85
- Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Default Destination Address Prefix Static Mapping on page 82
- Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Static Destination Address One-to-One Mapping on page 78
- Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Default Destination Address Prefix Static Mapping on page 75

## IPv6 NAT-PT Communication Overview

**NAT-PT communication with static mapping**— Network Address Translation-Protocol Translation (NAT-PT) can be done in two directions, from IPv6 to IPv4 and vice versa. For each direction, static NAT is used to map the destination host to a local address and a source address NAT is used to translate the source address. There are two types of static NAT and source NAT mapping: one-to-one mapping and prefix-based mapping.



**NAT-PT communication with DNS ALG**—A DNS-based mechanism dynamically maps IPv6 addresses to IPv4-only servers. NAT-PT uses the DNS ALG to transparently do the translations. For example, a company using an internal IPv6 network needs to be able to communicate with external IPv4 servers that do not yet have IPv6 addresses.

To support the dynamic address binding, a DNS should be used for name resolution. The IPv4 host looks up the name of the IPv6 node in its local configured IPv4 DNS server, which then passes the query to the IPv6 DNS server through an SRX Series device using NAT-PT.

The DNS ALG in NAT device :

- Translates the IPv6 address resolution back to IPv4 address resolution.
- Allocates an IPv6 address for the mapping.
- Stores a mapping of the allocated IPv4 address to the IPv6 address returned in the IPv6 address resolution so that the session can be established from any-IPv4 hosts to the IPv6 host.

#### Related Documentation

- [IPv6 NAT Overview on page 71](#)
- [IPv6 NAT PT Overview on page 73](#)
- [Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Static Destination Address One-to-One Mapping on page 85](#)
- [Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Default Destination Address Prefix Static Mapping on page 82](#)
- [Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Static Destination Address One-to-One Mapping on page 78](#)
- [Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Default Destination Address Prefix Static Mapping on page 75](#)

### Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Default Destination Address Prefix Static Mapping

This example shows how to configure an IPv4-initiated connection to an IPv6 node using default destination address prefix static mapping.

- [Requirements on page 75](#)
- [Overview on page 76](#)
- [Configuration on page 76](#)
- [Verification on page 78](#)

#### Requirements

Before you begin, configure the interfaces and assign the interfaces to security zones.

## Overview

The following example describes how to configure an IPv4-initiated connection to an IPv6 node that has a static mapping /96-based IPv6 address defined on its interface and static mapping /96 set up on the device. This example assumes the IPv6 addresses to be mapped IPv4 addresses, making the IPv4 addresses a part of the IPv6 address space.

Configuring an IPv4-initiated connection to an IPv6 node is useful when the devices on the IPv4 network must be interconnected to the devices on the IPv6 network and during migration of an IPv4 network to an IPv6 network. The mapping can be used for DNS ALG for reverse lookup of IPv4 addresses from IPv6 addresses, for the traffic initiated from the IPv6 network. This also provides connectivity for sessions initiated from IPv4 nodes with IPv6 nodes on the other side of the NAT/PT device.

## Configuration

### CLI Quick Configuration

To quickly configure an IPv4-initiated connection to an IPv6 node using default destination address static mapping, copy the following commands and paste them into the CLI.

```
[edit]
set security nat static rule-set test_rs from interface ge-0/0/1
set security nat static rule-set test_rs rule test_rule match destination-address 10.1.1.45/30
set security nat static rule-set test_rs rule test_rule then static-nat prefix 27a6::/96
set security nat source pool myipv6_prefix address 27a6::/96
set security nat source rule-set myipv6_rs from interface ge-0/0/1
set security nat source rule-set myipv6_rs to interface ge-0/0/2
set security nat source rule-set myipv6_rs rule ipv6_rule match source-address 10.1.1.1/30
set security nat source rule-set myipv6_rs rule ipv6_rule match destination-address
  27a6::a0a:a2d/126
set security nat source rule-set myipv6_rs rule ipv6_rule then source-nat pool myipv6_prefix
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure an IPv6-initiated connection to an IPv4 node using static destination address one-to-one mapping:

1. Configure the static NAT rule set for an interface.

```
[edit security nat static]
user@host# set rule-set test_rs from interface ge-0/0/1
```

2. Define the rule to match the destination address prefix.

```
[edit security nat static rule-set test_rs]
user@host# set rule test_rule match destination-address 10.1.1.45/30
```

3. Define the static NAT prefix for the device.

```
[edit security nat static rule-set test_rs]
user@host# set rule test_rule then static-nat prefix 27a6::/96
```

4. Configure the source NAT pool with an IPv6 address prefix.

```
[edit security nat source]
```

- ```
user@host# set pool myipv6_prefix address 27a6::/96
```
5. Configure the source NAT rule set for the interface.
 

```
[edit security nat source]
user@host# set rule-set myipv6_rs from interface ge-0/0/1
user@host# set rule-set myipv6_rs to interface ge-0/0/2
```
  6. Configure the IPv6 source NAT source address.
 

```
[edit security nat source rule-set myipv6_rs]
user@host# set rule ipv6_rule match source-address 10.1.1/30
```
  7. Configure the IPv6 source NAT destination address.
 

```
[edit security nat source rule-set myipv6_rs]
user@host# set rule ipv6_rule match destination-address 27a6::a0a:a2d/126
```
  8. Define the configured source NAT IPv6 pool in the rule.
 

```
[edit security nat source rule-set myipv6_rs]
user@host# set rule ipv6_rule then source-nat pool myipv6_prefix
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
source {
  pool myipv6_prefix {
    address {
      27a6::/96;
    }
  }
  rule-set myipv6_rs {
    from interface ge-0/0/1.0;
    to interface ge-0/0/2.0;
    rule ipv6_rule {
      match {
        source-address 10.1.1/30;
        destination-address 27a6::a0a:a2d/126;
      }
      then {
        source-nat {
          pool {
            myipv6_prefix;
          }
        }
      }
    }
  }
}
static {
  rule-set test_rs {
    from interface ge-0/0/1.0;
    rule test_rule {
      match {
```

```

        destination-address 10.1.1.45/30;
    }
    then {
        static-nat prefix 27a6::/96;
    }
}
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying That Static NAT Is Configured on page 78
- Verifying That Source NAT Is Configured on page 78

#### *Verifying That Static NAT Is Configured*

**Purpose** Verify whether static NAT is configured with an interface, a destination address, and a prefix.

**Action** From operational mode, enter the **show security nat static** command.

#### *Verifying That Source NAT Is Configured*

**Purpose** Verify whether source NAT is configured.

**Action** From operational mode, enter the **show security nat source** command.

#### Related Documentation

- IPv6 NAT Overview on page 71
- IPv6 NAT PT Overview on page 73
- IPv6 NAT-PT Communication Overview on page 74
- Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Static Destination Address One-to-One Mapping on page 85
- Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Default Destination Address Prefix Static Mapping on page 82
- Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Static Destination Address One-to-One Mapping on page 78

### Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Static Destination Address One-to-One Mapping

This example shows how to configure an IPv4-initiated connection to an IPv6 node using static destination address one-to-one mapping.

- Requirements on page 79
- Overview on page 79

- Configuration on page 79
- Verification on page 81

## Requirements

Before you begin, configure the interfaces and assign the interfaces to security zones.

## Overview

The following example describes how to configure an IPv4 node to communicate with an IPv6 node using one-to-one static NAT on the device.

The communication of an IPv4 node with an IPv6 node is useful for IPv4 hosts accessing an IPv6 server, for new servers that support IPv6 only and that need to be connected to the IPv6 network, and for migrating of old hosts to the new server when most of the machines have already moved to IPv6. For example, you can use this feature to connect an IPv4-only node to an IPv6-only printer. This mapping can also be used for DNS ALG for reverse lookup of IPv4 addresses from IPv6 addresses for traffic that is initiated from the IPv6 network.

## Configuration

### CLI Quick Configuration

To quickly configure an IPv4-initiated connection to an IPv6 node using static destination address one-to-one mapping, copy the following commands and paste them into the CLI.

```
[edit]
set security nat static rule-set test_rs from interface ge-0/0/1
set security nat static rule-set test_rs rule test_rule match destination-address 10.1.1.25/32
set security nat static rule-set test_rs rule test_rule then static-nat prefix 3ffe::25/128
set security nat source pool myipv6_prefix address 27a6::/96
set security nat source rule-set myipv6_rs from interface ge-0/0/1
set security nat source rule-set myipv6_rs to interface ge-0/0/2
set security nat source rule-set myipv6_rs rule ipv6_rule match source-address
  10.10.10.1/30
set security nat source rule-set myipv6_rs rule ipv6_rule match destination-address
  322f::25
set security nat source rule-set myipv6_rs rule ipv6_rule then source-nat pool myipv6_prefix
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure an IPv4-initiated connection to an IPv6 node using static destination address one-to-one mapping:

1. Configure the static NAT rule set for an interface.

```
[edit security nat static]
user@host# set rule-set test_rs from interface ge-0/0/1
```

2. Define the rule and the destination address.

```
[edit security nat static rule-set test_rs]
user@host# set rule test_rule match destination-address 10.1.1.25/32
```

3. Define the static NAT prefix.
 

```
[edit security nat static rule-set test_rs]
user@host# set rule test_rule then static-nat prefix 3ffe::25/128
```
4. Configure a source NAT pool with an IPv6 prefix address.
 

```
[edit security]
user@host# set nat source pool myipv6_prefix address 27a6::/96
```
5. Configure the source NAT rule set.
 

```
[edit security nat source]
user@host# set rule-set myipv6_rs from interface ge-0/0/1
user@host# set rule-set myipv6_rs from interface ge-0/0/2
```
6. Configure the source NAT source address.
 

```
[edit security nat source rule-set myipv6_rs]
user@host# set rule ipv6_rule match source-address 10.10.10.1/30
```
7. Configure the source NAT destination address.
 

```
[edit security nat source rule-set myipv6_rs]
user@host# set rule ipv6_rule match destination-address 322f::25
```
8. Define a configured source NAT IPv6 pool in the rule.
 

```
[edit security nat source rule-set myipv6_rs]
user@host# set rule ipv6_rule then source-nat pool myipv6_prefix
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
netscreen@srx220-tp# show security nat
source {
  pool myipv6_prefix {
    address {
      27a6::/96;
    }
  }
}
rule-set myipv6_rs {
  from interface ge-0/0/1.0;
  to interface ge-0/0/2.0;
  rule ipv6_rule {
    match {
      source-address 10.10.10.1/30;
      destination-address 322f::25/128;
    }
    then {
      source-nat {
        pool {
          myipv6_prefix;
        }
      }
    }
  }
}
```

```

    }
  }
  static {
    rule-set test_rs {
      from interface ge-0/0/1.0;
      rule test_rule {
        match {
          destination-address 10.1.1.25/32;
        }
        then {
          static-nat prefix 3ffe::25/128;
        }
      }
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying That Static NAT Is Configured on page 81
- Verifying That Source NAT Is Configured on page 81

#### *Verifying That Static NAT Is Configured*

**Purpose** Verify whether static NAT is configured with an interface, a destination address, and a prefix.

**Action** From operational mode, enter the **show security nat static** command.

#### *Verifying That Source NAT Is Configured*

**Purpose** Verify whether source NAT is configured.

**Action** From operational mode, enter the **show security nat source** command.

#### Related Documentation

- IPv6 NAT Overview on page 71
- IPv6 NAT PT Overview on page 73
- IPv6 NAT-PT Communication Overview on page 74
- Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Static Destination Address One-to-One Mapping on page 85
- Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Default Destination Address Prefix Static Mapping on page 82
- Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Default Destination Address Prefix Static Mapping on page 75

## Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Default Destination Address Prefix Static Mapping

This example shows how to configure an IPv6-initiated connection to an IPv4 node using default destination address prefix static mapping.

- Requirements on page 82
- Overview on page 82
- Configuration on page 82
- Verification on page 84

### Requirements

Before you begin, configure the interfaces and assign the interfaces to security zones.

### Overview

The following example describes the communication of an IPv6 node with an IPv4 node that has prefix-based static NAT defined on the device. The static NAT assumes that the IPv4 network is a special IPv6 network (that is, an IPv4-mapped IPv6 network), and hides the entire IPv4 network behind an IPv6 prefix.

The communication of an IPv6 node with an IPv4 node is useful when IPv6 is used in the network and must be connected to the IPv4 network, or when both IPv4 and IPv6 are used in the network and a mechanism is required to interconnect the two networks during migration. This also provides connectivity for sessions initiated from IPv6 nodes with IPv4 nodes on the other side of the NAT/PT device.

### Configuration

#### CLI Quick Configuration

To quickly configure an IPv6-initiated connection to an IPv4 node using default destination address prefix static mapping, copy the following commands and paste them into the CLI.

```
[edit]
set security nat static rule-set test_rs from interface ge-0/0/1
set security nat static rule-set test_rs rule test_rule match destination-address 27a6::/96
set security nat static rule-set test_rs rule test_rule then static-nat inet
set security nat source pool myipv4 address 1.1.1.2 to 1.1.1.5
set security nat source rule-set myipv4_rs from interface ge-0/0/1
set security nat source rule-set myipv4_rs to interface ge-0/0/2
set security nat source rule-set myipv4_rs rule ipv4_rule match destination-address
  10.1.1.15/30
set security nat source rule-set myipv4_rs rule ipv4_rule match source-address 2ffe::/96
set security nat source rule-set myipv4_rs rule ipv4_rule then source-nat pool myipv4
```



**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure an IPv6-initiated connection to an IPv4 node using default destination address prefix static mapping:

1. Configure the static NAT for an interface.
 

```
[edit security nat static]
user@host# set rule test_rs from interface ge-0/0/1
```
2. Define the rule and destination address with the prefix for the static NAT translation defined on the device.
 

```
[edit security nat static rule-set test_rs]
user@host# set rule test_rule match destination-address 27a6::/96
```
3. Define the static NAT as inet to translate to an IPv4 address.
 

```
[edit security nat static rule-set test_rs]
user@host# set rule test_rule then static-nat inet
```
4. Configure the IPv4 source NAT pool address.
 

```
[edit security nat source]
user@host# set pool myipv4 address 1.1.1.2 to 1.1.1.5
```
5. Configure the source NAT rule set.
 

```
[edit security nat source ]
user@host# set rule-set myipv4_rs from interface ge-0/0/1
user@host# set rule-set myipv4_rs from interface ge-0/0/2
```
6. Configure the IPv4 source NAT destination address.
 

```
[edit security nat source rule-set myipv4_rs]
user@host# set rule ipv4_rule match destination-address 10.1.1.15/30
```
7. Define the source address with the prefix for the source NAT defined on the device.
 

```
[edit security nat source rule-set myipv4_rs]
user@host# set rule ipv4_rule match source-address 2ffe::/96
```
8. Define a configured source NAT IPv4 pool in the rule.
 

```
[edit security nat source rule-set myipv4_rs]
user@host# sset rule ipv4_rule then source-nat pool myipv4
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
source {
  pool myipv4 {
    address {
      1.1.1.2/32 to 1.1.1.5/32;
    }
  }
}
```

```

}
rule-set myipv4_rs {
  from interface ge-0/0/1.0;
  to interface ge-0/0/2.0;
  rule ipv4_rule {
    match {
      source-address 2ffe::/96;
      destination-address 10.1.1.15/30;
    }
    then {
      source-nat {
        pool {
          myipv4;
        }
      }
    }
  }
}
}
}
static {
  rule-set test_rs {
    from interface ge-0/0/1.0;
    rule test_rule {
      match {
        destination-address 27a6::/96;
      }
      then {
        static-nat inet;
      }
    }
  }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying That Static NAT Is Configured on page 84
- Verifying That Source NAT Is Configured on page 84

#### *Verifying That Static NAT Is Configured*

**Purpose** Verify whether static NAT is configured with an interface, a destination address, and a prefix.

**Action** From operational mode, enter the **show security nat static** command.

#### *Verifying That Source NAT Is Configured*

**Purpose** Verify whether source NAT is configured.

**Action** From operational mode, enter the **show security nat source** command.

- Related Documentation**
- IPv6 NAT Overview on page 71
  - IPv6 NAT PT Overview on page 73
  - IPv6 NAT-PT Communication Overview on page 74
  - Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Static Destination Address One-to-One Mapping on page 85
  - Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Static Destination Address One-to-One Mapping on page 78
  - Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Default Destination Address Prefix Static Mapping on page 75

### Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Static Destination Address One-to-One Mapping

This example shows how to configure an IPv6-initiated connection to an IPv4 node using static destination address one-to-one mapping.

- Requirements on page 85
- Overview on page 85
- Configuration on page 85
- Verification on page 87

#### Requirements

Before you begin, configure the interfaces and assign the interfaces to security zones.

#### Overview

The following example describes the communication of an IPv6 node with an IPv4 node that has a one-to-one static NAT address defined on the device. The communication of an IPv6 node with an IPv4 node allows IPv6 hosts to access an IPv4 server when neither of the devices has a dual stack and must depend on the NAT/PT device to communicate. This enables some IPv4 legacy server applications to work even after the network has migrated to IPv6.

#### Configuration

- CLI Quick Configuration**
- To quickly configure an IPv6-initiated connection to an IPv4 node using static destination address one-to-one mapping, copy the following commands and paste them into the CLI.

```
[edit]
set security nat static rule test_rs from interface ge-0/0/1
set security nat static rule test_rs rule test_rule match destination-address 27a6::15/128
set security nat static rule test_rs rule test_rule then static-nat prefix 10.2.2.15/32
set security nat source pool myipv4 address 1.1.1.2 to 1.1.1.3
set security nat source rule myipv4_rs from interface ge-0/0/1
set security nat source rule myipv4_rs to interface ge-0/0/2
set security nat source rule myipv4_rs rule ipv4_rule match source-address 27a6::/96
set security nat source rule myipv4_rs rule ipv4_rule match destination-address 10.2.2.15
set security nat source rule myipv4_rs rule ipv4_rule then source-nat pool myipv4
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure an IPv6-initiated connection to an IPv4 node using static destination address one-to-one mapping:

1. Configure the static NAT rule set for an interface.

```
[edit security nat static]
user@host# set rule-set test_rs from interface ge-0/0/1
```

2. Define a rule to match the destination address.

```
[edit security nat static rule-set test_rs]
user@host# set rule test_rule match destination-address 27a6::15/128
```

3. Define the static NAT prefix to the rule.

```
[edit security nat static rule-set test_rs]
user@host# set rule test_rule then static-nat prefix 10.2.2.15/32
```

4. Configure a source NAT pool with an IPv4 addresses.

```
[edit security nat]
user@host# set source pool myipv4 address 1.1.1.2 1.1.1.3
```

5. Configure the IPv4 address for the interface.

```
[edit security nat source ]
user@host# set rule-set myipv4_rs from interface ge-0/0/1
```

6. Configure the source address to the IPv4 source NAT address.

```
[edit security nat source rule-set myipv4_rs]
user@host# set rule ipv4_rule match source-address 27a6::/96
```

7. Configure the destination address to IPv4 source NAT address.

```
[edit security nat source rule-set myipv4_rs]
user@host# set rule ipv4_rule match destination-address 10.2.2.15
```

8. Define the configured source NAT IPv4 pool in the rule.

```
[edit security nat source rule-set myipv4_rs]
user@host# set rule ipv4_rule then source-nat pool myipv4
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
source {
  pool myipv4 {
    address {
      1.1.1.2/32 to 1.1.1.3/32;
    }
  }
}
rule-set myipv4_rs {
```

```

from interface ge-0/0/1.0;
to interface ge-0/0/2.0;
rule ipv4_rule {
  match {
    source-address 27a6::/96;
    destination-address 10.2.2.15/32;
  }
  then {
    source-nat {
      pool {
        myipv4;
      }
    }
  }
}
}
static {
  rule-set test_rs {
    from interface ge-0/0/1.0;
    rule test_rule {
      match {
        destination-address 27a6::15/128;
      }
      then {
        static-nat prefix 10.2.2.15/32;
      }
    }
  }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying That Static NAT Is Configured on page 87
- Verifying That Source NAT Is Configured on page 87

#### *Verifying That Static NAT Is Configured*

**Purpose** Verify whether static NAT is configured with an interface, a destination address, and a prefix.

**Action** From operational mode, enter the **show security nat static** command.

#### *Verifying That Source NAT Is Configured*

**Purpose** Verify whether source NAT is configured.

**Action** From operational mode, enter the **show security nat source** command.

- Related Documentation**
- IPv6 NAT Overview on page 71
  - IPv6 NAT PT Overview on page 73
  - IPv6 NAT-PT Communication Overview on page 74
  - Example: Configuring an IPv6-Initiated Connection to an IPv4 Node Using Default Destination Address Prefix Static Mapping on page 82
  - Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Static Destination Address One-to-One Mapping on page 78
  - Example: Configuring an IPv4-Initiated Connection to an IPv6 Node Using Default Destination Address Prefix Static Mapping on page 75

## IPv6 ALGs

---

- IPv6 DNS ALG for Routing, NAT, and NAT-PT on page 88
- IPv6 FTP ALG for Routing on page 90
- Understanding IPV6 ALG support for ICMP on page 91

### IPv6 DNS ALG for Routing, NAT, and NAT-PT

Domain Name System (DNS) is the part of the ALG that handles DNS traffic, monitors DNS query and reply packets, and closes the session if the DNS flag indicates the packet is a reply message.

The DNS ALG module has been working as expected for IPv4. In Junos OS Release 10.4, this feature implements IPv6 support on DNS ALG for routing, Network Address Translation (NAT), and Network Address Translation-Protocol Translation (NAT-PT).

When DNS ALG receives a DNS query from the DNS client, a security check is done on the DNS packet. When the DNS ALG receives a DNS reply from the DNS server, a similar security check is done, and then the session for the DNS traffic closes.

#### IPv6 DNS ALG Traffic in NAT mode

---

IPv6 NAT provides address translation between IPv4 and IPv6 addressed network devices. It also provides address translation between IPv6 hosts. NAT between IPv6 hosts is done in a similar manner and for similar purposes as IPv4 NAT.

When the DNS traffic works in NAT mode, the DNS ALG translates the public address in a DNS reply to a private address when the DNS client is on private network, and similarly translates a private address to a public address when the DNS client is on a public network.

In Junos OS Release 10.4 IPv6 NAT supports:

- Source NAT translations
- Destination NAT mappings
- Static NAT mappings



**NOTE:** IPv6 DNS ALG NAT supports only static NAT mapping.

### IPv6 DNS ALG Traffic in NAT-PT mode

IPv6 NAT-PT provides address allocation and protocol translation between IPv4 and IPv6 addressed network devices. The translation process is based on the Stateless IP/ICMP Translation (SIIT) method; however, the state and the context of each communication is retained during the session lifetime. IPv6 NAT-PT supports Internet Control Message Protocol (ICMP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) packets.

IPv6 NAT-PT supports the following types of NAT-PT:

- Traditional NAT-PT
- Bidirectional NAT-PT

A DNS-based mechanism dynamically maps IPv6 addresses to IPv4-only servers. NAT-PT uses the DNS ALG to transparently do the translations.

For example, a company using an internal IPv6 network needs to be able to communicate with external IPv4 servers that do not have IPv6 addresses.

To support the dynamic address binding, a DNS should be used for name resolution. The IPv4 host looks up the name of the IPv6 node in its local configured IPv4 DNS server, which then passes the query to the IPv6 DNS server through an SRX Series or a J Series device using NAT-PT.

When DNS traffic works in NAT-PT mode, the DNS ALG translates the IP address in a DNS reply packet between the IPv4 address and the IPv6 address when the DNS client is in an IPv6 network and the server is in an IPv4 network, and vice versa.



**NOTE:** In NAT-PT mode, only IPv4 to IPv6 addresses translation is supported in DNS ALG. To support NAT-PT mode in a DNS ALG, the NAT module should support NAT-PT.

When the DNS ALG receives a DNS query from the DNS client, the DNS ALG performs the following security and sanity checks on the DNS packets:

- Enforces the maximum DNS message length (the default is 512 bytes and the maximum length is 8KB)
- Enforces a domain-name length of 255 bytes and a label length of 63 bytes
- Verifies the integrity of the domain-name referred to by the pointer if compression pointers are encountered in the DNS message
- Checks to see if a compression pointer loop exists

Similar sanity checks are done when the DNS ALG receives a DNS reply from the DNS Server, after which the session for this DNS traffic gets closed.

## IPv6 FTP ALG for Routing

File Transfer Protocol (FTP) is the part of the ALG that handles FTP traffic. The PORT/PASV requests and corresponding 200/227 responses in FTP are used to announce the TCP port, which the host listens to for the FTP data connection.

EPRT/EPSV/229 commands are used for these requests and responses. FTP ALG supports EPRT/EPSV/229 already, but only for IPv4 addresses.

In Junos OS Release 10.4, EPRT/EPSV/229 commands have been updated to support both IPv4 and IPv6 addresses.

FTP ALG uses preallocated objcache to store its session cookies. When both IPv4 and IPv6 addresses are supported on FTP ALG, the session cookie structure will enlarge by 256 bits (32 bytes) to store IPv6 address.

### EPRT mode

The EPRT command allows for the specification of an extended address for the data connection. The extended address must consist of the network protocol as well as the network and transport addresses.

The format of EPRT is:

```
EPRT <space><d><net-prt><d><net-addr><d><tcp-port><d>
```

- <net-prt>: An address family number defined by IANA
- <net-addr>: A protocol specific string of the network address
- <tcp-port>: A TCP port number

The following are sample EPRT commands for IPv6:

```
EPRT |2|1080::8:800:200C:417A|5282|
```

In this mode, FTP ALG focuses only on the EPRT command; it extracts the IPv6 address and port from the EPRT command and opens the pinhole.

### EPSV mode

The EPSV command requests that a server be listening on a data port and waiting for a connection. The response to this command includes only the TCP port number of the listening connection.

An example response string is follows:

```
Entering Extended Passive Mode (||6446|)
```



**NOTE:** The response code for entering passive mode using an extended address must be 229. You should extract the TCP port in 229 payloads and use it to open the pinhole.



## Understanding IPV6 ALG support for ICMP

The Internet Control Message Protocol (ICMP) Application Layer Gateway (ALG) is one of the ALG's that handle ICMP traffic.

IPv6 nodes use the ICMPv6 protocol to report errors encountered in processing packets and to perform other Internet-layer functions such as diagnostics. ICMPv6 is an integral part of IPv6 and must be fully implemented by every IPv6 node; therefore the ALG layer is always enabled for ICMPv6.

### ICMP Error Messages

---

ICMPv6 messages are grouped into two classes:

- ICMPv6 error messages
  - Destination unreachable
  - Packet too big
  - Time exceeded
  - Parameter problem
- ICMPv6 informational (or ping) messages
  - Echo request
  - Echo reply

The ICMP ALG monitors all these messages, and then does the following :

- Closes the session
- Modifies the payload

The ICMP ALG closes a session if it meets the following conditions:

- Receives echo reply message.
- Receives a destination unreachable error message and has not received any replies yet.



**NOTE:** The ICMP ALG checks if the session has received any replies from destination node. If it has received any reply , the destination should be reachable and the ICMP error message is not credible, therefore it does not close the session. This is to avoid hackers from sniffing the TCP/UDP packet and forging an ICMP destination unreachable packet to kill the session.

### ICMP ALG Functionality

---

ICMP ALG behaves differently in various modes.

ICMP ALG functionality in NAT mode:

1. Close the session.
2. Modify the identifier, the sequence number or both of the echo request.
3. Resume the original identifier and sequence number for the echo reply.
4. NAT translates the embedded IPv6 packet for the ICMPv6 error message.

ICMP ALG functionality in NAT-PT support mode:

1. Close the session.
2. Translate the ICMPv4 ping message to the ICMPv6 ping message.
3. Translate the ICMPv6 ping message to the ICMPv4 ping message.
4. Translate the ICMPv4 error message to the ICMPv6 error message and translate its embedded IPv4 packet to an IPv6 packet.
5. Translate the ICMPv6 error message to the ICMPv4 error message and translate its embedded IPv6 packet to an IPv4 packet .

**Related  
Documentation**

- Understanding How SRX Series and J-series Devices Handle Packet Fragmentation for IPv6 Flows on page 62
- Understanding IP Version 6 (IPv6) on page 48

## CHAPTER 3

# Introducing Junos OS for J Series Services Routers

- Understanding Stateful and Stateless Data Processing for J Series Services Routers on page 93
- Session Characteristics for J Series Services Routers on page 97
- Understanding the Data Path for J Series Services Routers on page 104

## Understanding Stateful and Stateless Data Processing for J Series Services Routers

Junos OS for J Series Services Routers integrates the world-class network security and routing capabilities of Juniper Networks Operating System.

Traffic that enters and exits a services router running Junos OS is processed according to features you configure, such as security policies, packet filters, and screens. For example, the software can determine:

- Whether the packet is allowed into the router
- Which class of service (CoS) to apply to the packet, if any
- Which firewall screens to apply to the packet
- Whether to send the packet through an IPsec tunnel
- Whether the packet requires an Application Layer Gateway (ALG)
- Whether to apply Network Address Translation (NAT) to translate the packet's address
- Which route the packet uses to reach its destination

Packets that enter and exit a services router running Junos OS undergo both packet-based and flow-based processing. A device always processes packets discretely. Packet treatment depends on characteristics that were established for the first packet of the packet stream.

Branch devices implement both packet-based and flow-based modes, concurrently. Flow-based and packet-based processing are described in the following sections:

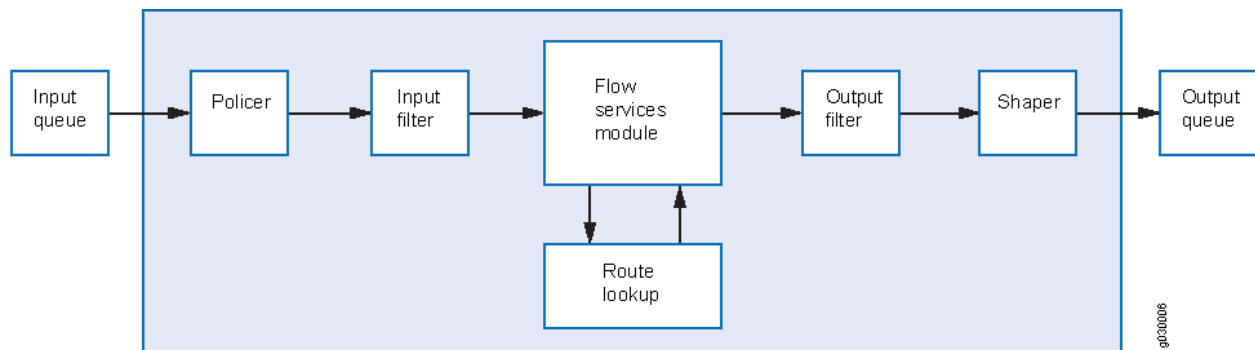
- Understanding Flow-Based Processing on page 94
- Understanding Packet-Based Processing on page 96

## Understanding Flow-Based Processing

A packet undergoes flow-based processing after any packet-based filters and policers have been applied to it.

Figure 6 on page 94 shows an architectural overview of traffic flow in a Juniper Networks device running Junos OS. See “Understanding the Data Path for J Series Services Routers” on page 104 to follow the path of the traffic as it traverses through the flow services module.

**Figure 6: Traffic Flow for Flow-Based Processing**



A flow is defined as a set of packets coming from the same source/destination addresses, source/destination ports (when applicable), protocol, and ingress/egress zones. Flows are time bound so it is possible to have packets that, while fitting the previous definition, belong to different flows. For example, when an existing session is initiated and terminated, after which a new session is established using the exact same parameters as the previous session, the packets would belong to different flows.

Configuration settings that determine the fate of a packet—such as the security policy that applies to it, whether the packet is sent through an IPsec tunnel, if it requires an Application Layer Gateway (ALG), if Network Address Translation (NAT) is applied to translate the packet's address—are assessed for the first packet of a flow. The settings are then applied to the rest of the packets in the flow.

To determine if a packet belongs to an existing flow, the router attempts to match the packet's information to that of an existing flow based on the following six match criteria:

- Source address
- Destination address
- Source port
- Destination port
- Protocol
- Session token—An internal parameter not extracted from the packet's header

If the packet matches an existing flow, processing for the packet is determined by the flow state (maintained by the flow's session). If the packet does not match the session

for an existing flow, the packet's information is used to create a new flow state and a session is allocated for it (a session is allocated only if this is permitted by the security policy). Sessions used for the first packet of a flow is cached in a flow table for use with the same flow and closely related flows.



**NOTE:** A new session is allocated for the new flow state only if this is permitted by the security policy. For TCP, only SYN packets will trigger creating a new session (unless SYN checking is not enabled).

### Zones and Policies

Security policies are associated with zones. A zone is a collection of interfaces that define a security boundary. A packet's incoming zone, as determined by the interface through which it arrived, and its outgoing zone, as determined by the forwarding lookup, together determine which policy is used for packets of the flow.

### Flows and Sessions

Flow-based packet processing, which is stateful, requires the creation of sessions. A session is created, based on the characteristics assessed for the first packet of a flow, for the following purposes:

- To store the security measures to be applied to the packets of the flow
- To cache information about the state of the flow

For example, logging and counting information for a flow is cached in its session. (Some stateful firewall screens rely on threshold values that pertain to individual sessions or across all sessions.)

- To allocate required resources for the flow for features such as Network Address Translation (NAT) and IPsec tunnels
- To provide a framework for features such as Application Layer Gateways (ALGs) and firewall features

Most packet processing occurs in the context of a flow. The flow engine and session bring together the following features and events that affect a packet as it undergoes flow-based processing:

- Flow-based forwarding
- Session management, including session aging and changes in routes, policy, and interfaces
- Management of virtual private networks (VPNs), ALGs, and authentication
- Management of policies, NAT, zones, and screens

Policies can be configured to log session permit, close, and deny events.

## Understanding Packet-Based Processing

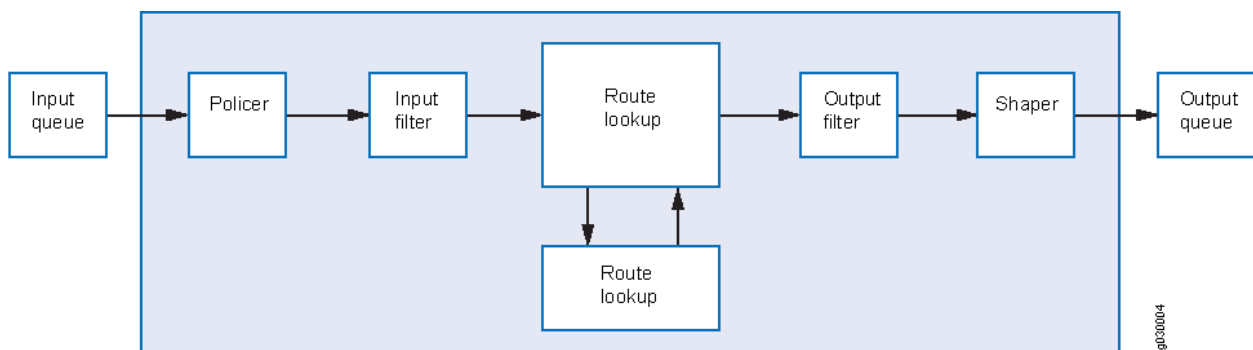
A packet undergoes packet-based processing when it is dequeued from its input (ingress) interface and before it is enqueued on its output (egress) interface.

Packet-based processing applies stateless firewall filters and class-of-service (CoS) features to discrete packets. You can apply a firewall filter to an ingress or egress interface, or to both.

- When a packet arrives at an interface on the device, any packet-based filters and policers associated with the interface are applied to the packet before any security policies are evaluated.
- Before a packet leaves the device, any packet-based filters and traffic shapers associated with the output interface are applied to the packet after any security policies have been evaluated.

Figure 7 on page 96 shows an architectural overview of traffic flow in a Juniper Networks device running Junos OS.

**Figure 7: Traffic Flow for Packet-Based Processing**



Filters and CoS features are typically associated with one or more interfaces to influence which packets are allowed to transit the system and to apply special actions to packets as necessary.



**NOTE:** Packet-based processing occurs only if you configure filters, CoS, IPv6, and MPLS features for an interface that handles the packet.

The following sections describe the kinds of packet-based features that you can configure and apply to transit traffic. For details on specific stateless firewall filters and CoS features, see the *Junos OS Routing Protocols and Policies Configuration Guide for Security Devices*, the *Junos OS Class of Service Configuration Guide for Security Devices*, and the *Junos OS CLI Reference*.

### Stateless Firewall Filters

Also referred to as access control lists (ACLs), stateless firewall filters control access and limit traffic rates. They statically evaluate the contents of packets transiting the

device from a source to a destination, or packets originating from or destined for the Routing Engine. A stateless firewall filter evaluates every packet, including fragmented packets.

You can apply a stateless firewall filter to an input or output interface, or to both. A filter contains one or more terms, and each term consists of two components—match conditions and actions. By default, a packet that does not match a firewall filter is discarded.

You can plan and design stateless firewall filters to be used for various purposes—for example, to limit traffic to certain protocols, IP source or destination addresses, or data rates.

### Class-of-Service Features

CoS features allow you to police and shape traffic.

- **Policing traffic**—Policers allow you to limit traffic of a certain class to a specified bandwidth and burst size. Packets exceeding the policer limits can be discarded or assigned to a different forwarding class, a different loss priority, or both. You can use policers to limit the amount of traffic passing into or out of an interface.
- **Traffic shaping**—You can shape traffic by assigning service levels with different delay, jitter, and packet loss characteristics to particular applications served by specific traffic flows. Traffic shaping is especially useful for real-time applications, such as voice and video transmission.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Session Characteristics for J Series Services Routers on page 97](#)
- [Understanding the Data Path for J Series Services Routers on page 104](#)
- [Monitoring Policy Statistics on page 177](#)
- [ALG Overview on page 217](#)
- [NAT Overview on page 1335](#)

## Session Characteristics for J Series Services Routers

- [Understanding Session Characteristics for J Series Services Routers on page 97](#)
- [Example: Controlling Session Termination for J Series Services Routers on page 98](#)
- [Example: Disabling TCP Packet Security Checks for J Series Services Routers on page 101](#)
- [Example: Accommodating End-to-End TCP Communication for J Series Services Routers on page 102](#)

### Understanding Session Characteristics for J Series Services Routers

Sessions are created, based on routing and other classification information, to store information and allocate resources for a flow. Sessions have characteristics, some of which you can change, such as when they are terminated. For example, you might want

to ensure that a session table is never entirely full to protect against an attacker's attempt to flood the table and thereby prevent legitimate users from starting sessions.

Depending on the protocol and service, a session is programmed with a timeout value. For example, the default timeout for TCP is 1800 seconds. The default timeout for UDP is 60 seconds. When a flow is terminated, it is marked as invalid, and its timeout is reduced to 10 seconds.

If no traffic uses the session before the service timeout, the session is aged out and freed to a common resource pool for reuse. You can affect the life of a session in the following ways:

- You can specify circumstances for terminating sessions using any of the following methods:
  - Aggressively age out invalid sessions based on a timeout value
  - Age out sessions based on how full the session table is
  - Set an explicit timeout for aging out TCP sessions
  - Configure a TCP session to be invalidated when it receives a TCP RST (reset) message
- You can configure sessions to accommodate other systems as follows:
  - Disable TCP packet security checks
  - Accommodate end-to-end communication

The following topics show you how to modify a session's characteristics. For details, see the *Junos OS CLI Reference*.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Stateful and Stateless Data Processing for J Series Services Routers on page 93](#)
- [Example: Controlling Session Termination for J Series Services Routers on page 98](#)
- [Example: Disabling TCP Packet Security Checks for J Series Services Routers on page 101](#)
- [Example: Accommodating End-to-End TCP Communication for J Series Services Routers on page 102](#)

### Example: Controlling Session Termination for J Series Services Routers

This example shows how to terminate sessions based on a timeout value or the number of sessions in the session table.

- [Requirements on page 99](#)
- [Overview on page 99](#)
- [Configuration on page 99](#)
- [Verification on page 100](#)



## Requirements

Before you begin:

- Configure security zones. See “Security Zones and Interfaces Overview” on page 111.
- Configure security policies. See “Security Policies Configuration Overview” on page 151.

## Overview

Junos OS terminates sessions normally under certain circumstances—for example, after receiving a TCP FINish Close or a RST (reset) message, when encountering ICMP errors for UDP, and when no matching traffic is received before the service timeout. When sessions are terminated, their resources are freed up for use by other sessions.

To control when sessions are terminated, you configure the router to age out sessions after a certain period of time, when the number of sessions in the session table reaches a specified percentage, or both. When the number of sessions in the session table reaches this percentage, the router begins to age sessions aggressively. When the number of sessions in the session table reaches the low-water mark, the router stops aggressively aging sessions.

## Configuration

### CLI Quick Configuration

To quickly terminate sessions based on a timeout value or the number of sessions in the session table, copy the following commands and paste them into the CLI:

```
[edit]
```

```
set security flow aging early-ageout 2
```

```
set security flow aging high-watermark 90 low-watermark 50
```

```
set security flow tcp-session tcp-initial-timeout 280
```

```
set security flow tcp-session rst-invalidate-session
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To terminate sessions based on a timeout value or the number of sessions in the session table:

1. Specify the number of seconds after which a session is invalidated.

```
[edit security flow]
user@host# set aging early-ageout 2
```

2. Specify a percentage of sessions.

```
[edit security flow]
user@host# set aging high-watermark 90 low-watermark 50
```

3. Configure an explicit timeout value to remove a TCP session from the session table.

```
[edit security flow]
```

```
user@host# set tcp-session tcp-initial-timeout 280
```

4. Configure any session that receives a TCP RST message to be invalidated.

```
[edit security flow]
```

```
user@host# set tcp-session rst-invalidate-session
```

**Results** From configuration mode, confirm your configuration by entering the **show security flow** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security flow
aging {
  early-ageout 2;
  low-watermark 50;
  high-watermark 90;
}
tcp-session {
  rst-invalidate-session;
  tcp-initial-timeout 280;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Troubleshooting with Logs on page 100

#### *Troubleshooting with Logs*

**Purpose** Use these logs to identify any issues.

**Action** From operational mode, enter the **show log messages** command and the **show log dcd** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Session Characteristics for J Series Services Routers on page 97](#)
- [Example: Accommodating End-to-End TCP Communication for J Series Services Routers on page 102](#)
- [Example: Disabling TCP Packet Security Checks for J Series Services Routers on page 101](#)

## Example: Disabling TCP Packet Security Checks for J Series Services Routers

This example shows how to disable TCP SYN checks and TCP sequence checks on all TCP sessions.

- Requirements on page 101
- Overview on page 101
- Configuration on page 101
- Verification on page 101

### Requirements

Before you begin, review TCP packets and security checks. See *Junos OS Feature Support Reference for SRX Series and J Series Devices*.

### Overview

Junos OS provides a mechanism to disable security checks on TCP packets to ensure interoperability with hosts and routers with faulty TCP implementations.

### Configuration

#### Step-by-Step Procedure

To disable TCP SYN checks and TCP sequence checks on all TCP sessions:

1. Disable TCP SYN checks on all TCP sessions.
 

```
[edit security flow]
user@host# set tcp-session no-syn-check
```
2. Disable TCP sequence checks on all TCP sessions.
 

```
[edit security flow]
user@host# set tcp-session no-sequence-check
```
3. If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security flow** command.

#### Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Session Characteristics for J Series Services Routers on page 97
- Example: Controlling Session Termination for J Series Services Routers on page 98
- Example: Accommodating End-to-End TCP Communication for J Series Services Routers on page 102

## Example: Accommodating End-to-End TCP Communication for J Series Services Routers

This example shows how to change the maximum segment size (MSS) for TCP packets to be sent or received over GRE and IPsec tunnels.

- Requirements on page 102
- Overview on page 102
- Configuration on page 102
- Verification on page 104

### Requirements

---

Before you begin, review TCP packets and security checks. See *Junos OS Feature Support Reference for SRX Series and J Series Devices*.

### Overview

---

End-to-end TCP communication in a customer network might not work for large packets approaching 1500 bytes because of GRE or IPsec tunneling encapsulation. You can configure sessions to accommodate other systems and segment sizes.

### Configuration

---

#### CLI Quick Configuration

To quickly change the MSS for TCP packets to be sent or received over GRE and IPsec tunnels, copy the following commands and paste them into the CLI:

[edit ]

```
set security flow tcp-mss ipsec-vpn mss 1400
```

```
set security flow tcp-mss gre-in mss 1364
```

```
set security flow tcp-mss gre-out mss 1364
```

```
set security flow tcp-mss all-tcp mss 1400
```

```
set security flow allow-dns-reply
```

```
set security flow route-change-timeout 62
```

```
set security flow syn-flood-protection-mode syn-proxy
```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To change the MSS for TCP packets to be sent or received over GRE and IPsec tunnels:

1. Set the tunnel sessions.

[edit security flow]

```
user@host# set tcp-mss ipsec-vpn mss 1400
```

```
user@host# set tcp-mss gre-in mss 1364
```

```
user@host# set tcp-mss gre-out mss 1364
```

2. Configure TCP MSS for all TCP sessions.

```
[edit security flow]
```

```
user@host# set tcp-mss all-tcp mss 1400
```

3. Allow an unmatched incoming DNS reply packet.

```
[edit security flow]
```

```
user@host# set allow-dns-reply
```

4. Set the timeout value for route change to nonexistent route.

```
[edit security flow]
```

```
user@host# set route-change-timeout 62
```

5. Enable TCP SYN flood protection mode.

```
[edit security flow]
```

```
user@host# set syn-flood-protection-mode syn-proxy
```

**Results** From configuration mode, confirm your configuration by entering the **show security flow** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security flow
allow-dns-reply;
route-change-timeout 62;
syn-flood-protection-mode syn-proxy;
tcp-mss {
  all-tcp {
    mss 1400;
  }
  ipsec-vpn {
    mss 1400;
  }
  gre-in {
    mss 1364;
  }
  gre-out {
    mss 1364;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform this task:

- Troubleshooting with Logs on page 104

### Troubleshooting with Logs

**Purpose** Use these logs to identify any issues.

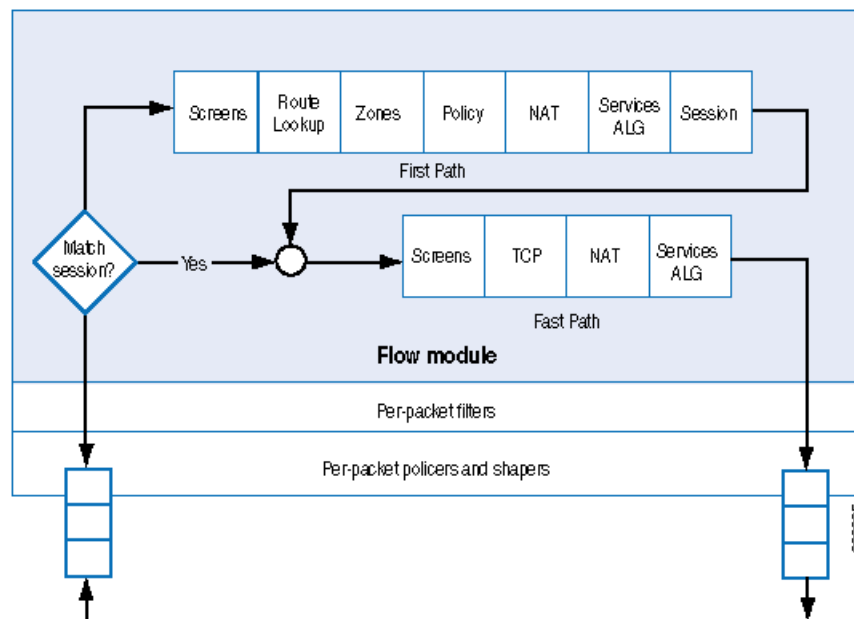
**Action** From operational mode, enter the **show log messages** command and the **show log dcd** command.

- Related Documentation**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Example: Disabling TCP Packet Security Checks for J Series Services Routers on page 101
  - Example: Controlling Session Termination for J Series Services Routers on page 98
  - Understanding Session Characteristics for J Series Services Routers on page 97

## Understanding the Data Path for J Series Services Routers

Figure 8 on page 104 shows the path of a data packet as it traverses the services router. Refer to “Understanding Stateful and Stateless Data Processing for J Series Services Routers” on page 93 to see how the flow module in Figure 8 on page 104 fits in with the Junos operating system (Junos OS) architecture of the software.

Figure 8: Data Packet Traversing the Flow Module on the Services Router



As a packet transits the router, it takes the following path. This packet walk brings together the packet-based processing and flow-based processing that Junos OS performs on the packet.

- Understanding the Forwarding Processing on page 105
- Understanding the Session-Based Processing on page 105
- Understanding Forwarding Features on page 107

## Understanding the Forwarding Processing

Junos OS performs forwarding processing as follows:

1. The packet enters the system and is treated on a per-packet basis.
2. The system applies stateless policing filters and class-of-service (CoS) classification to the packet.

For details, see the *Junos OS Routing Protocols and Policies Configuration Guide for Security Devices*, the *Junos OS Class of Service Configuration Guide for Security Devices*, and the *Junos OS CLI Reference*.

## Understanding the Session-Based Processing

After forwarding processing, Junos OS performs session lookup and either first-packet processing or fast-path processing on the packet.

### Session Lookup

---

If the packet has not already been dropped, Junos OS performs session lookup to determine whether the packet belongs to an existing session. The system uses six match criteria to perform the session lookup:

- Session token
- Source and destination IP addresses
- Source and destination ports
- Protocol

If the packet does not match an existing session, the system creates a new session for it. This process is called the first-packet path. (See “First-Packet Path Processing” on page 105.)

If the packet matches a session, fast-path processing is performed. (See “Fast-Path Processing” on page 106.)

### First-Packet Path Processing

---

If a packet does not match an existing session, Junos OS creates a new session for it as follows:

1. For the first packet, the system creates a session based on the routing for the packet and the policy lookup so that the packet becomes the first packet of a flow.
2. Depending on the protocol and whether the service is TCP or UDP, the session is programmed with a timeout value.
  - For TCP, the default timeout is 1800 seconds.
  - For UDP, the default timeout is 60 seconds.

You can configure these timeouts to be more or less aggressive. If you have changed the session timeout value, the new value is applied here. If no traffic uses the session during the service timeout period, the router ages out the session and releases its memory for reuse.

3. Firewall screens are applied.

Session initialization screens are applied.
4. Route lookup is performed.
5. The destination zone is determined:
  - a. The system determines a packet's *incoming* zone by the interface through which it arrives.
  - b. The system determines a packet's *outgoing* zone by route lookup.

Together they determine which policy is applied to the packet.
6. Policy lookup is performed.

The system checks the packet against policies you have defined to determine how the packet is to be treated.
7. If Network Address Translation (NAT) is used, the system performs address allocation.
8. The system sets up the Application Layer Gateway (ALG) service vector.
9. The system creates and installs the session.

Decisions made for the first packet of a flow are cached in a flow table for use with following related flows.

For example, the system determines asymmetric traffic by doing a reverse route lookup on the packet. If the first packet of a flow has ingress on an interface for a zone, then the reply traffic for this flow needs to egress out of the same interface on which the first packet ingress; otherwise, the traffic is considered asymmetric and will be dropped.

10. Fast-path processing is applied to the packet.

### Fast-Path Processing

---

If a packet matches a session, Junos OS performs fast-path processing as follows:

1. Configured screens are applied.
2. TCP checks are performed.



3. NAT is applied.
4. Forwarding features are applied. See “Understanding Forwarding Features” on page 107.

## Understanding Forwarding Features

After the packet has passed through session-based processing, Junos OS prepares the packet and transmits it as follows:

1. Routing packet filters are applied.
2. Traffic shaping is applied.
3. The packet is transmitted.

For information about packet filters and CoS traffic shaping, see the *Junos OS Class of Service Configuration Guide for Security Devices*.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Stateful and Stateless Data Processing for J Series Services Routers on page 93](#)
- [NAT Overview on page 1335](#)
- [Security Policies Overview on page 145](#)
- [ALG Overview on page 217](#)



## PART 2

# Security Zones and Interfaces

- Security Zones and Interfaces on page 111
- Address Books and Address Sets on page 133



# Security Zones and Interfaces

- Security Zones and Interfaces Overview on page 111
- Security Zones on page 112
- Host Inbound Traffic on page 116
- Protocols on page 121
- TCP-Reset Parameters on page 125
- DNS on page 127

## Security Zones and Interfaces Overview

---

Interfaces act as a doorway through which traffic enters and exits a Juniper Networks device. Many interfaces can share exactly the same security requirements; however, different interfaces can also have different security requirements for inbound and outbound data packets. Interfaces with identical security requirements can be grouped together into a single *security zone*.

A *security zone* is a collection of one or more network segments requiring the regulation of inbound and outbound traffic through policies.

Security zones are logical entities to which one or more interfaces are bound. With many types of Juniper Networks devices, you can define multiple security zones, the exact number of which you determine based on your network needs.

On a single device, you can configure multiple security zones, dividing the network into segments to which you can apply various security options to satisfy the needs of each segment. At a minimum, you must define two security zones, basically to protect one area of the network from the other. On some security platforms, you can define many security zones, bringing finer granularity to your network security design—and without deploying multiple security appliances to do so.

From the perspective of security policies, traffic enters into one security zone and goes out on another security zone. This combination of a **from-zone** and a **to-zone** is defined as a *context*. Each context contains an ordered list of policies. For more information on policies, see “Security Policies Overview” on page 145.

This topic includes the following sections:

- Understanding Security Zone Interfaces on page 112
- Understanding Interface Ports on page 112

## Understanding Security Zone Interfaces

An interface for a security zone can be thought of as a doorway through which TCP/IP traffic can pass between that zone and any other zone.

Through the policies you define, you can permit traffic between zones to flow in one direction or in both. With the routes that you define, you specify the interfaces that traffic from one zone to another must use. Because you can bind multiple interfaces to a zone, the routes you chart are important for directing traffic to the interfaces of your choice.

An interface can be configured with an IPv4 address, IPv6 address, or both.

## Understanding Interface Ports

On J Series Services Routers, interface ports for the system are located on Physical Interface Modules (PIMs) that you can install in slots on the device. In addition, each device has four built-in Gigabit Ethernet ports in slot 0. Each physical port can have many logical interfaces configured with properties different from the port's other logical units.

Interfaces are named by type, slot number, module number (always 0), port number, and the logical unit number. Port numbering starts with 0. Interface names have the following format:

```
type-pim/slot/port.logical-unit-number
```

For example, an interface on port 1 of a T1 PIM installed in slot 3 is named t1-3/0/1. Logical unit 1 on the interface is named t1-3/0/1.1. The built-in Gigabit Ethernet interfaces are named ge-0/0/0 through ge-0/0/3.

For more information about interfaces and interface names, see the [Junos OS Interfaces Configuration Guide for Security Devices](#).

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Functional Zones on page 113
- Understanding Security Zones on page 113
- Example: Creating Security Zones on page 114
- Understanding How to Control Inbound Traffic Based on Traffic Types on page 116

---

## Security Zones

- Understanding Functional Zones on page 113
- Understanding Security Zones on page 113
- Example: Creating Security Zones on page 114

## Understanding Functional Zones

A functional zone is used for special purposes, like management interfaces. Currently, only the management (MGT) zone is supported. Management zones have the following properties:

- Management zones host management interfaces.
- Traffic entering management zones does not match policies; therefore, traffic cannot transit out of any other interface if it was received in the management interface.
- Management zones can only be used for dedicated management interfaces.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Zones and Interfaces Overview on page 111
- Understanding Security Zones on page 113
- Example: Creating Security Zones on page 114

## Understanding Security Zones

Security zones are the building blocks for policies; they are logical entities to which one or more interfaces are bound. Security zones provide a means of distinguishing groups of hosts (user systems and other hosts, such as servers) and their resources from one another in order to apply different security measures to them.

Security zones have the following properties:

- Policies—Active security policies that enforce rules for the transit traffic, in terms of what traffic can pass through the firewall, and the actions that need to take place on the traffic as it passes through the firewall. For more information, see “Security Policies Overview” on page 145.
- Screens—A Juniper Networks stateful firewall secures a network by inspecting, and then allowing or denying, all connection attempts that require passage from one security zone to another. For every security zone, you can enable a set of predefined screen options that detect and block various kinds of traffic that the device determines as potentially harmful. For more information, see “Reconnaissance Deterrence Overview” on page 1019.
- Address books—IP addresses and address sets that make up an address book to identify its members so that you can apply policies to them. Address book entries can include any combination of IPv4 addresses, IPv6 addresses, and Domain Name System (DNS) names. For more information, see “Example: Configuring Address Books and Address Sets” on page 139.
- TCP-RST—When this feature is enabled, the system sends a TCP segment with the RESET flag set when traffic arrives that does not match an existing session and does not have the SYNchronize flag set.
- Interfaces—List of interfaces in the zone.

Security zones have the following preconfigured zone:

- Trust zone—Available only in the factory configuration and is used for initial connection to the device. After you commit a configuration, the trust zone can be overridden.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Zones and Interfaces Overview on page 111
- Understanding Functional Zones on page 113
- Example: Creating Security Zones on page 114

## Example: Creating Security Zones

This example shows how to configure zones and assign interfaces to them. When you configure a security zone, you can specify many of its parameters at the same time.

- Requirements on page 114
- Overview on page 114
- Configuration on page 114
- Verification on page 115

### Requirements

Before you begin, configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

### Overview

An interface for a security zone can be thought of as a doorway through which TCP/IP traffic can pass between that zone and any other zone.



**NOTE:** By default, interfaces are in the null zone. The interfaces will not pass traffic until they have been assigned to a zone.

### Configuration

#### CLI Quick Configuration

To quickly create zones and assign interfaces to them, copy the following commands and paste them into the CLI:

[edit]

```
set interfaces ge-0/0/1 unit 0 family inet address 10.12.12.1/24
```

```
set interfaces ge-0/0/1 unit 0 family inet6 address fa:43::21/96
```

```
set security security-zone ABC interfaces ge-0/0/1.0
```



**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To create zones and assign interfaces to them:

1. Configure an Ethernet interface and assign an IPv4 address to it.  

```
[edit]

user@host# set interfaces ge-0/0/1 unit 0 family inet address 10.12.12.1/24
```
2. Configure an Ethernet interface and assign an IPv6 address to it.  

```
user@host# set interfaces ge-0/0/1 unit 0 family inet6 address fa:43::21/96
```
3. Configure a security zone and assign it to an Ethernet interface.  

```
user@host# set security security-zone ABC interfaces ge-0/0/1.0
```

**Results** From configuration mode, confirm your configuration by entering the **show security zones security-zone ABC** and **show interfaces ge-0/0/1** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
user@host# show security zones security-zone ABC
...
  interfaces {
    ge-0/0/1.0 {
      ...
    }
  }

[edit]
user@host# show interfaces ge-0/0/1
...
  unit 0 {
    family inet {
      address 10.12.12.1/24;
    }
    family inet6 {
      address fe:43::21/96;
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Troubleshooting with Logs on page 116

### ***Troubleshooting with Logs***

**Purpose** Use these logs to identify any issues.

**Action** From operational mode, enter the **show log messages** command and the **show log dcd** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Security Zones and Interfaces Overview on page 111
  - Understanding Functional Zones on page 113
  - Understanding Security Zones on page 113

## Host Inbound Traffic

---

- Understanding How to Control Inbound Traffic Based on Traffic Types on page 116
- Supported System Services for Host Inbound Traffic on page 117
- Example: Controlling Inbound Traffic Based on Traffic Types on page 118

### Understanding How to Control Inbound Traffic Based on Traffic Types

This topic describes how to configure zones to specify the kinds of traffic that can reach the device from systems that are directly connected to its interfaces.

Note the following:

- You can configure these parameters at the zone level, in which case they affect all interfaces of the zone, or at the interface level. (Interface configuration overrides that of the zone.)
- You must enable all expected host-inbound traffic. Inbound traffic destined to this device is dropped by default.
- You can also configure a zone's interfaces to allow for use by dynamic routing protocols.

This feature allows you to protect the device against attacks launched from systems that are directly or indirectly connected to any of its interfaces. It also enables you to selectively configure the device so that administrators can manage it using certain applications on certain interfaces. You can prohibit use of other applications on the same or different interfaces of a zone. For example, most likely you would want to ensure that outsiders not use the Telnet application from the Internet to log into the device because you would not want them connecting to your system.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Security Zones and Interfaces Overview on page 111
  - Supported System Services for Host Inbound Traffic on page 117
  - Understanding How to Identify Duplicate Sessions Using the TCP-Reset Parameter on page 125

- Example: Controlling Inbound Traffic Based on Traffic Types on page 118

## Supported System Services for Host Inbound Traffic

This topic describes the supported system services for host inbound traffic on the specified zone or interface.

For example, suppose a user whose system was connected to interface **1.3.1.4** in zone **ABC** wanted to telnet into interface **2.1.2.4** in zone **ABC**. For this action to be allowed, the Telnet application must be configured as an allowed inbound service on both interfaces and a policy must permit the traffic transmission.

Table 11 on page 117 shows the system services that can be used for host inbound traffic.

**Table 11: System Services for Host Inbound Traffic**

| Host Inbound System Services |                |
|------------------------------|----------------|
| all                          | any-service    |
| dns                          | finger         |
| ftp                          | http           |
| https                        | indent-reset   |
| ike                          | netconf        |
| ntp                          | ping           |
| reverse-ssh                  | reverse-telnet |
| rlogin                       | rpm            |
| rsh                          | sip            |
| snmp                         | snmp-trap      |
| ssh                          | telnet         |
| tftp                         | traceroute     |
| xnm-clear-text               | xnm-ssl        |

Table 12 on page 118 shows the supported protocols that can be used for host inbound traffic.

Table 12: Protocols for Host Inbound Traffic

| Protocols |       |
|-----------|-------|
| all       | bfd   |
| bgp       | dvmrp |
| igmp      | msdp  |
| ndp       | nhrp  |
| ospf      | ospf3 |
| pgm       | pim   |
| rip       | ripng |
| sap       | vrrp  |



**NOTE:** All services (except DHCP and BOOTP) can be configured either per zone or per interface. A DHCP server is configured only per interface because the incoming interface must be known by the server to be able to send out DHCP replies.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding How to Control Inbound Traffic Based on Traffic Types](#) on page 116
- [Example: Controlling Inbound Traffic Based on Traffic Types](#) on page 118

### Example: Controlling Inbound Traffic Based on Traffic Types

This example shows how to configure inbound traffic based on traffic types.

- [Requirements](#) on page 118
- [Overview](#) on page 119
- [Configuration](#) on page 119
- [Verification](#) on page 121

#### Requirements

Before you begin:

- Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Understand Inbound traffic types. See “Understanding How to Control Inbound Traffic Based on Traffic Types” on page 116.

## Overview

By allowing system services to run, you can configure zones to specify different types of traffic that can reach the device from systems that are directly connected to its interfaces. You can configure the different system services at the zone level, in which case they affect all interfaces of the zone, or at the interface level. (Interface configuration overrides that of the zone.)

You must enable all expected host-inbound traffic. Inbound traffic from devices directly connected to the device's interfaces is dropped by default.

## Configuration

### CLI Quick Configuration

To quickly configure inbound traffic based on traffic types, copy the following commands and paste them into the CLI:

```
[edit]
```

```
set security zones security-zone ABC host-inbound-traffic system-services all
```

```
set security zones security-zone ABC interfaces ge-0/0/1.3 host-inbound-traffic
system-services telnet
```

```
set security zones security-zone ABC interfaces ge-0/0/1.3 host-inbound-traffic
system-services ftp
```

```
set security zones security-zone ABC interfaces ge-0/0/1.3 host-inbound-traffic
system-services snmp
```

```
set security zones security-zone ABC interfaces ge-0/0/1.0 host-inbound-traffic
system-services all
```

```
set security zones security-zone ABC interfaces ge-0/0/1.0 host-inbound-traffic
system-services ftp except
```

```
set security zones security-zone ABC interfaces ge-0/0/1.0 host-inbound-traffic
system-services http except
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see "Using the CLI Editor in Configuration Mode."

To configure inbound traffic based on traffic types:

1. Configure a security zone.

```
[edit]
```

```
user@host# edit security zones security-zone ABC
```

2. Configure the security zone to support inbound traffic for all system services.

```
[edit security zones security-zone ABC]
```

```
user@host# set host-inbound-traffic system-services all
```

3. Configure the Telnet, FTP, and SNMP system services at the interface level (not the zone level) for the first interface.

```
[edit security zones security-zone ABC]
```

```
user@host# set interfaces ge-0/0/1.3 host-inbound-traffic system-services telnet
```

```
user@host# set interfaces ge-0/0/1.3 host-inbound-traffic system-services ftp
```

```
user@host# set interfaces ge-0/0/1.3 host-inbound-traffic system-services snmp
```

4. Configure the security zone to support inbound traffic for all system services for a second interface.

```
[edit security zones security-zone ABC]
```

```
user@host# set interfaces ge-0/0/1.0 host-inbound-traffic system-services all
```

5. Exclude the FTP and HTTP system services from the second interface.

```
[edit security zones security-zone ABC]
```

```
user@host# set interfaces ge-0/0/1.0 host-inbound-traffic system-services ftp
except
```

```
user@host# set interfaces ge-0/0/1.0 host-inbound-traffic system-services http
except
```

**Results** From configuration mode, confirm your configuration by entering the **show security zones security-zone ABC**. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security zones security-zone ABC
host-inbound-traffic {
  system-services {
    all;
  }
}
interfaces {
  ge-0/0/1.3 {
    host-inbound-traffic {
      system-services {
        ftp;
        telnet;
        snmp;
      }
    }
  }
  ge-0/0/1.0 {
    host-inbound-traffic {
      system-services {
        all;
        ftp {
          except;
        }
      }
    }
  }
}
```

```

    }
    http {
      except;
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Troubleshooting with Logs on page 121

#### **Troubleshooting with Logs**

**Purpose** Use these logs to identify any issues.

**Action** From operational mode, enter the **show log messages** command and the **show log dcd** command.

#### **Related Documentation**

- [Junos OS CLI Reference](#)
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding How to Control Inbound Traffic Based on Traffic Types on page 116
- Supported System Services for Host Inbound Traffic on page 117

## Protocols

- Stream Control Transmission Protocol Overview on page 121
- Understanding How to Control Inbound Traffic Based on Protocols on page 122
- Example: Controlling Inbound Traffic Based on Protocols on page 123

### Stream Control Transmission Protocol Overview

Stream Control Transmission Protocol (SCTP) is an IP Transport Layer protocol. SCTP is a reliable transport protocol operating on top of a connectionless packet network such as IP and supports data transfer across the network in single IP or multi-IP cases. SCTP provides the following services:

- Aggregate Server Access Protocol (ASAP)
- Bearer Independent Call Control (BICC)
- Direct Data Placement Segment chunk (DDP-segment)
- Direct Data Placement Stream session control (DDP-stream)
- DPNSS/DASS 2 extensions to IUA Protocol (DUA)
- Endpoint Handleescape Redundancy Protocol (ENRP)

- H.248 Protocol (H248)
- H.323 Protocol (H323)
- ISDN User Adaptation Layer (IUA)
- MTP2 User Peer-to-Peer Adaptation Layer (M2PA)
- MTP2 User Adaptation Layer (M2UA)
- MTP3 User Adaptation Layer (M3UA)
- Q.IPC
- Reserved
- Simple Middlebox Configuration (SIMCO)
- SCCP User Adaptation Layer (SUA)
- Transport Adapter Layer Interface (TALI)
- v5.2 User Adaptation Layer (V5UA)

SCTP can transport signaling messages to and from Signaling System 7 (SS7) for 3G mobile network through M3UA, M2UA or SUA. SCTP is a packet-based transport protocol. SCTP provide reliable and secure transport, minimized end-to-end delay, short failover time in case of network failures and both sequence and no-sequence transport.

### Configuration Overview

---

You should configure at least one SCTP profile to enable the security device to perform stateful inspection on all SCTP traffic. The stateful inspection of SCTP traffic will drop some anomalous SCTP packets. The SCTP firewall supports deeper inspection:

- **Packet filtering**—The profile configuration of drop packets for special SCTP payload protocol and M3UA service enables packet filtering.
- **Limit-rate**—Controls the packets rate of SCCP in M3UA service..

The SCTP deeper inspection requires the following setting:

- Creating an SCTP profile
- Configuring the filtering and limit parameters
- Binding the SCTP profile to a policy



**NOTE:** The policy should permit SCTP traffic.

---

For detailed information about the configuration commands, see the *Junos OS CLI Reference*.

## Understanding How to Control Inbound Traffic Based on Protocols

This topic describes the inbound system protocols on the specified zone or interface.



Any host-inbound traffic that corresponds to a protocol listed under the host-inbound traffic option is allowed. For example, if anywhere in the configuration, you map a protocol to a port number other than the default, you can specify the protocol in the host-inbound traffic option, and the new port number will be used. Table 13 on page 123 lists the supported protocols. A value of **all** indicates that traffic from all of the following protocols is allowed inbound on the specified interfaces (of the zone, or a single specified interface).

**Table 13: Supported Inbound System Protocols**

| Supported System Services |       |       |      |
|---------------------------|-------|-------|------|
| all                       | igmp  | pim   | sap  |
| bfd                       | ldp   | rip   | vrrp |
| bgp                       | msdp  | ripng | nhrp |
| router-discovery          | dvmrp | ospf  | rsvp |
| ndp                       | pgm   | ospf3 |      |



**NOTE:** If DVMRP or PIM is enabled for an interface, IGMP and MLD host-inbound traffic is enabled automatically. Because ISIS uses OSI addressing and should not generate any IP traffic, there is no host-inbound traffic option for the ISIS protocol.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Security Zones and Interfaces Overview](#) on page 111
- [Understanding How to Control Inbound Traffic Based on Traffic Types](#) on page 116
- [Understanding How to Identify Duplicate Sessions Using the TCP-Reset Parameter](#) on page 125
- [Example: Controlling Inbound Traffic Based on Protocols](#) on page 123

### Example: Controlling Inbound Traffic Based on Protocols

This example shows how to enable inbound traffic for an interface.

- [Requirements](#) on page 123
- [Overview](#) on page 124
- [Configuration](#) on page 124
- [Verification](#) on page 125

#### Requirements

Before you begin:

- Configure security zones. See “Example: Creating Security Zones” on page 114.
- Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

### Overview

---

Any host-inbound traffic that corresponds to a protocol listed under the host-inbound traffic option is allowed. For example, if anywhere in the configuration you map a protocol to a port number other than the default, you can specify the protocol in the host-inbound traffic option, and the new port number will be used.

A value of **all** indicates that traffic from all of the protocols is allowed inbound on the specified interfaces (of the zone, or a single specified interface).

### Configuration

---

#### CLI Quick Configuration

To quickly configure inbound traffic based on protocols, copy the following commands and paste them into the CLI:

```
[edit]
```

```
set security zones security-zone ABC interfaces ge-0/0/1.0 host-inbound-traffic protocols ospf
```

```
set security zones security-zone ABC interfaces ge-0/0/1.0 host-inbound-traffic protocols ospf3
```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To configure inbound traffic based on protocols:

1. Configure a security zone.

```
[edit]
```

```
user@host# edit security zones security-zone ABC
```

2. Configure the security zone to support inbound traffic based on the ospf protocol for an interface.

```
[edit security zones security-zone ABC]
```

```
user@host# set interfaces ge-0/0/1.0 host-inbound-traffic protocols ospf
```

3. Configure the security zone to support inbound traffic based on the ospf3 protocol for an interface.

```
[edit security zones security-zone ABC]
```

```
user@host# set interfaces ge-0/0/1.0 host-inbound-traffic protocols ospf3
```

**Results** From configuration mode, confirm your configuration by entering the **show security zones security-zone ABC**. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security zones security-zone ABC
interfaces {
  ge-0/0/1.0 {
    host-inbound-traffic {
      protocols {
        ospf;
        ospf3;
      }
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Troubleshooting with Logs on page 125

#### **Troubleshooting with Logs**

**Purpose** Use these logs to identify any issues.

**Action** From operational mode, enter the **show log messages** command and the **show log dcd** command.

- Related Documentation**
- [Junos OS CLI Reference](#)
  - [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding How to Control Inbound Traffic Based on Protocols on page 122

## TCP-Reset Parameters

- Understanding How to Identify Duplicate Sessions Using the TCP-Reset Parameter on page 125
- Example: Configuring the TCP-Reset Parameter on page 126

### Understanding How to Identify Duplicate Sessions Using the TCP-Reset Parameter

When the TCP-RST feature is enabled, the system sends a TCP segment with the RESET flag set when traffic arrives that does not match an existing session and does not have the SYNchronize flag set.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Security Zones and Interfaces Overview on page 111
  - Understanding How to Control Inbound Traffic Based on Traffic Types on page 116

- [Understanding How to Control Inbound Traffic Based on Protocols](#) on page 122
- [Example: Configuring the TCP-Reset Parameter](#) on page 126

## Example: Configuring the TCP-Reset Parameter

This example shows how to configure the TCP-Reset parameter for a zone.

- [Requirements](#) on page 126
- [Overview](#) on page 126
- [Configuration](#) on page 126
- [Verification](#) on page 126

---

### Requirements

Before you begin, configure security zones. See “[Example: Creating Security Zones](#)” on page 114.

---

### Overview

When the TCP-Reset parameter feature is enabled, the system sends a TCP segment with the RESET flag set when traffic arrives that does not match an existing session and does not have the SYNchronize flag set.

---

### Configuration

#### Step-by-Step Procedure

To configure the TCP-Reset parameter for a zone:

1. Configure a security zone.  
[edit]  
  
user@host# edit security zones security-zone ABC
2. Configure the TCP-Reset parameter for the zone.  
[edit security zones security-zone ABC]  
  
user@host# set tcp-rst
3. If you are done configuring the device, commit the configuration.  
[edit]  
  
user@host# commit

---

### Verification

To verify the configuration is working properly, enter the **show security zones** command.

#### Related Documentation

- [Junos OS CLI Reference](#)
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

- Understanding How to Identify Duplicate Sessions Using the TCP-Reset Parameter on page 125

## DNS

---

- DNS Overview on page 127
- Example: Configuring the TTL Value for DNS Server Caching on page 128
- Example: Configuring a Forwarder for a DNS server on page 129
- DNSSEC Overview on page 129
- Example: Configuring DNSSEC on page 129
- Example: Configuring Keys for DNSSEC on page 130
- Example: Configuring Secure Domains and Trusted Keys for DNSSEC on page 130

### DNS Overview

A Domain Name System (DNS) is a distributed hierarchical system that converts hostnames to IP addresses. The DNS is divided into sections called zones. Each zone has name servers that respond to the queries belonging to their zones.

This topic includes the following sections:

- DNS Components on page 127
- DNS Server Caching on page 127
- Forwarders on page 127

#### DNS Components

---

DNS includes three main components:

- DNS resolver — Resides on the client side of the DNS. When a user sends a hostname request, the resolver sends a DNS query request to the name servers to request the hostname's IP address.
- Name servers — Processes the DNS query requests received from the DNS resolver and returns the IP address to the resolver.
- Resource records — Data elements that define the basic structure and content of the DNS.

#### DNS Server Caching

---

DNS name servers are responsible for providing the hostname IP address to users. The TTL field in the resource record defines the period for which DNS query results are cached. When the TTL value expires, the name server sends a fresh DNS query and updates the cache.

#### Forwarders

---

When a DNS server cannot resolve a query, it forwards the query to another DNS server that is configured as a forwarder. You can use the CLI to configure a DNS server to act

as a forwarder. The DNS server forwards the queries only to the servers that are configured as forwarders.

**Related Documentation**

- Example: Configuring the TTL Value for DNS Server Caching on page 128
- Example: Configuring a Forwarder for a DNS server on page 129
- DNSSEC Overview on page 129

## Example: Configuring the TTL Value for DNS Server Caching

This example shows how to configure the TTL value for a DNS server cache to define the period for which DNS query results are cached.

- Requirements on page 128
- Overview on page 128
- Configuration on page 128
- Verification on page 129

### Requirements

---

No special configuration beyond device initialization is required before performing this task.

### Overview

---

The DNS name server stores DNS query responses in its cache for the TTL period specified in the TTL field of the resource record. When the TTL value expires, the name server sends a fresh DNS query and updates the cache. You can configure the TTL value from 0 to 604,800 seconds. You can also configure the TTL value for cached negative responses. Negative caching is the storing of the record that a value does not exist. In this example, you set the maximum TTL value for cached (and negative cached) responses to 86,400 seconds.

### Configuration

---

**Step-by-Step Procedure**

To configure the TTL value for a DNS server cache:

1. Specify the maximum TTL value for cached responses, in seconds.  
[edit]  
user@host# **set system services dns max-cache-ttl 86400**
2. Specify the maximum TTL value for negative cached responses, in seconds.  
[edit]  
user@host# **set system services dns max-ncache-ttl 86400**
3. If you are done configuring the device, commit the configuration.  
[edit]  
user@host# **commit**

### Verification

To verify the configuration is working properly, enter the **show system services** command.

- [\[xref target has no title\]](#)

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- DNS Overview on page 127

### Example: Configuring a Forwarder for a DNS server

You can configure a DNS server to act as a forwarder. A DNS server will forward any DNS query it cannot handle to another server that is configured as a forwarder. The following example shows how to configure a DNS server with IP address 10.100.11.24 to act as a forwarder:

```
edit
user@host# set system services dns forwarders 10.100.11.24
```

#### Related Documentation

- DNS Overview on page 127

### DNSSEC Overview

Junos OS devices support the domain name service security extensions (DNSSEC) standard. DNSSEC is an extension of DNS that provides authentication and integrity verification of data by using public-key based signatures.

In DNSSEC, all the resource records in a DNS are signed with the private key of the zone owner. The DNS resolver uses the public key of the owner to validate the signature. The zone owner generates a private key to encrypt the hash of a set of resource records. The private key is stored in RRSIG record. The corresponding public key is stored in the DNSKEY record. The resolver uses the public key to decrypt the RRSIG and compares the result with the hash of the resource record to verify that it has not been altered.

Similarly, the hash of the public DNSKEY is stored in a DS record in a parent zone. The zone owner generates a private key to encrypt the hash of the public key. The private key is stored in the RRSIG record. The resolver retrieves the DS record and its corresponding RRSIG record and public key. Using the public key, the resolver decrypts the RRSIG record and compares the result with the hash of the public DNSKEY to verify that it has not been altered. This establishes a chain of trust between the resolver and the name servers.

#### Related Documentation

- DNS Overview on page 127
- Example: Configuring Keys for DNSSEC on page 130
- Example: Configuring Secure Domains and Trusted Keys for DNSSEC on page 130

### Example: Configuring DNSSEC

DNS-enabled devices run a DNS resolver (proxy) that listens on loopback address 127.0.0.1 or ::1. The DNS resolver performs a hostname resolution for DNSSEC. Users need to set

name server IP address to 127.0.0.1 or ::1 so the DNS resolver forwards all DNS queries to DNSSEC instead of to DNS. If the name server IP address is not set, DNS will handle all queries instead of to DNSSEC.

The following example shows how to set the server IP address to 127.0.0.1:

```
[edit]
user@host# set system name-server 127.0.0.1
```

The DNSSEC feature is enabled by default. You can disable DNSSEC in the server by using the following CLI command:

```
[edit ]
set system services dns dnssec disable
```

**Related Documentation**

- [DNSSEC Overview on page 129](#)

### Example: Configuring Keys for DNSSEC

You can load a public key from a file or you can copy and paste the key file from a terminal. In both cases, you must save the keys to the configuration instead of to a file. The following example shows how to load a key from a file:

```
[edit system services dns dnssec trusted-keys]
#load-key filename
```

The following example explains how to load the key from a terminal:

```
[edit system services dns dnssec trusted-keys]
# set key "...pasted-text..."
```

If you are done loading the keys from the file or terminal, click **commit** in the CLI editor.

**Related Documentation**

- [Example: Configuring Secure Domains and Trusted Keys for DNSSEC on page 130](#)

### Example: Configuring Secure Domains and Trusted Keys for DNSSEC

This example shows how to configure secure domains and trusted keys for DNSSEC.

- [Requirements on page 130](#)
- [Overview on page 130](#)
- [Configuration on page 131](#)
- [Verification on page 132](#)

#### Requirements

---

Set the name server IP address so the DNS resolver forwards all DNS queries to DNSSEC instead of DNS. See "Example: Configuring DNSSEC" on page 129 for more information.

#### Overview

---

You can configure secure domains and assign trusted keys to the domains. Both signed and unsigned responses can be validated when DNSSEC is enabled.



When you configure a domain as a secure domain and if DNSSEC is enabled, all unsigned responses to that domain are ignored and the server returns a SERVFAIL error code to the client for the unsigned responses. If the domain is not configured as a secure domain, unsigned responses will be accepted.

When the server receives a signed response, it checks if the DNSKEY in the response matches any of the trusted keys that are configured. If it finds a match, the server accepts the signed response.

You can also attach a DNS root zone as a trusted anchor to a secure domain to validate the signed responses. When the server receives a signed response, it queries the DNS root zone for a DS record. When it receives the DS record, it checks if the DNSKEY in the DS record matches the DNSKEY in the signed response. If it finds a match, the server accepts the signed response.

### Configuration

#### CLI Quick Configuration

To quickly configure secure domains for DNSSEC and secure them with a trusted key and trusted anchor, copy the following commands and paste them into the CLI.

```
[edit]
set system services dns dnssec secure-domains domain1.net
set system services dns dnssec secure-domains domain2.net
set system services dns dnssec trusted-keys key domain1.net.25633CJ5K3h
set system services dns dnssec dlz domain domain2.net trusted-anchor dlz.isc.org
```

#### Step-by-Step Procedure

To configure secure domains and trusted keys for DNSSEC:

1. Configure domain1.net and domain2.net as secure domains.

```
[edit]
user@host# set system services dns dnssec secure-domains domain1.net
user@host# set system services dns dnssec secure-domains domain2.net
```

2. Configure trusted keys to domain1.net.

```
[edit]
user@host# set system services dns dnssec trusted-keys key
"domain1.net.25633CJ5K3h"
```

3. Attach a root zone div.isc.org as a trusted anchor to a secure domain.

```
[edit]
user@host# set system services dns dnssec dlz domain domain2.net trusted-anchor
dlz.isc.org
```

#### Results

From configuration mode, confirm your configuration by entering the `show system services` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
dns {
  dnssec {
    trusted-keys {
      key domain1.net.25633CJ5K3h; ## SECRET-DATA
    }
  }
  dlz {
```

```
        domain domain2.net trusted-anchor dlv.isc.org;
    }
    secure-domains {
        domain1.net;
        domain2.net;
    }
}
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

---

To confirm that the configuration is working properly, perform this task:

- [Verifying Secure Domains and Trusted Keys for DNSSEC Configuration on page 132](#)

#### ***Verifying Secure Domains and Trusted Keys for DNSSEC Configuration***

**Purpose** Verify information about secure domains and trusted keys for the DNSSEC configuration.

**Action** From operational mode, enter the **show ...** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [DNSSEC Overview on page 129](#)
  - [Example: Configuring Keys for DNSSEC on page 130](#)

## CHAPTER 5

# Address Books and Address Sets

- [Security Policy Address Books and Address Sets Overview on page 133](#)
- [Understanding Address Books on page 134](#)
- [Understanding Address Sets on page 135](#)
- [Limitations of Addresses and Address Sets on page 138](#)
- [Example: Configuring Address Books on page 139](#)
- [Verifying Address Book Configuration on page 141](#)

## Security Policy Address Books and Address Sets Overview

Each security zone contains an address book. Before you can set up policies between two zones, you must define the addresses for each of the zone's address books. To manage an address book with large numbers of addresses, you can create groups of addresses called address sets.

A security zone is a logical group of interfaces with identical security requirements. Each security zone contains an address book. Before you can set up policies between two zones, you must define the addresses for each of the zone's address books. A zone's address book must contain entries for the addressable networks and end hosts (and, thus, users) belonging to the zone.

### **Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Security Policies Overview on page 145](#)
- [Understanding Address Books on page 134](#)
- [Understanding Address Sets on page 135](#)
- [Example: Configuring Address Books and Address Sets on page 139](#)
- [Verifying Address Book Configuration on page 141](#)

## Understanding Address Books

The following guidelines apply to address books:

- An address book for a security zone contains the IP address or wildcard address or domain names of hosts and subnets whose traffic is either allowed, blocked, encrypted, or user-authenticated.
- Address books can have address sets. Each address set has a name and a list of address names.
- Addresses and address sets in the same zone must have distinct names.
- Addresses must conform to the security requirements of the zone.
- Address book entries can include any combination of IPv4 addresses, IPv6 addresses, wildcard addresses, and Domain Name System (DNS) names.



**NOTE:** Support for IPv6 addresses in active/active chassis cluster configurations (in addition to the existing support of active/passive chassis cluster configurations) has been added in Junos OS Release 10.4. Support for wildcard addresses has been added in Junos OS Release 11.1. IPv6 wildcard address configuration is not supported in Junos OS Release 11.1.

- By default, you can resolve IPv4 and IPv6 addresses for a DNS. If IPv4 or IPv6 addresses are designated, you can resolve only those addresses by using the keywords **ipv4-only** and **ipv6-only**, respectively.
- The predefined addresses, **any-ipv4** and **any-ipv6** are automatically created for each security zone.
- The address book of a security zone must contain all IP addresses that are reachable within that zone.

Policies contain both source and destination zones and addresses. An address is referred to in a policy by the name you give it in the zone address book.

- When traffic is sent to a zone, the zone and address to which the traffic is sent are used as the destination zone and address-matching criteria in policies.
- When traffic is sent from a zone, the zone and address from which it is sent are used as the matching source zone and address in policies.

For more information on the address book configuration syntax and options, see the [Junos OS CLI Reference](#).



**NOTE:** Specify addresses as network prefixes in the `prefix/length` format. For example, `1.2.3.0/24` is an acceptable address book address because it translates to a network prefix. However, `1.2.3.4/24` is not acceptable for an address book because it exceeds the subnet length of 24 bits. Everything beyond the subnet length must be entered as 0 (zero). In special scenarios, you can enter a hostname because it can use the full 32-bit address length.

An IPv6 address prefix is a combination of an IPv6 prefix (address) and a prefix length. The prefix takes the form `ipv6-prefix/prefix-length` and represents a block of address space (or a network). The `ipv6-prefix` variable follows general IPv6 addressing rules. The `/prefix-length` variable is a decimal value that indicates the number of contiguous, higher-order bits of the address that make up the network portion of the address. For example, `10FA:6604:8136:6502::/64` is a possible IPv6 prefix. For more information on text representation of IPv6 addresses and address prefixes, see RFC 4291, *IP Version 6 Addressing Architecture*.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policy Address Books and Address Sets Overview on page 133
- Understanding Address Sets on page 135
- Example: Configuring Address Books and Address Sets on page 139
- Example: Configuring Schedulers on page 182

## Understanding Address Sets

An address book can grow to contain large numbers of addresses and become difficult to manage. To manage an address book with large numbers of addresses, you can create groups of addresses called *address sets*. You can reference an address set in a policy as you would an individual address book entry.

The following example shows addresses and address sets in the green zone:

```

user@host# set security zones security-zone green address-book address src_addr1 64.10.4.44/32
user@host# set security zones security-zone green address-book address src_addr2 64.10.9.28/32
user@host# set security zones security-zone green address-book address src_addr3 10.10.10.0/24
user@host# set security zones security-zone green address-book address src_addr4 fa:43::/96
user@host# set security zones security-zone green address-book address src_addr5
fe80::210:dbff:feff:1000/64
user@host# set security zones security-zone green address-book address src_addr6
0001:db8:1::1/127
user@host# set security zones security-zone green address-book address bbc dns-name
www.bbc.com
user@host# set security zones security-zone green address-book address-set
my_source_addresses address src_addr1
user@host# set security zones security-zone green address-book address-set
my_source_addresses address src_addr2
user@host# set security zones security-zone green address-book address-set
my_source_addresses address src_addr3

```

```

user@host# set security zones security-zone green address-book address-set
my_source_addresses address src_addr4
user@host# set security zones security-zone green address-book address-set
my_source_addresses address src_addr5
user@host# set security zones security-zone green address-book address-set
my_source_addresses address src_addr6
user@host# show security zones security-zone green

```

```

address-book {
  address src_addr1 64.10.4.44/32;
  address src_addr2 64.10.9.28/32;
  address src_addr3 10.10.10.0/24;
  address src_addr4 fa:43::/96;
  address src_addr5 fe80::210:dbff:feff:1000/64;
  address src_addr6 0001:db8:1::1/127;
  address bbc {
    dns-name www.bbc.com;
  }
  address-set my_source_addresses {
    address src_addr1;
    address src_addr2;
    address src_addr3;
    address src_addr4;
    address src_addr5;
    address src_addr6;
  }
}

```

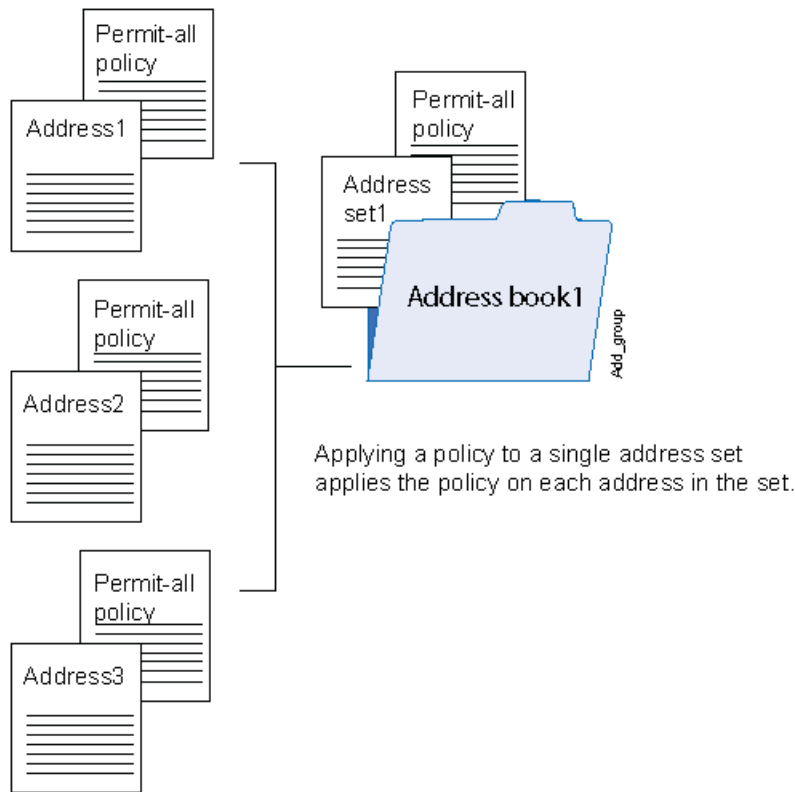
For more information on the address set configuration syntax and options, see the [Junos OS CLI Reference](#).



**NOTE:** Consider that for each address set, the system creates individual rules for its members. It creates an internal rule for each member in the group as well as for each service configured for each user. If you configure address books without taking this into account, you can exceed the number of available policy resources, especially if both the source and destination addresses are address groups and the specified service is a service group.

When you add addresses to policies, sometimes the same subset of addresses can be present in multiple policies, making it difficult to manage how policies affect each address entry. Junos OS allows you to create groups of addresses called *address sets*. Address sets simplify the process by allowing you to add multiple addresses within an address set and therefore manage a small number of address sets, rather than manage a large number of individual address entries. See Figure 9 on page 137.

Figure 9: Address Sets



The address set option has the following features:

- You can create address sets in any zone.
- You can create address sets with existing users, or you can create empty address sets and later fill them with users.
- You can reference an address set entry in a policy like an individual address book entry.



**NOTE:** Junos OS applies policies automatically to each address set member, so you do not have to create them one by one for each address.

- When you delete an individual address book entry from the address book, you must remove the address (wherever it is referred) from all the address sets.

The following constraints apply to address sets:

- To configure an address set, you need more than an address in the address book.
- Address sets can only contain address names that belong to the same security zone.
- Address names cannot be the same as address set names. For example, if the name **Paris** is used for an address in an individual address entry, it cannot be used for an address set name.

- If an address set is referenced in a policy, the address set cannot be removed without removing its reference in the policy. It can, however, be edited.
- The predefined address set, **any**, which contains both **any-ipv4** and **any-ipv6** addresses is automatically created for each security zone.
- You cannot add the predefined address **any** to an address book.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policy Address Books and Address Sets Overview on page 133
- Understanding Address Books on page 134
- Example: Configuring Address Books and Address Sets on page 139
- Example: Configuring Schedulers on page 182
- Limitations of Addresses and Address Sets on page 138

## Limitations of Addresses and Address Sets

On SRX Series and J Series devices, the limitation on the number of addresses in address-set has been increased. The number of addresses in address-set now depends on the device and is equal to the number of addresses supported by the policy.

Table 14 on page 138 provides the address-set details per device to increase the configuration limitation.

**Table 14: Number of Addresses in address-set on SRX Series and J Series Devices**

| Device             | address-set |
|--------------------|-------------|
| Default            | 1024        |
| SRX100 High Memory | 1024        |
| SRX100 Low Memory  | 512         |
| SRX210 High Memory | 1024        |
| SRX210 Low Memory  | 512         |
| SRX240 High Memory | 1024        |
| SRX240 Low Memory  | 512         |
| SRX650             | 1024        |
| SRX3400            | 1024        |
| SRX3600            | 1024        |



Table 14: Number of Addresses in address-set on SRX Series and J Series Devices (*continued*)

| Device   | address-set |
|----------|-------------|
| SRX5600  | 1024        |
| SRX5800  | 1024        |
| J Series | 1024        |

## Example: Configuring Address Books

This example describes how to configure address books and address sets for a zone.

- Requirements on page 139
- Overview on page 139
- Configuration on page 139
- Verification on page 141

### Requirements

Before you begin, configure the zones required in this example. See “Example: Creating Security Zones” on page 114.

### Overview

In this example, you configure addresses, wildcard addresses, and address sets for address books in the IntranetGREEN zone. This zone contains servers that belong to the same subnet. You can add individual addresses for the servers to the zone address list to accommodate users with access rights to one server but not the other. You can also add an address set to combine the servers into a single addressable entity.

### Configuration

**CLI Quick Configuration** To quickly configure address book entries for the IntranetGREEN zone, copy the following commands and paste them into the CLI.

```
[edit]
set security zones security-zone IntranetGREEN address-book address G1 10.1.10.0/24
set security zones security-zone IntranetGREEN address-book address G2 192.168.0.0/16
set security zones security-zone IntranetGREEN address-book address G3
  wildcard-address 192.168.0.11/255.255.0.255
set security zones security-zone IntranetGREEN address-book address-set SerAll address
  G1
set security zones security-zone IntranetGREEN address-book address-set SerAll address
  G2
set security zones security-zone IntranetGREEN address-book address-set SerAll address
  G3
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To configure address book entries:

1. Create a security zone.

```
[edit]
user@host# set security zones security-zone IntranetGREEN
```

2. Create an address book and assign an address entry.

```
[edit security zones security-zone IntranetGREEN]
user@host# set address-book address G1 10.1.10.0/24
```

3. Create another address book and assign an address entry.

```
[edit security zones security-zone IntranetGREEN]
user@host# set address-book address G2 192.168.0.0/16
```

4. Create another address book and assign a wildcard address entry.

```
[edit security zones security-zone IntranetGREEN]
user@host# set address-book address G3 wildcard-address
192.168.0.11/255.255.0.255
```

5. Configure an address set for all of the entries in Step 2.

```
[edit security zones security-zone IntranetGREEN]
user@host# set address-book address-set serAll address G1
```

6. Configure another address set for the entries in Step 3.

```
[edit security zones security-zone IntranetGREEN]
user@host# set address-book address-set serAll address G2
```

7. Configure another address set for the entries in Step 4.

```
[edit security zones security-zone IntranetGREEN]
user@host# set address-book address-set serAll address G3
```

**Results** From configuration mode, confirm your configuration by entering the **show security zones security-zone IntranetGREEN** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
[user@host]show security zones security-zone IntranetGREEN
address-book {
  address G1 10.1.10.0/24;
  address G2 192.168.0.0/16;
  address G3 {
    wildcard-address 192.168.0.11/255.255.0.255;
  }
  address-set serAll {
    address G1;
    address G2;
    address G3;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform this task:

- Verifying the Address book Entries on page 141

### Verifying the Address book Entries

---

**Purpose** Verify the list of address book entries currently configured in the device.

**Action** From operational mode, enter the **show security zones** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Security Policy Address Books and Address Sets Overview on page 133
  - Verifying Address Book Configuration on page 141
  - Understanding Wildcard Addresses on page 150

## Verifying Address Book Configuration

---

**Purpose** Display information about address books and zones.

**Action** Use the **show security zones** CLI command to verify the address book and address set configuration. You get the following output:

```
user@host# show security zones security-zone green
```

```
address-book {
  address src_addr1 64.10.4.44/32;
  address src_addr2 64.10.9.28/32;
  address src_addr3 10.10.10.10/24;
  address bbc {
    dns-name www.bbc.com;
  }
  address-set my_source_addresses {
    address src_addr1;
    address src_addr2;
    address src_addr3;
  }
}
```

**Meaning** The output displays information about all the addresses configured in an address book in the specified. Verify the following information:

- Configured addresses belong to the correct address book.
- Configured address book belongs to the correct zone.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Security Policy Address Books and Address Sets Overview on page 133

- [Limitations of Addresses and Address Sets on page 138](#)
- [Example: Configuring Address Books and Address Sets on page 139](#)
- [Example: Configuring Schedulers on page 182](#)

## PART 3

# Security Policies

- Security Policies on page 145
- Security Policy Schedulers on page 181
- Security Policy Applications on page 187



## CHAPTER 6

# Security Policies

- Security Policies Overview on page 145
- Understanding Security Policy Rules on page 148
- Understanding Security Policy Elements on page 151
- Security Policies Configuration Overview on page 151
- Configuring Policies Using the Firewall Wizard on page 152
- Example: Configuring a Security Policy to Permit or Deny All Traffic on page 152
- Example: Configuring a Security Policy to Permit or Deny Selected Traffic on page 156
- Example: Configuring a Security Policy to Permit or Deny Wildcard Address Traffic on page 161
- Application Firewall Overview on page 164
- Example: Configuring Application Firewall Rule Sets Within Security Policy (CLI) on page 167
- Understanding Security Policy Ordering on page 173
- Example: Reordering the Policies on page 175
- Troubleshooting Security Policies on page 176
- Monitoring Policy Statistics on page 177
- Matching Security Policies on page 178

## Security Policies Overview

---

With the advent of the Internet, the need for a secure network has become vital for businesses with an Internet connection. Before a network can be secured for a business, a network security policy has to outline all the network resources within that business and identify the required security level for those resources. The network security policy also defines the security threats and the actions taken for such threats. Junos OS stateful firewall policy provides a set of tools to network administrators, enabling them to implement network security for their organizations.

In a Junos OS stateful firewall, the security policies enforce rules for transit traffic, in terms of what traffic can pass through the firewall, and the actions that need to take place on traffic as it passes through the firewall. From the perspective of security policies, the traffic enters one security zone and exits another security zone. This combination of

a *from-zone* and *to-zone* is called a *context*. Each context contains an *ordered list* of policies. Each policy is processed in the order that it is defined within a context.

A security policy, which can be configured from the user interface, controls the traffic flow from one zone to another zone by defining the kind(s) of traffic permitted from specified IP sources to specified IP destinations at scheduled times.

Policies allow you to deny, permit, reject (deny and send a TCP RST or ICMP port unreachable message to the source host), encrypt and decrypt, authenticate, prioritize, schedule, filter, and monitor the traffic attempting to cross from one security zone to another. You decide which users and what data can enter and exit, and when and where they can go.



**NOTE:** For a J Series or an SRX Series device that supports virtual systems, policies set in the root system do not affect policies set in virtual systems.

---

A J Series or an SRX Series device secures a network by inspecting, and then allowing or denying, all connection attempts that require passage from one security zone to another.

Logging capability can also be enabled with security policies during session initialization (**session-init**) or session close (**session-close**) stage.

- To view logs from denied connections, enable log on **session-init**.
  - To log sessions after their conclusion/tear-down, enable log on **session-close**.
- 



**NOTE:** Session log is enabled at real time in the flow code which impacts the user performance. If both **session-close** and **session-init** are enabled, performance is further degraded as compared to enabling **session-init** only.

---

By default, a device denies all traffic in all directions, including intra-zone and inter-zone directions. Through the creation of policies, you can control the traffic flow from zone to zone by defining the kinds of traffic permitted to pass from specified sources to specified destinations at scheduled times.

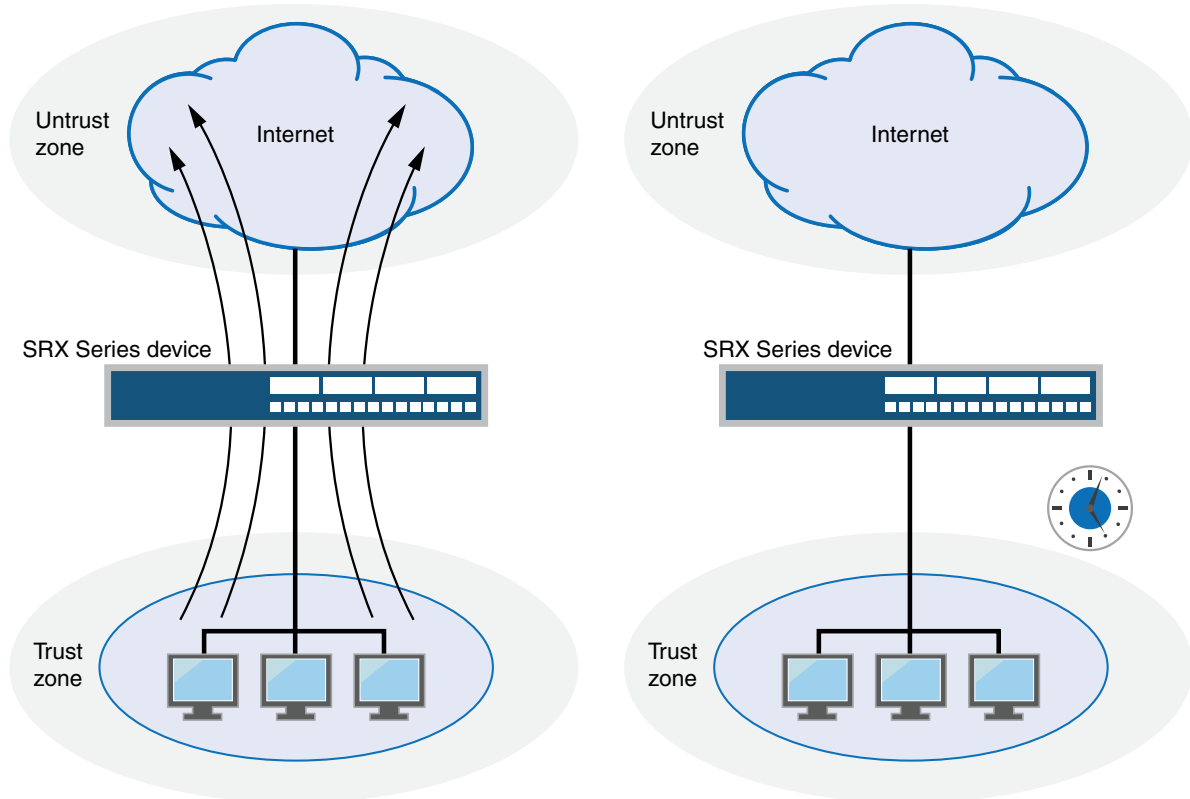
At the broadest level, you can allow all kinds of traffic from any source in one zone to any destination in all other zones without any scheduling restrictions. At the narrowest level, you can create a policy that allows only one kind of traffic between a specified host in one zone and another specified host in another zone during a scheduled interval of time. See Figure 10 on page 147.



Figure 10: Default Policy

Broadly defined Internet access: Any service from any point in the trust zone to any point in the untrust zone at any time.

Narrowly defined Internet access: SMTP service from a mail server in the trust zone to a mail server in the untrust zone from 5:00 AM to 7:00 PM.



g030677

Every time a packet attempts to pass from one zone to another or between two interfaces bound to the same zone, the device checks for a policy that permits such traffic (see “Understanding Security Zones” on page 113 and “Policy Application Sets Overview” on page 188). To allow traffic to pass from one security zone to another—for example, from zone A to zone B—you must configure a policy that permits zone A to send traffic to zone B. To allow traffic to flow the other way, you must configure another policy permitting traffic from zone B to zone A.

To allow data traffic to pass between zones, you must configure firewall policies.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Security Policy Rules on page 148](#)
- [Understanding Security Policy Elements on page 151](#)
- [Security Policies Configuration Overview on page 151](#)
- [Understanding Security Policy Ordering on page 173](#)
- [Security Zones and Interfaces Overview on page 111](#)

## Understanding Security Policy Rules

---

The security policy applies the security rules to the transit traffic within a context (**from-zone** to **to-zone**). Each policy is uniquely identified by its name. The traffic is classified by matching its source and destination zones, the source and destination addresses, and the application that the traffic carries in its protocol headers with the policy database in the data plane.

Each policy is associated with the following characteristics:

- A source zone
- A destination zone
- One or many source address names or address set names
- One or many destination address names or address set names
- One or many application names or application set names

These characteristics are called the *match criteria*. Each policy also has actions associated with it: permit, deny, reject, count, log, and VPN tunnel. You have to specify the match condition arguments when you configure a policy, source address, destination address, and application name.

You can specify to configure a policy with IPv4 or IPv6 addresses using the wildcard entry **any**. When flow support is not enabled for IPv6 traffic, **any** matches IPv4 addresses. For example, if you want to include both IPv4 and IPv6 addresses in the match criteria, then **any** is used. You can also specify the wildcard **any-ipv4** or **any-ipv6** for the source and destination address match criteria to include only IPv4 or only IPv6 addresses, respectively.

If you do not want to specify a specific application, enter **any** as the default application. To look up the default applications, from configuration mode, enter **show groups junos-defaults | find applications (predefined applications)**. For example, if you do not supply an application name, the policy is installed with the application as a wildcard (default). Therefore, any data traffic that matches the rest of the parameters in a given policy would match the policy regardless of the application type of the data traffic.

The action of the first policy that the traffic matches is applied to the packet. If there is no matching policy, the packet is dropped. Policies are searched from top to bottom, so it is a good idea to place more specific policies near the top of the list. You should also place IPsec VPN tunnel policies near the top. Place the more general policies, such as one that would allow certain users access to all Internet applications, at the bottom of the list. For example, place deny-all or reject-all policies at the bottom after all of the specific policies have been parsed before and legitimate traffic has been allowed/count/logged.



**NOTE:** Support for IPv6 addresses added in Release 10.2 of Junos OS and support for IPv6 addresses in active/active chassis cluster configurations (in addition to the existing support of active/passive chassis cluster configurations) has been added in Junos OS Release 10.4.

Policies are looked up during flow processing after firewall filters and screens have been processed and route look up has been completed by the Services Processing Unit (SPU) (for high-end SRX Series devices). Policy look up determines the destination zone, destination address, and egress interface.

When you are creating a policy, the following policy rules apply:

- Security policies are configured in a **from-zone** to **to-zone** direction. Under a specific zone direction, each security policy contains a name, match criteria, an action, and miscellaneous options.
- The policy name, match criteria, and action are required.
- The policy name is a keyword.
- The source address in the match criteria is composed of one or more address names or address set names in the **from-zone**.
- The destination address of the match criteria is composed of one or more address names or address set names in the **to-zone**.
- The application name in the match criteria is composed of the name of one or more applications or application sets.
- One of the following actions is required: permit, deny, reject, count, or log.
- When logging is enabled, the system logs at session close (**session-close**) time by default. To enable logging at session creation, use the **session-init** command.
- When the count alarm is turned on, you can, optionally, specify alarm thresholds in bytes per second and kilobytes per minute.
- You cannot specify **global** as either the **from-zone** or the **to-zone** except under following condition:
 

Any policy configured with the **to-zone** as a global zone must have a single destination address to indicate that either static NAT or incoming NAT has been configured in the policy.
- In SRX Series Services Gateways, the policy permit option with NAT is simplified. Each policy will optionally indicate whether it allows NAT translation, does not allow NAT translation, or does not care.
- Address names cannot begin with the following reserved prefixes. These are used only for address NAT configuration:
  - **static\_nat\_**
  - **incoming\_nat\_**

- `junos_`
- Application names cannot begin with the `junos_` reserved prefix.

## Understanding Wildcard Addresses

Source and destination addresses are two of the five match criteria that should be configured in a security policy. You can now configure wildcard addresses for the source and destination address match criteria in a security policy. A wildcard address is represented as A.B.C.D/wildcard-mask. The wildcard mask determines which of the bits in the IP address A.B.C.D should be ignored by the security policy match criteria. For example, the source IP address 192.168.0.11/255.255.0.255 in a security policy implies that the security policy match criteria can discard the third octet in the IP address (symbolically represented as 192.168.\*.11). Therefore, packets with source IP addresses such as 192.168.1.11 and 192.168.22.11 conform to the match criteria. However, packets with source IP addresses such as 192.168.0.1 and 192.168.1.21 do not satisfy the match criteria.

The wildcard address usage is not restricted to full octets only. You can configure any wildcard address. For example, the wildcard address 192.168. 7.1/255.255.7.255 implies that you need to ignore only the first 5 bits of the third octet of the wildcard address while making the policy match. If the wildcard address usage is restricted to full octets only, then wildcard masks with either 0 or 255 in each of the four octets only will be permitted.



**NOTE:** The first octet of the wildcard mask should be greater than 128. For example, a wildcard mask represented as 0.255.0.255 or 1.255.0.255 is invalid.

A wildcard security policy is a simple firewall policy that allows you to permit, deny, and reject the traffic trying to cross from one security zone to another. You should not configure security policy rules using wildcard addresses for services, such as Intrusion Detection and Prevention (IDP), Unified Threat Management (UTM), and IP Security (IPsec).

If wildcard security policies are configured on a device, performance degrades based on the number of wildcard addresses configured across all the policies.



**NOTE:** IPv6 wildcard address configuration is not supported in this release.

For detailed information on the wildcard address configuration syntax and options, see the *Junos OS CLI Reference Guide*.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policies Overview on page 145
- Understanding Security Policy Elements on page 151
- Security Policies Configuration Overview on page 151
- Understanding Security Policy Ordering on page 173

## Understanding Security Policy Elements

A policy permits, denies, or tunnels specified types of traffic unidirectionally between two points.

Each policy consists of:

- A unique name for the policy.
- A **from-zone** and a **to-zone**, for example: `user@host# set security policy from-zone untrust to-zone untrust`
- A set of match criteria defining the conditions that must be satisfied to apply the policy rule. The match criteria are based on a source IP address, destination IP address, and applications.
- A set of actions to be performed in case of a match—permit, deny, or reject.
- Accounting and auditing elements—counting, logging, or structured system logging.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Junos OS CLI Reference](#)
- [Security Policies Overview on page 145](#)
- [Understanding Security Policy Rules on page 148](#)
- [Security Policies Configuration Overview on page 151](#)
- [Understanding Security Policy Ordering on page 173](#)

## Security Policies Configuration Overview

You must complete the following tasks to create a security policy:

1. Create zones. See “Example: Creating Security Zones” on page 114.
2. Configure an address book with addresses for the policy. See “Example: Configuring Address Books and Address Sets” on page 139.
3. Create an application (or application set) that indicates that the policy applies to traffic of that type. See “Example: Configuring Applications and Application Sets” on page 189.
4. Create the policy. See “Example: Configuring a Security Policy to Permit or Deny All Traffic” on page 152, “Example: Configuring a Security Policy to Permit or Deny Selected Traffic” on page 156, and “Example: Configuring a Security Policy to Permit or Deny Wildcard Address Traffic” on page 161.
5. Create schedulers if you plan to use them for your policies. See “Example: Configuring Schedulers” on page 182.

The Firewall Policy Wizard enables you to perform basic security policy configuration. For more advanced configuration, use the J-Web interface or the CLI.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Security Policy Rules on page 148](#)
- [Understanding Security Policy Elements on page 151](#)
- [Troubleshooting Security Policies on page 176](#)

---

## Configuring Policies Using the Firewall Wizard

The Firewall Policy Wizard enables you to perform basic security policy configuration. For more advanced configuration, use the J-Web interface or the CLI.

To configure policies using the Firewall Policy Wizard:

1. Select **Configure>Wizards>FW Policy Wizard** in the J-Web interface.
2. Click the Firewall Wizard button to launch the wizard.
3. Follow the prompts in the wizard.

The upper left area of the wizard page shows where you are in the configuration process. The lower left area of the page shows field-sensitive help. When you click a link under the Resources heading, the document opens in your browser. If the document opens in a new tab, be sure to close only the tab (not the browser window) when you close the document.

**Related  
Documentation**

- 

---

## Example: Configuring a Security Policy to Permit or Deny All Traffic

This example shows how to configure a security policy to permit or deny traffic.

- [Requirements on page 152](#)
- [Overview on page 153](#)
- [Configuration on page 153](#)
- [Verification on page 156](#)

### Requirements

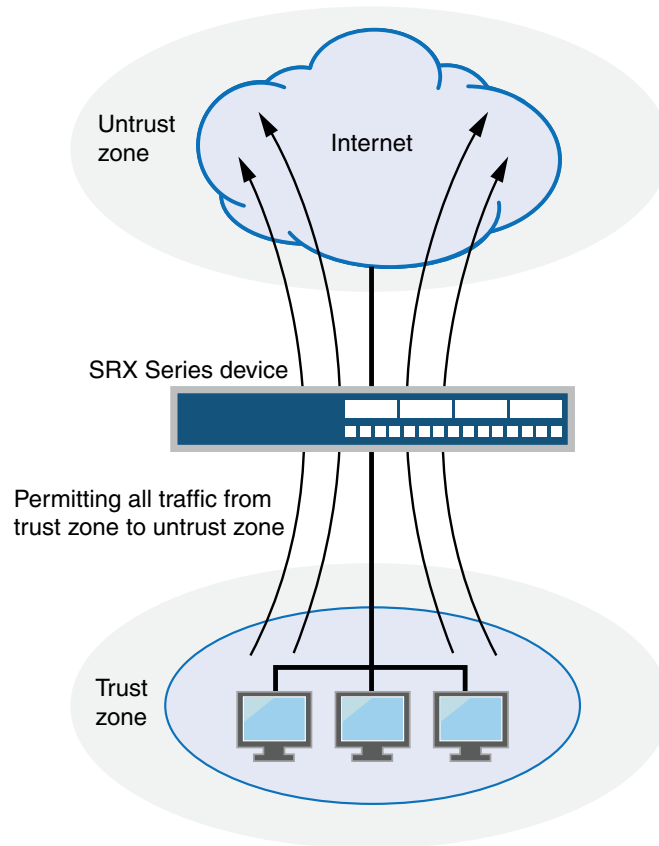
Before you begin:

- Create zones. See “Example: Creating Security Zones” on page 114.
- Configure an address book and create addresses for use in the policy. See “Example: Configuring Address Books and Address Sets” on page 139.
- Create an application (or application set) that indicates that the policy applies to traffic of that type. See “Example: Configuring Applications and Application Sets” on page 189.

## Overview

In a Junos OS, security policies enforce rules for transit traffic, in terms of what traffic can pass through the device, and the actions that need to take place on traffic as it passes through the device. From the perspective of security policies, the traffic enters one security zone and exits another security zone. In this example, you configure the trust and untrust interfaces, ge-0/0/2 and ge-0/0/1. See Figure 11 on page 153.

Figure 11: Permitting All Traffic



This configuration example shows how to:

- Permit or deny all traffic from the trust zone to the untrust zone but block everything from the untrust zone to the trust zone.
- Permit or deny selected traffic from a host in the trust zone to a server in the untrust zone at a particular time.

## Configuration

**CLI Quick Configuration** To quickly configure a security policy to permit or deny all traffic, copy the following commands and paste them into the CLI:

[edit]

```
set security zones security-zone trust interfaces ge-0/0/2 host-inbound-traffic
system-services all
```

```
set security zones security-zone untrust interfaces ge-0/0/1 host-inbound-traffic
system-services all
```

```
set security policies from-zone trust to-zone untrust policy permit-all match
source-address any
```

```
set security policies from-zone trust to-zone untrust policy permit-all match
destination-address any
```

```
set security policies from-zone trust to-zone untrust policy permit-all match application
any
```

```
set security policies from-zone trust to-zone untrust policy permit-all then permit
```

```
set security policies from-zone untrust to-zone trust policy deny-all match source-address
any
```

```
set security policies from-zone untrust to-zone trust policy deny-all match
destination-address any
```

```
set security policies from-zone untrust to-zone trust policy deny-all match application
any
```

```
set security policies from-zone untrust to-zone trust policy deny-all then deny
```

**Step-by-Step  
Procedure**

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To configure a security policy to permit or deny all traffic:

1. Configure the interfaces and security zones.

```
[edit security zones]
```

```
user@host# set security-zone trust interfaces ge-0/0/2 host-inbound-traffic
system-services all
```

```
user@host# set security-zone untrust interfaces ge-0/0/1 host-inbound-traffic
system-services all
```

2. Create the security policy to permit traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
```

```
user@host# set policy permit-all match source-address any
```

```
user@host# set policy permit-all match destination-address any
```



```
user@host# set policy permit-all match application any
```

```
user@host# set policy permit-all then permit
```

3. Create the security policy to deny traffic from the untrust zone to the trust zone.

```
[edit security policies from-zone untrust to-zone trust]
```

```
user@host# set policy deny-all match source-address any
```

```
user@host# set policy deny-all match destination-address any
```

```
user@host# set policy deny-all match application any
```

```
user@host# set policy deny-all then deny
```

**Results** From configuration mode, confirm your configuration by entering the **show security policies** and **show security zones** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.



**NOTE:** The configuration example is a default permit-all from the trust zone to the untrust zone.

```
[edit]
user@host# show security policies
from-zone trust to-zone untrust {
  policy permit-all {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone untrust to-zone trust {
  policy deny-all {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      deny;
    }
  }
}
user@host# show security zones
```

```

security-zone trust {
  interfaces {
    ge-0/0/2.0 {
      host-inbound-traffic {
        system-services {
          all;
        }
      }
    }
  }
}
security-zone untrust {
  interfaces {
    ge-0/0/1.0 {
      host-inbound-traffic {
        system-services {
          all;
        }
      }
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform this task:

- [Verifying Policy Configuration on page 156](#)

### [Verifying Policy Configuration](#)

|                |                                                                                                                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | Verify information about security policies.                                                                                                                                                                                                            |
| <b>Action</b>  | From operational mode, enter the <b>show security policies detail</b> command to display a summary of all security policies configured on the device.                                                                                                  |
| <b>Meaning</b> | The output displays information about policies configured on the system. Verify the following information: <ul style="list-style-type: none"> <li>• From and to zones</li> <li>• Source and destination addresses</li> <li>• Match criteria</li> </ul> |

## [Example: Configuring a Security Policy to Permit or Deny Selected Traffic](#)

This example shows how to configure a security policy to permit or deny selected traffic.

- [Requirements on page 157](#)
- [Overview on page 157](#)

- Configuration on page 158
- Verification on page 160

## Requirements

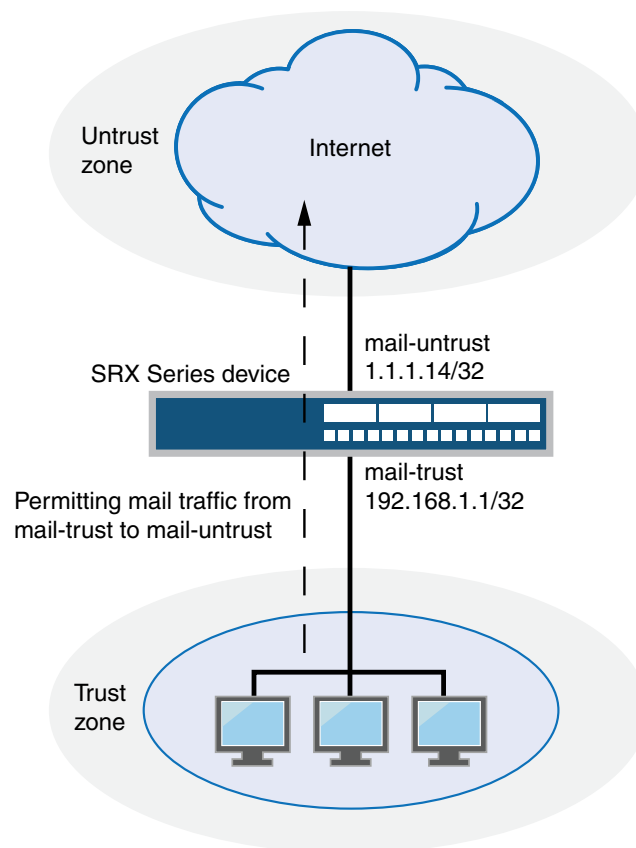
Before you begin:

- Create zones. See “Example: Creating Security Zones” on page 114.
- Configure an address book and create addresses for use in the policy. See “Example: Configuring Address Books and Address Sets” on page 139.
- Create an application (or application set) that indicates that the policy applies to traffic of that type. See “Example: Configuring Applications and Application Sets” on page 189.
- Permit traffic to and from trust and untrust zones. See “Example: Configuring a Security Policy to Permit or Deny All Traffic” on page 152.

## Overview

In a Junos OS, security policies enforce rules for the transit traffic, in terms of what traffic can pass through the device, and the actions that need to take place on the traffic as it passes through the device. From the perspective of security policies, the traffic enters one security zone and exits another security zone. In this example, you configure a specific security to allow only e-mail traffic from a host in the trust zone to a server in the untrust zone. No other traffic is allowed. See Figure 12 on page 158.

Figure 12: Permitting Selected Traffic



## Configuration

### CLI Quick Configuration

To quickly configure a security policy to allow selected traffic, copy the following commands and paste them into the CLI:

[edit]

```
set security zones security-zone trust interfaces ge-0/0/2 host-inbound-traffic
system-services all
```

```
set security zones security-zone untrust interfaces ge-0/0/1 host-inbound-traffic
system-services all
```

```
set security zones security-zone untrust address-book address mail-untrust 1.1.1.24/32
```

```
set security zones security-zone trust address-book address mail-trust 192.168.1.1/32
```

```
set security policies from-zone trust to-zone untrust policy permit-mail match
source-address mail-trust
```

```
set security policies from-zone trust to-zone untrust policy permit-mail match
destination-address mail-untrust
```

```
set security policies from-zone trust to-zone untrust policy permit-mail match application
  junos-mail
```

```
set security policies from-zone trust to-zone untrust policy permit-mail then permit
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To configure a security policy to allow selected traffic:

1. Configure the interfaces and security zones.

```
[edit security zones]
```

```
user@host# set security-zone trust interfaces ge-0/0/2 host-inbound-traffic
  system-services all
```

```
user@host# set security-zone untrust interfaces ge-0/0/1 host-inbound-traffic
  system-services all
```

2. Create address book entries for both client and server.

```
[edit security zones]
```

```
user@host# set security-zone untrust address-book address mail-untrust 1.1.1.24/32
```

```
user@host# set security-zone trust address-book address mail-trust 192.168.1.1/32
```

3. Define the policy to permit mail traffic.

```
[edit security policies from-zone trust to-zone untrust]
```

```
user@host# set policy permit-mail match source-address mail-trust
```

```
user@host# set policy permit-mail match destination-address mail-untrust
```

```
user@host# set policy permit-mail match application junos-mail
```

```
user@host# set policy permit-mail then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security policies** and **show security zones** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security policies
from-zone trust to-zone untrust {
  policy permit-mail {
    match {
      source-address mail-trust;
      destination-address mail-untrust;
      application junos-mail;
    }
  }
}
```

```

    }
    then {
        permit;
    }
}
}

user@host# show security zones
security-zone trust {
    address-book {
        address mail-trust 192.168.1.1/32;
    }
    host-inbound-traffic {
        system-services {
            all;
        }
        interfaces {
            ge-0/0/2 {
                host-inbound-traffic {
                    system-services {
                        all;
                    }
                }
            }
        }
    }
}
security-zone untrust {
    address-book {
        address mail-untrust 1.1.1.24/32;
    }
    interfaces {
        ge-0/0/1 {
            host-inbound-traffic {
                system-services {
                    all;
                }
            }
        }
    }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Policy Configuration on page 160

### Verifying Policy Configuration

**Purpose** Verify information about security policies.

**Action** From operational mode, enter the **show security policies detail** command to display a summary of all security policies configured on the device.

**Meaning** The output displays information about policies configured on the system. Verify the following information:

- From and to zones
- Source and destination addresses
- Match criteria

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policies Overview on page 145
- Example: Configuring a Security Policy to Permit or Deny All Traffic on page 152

## Example: Configuring a Security Policy to Permit or Deny Wildcard Address Traffic

This example shows how to configure a security policy to permit or deny wildcard address traffic.

- Requirements on page 161
- Overview on page 161
- Configuration on page 162
- Verification on page 164

### Requirements

Before you begin:

- Understand wildcard addresses. See “Understanding Wildcard Addresses” on page 150.
- Create zones. See “Example: Creating Security Zones” on page 114.
- Configure an address book and create addresses for use in the policy. See “Example: Configuring Address Books and Address Sets” on page 139.
- Create an application (or application set) that indicates that the policy applies to traffic of that type. See “Example: Configuring Applications and Application Sets” on page 189.
- Permit traffic to and from trust and untrust zones. See “Example: Configuring a Security Policy to Permit or Deny All Traffic” on page 152.
- Permit e-mail traffic to and from trust and untrust zones. See “Example: Configuring a Security Policy to Permit or Deny Selected Traffic” on page 156

### Overview

In the Junos operating system (Junos OS), security policies enforce rules for the transit traffic, in terms of what traffic can pass through the device, and the actions that need to take place on the traffic as it passes through the device. From the perspective of security policies, the traffic enters one security zone and exits another security zone. In this example, you configure a specific security to allow only wildcard address traffic from a host in the trust zone to the untrust zone. No other traffic is allowed.

## Configuration

**CLI Quick Configuration** To quickly configure a security policy to allow wildcard address traffic, copy the following commands and paste them into the CLI:

```
[edit]
```

```
set security zones security-zone trust interfaces ge-0/0/2 host-inbound-traffic  
system-services all
```

```
set security zones security-zone untrust interfaces ge-0/0/1 host-inbound-traffic  
system-services all
```

```
set security zones security-zone trust address-book address wildcard-trust  
wildcard-address 192.168.0.11/255.255.0.255
```

```
set security policies from-zone trust to-zone untrust policy permit-wildcard match  
source-address wildcard-trust
```

```
set security policies from-zone trust to-zone untrust policy permit-wildcard match  
destination-address any
```

```
set security policies from-zone trust to-zone untrust policy permit-wildcard match  
application any
```

```
set security policies from-zone trust to-zone untrust policy permit-wildcard then permit
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To configure a security policy to allow selected traffic:

1. Configure the interfaces and security zones.

```
[edit security zones]
```

```
user@host# set security-zone trust interfaces ge-0/0/2 host-inbound-traffic  
system-services all
```

```
user@host# set security-zone untrust interfaces ge-0/0/1 host-inbound-traffic  
system-services all
```

2. Create an address book entry for the host.

```
[edit security zones]
```

```
user@host# set security-zone trust address-book address wildcard-trust  
wildcard-address 192.168.0.11/255.255.0.255
```

3. Define the policy to permit wildcard address traffic.

```
[edit security policies from-zone trust to-zone untrust]
```



```
user@host# set policy permit-wildcard match source-address wildcard-trust
```

```
user@host# set policy permit-wildcard match destination-address any
```

```
user@host# set policy permit-wildcard match application any
```

```
user@host# set policy permit-wildcard then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security policies** and **show security zones** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security policies
from-zone trust to-zone untrust {
  policy permit-wildcard {
    match {
      source-address wildcard-trust;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}

user@host# show security zones
security-zone trust {
  address-book {
    address wildcard-trust {
      wildcard-address 192.168.0.11/255.255.0.255;
    }
  }
  host-inbound-traffic {
    system-services {
      all;
    }
  }
  interfaces {
    ge-0/0/2 {
      host-inbound-traffic {
        system-services {
          all;
        }
      }
    }
  }
}

security-zone untrust {
  interfaces {
    ge-0/0/1 {
      host-inbound-traffic {
        system-services {
```

```

    all;
  }
}
}
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Policy Configuration on page 164

### Verifying Policy Configuration

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>               | Verify information about address books and zones.                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Action</b>                | From operational mode, enter the <b>show security policies policy-name permit-wildcard detail</b> command to display details about the permit-wildcard security policy configured on the device.                                                                                                                                                                                                                                                  |
| <b>Meaning</b>               | The output displays information about the permit-wildcard policy configured on the system. Verify the following information: <ul style="list-style-type: none"> <li>• From and To zones</li> <li>• Source and destination addresses</li> <li>• Match criteria</li> </ul>                                                                                                                                                                          |
| <b>Related Documentation</b> | <ul style="list-style-type: none"> <li>• <a href="#">Junos OS Feature Support Reference for SRX Series and J Series Devices</a></li> <li>• Security Policies Overview on page 145</li> <li>• Understanding Wildcard Addresses on page 150</li> <li>• Example: Configuring a Security Policy to Permit or Deny All Traffic on page 152</li> <li>• Example: Configuring a Security Policy to Permit or Deny Selected Traffic on page 156</li> </ul> |

## Application Firewall Overview

Many dynamic applications use HTTP static ports to tunnel non-HTTP traffic through the network. Such applications can send traffic that might not be adequately controlled by standard network firewall policies, leading to a security threat. Standard policies function based on IP address and port and therefore are not effective with these dynamic applications. To avoid these security issues, an additional security control that functions based on the application ID has been introduced.

The application firewall is implemented as a plug-in with its own policies or rule set. To enable this application firewall feature, the security policy configuration is modified by

adding a reference to the application firewall rule set within the policy. The rule sets are defined independent of network policy.

To implement the application firewall feature, do the following:

- Defining application firewall rule sets
- Creating the rule sets reference within the security policy

This topic includes the following sections:

- Understanding Application Firewall Rule Sets on page 165
- Configuring an Application Firewall Within a Security Policy on page 166
- Application Firewall Support in Chassis Cluster on page 166

## Understanding Application Firewall Rule Sets

Application firewall policy is defined by a collection of rule sets with a rule that matches the application ID defined by the application signature.

There are two sources for the application signatures:

- Users can download a predefined signatures from the Juniper Networks Security Engineering website.
- Users can define their own signatures using the Junos OS configuration CLI.

The rule sets are defined with at least two rules. One is the default rule, and the other rule contains a list of applications to deny or permit.

A rule set permits or denies specified types of traffic between source and destination.

Each rule set consists of:

- A name for the rule set.
- A set of rules. Each rule consists of a rule name, match criteria, and an action. The match criteria define the conditions that must be satisfied to apply the rule. They are based on a list of dynamic applications. The action can be permit or deny.
- A default rule defining the action to be taken when the identified dynamic application is not specified in any rules of the rule set.

There is a limit for the overall number of rule sets and rules. There is no limit for the number of rules in a rule set. There is no limit for the number of dynamic applications in a rule. Each rule set must have a default rule. The default rule is applied when the application ID is not specified in any rules of the rule set.



**NOTE:** The default rule and the rule within the same rule set must be defined with different actions (permit or deny). If the actions in the rules are the same, a commit failure error message is displayed, indicating that the user needs to change the rule within the rule set.

The application firewall policy decides the application ID as an unknown application ID during the following cases:

- No application ID matches the traffic.
- The system encounters an error when identifying the application.
- Failover sessions.

When the application ID is identified as unknown, then the traffic is processed based on the action defined in the rule for unknown in the rule set. When there is no rule defined for unknown in the rule set, the default rule is applied for unknown dynamic applications.



**NOTE:** The `junos:UNKNOWN` keyword is reserved for unknown dynamic applications.

## Configuring an Application Firewall Within a Security Policy

The application firewall is enabled by creating a reference to the rule set within the security policy. When the policy with the application firewall rule set is selected, the application firewall rule set is triggered. The traffic is processed by the rules defined in the rule sets, based on the application ID.

During packet processing, when the policy search returns with a policy enabled with the application firewall, the application ID of the traffic is identified. The rule set matching the application ID is selected to process the traffic with permit or deny actions. When the action of the rule determines that the packet should be dropped, the session is closed and all the services configured within the policy are terminated. The application firewall acts as a filter to decide whether to permit or deny the traffic, based on the application running on the session.

## Application Firewall Support in Chassis Cluster

When the application ID is not identified during failover sessions, the ID is considered an unknown application ID. During this session, the traffic is processed based on the action defined in a rule specified for unknown. If there is no rule defined for unknown, then the default rule is applied.

When the application ID is identified before sessions failover,, the same action that is taken before the failover is effective after the failover. The application firewall action taken before and after failover depends on the application ID state, as shown in Table 15 on page 166.

**Table 15: Application Firewall Actions**

| Before Failover      |                             | After Failover       |                             |
|----------------------|-----------------------------|----------------------|-----------------------------|
| Application ID State | Application Firewall Action | Application ID State | Application Firewall Action |
| Success              | Deny                        | Success              | Deny                        |
| Success              | Permit                      | Success              | Permit                      |

Table 15: Application Firewall Actions (*continued*)

| Before Failover |   | After Failover |                                                          |
|-----------------|---|----------------|----------------------------------------------------------|
| Pending         | — | UNKNOWN        | Action based on the rule defined for unknown application |



**NOTE:** In-service software upgrade (unified ISSU) is not supported due to lack of chassis cluster infrastructure support. Thus, the failover event is controlled through the application firewall policy by allowing or denying the unknown dynamic applications.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Security Policies Overview on page 145](#)
- [Understanding Security Policy Rules on page 148](#)
- [Understanding Security Policy Elements on page 151](#)
- [Security Policies Configuration Overview on page 151](#)

## Example: Configuring Application Firewall Rule Sets Within Security Policy (CLI)

This example shows how to configure application firewall rule sets within the security policy.

- [Requirements on page 167](#)
- [Overview on page 167](#)
- [Configuration on page 168](#)
- [Verification on page 172](#)

### Requirements

- Create zones. See “Example: Creating Security Zones” on page 114.
- Configure an address book with addresses for the policy. See “Example: Configuring Address Books and Address Sets” on page 139.
- Create an application (or application set) that indicates that the policy applies to traffic of that type. See “Example: Configuring Applications and Application Sets” on page 189.

### Overview

In Junos OS, the security policies provide firewall security functionality by enforcing rules for the traffic so that traffic passing through the device is permitted or denied based on the action defined in the rules. The application firewall support in the policies provides additional security control for dynamic applications.

The application firewall is defined by a collection of rule sets. These rule sets can be defined independently and shared across network security policies. A rule set defines the rules that match the application ID detected, based on the application signature.

This configuration example shows how to:

- Permit or deny selected traffic from the untrust zone to the trust zone, based on the application firewall rule sets defined with the rules matching the dynamic applications.

## Configuration

### CLI Quick Configuration

To quickly configure the security policies with application firewall rule sets, copy the following commands and paste them into the CLI:

```
[edit]
set security policies from-zone untrust to-zone trust policy policy1 match source-address
  1.1.1.0
set security policies from-zone untrust to-zone trust policy policy1 match
  destination-address 2.2.2.0
set security policies from-zone untrust to-zone trust policy policy1 match application
  junos-http
set security policies from-zone untrust to-zone trust policy policy1 then permit
  application-services application-firewall rule-set rs1
set security policies from-zone untrust to-zone trust policy policy2 match source-address
  1.1.1.0
set security policies from-zone untrust to-zone trust policy policy2 match
  destination-address 2.2.2.0
set security policies from-zone untrust to-zone trust policy policy2 match application any
set security policies from-zone untrust to-zone trust policy policy2 then permit
  application-services application-firewall rule-set rs2
set security application-firewall rule-sets rs1 rule r1 match dynamic-application
  [junos:KAZZA junos:EDONKEY junos:YSMG]
set security application-firewall rule-sets rs1 rule r1 then deny
set security application-firewall rule-sets rs1 default-rule permit
set security application-firewall rule-sets rs2 rule r1 match dynamic-application
  [junos:FACEBOOK-ACCESS junos:GOOGLE-TALK junos:MEEBO junos:UNKNOWN]
set security application-firewall rule-sets rs2 rule r1 then permit
set security application-firewall rule-sets rs2 default-rule deny
```

### J-Web Quick Configuration

To use the J-Web interface to configure two security policies with application firewall rule sets:

1. Configure rule set rs1 and a single associated rule:
  - a. Select **Configure>Security>Application FW** to display the Application Firewall configuration page.
  - b. Click **Add** in the upper pane, and enter the following information:
    - Rule Set Name: **rs1**
    - Default Rule: **permit**
  - c. Click **Add** in the Rules pane, and enter the following information:
    - Rule Name: **r1**

- Match Dynamic Application: **junos:Kazza junos:EDONKEY junos:YSMG**
  - Action: **deny**
- d. Click **OK** to return to the Application Firewall configuration page.
2. Configure rule set rs2 and a single associated rule:
- a. Click **Add** in the upper pane, and enter the following information:
- Rule Set Name: **rs2**
  - Default Rule: **deny**
- b. Click **Add** in the Rule pane, and enter the following information.
- Rule Name: **r1**
  - Match Dynamic Application: **junos:FACEBOOK-ACCESS junos:GOOGLE-TALK junos:MEEBO junos:UNKNOWN**
  - Action: **permit**
- c. Click **OK** to return to the Application Firewall configuration page.
3. Configure policy1 and assign the application service rule set rs1:
- a. Select **Configure>Security>Policy>FW Policies** to display the Security Policy page.
- b. Click **Add**, and click the **Policy** tab.
- c. Enter the following policy information:
- Policy Name: **policy1**
  - Policy Action: **permit**
  - From Zone: **untrust**
  - To Zone: **trust**
  - Source Address: **1.1.1.0**
  - Destination Address: **2.2.2.0**
  - Applications: **junos-http**
- d. Click the **Application Services** tab, and enter the following application firewall information:

- Application Firewall - Rule Set: **rs1**
- e. Click **OK**.
4. Configure policy2 and assign the application service rule set rs2:
    - a. Click **Add**, and click the **Policy** tab.
    - b. Enter the following policy information:
      - Policy Name: **policy2**
      - Policy Action: **permit**
      - From Zone: **untrust**
      - To Zone: **trust**
      - Source Address: **1.1.1.0**
      - Destination Address: **2.2.2.0**
      - Applications: **any**
    - c. Click the **Application Services** tab, and enter the following application firewall information:
      - Application Firewall - Rule Set: **rs2**
    - d. Click **OK**.
  5. If your configuration is complete, select **Commit Options>Commit**.

**Step-by-Step  
Procedure**

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure two security policies with application firewall rule sets that permit or deny traffic from different dynamic applications:

1. Configure a policy to process the traffic that goes to the HTTP static ports with the application firewall rule set rs1.

```
[edit security policies from-zone untrust to-zone trust policy policy1]
user@host# set match source-address 1.1.1.0
user@host# set match destination-address 2.2.2.0
user@host# set match application junos-http
user@host# set then permit application-services application-firewall rule-set rs1
```

2. Configure another policy to process any traffic that does not go to the HTTP static ports with the application firewall rule set rs2.

```
[edit security policies from-zone untrust to-zone trust policy policy2]
user@host# set match source-address 1.1.1.0
user@host# set match destination-address 2.2.2.0
user@host# set match application any
user@host# set then permit application-services application-firewall rule-set rs2
```



3. Define the application firewall rule set rs1 to deny traffic from selected dynamic applications.

```
[edit security application-firewall rule-sets rs1]
user@host# set rule r1 match dynamic-application [junos:KAZZA junos:EDONKEY
  junos:YSMG]
user@host# set rule r1 then deny
user@host# set default-rule permit
```

4. Define the application firewall rule set rs2 to permit traffic from selected dynamic applications.

```
[edit security application-firewall rule-sets rs2]
user@host# set rule r1 match dynamic-application [junos:FACEBOOK-ACCESS
  junos:GOOGLE-TALK junos:MEEBO junos:UNKNOWN]
user@host# set rule r1 then permit
user@host# set default-rule deny
```

**Results** From configuration mode, confirm your configuration by entering the **show security policies** and **show security application-firewall** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security policies
  from-zone untrust to-zone trust {
    policy 1 {
      match {
        source-address 1.1.1.0;
        destination-address 2.2.2.0;
        application junos-http;
      }
      then {
        permit {
          application-services {
            application-firewall {
              rule-set rs1;
            }
          }
        }
      }
    }
  }
  policy 2 {
    match {
      source-address 1.1.1.0;
      destination-address 2.2.2.0;
      application any;
    }
    then {
      permit {
        application-services {
          application-firewall {
            rule-set rs2;
          }
        }
      }
    }
  }
}
```

```

    }
  }
}
user@host# show security application-firewall
rule-sets rs1 {
  rule r1 {
    match {
      dynamic-application [junos:KAZZA junos:EDONKEY junos:YSMG];
    }
    then {
      deny;
    }
  }
  default-rule {
    permit;
  }
}
rule-sets rs2 {
  rule r1 {
    match {
      dynamic-application [junos:FACEBOOK-ACCESS junos:GOOGLE-TALK
        junos:MEEBO junos:UNKNOWN];
    }
    then {
      permit;
    }
  }
  default-rule {
    deny;
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Application Firewall Configuration on page 172

### Verifying Application Firewall Configuration

**Purpose** Verify information about application firewall support enabled under the security policy.

**Action** To verify the security policy configuration enabled with application firewall, enter the **show security policies** and **show security policies detail** commands. To verify all the application firewall rule sets configured on the device, enter the **show security application-firewall rule-set all** command.

**Meaning** The output displays information about application firewall enabled policies configured on the system. Verify the following information.

- Rule sets
- Rules

- Match criteria

**Related  
Documentation**

- Security Policies Overview on page 145
- Application Firewall Overview on page 164

## Understanding Security Policy Ordering

Junos OS offers a tool for verifying that the order of policies in the policy list is valid.

It is possible for one policy to eclipse, or *shadow*, another policy. Consider the following examples:

Example 1

[edit]

```
user@host# set security zones security-zone trust interfaces ge-0/0/2 host-inbound-traffic
system-services all
```

```
user@host# set security zones security-zone untrust interfaces ge-0/0/1
host-inbound-traffic system-services all
```

```
user@host# set security policies from-zone trust to-zone untrust policy permit-all match
source-address any
```

```
user@host# set security policies from-zone trust to-zone untrust match
destination-address any
```

```
user@host# set security policies from-zone trust to-zone untrust match application any
```

```
user@host# set security policies from-zone trust to-zone untrust set then permit
```

```
user@host# set security policies from-zone untrust to-zone trust policy deny-all match
source-address any
```

```
user@host# set security policies from-zone untrust to-zone trust policy deny-all match
destination-address any
```

```
user@host# set security policies from-zone untrust to-zone trust policy deny-all match
application any
```

```
user@host# set security policies from-zone untrust to-zone trust policy deny-all then
deny
```

Example 2

[edit]

```
user@host# set security zones security-zone trust interfaces ge-0/0/2.0
host-inbound-traffic system-services all
```

```
user@host# set security zones security-zone untrust interfaces ge-0/0/1.0
host-inbound-traffic system-services all
```

```
user@host# set security zones security-zone untrust address-book address mail-untrust
1.1.1.24/32
```

```
user@host# set security zones security-zone trust address-book address mail-trust
192.168.1.1/32
```

```
user@host# set security policies from-zone trust to-zone untrust policy permit-mail match
source-address mail-trust
```

```
user@host# set security policies from-zone trust to-zone untrust policy permit-mail match
destination-address mail-untrust
```

```
user@host# set security policies from-zone trust to-zone untrust policy permit-mail match
application junos-mail
```

```
user@host# set security policies from-zone trust to-zone untrust policy permit-mail then
permit
```

In examples 1 and 2, where policy **permit-mail** is configured after policy **permit-all** from zone **trust** to zone **untrust**. All traffic coming from zone **untrust** matches the first policy **permit-all** and is allowed by default. No traffic matches policy **permit-mail**.

Because Junos OS performs a policy lookup starting from the top of the list, when it finds a match for traffic received, it does not look any lower in the policy list. To correct the previous example, you can simply reverse the order of the policies, putting the more specific one first:

```
[edit]
```

```
user@host# insert security policies from-zone trust to-zone untrust policy permit-mail
before policy permit-all
```

In cases where there are dozens or hundreds of policies, the eclipsing of one policy by another might not be so easy to detect. To check if policies are being shadowed, enter the following command:

```
[edit]
```

```
user@host# show policy-options <policy-name>
```

This command reports the shadowing and shadowed policies. It is then the administrator's responsibility to correct the situation.



**NOTE:** The concept of policy *shadowing* refers to the situation where a policy higher in the policy list always takes effect before a subsequent policy. Because the policy lookup always uses the first policy it finds that matches the five-part tuple of the source and destination zone, source and destination address, and application type, if another policy applies to the same tuple (or a subset of the tuple), the policy lookup uses the first policy in the list and never reaches the second one.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policies Configuration Overview on page 151
- Example: Configuring a Security Policy to Permit or Deny All Traffic on page 152
- Example: Configuring a Security Policy to Permit or Deny Selected Traffic on page 156

## Example: Reordering the Policies

This example shows how to move policies around after they have been created.

- Requirements on page 175
- Overview on page 175
- Configuration on page 175
- Verification on page 176

### Requirements

Before you begin:

- Create zones. See “Example: Creating Security Zones” on page 114.
- Configure the address book and create addresses for use in the policy. See “Example: Configuring Address Books and Address Sets” on page 139.

### Overview

To reorder policies to correct shadowing, you can simply reverse the order of the policies, putting the more specific one first.

### Configuration

#### Step-by-Step Procedure

To reorder existing policies:

1. Reorder two existing policies by entering the following command:
 

```
[edit]
user@host# insert security policies from-zone trust to-zone untrust policy
permit-mail before policy permit-all
```
2. If you are done configuring the device, commit the configuration.
 

```
[edit]
```

```
user@host# commit
```

## Verification

To verify the configuration is working properly, enter the **show security policies** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policies Overview on page 145
- Understanding Security Policy Ordering on page 173

## Troubleshooting Security Policies

---

- Checking a Security Policy Commit Failure on page 176
- Verifying a Security Policy Commit on page 176
- Debugging Policy Lookup on page 177

### Checking a Security Policy Commit Failure

**Problem** Most policy configuration failures occur during a commit or runtime.

Commit failures are reported directly on the CLI when you execute the CLI command **commit-check** in configuration mode. These errors are configuration errors, and you cannot commit the configuration without fixing these errors.

**Solution** To fix these errors, do the following:

1. Review your configuration data.
2. Open the file `/var/log/nsd_chk_only`. This file is overwritten each time you perform a commit check and contains detailed failure information.

### Verifying a Security Policy Commit

**Problem** Upon performing a policy configuration commit, if you notice that the system behavior is incorrect, use the following steps to troubleshoot this problem:

**Solution**

1. Operational **show** Commands—Execute the operational commands for security policies and verify that the information shown in the output is consistent with what you expected. If not, the configuration needs to be changed appropriately.
2. Traceoptions—Set the **traceoptions** command in your policy configuration. The flags under this hierarchy can be selected as per user analysis of the **show** command output. If you cannot determine what flag to use, the flag option **all** can be used to capture all trace logs.

```
user@host# set security policies traceoptions <flag all>
```

You can also configure an optional filename to capture the logs.

```
user@host# set security policies traceoptions <filename>
```

If you specified a filename in the trace options, you can look in the `/var/log/<filename>` for the log file to ascertain if any errors were reported in the file. (If you did not specify a filename, the default filename is `eventd`.) The error messages indicate the place of failure and the appropriate reason.

After configuring the trace options, you must recommit the configuration change that caused the incorrect system behavior.

## Debugging Policy Lookup

**Problem** When you have the correct configuration, but some traffic was incorrectly dropped or permitted, you can enable the **lookup** flag in the security policies traceoptions. The **lookup** flag logs the lookup related traces in the trace file.

**Solution** `user@host# set security policies traceoptions <flag lookup>`

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policies Overview on page 145
- Checking a Security Policy Commit Failure on page 176
- Verifying a Security Policy Commit on page 176
- Debugging Policy Lookup on page 177
- Monitoring Policy Statistics on page 177

## Monitoring Policy Statistics

**Purpose** Monitor and record traffic that Junos OS permits or denies based on previously configured policies.

**Action** To monitor traffic, enable the count and log options.

**Count**—Configurable in an individual policy. If count is enabled, statistics are collected for the number of packets, bytes, and sessions that enter the firewall for a given policy. For counts (only for packets and bytes), you can specify that alarms be generated whenever the traffic exceeds specified thresholds.

**Log**—Logging capability can be enabled with security policies during session initialization (**session-init**) or session close (**session-close**) stage.

- To view logs from denied connections, enable log on **session-init**.
- To log sessions after their conclusion/tear-down, enable log on **session-close**.



**NOTE:** Session log is enabled at real time in the flow code which impacts the user performance. If both **session-close** and **session-init** are enabled, performance is further degraded as compared to enabling **session-init** only.

For details about information collected for session logs, see “Information Provided in Session Log Entries for SRX Series Services Gateways” on page 16.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policies Overview on page 145
- Troubleshooting Security Policies on page 176
- Checking a Security Policy Commit Failure on page 176
- Verifying a Security Policy Commit on page 176
- Debugging Policy Lookup on page 177

## Matching Security Policies

The **show security match-policies** command allows you to troubleshoot traffic problems using the match criteria: source port, destination port, source IP address, destination IP address, and protocol. For example, if your traffic is not passing because either an appropriate policy is not configured or the match criteria is incorrect, the **show security match-policies** command allows you to work offline and identify where the problem actually exists. It uses the search engine to identify the problem and thus enables you to use the appropriate match policy for the traffic.

The **result-count** option specifies how many policies to display. The first enabled policy in the list is the policy that is applied to all matching traffic. Other policies below it are “shadowed” by the first and are never encountered by matching traffic.



**NOTE:** The **show security match-policies** command is applicable only to security policies; IDP policies are not supported.

#### Example 1: show security match-policies

```
user@host> show security match-policies from-zone z1, to-zone z2 source-ip 10.10.10.1
destination-ip 30.30.30.1 source-port 1 destination-port 21 protocol tcp
Policy: p1, action-type: permit, State: enabled, Index: 4
Sequence number: 1
From zone: z1, To zone: z2
Source addresses:
  a2: 20.20.0.0/16
  a3: 10.10.10.1/32
Destination addresses:
  d2: 40.40.0.0/16
  d3: 30.30.30.1/32
Application: junos-ftp
IP protocol: tcp, ALG: ftp, Inactivity timeout: 1800
Source port range: [0-0]
Destination port range: [21-21]
```

#### Example 2: Using the result-count Option

By default, the output list contains the policy that will be applied to traffic with the specified characteristics. To list more than one policy that match the criteria, use the **result-count** option. The first policy listed is always the policy that will be applied to matching traffic. If the **result-count** value is from 2 to 16, the output includes all policies



that match the criteria up to the specified **result-count**. All policies listed after the first are “shadowed” by the first policy and are never applied to matching traffic.

Use this option to test the positioning of a new policy or to troubleshoot a policy that is not applied as expected for particular traffic.

In the following example, the traffic criteria matches two policies. The first policy listed, **p1**, contains the action applied to the traffic. Policy **p15** is shadowed by the first policy, and its action, therefore, will not be applied to matching traffic.

```
user@host> show security match-policies source-ip 10.10.10.1 destination-ip 20.20.20.5
source_port 1004 destination_port 80 protocol tcp result_count 5
Policy: p1, action-type: permit, State: enabled, Index: 4
  Sequence number: 1
  From zone: zone-A, To zone: zone-B
  Source addresses:
    sa1: 10.10.0.0/16
  Destination addresses:
    da5: 20.20.0.0/16
  Application: any
  IP protocol: 1, ALG: 0, Inactivity timeout: 0
  Source port range: [1000-1030]
  Destination port range: [80-80]

Policy: p15, action-type: deny, State: enabled, Index: 18
  Sequence number: 15
  From zone: zone-A, To zone: zone-B
  Source addresses:
    sa11: 10.10.10.1/32
  Destination addresses:
    da15: 20.20.20.5/32
  Application: any
  IP protocol: 1, ALG: 0, Inactivity timeout: 0
  Source port range: [1000-1030]
  Destination port range: [80-80]
```

For more information on matching policies and descriptions of the options and output fields, see the [Junos OS CLI Reference](#).



## CHAPTER 7

# Security Policy Schedulers

- Security Policy Schedulers Overview on page 181
- Example: Configuring Schedulers on page 182
- Verifying Scheduled Policies on page 184

### Security Policy Schedulers Overview

---

Schedulers are powerful features that allow a policy to be activated for a specified duration. You can define schedulers for a single (nonrecurrent) or recurrent time slot within which a policy is active. You can create schedulers irrespective of a policy, meaning that a scheduler cannot be used by any policies. However, if you want a policy to be active within a scheduled time, then you must first create a scheduler.

When a scheduler times out, the associated policy is deactivated and all sessions associated with the policy are also timed out.

If a policy contains a reference to a scheduler, the schedule determines when the policy is active, that is, when it can be used as a possible match for traffic. Schedulers allow you to restrict access to a resource for a period of time or remove a restriction.

The following guidelines apply to schedulers:

- A scheduler can have multiple policies associated with it; however, a policy cannot be associated with multiple schedulers.
- A policy is active during the time when the scheduler it refers to is also active.
- When a scheduler is off, the policy is unavailable for policy lookup.
- A scheduler can be configured as one of the following:
  - Scheduler can be active for a single time slot, as specified by a start date and time and a stop date and time.
  - Scheduler can be active forever (recurrent), but as specified by the daily schedule. The schedule on a specific day (time slot) takes priority over the daily schedule.
  - Scheduler can be active within a time slot as specified by the weekday schedule.
  - Scheduler can have a combination of two time slots (daily and timeslot).

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Security Policies Overview on page 145](#)
  - [Example: Configuring Schedulers on page 182](#)
  - [Verifying Scheduled Policies on page 184](#)

## Example: Configuring Schedulers

---

This example shows how to configure schedulers.

- [Requirements on page 182](#)
- [Overview on page 182](#)
- [Configuration on page 182](#)
- [Verification on page 184](#)

### Requirements

Before you begin:

- Understand security policies schedulers. See “Security Policies Overview” on page 145.
- Configure security zones before applying this configuration.

### Overview

Schedulers are powerful features that allow a policy to be activated for a specified duration. You can define schedulers for a single (nonrecurrent) or recurrent time slot within which a policy is active. If you want a policy to be active within a scheduled time, then you must first create a scheduler.

To configure a scheduler, you enter a meaningful name and a start and stop time for the scheduler. You can also attach comments.

In this example, you:

- Specify the scheduler, sch1, that allows a policy, which refers to it, to be used for packet match checks from 8 AM to 9 PM all days of the week from October 1, 2009 to June, 2010 except Sundays.
- Configure another scheduler, SunHrs, to check for packet matches from noon to 6 PM on Sundays.
- Create a policy, abc, and specify the match conditions and action to be taken on traffic that matches the specified conditions. and bind the schedulers to the policy to allow access during the specified weekend hours.

### Configuration

- CLI Quick Configuration** To quickly configure schedulers, copy the following commands and paste them into the CLI.

```
[edit]
set schedulers scheduler sch1 start-date 2009-10-01.08:00 stop-date 2010-06-01.21:00
set schedulers scheduler sch1 sunday exclude
set schedulers scheduler SunHrs sunday start-time 12:00 stop-time 18:00
set security policies from-zone green to-zone red policy abc match source-address any
set security policies from-zone green to-zone red policy abc match destination-address
  any
set security policies from-zone green to-zone red policy abc match application any
set security policies from-zone green to-zone red policy abc then permit
set security policies from-zone green to-zone red policy abc scheduler-name sch1
set security policies default-policy permit-all
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a scheduler:

1. Set a scheduler.

```
[edit schedulers ]
user@host# set scheduler sch1 start-date 2009-10-01.08:00 stop-date
2010-06-01.21:00
user@host# set scheduler sch1 sunday exclude
```

2. Set another scheduler.

```
[edit schedulers]
user@host# set scheduler SunHrs sunday start-time 12:00 stop-time 18:00
```

3. Specify the match conditions for the policy.

```
[edit security policies from-zone green to-zone red policy abc]
user@host# set match source-address any destination-address any application
any
```

4. Specify the action.

```
[edit security policies from-zone green to-zone red policy abc]
user@host# set then permit
```

5. Associate the scheduler to the policy.

```
[edit security policies from-zone green to-zone red policy abc ]
user@host# set scheduler-name sch1
```

**Results** From configuration mode, confirm your configuration by entering the **show schedulers** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
[user@host]show schedulers
scheduler sch1 {
  start-date 2009-10-01.08:00 stop-date 2010-06-01.21:00;
  sunday exclude;
}
scheduler SunHrs {
  sunday {
```

```

        start-time 12:00 stop-time 18:00;
    }
}
[edit]
[user@host]show security policies
from-zone green to-zone red {
  policy abc {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
    scheduler-name sch1;
  }
}
default-policy {
  permit-all;
}

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Schedulers are Active on page 184
- Verifying Policies on page 184

### Verifying Schedulers are Active

---

- Purpose** Verify if schedulers are enabled or not.
- Action** From operational mode, enter the **show schedulers** command.

### Verifying Policies

---

- Purpose** Verify if the policies are working.
- Action** From operational mode, enter the **show security policies** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Verifying Scheduled Policies on page 184

## Verifying Scheduled Policies

---

- Purpose** Display information about address books and zones.

**Action** Use the `show schedulers` CLI command to display information about schedulers configured on the system. If a specific scheduler is identified, detailed information is displayed for that scheduler only.

```

user@host# show schedulers
scheduler sche1 {
  /* This is sched1 */
  start-date 2006-11-02.12:12 stop-date 2007-11-02.12:11;
}
scheduler sche2 {
  daily {
    all-day;
  }
  sunday {
    start-time 16:00 stop-time 17:00;
  }
  friday {
    exclude;
  }
}
scheduler sche3 {
  start-date 2006-11-02.12:12 stop-date 2007-11-02.12:11;
  daily {
    start-time 10:00 stop-time 17:00
  }
  sunday {
    start-time 12:00 stop-time 14:00;
    start-time 16:00 stop-time 17:00;
  }
  monday {
    all-day;
  }
  friday {
    exclude;
  }
}

```

**Meaning** The output displays information about schedulers configured on the system. Verify the following information:

- Daily (recurrent) and one-time only (nonrecurrent) schedulers are configured correctly.
- Schedulers are active if policies are associated.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policies Overview on page 145
- Example: Configuring Schedulers on page 182





## CHAPTER 8

# Security Policy Applications

- Security Policy Applications Overview on page 187
- Policy Application Sets Overview on page 188
- Example: Configuring Applications and Application Sets on page 189
- Custom Policy Applications on page 190
- Policy Application Timeouts on page 194
- Understanding the ICMP Predefined Policy Application on page 198
- Default Behaviour of ICMP Unreachable Errors on page 202
- Understanding Internet-Related Predefined Policy Applications on page 202
- Understanding Microsoft Predefined Policy Applications on page 204
- Understanding Dynamic Routing Protocols Predefined Policy Applications on page 205
- Understanding Streaming Video Predefined Policy Applications on page 206
- Understanding Sun RPC Predefined Policy Applications on page 206
- Understanding Security and Tunnel Predefined Policy Applications on page 207
- Understanding IP-Related Predefined Policy Applications on page 208
- Understanding Instant Messaging Predefined Policy Applications on page 209
- Understanding Management Predefined Policy Applications on page 209
- Understanding Mail Predefined Policy Applications on page 211
- Understanding UNIX Predefined Policy Applications on page 211
- Understanding Miscellaneous Predefined Policy Applications on page 212

## Security Policy Applications Overview

---

Applications are types of traffic for which protocol standards exist. Each application has a transport protocol and destination port number(s) associated with it, such as TCP/port 21 for FTP and TCP/port 23 for Telnet. When you create a policy, you must specify an application for it.

You can select one of the predefined applications from the application book, or a custom application or application set that you created. You can see which application you can use in a policy by using the **show application** CLI command.



**NOTE:** Each predefined application has a source port range of 1–65535, which includes the entire set of valid port numbers. This prevents potential attackers from gaining access by using a source port outside of the range. If you need to use a different source port range for any predefined application, create a custom application. For information, see “Understanding Custom Policy Applications” on page 190.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policies Overview on page 145
- Understanding Security Policy Rules on page 148
- Understanding Security Policy Elements on page 151
- Policy Application Sets Overview on page 188

## Policy Application Sets Overview

When you create a policy, you must specify an application, or service, for it to indicate that the policy applies to traffic of that type. Sometimes the same applications or a subset of them can be present in multiple policies, making it difficult to manage. Junos OS allows you to create groups of applications called *application sets*. Application sets simplify the process by allowing you to manage a small number of application sets, rather than a large number of individual application entries.

The application (or application set) is referred to by security policies as match criteria for packets initiating sessions. If the packet matches the application type specified by the policy and all other criteria match, then the policy action is applied to the packet.

You can specify the name of an application set in a policy. In this case, if all of the other criteria match, any one of the applications in the application set serves as valid matching criteria; **any** is the default application name that indicates all possible applications.

Applications are created in the `.../applications/application/application-name` directory. You do not need to configure an application for any of the services that are predefined by the system.

In addition to predefined services, you can configure a custom service. After you create a custom service, you can refer to it in a policy.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policy Applications Overview on page 187
- Custom Application Mappings on page 190
- Understanding Policy Application Timeout Configuration and Lookup on page 194
- Example: Configuring Applications and Application Sets on page 189

## Example: Configuring Applications and Application Sets

This example shows how to configure applications and application sets.

- Requirements on page 189
- Overview on page 189
- Configuration on page 189
- Verification on page 190

### Requirements

Before you begin, configure the required applications. See “Policy Application Sets Overview” on page 188.

### Overview

Rather than creating or adding multiple individual application names to a policy, you can create an application set and refer to the name of the set in a policy. For example, for a group of employees, you can create an application set that contains all the approved applications.

In this example, you create an application set that are used to log into the servers in the ABC (intranet) zone, to access the database, and to transfer files.

- Define the applications in the configured application set.
- Managers in zone A and managers in zone B use these services. Therefore, give the application set a generic name, such as MgrAppSet.
- Create an application set for the applications that are used for e-mail and Web-based applications that are delivered by the two servers in the external zone.

### Configuration

#### Step-by-Step Procedure

To configure an application and application set:

1. Create an application set for managers.
 

```
[edit applications]
user@host# set application-set MgrAppSet application junos-ssh
user@host# set application-set MgrAppSet application junos-telnet
```
2. Create another application set for e-mail and Web-based Example: Setting a Policy Application Time applications.
 

```
[edit applications]
user@host# set application-set WebMailApps application junos-smtp
user@host# set application-set WebMailApps application junos-pop3
```
3. If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

## Verification

To verify the configuration is working properly, enter the **show applications** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policies Overview on page 145
- Security Policy Applications Overview on page 187

## Custom Policy Applications

---

- Understanding Custom Policy Applications on page 190
- Custom Application Mappings on page 190
- Example: Adding and Modifying Custom Policy Applications on page 191
- Example: Defining a Custom ICMP Application on page 192

## Understanding Custom Policy Applications

If you do not want to use predefined applications in your policy, you can easily create custom applications.

You can assign each custom application the following attributes:

- Name
- Transport protocol
- Source and destination port numbers for applications using TCP or UDP
- Type and code values for applications using ICMP
- Timeout value

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policy Applications Overview on page 187
- Custom Application Mappings on page 190
- Understanding Policy Application Timeout Configuration and Lookup on page 194
- Understanding Policy Application Timeouts Contingencies on page 195
- Example: Adding and Modifying Custom Policy Applications on page 191

## Custom Application Mappings

The application option specifies the Layer 7 application that maps to the Layer 4 application that you reference in a policy. A predefined application already has a mapping to a Layer 7 application. However, for custom applications, you must link the application to an application explicitly, especially if you want the policy to apply an Application Layer Gateway (ALG) or deep inspection to the custom application.



**NOTE:** Junos OS supports ALGs for numerous applications, including DNS, FTP, H.323, HTTP, RSH, SIP, Telnet, and TFTP.

Applying an ALG to a custom application involves the following two steps:

- Define a custom application with a name, timeout value, transport protocol, and source and destination ports.
- When configuring a policy, reference that application and the application type for the ALG that you want to apply.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policy Applications Overview on page 187
- Understanding Custom Policy Applications on page 190
- Understanding Policy Application Timeout Configuration and Lookup on page 194
- Understanding Policy Application Timeouts Contingencies on page 195
- Example: Adding and Modifying Custom Policy Applications on page 191

### Example: Adding and Modifying Custom Policy Applications

This example shows how to add and modify custom policy applications.

- Requirements on page 191
- Overview on page 191
- Configuration on page 192
- Verification on page 192

#### Requirements

Before you begin, create addresses and security zones. See “Example: Creating Security Zones” on page 114.

#### Overview

In this example, you create a custom application using the following information:

- A name for the application, such as **cust-telnet**.
- A range of source port numbers: **1** through **65535**.
- A range of destination port numbers to receive the application request, such as **1** through **65535**.
- Whether the application uses TCP or UDP, or some other protocol as defined by the Internet specifications.

## Configuration

**Step-by-Step Procedure** The following example requires you to navigate through various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To add and modify a custom policy application:

1. Configure TCP and specify the source port and destination port.

```
[edit applications application cust-telnet]
user@host# set protocol tcp source-port 1-65535 destination-port 23000
```

2. Specify the length of time that the application is inactive.

```
[edit applications application cust-telnet]
user@host# set inactivity-timeout 1800
```

3. Modify a custom policy application.

```
[edit applications application cust-telnet]
user@host# delete protocol tcp
user@host# set application-protocol ftp
```

4. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

## Verification

To verify the configuration is working properly, enter the **show applications application** command.



**NOTE:** The timeout value is in seconds. If you do not set it, the timeout value of a custom application is 1800 seconds. If you do not want an application to time out, type **never**.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Security Policies Overview on page 145](#)
- [Security Policy Applications Overview on page 187](#)
- [Understanding Custom Policy Applications on page 190](#)
- [Example: Defining a Custom ICMP Application on page 192](#)

## Example: Defining a Custom ICMP Application

This example shows how to define a custom ICMP application.

- [Requirements on page 193](#)
- [Overview on page 193](#)

- Configuration on page 194
- Verification on page 194

### Requirements

Before you begin:

- Understand custom policy application. See “Understanding Custom Policy Applications” on page 190.
- Understand the ICMP predefined policy application. See “Understanding the ICMP Predefined Policy Application” on page 198.

### Overview

Junos OS supports ICMP—as well as several ICMP messages—as predefined or custom applications. When configuring a custom ICMP application, you define a type and code.

- There are different message types within ICMP. For example:
  - type 0 = Echo Request message
  - type 3 = Destination Unreachable message
- An ICMP message type can also have a message code. The code provides more specific information about the message, as shown in Table 16 on page 193.

**Table 16: Message Descriptions**

| Message Type             | Message Code                                                  |
|--------------------------|---------------------------------------------------------------|
| 5 = Redirect             | 0 = Redirect datagram for the network (or subnet)             |
|                          | 1 = Redirect datagram for the host                            |
|                          | 2 = Redirect datagram for the type of application and network |
|                          | 3 = Redirect datagram for the type of application and host    |
| 11 = Time Exceeded Codes | 0 = Time to live exceeded in transit                          |
|                          | 1 = Fragment reassembly time exceeded                         |

Junos OS supports any type or code within the range of 0 through 55.

In this example, you define a custom application named `host-unreachable` using ICMP as the transport protocol. The type is 3 (for destination unreachable) and the code is 1 (for host unreachable). You set the timeout value at 4 minutes.



**NOTE:** For more information about ICMP types and codes, refer to RFC 792, *Internet Control Message Protocol*.

## Configuration

---

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see [Using the CLI Editor in Configuration Mode](#).

To define a custom ICMP application:

1. Set the application type and code.

```
[edit applications application host-unreachable]
user@host# set icmp-type 5 icmp-code 0
```

2. Set the inactivity timeout value.

```
[edit applications application host-unreachable]
user@host# set inactivity-timeout 4
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

## Verification

---

To verify the configuration is working properly, enter the **show applications** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Security Policies Overview on page 145](#)

## Policy Application Timeouts

---

- [Understanding Policy Application Timeout Configuration and Lookup on page 194](#)
- [Understanding Policy Application Timeouts Contingencies on page 195](#)
- [Example: Setting a Policy Application Timeout on page 197](#)

### Understanding Policy Application Timeout Configuration and Lookup

The application timeout value you set for an application determines the session timeout. You can set the timeout threshold for a predefined or custom application; you can use the application default timeout, specify a custom timeout, or use no timeout at all. Application timeout behavior is the same in virtual systems (vsys) security domains as at the root level.

Application timeout values are stored in the application entry database and in the corresponding vsys TCP and UDP port-based timeout tables. When you set an application timeout value, Junos OS updates these tables with the new value. There are also default timeout values in the applications entry database, which are taken from predefined applications. You can set a timeout, but you cannot alter a default value.



Each custom application can be configured with its own custom application timeout. If multiple custom applications are configured with custom timeouts, then each application will have its own custom application timeout.

For single application entries, an application timeout lookup proceeds as follows:

1. The specified timeout in the application entry database, if set.
2. The default timeout in the application entry database, if specified in the predefined application.
3. The protocol-based default timeout table. See Table 17 on page 195.

**Table 17: Protocol-Based Default Timeout**

| Protocol | Default Timeout (seconds) |
|----------|---------------------------|
| TCP      | 1800                      |
| UDP      | 60                        |
| ICMP     | 60                        |
| OSPF     | 60                        |
| Other    | 1800                      |

For application groups, including hidden groups created in multicell policy configurations, and for the predefined application **ANY** (if timeout is not set), application timeout lookup proceeds as follows:

1. The vsys TCP and UDP port-based timeout table, if a timeout is set.
2. The protocol-based default timeout table.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policy Applications Overview on page 187
- Understanding Custom Policy Applications on page 190
- Understanding Policy Application Timeouts Contingencies on page 195
- Custom Application Mappings on page 190
- Example: Adding and Modifying Custom Policy Applications on page 191

## Understanding Policy Application Timeouts Contingencies

When setting timeouts, be aware of the following contingencies:

- If an application contains several application rule entries, all rule entries share the same timeout. The timeout table is updated for each rule entry that matches the protocol (for UDP and TCP—other protocols use the default). You need to define the application

timeout only once. For example, if you create an application with two rules, the following commands will set the timeout to 20 seconds for both rules:

```
user@host# set applications application test protocol tcp destination-port 1035-1035
inactivity-timeout 20
```

```
user@host# set applications application test term test protocol udp
```

```
user@host# set applications application test term test source-port 1-65535
```

```
user@host# set applications application test term test destination-port 1111-1111
```

- If multiple custom applications are configured with custom timeouts, then each application will have its own custom application timeout. For example:

```
user@host# set applications application ftp-1 protocol tcp source-port 0-65535
destination-port 2121-2121 inactivity-timeout 10
```

```
user@host# set applications application telnet-1 protocol tcp source-port 0-65535
designating-port 2100-2148 inactivity-timeout 20
```

With this configuration, Junos OS applies a 10-second timeout for destination port 2121 and a 20-second timeout for destination port 2100 in an application group.

- If you unset an application timeout, the default protocol-based timeout in the application entry database is used, and the timeout values in both the application entry and port-based timeout tables are updated with the default value.

If the modified application has overlapping destination ports with other applications, the default protocol-based timeout might not be the desired value. In that case, reboot Junos OS, or set the application timeout again for the desired timeout to take effect.

- When you modify a predefined application and reboot, the modified application might not be the last one in the configuration. This is because predefined applications are loaded before custom applications, and any change made to a custom application, even if made earlier, will show as later than the predefined application change when you reboot.

For example, suppose you create the following application:

```
user@host# set applications application my-application protocol tcp destination-port
179-179 inactivity-timeout 20
```

Later you modify the timeout of the predefined application BGP as follows:

```
user@host# set applications application bgp inactivity-timeout 75
```

The BGP application will use the 75-second timeout value, because it is now written to the application entry database. But the timeout for port 179, the port BGP uses, is also changed to 75 in the TCP port-based timeout table. After you reboot, the BGP application will continue to use the 75-second timeout that, as a single application, it gets from the application entry database. But the timeout in the TCP port-based table for port 179 will now be 60. You can verify this by entering the **show applications application bgp** command.

The BGP application has no effect on single applications. But if you add BGP or my\_application to an application group, the 60-second timeout value will be used for

destination port 179. This is because application group timeout is taken from the port-based timeout table, if one is set.

To ensure predictability when you modify a predefined application timeout, therefore, you can create a similar application, for example:

```
user@host# set applications application my-bgp protocol tcp destination-port 179-179
inactivity-timeout 75
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Security Policy Applications Overview on page 187](#)
- [Understanding Custom Policy Applications on page 190](#)
- [Custom Application Mappings on page 190](#)
- [Understanding Policy Application Timeout Configuration and Lookup on page 194](#)
- [Example: Adding and Modifying Custom Policy Applications on page 191](#)

### Example: Setting a Policy Application Timeout

This example shows how to set a policy application timeout value.

- [Requirements on page 197](#)
- [Overview on page 197](#)
- [Configuration on page 197](#)
- [Verification on page 198](#)

#### Requirements

Before you begin, understand policy application timeouts. See “Understanding Policy Application Timeout Configuration and Lookup” on page 194.

#### Overview

Application timeout values are stored in the application entry database and in the corresponding vsys TCP and UDP port-based timeout tables. In this example, you set the device for a policy application timeout to 75 minutes for the FTP predefined application.

When you set an application timeout value, Junos OS updates these tables with the new value.

#### Configuration

#### Step-by-Step Procedure

To set a policy application timeout:

1. Set the inactivity timeout value.

```
[edit applications application ftp]
user@host# set inactivity-timeout 75
```

2. Commit the configuration if you are done configuring the device.

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show applications** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policy Applications Overview on page 187

## Understanding the ICMP Predefined Policy Application

When you create a policy, you can specify the ICMP predefined application for the policy.

Internet Control Message Protocol (ICMP) is a part of IP and provides a way to query a network (ICMP query messages) and to receive feedback from the network for error patterns (ICMP error messages). ICMP does not, however, guarantee error message delivery or report all lost datagrams; and it is not a reliable protocol. ICMP codes and type codes describe ICMP query messages and ICMP error messages.

You can choose to permit or deny any or specific types of ICMP messages to improve network security. Some types of ICMP messages can be exploited to gain information about your network that might compromise security. For example, ICMP, TCP, or UDP packets can be constructed to return ICMP error messages that contain information about a network, such as its topology, and access list filtering characteristics. Table 18 on page 198 lists ICMP message names, the corresponding code, type, and description.

**Table 18: ICMP Messages**

| ICMP Message Name                                                                              | Type     | Code   | Description                                                                                                                                                                                                                                                                                                                                 |
|------------------------------------------------------------------------------------------------|----------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ICMP-ANY                                                                                       | all      | all    | <p>ICMP-ANY affects any protocol using ICMP.</p> <p>Denying ICMP-ANY impairs any attempt to ping or monitor a network using ICMP.</p> <p>Permitting ICMP-ANY allows all ICMP messages.</p>                                                                                                                                                  |
| ICMP-ADDRESS-MASK <ul style="list-style-type: none"> <li>• Request</li> <li>• Reply</li> </ul> | 17<br>18 | 0<br>0 | <p>ICMP address mask query is used for systems that need the local subnet mask from a bootstrap server.</p> <p>Denying ICMP address mask request messages can adversely affect diskless systems.</p> <p>Permitting ICMP address mask request messages might allow others to fingerprint the operating system of a host in your network.</p> |

Table 18: ICMP Messages (*continued*)

| ICMP Message Name       | Type | Code | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------|------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ICMP-DEST-UNREACH       | 3    | 0    | <p>ICMP destination unreachable error message indicates that the destination host is configured to reject the packets.</p> <p>Codes 0, 1, 4, or 5 can be from a gateway. Codes 2 or 3 can be from a host (RFC 792).</p> <p>Denying ICMP destination unreachable error messages can remove the assumption that a host is up and running behind a J Series or an SRX Series device.</p> <p>Permitting ICMP destination unreachable error messages can allow some assumptions, such as security filtering, to be made about the network.</p> |
| ICMP Fragment Needed    | 3    | 4    | <p>ICMP fragmentation error message indicates that fragmentation is needed but the don't fragment flag is set.</p> <p>We recommend denying these messages from the Internet to an internal network.</p>                                                                                                                                                                                                                                                                                                                                   |
| ICMP FragmentReassembly | 11   | 1    | <p>ICMP fragment reassembly time exceeded error indicates that a host reassembling a fragmented message ran out of time and dropped the packet. This message is sometimes sent.</p> <p>We recommend denying these messages from the Internet (external) to the trusted (internal) network.</p>                                                                                                                                                                                                                                            |
| ICMP-HOST-UNREACH       | 3    | 1    | <p>ICMP host unreachable error messages indicate that routing table entries do not list or list as infinity a particular host. Sometimes this error is sent by gateways that cannot fragment when a packet requiring fragmentation is received.</p> <p>We recommend denying these messages from the Internet to a trusted network.</p> <p>Permitting these messages allows others to be able to determine your internal hosts IP addresses by a process of elimination or make assumptions about gateways and fragmentation.</p>          |
| ICMP-INFO               | 15   | 0    | ICMP-INFO query messages allow diskless host systems to query the network and self-configure.                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| • Request               | 16   | 0    | <p>Denying ICMP address mask request messages can adversely affect diskless systems.</p> <p>Permitting ICMP address mask request messages might allow others to broadcast information queries to a network segment to determine computer type.</p>                                                                                                                                                                                                                                                                                        |
| • Reply                 |      |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

Table 18: ICMP Messages (*continued*)

| ICMP Message Name      | Type | Code | Description                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------|------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ICMP-PARAMETER-PROBLEM | 12   | 0    | <p>ICMP parameter problem error messages notify you when incorrect header parameters are present and have caused a packet to be discarded.</p> <p>We recommend denying these messages from the Internet to a trusted network.</p> <p>Permitting ICMP parameter problem error messages allows others to make assumptions about your network.</p>                                             |
| ICMP-PORT-UNREACH      | 3    | 3    | <p>ICMP port unreachable error messages indicate that gateways processing datagrams requesting certain ports are unavailable or unsupported in the network.</p> <p>We recommend denying these messages from the Internet to a trusted network.</p> <p>Permitting ICMP port unreachable error messages can allow others to determine which ports you use for certain protocols.</p>          |
| ICMP-PROTOCOL-UNREACH  | 3    | 2    | <p>ICMP protocol unreachable error messages indicate that gateways processing datagrams requesting certain protocols are unavailable or unsupported in the network.</p> <p>We recommend denying these messages from the Internet to a trusted network.</p> <p>Permitting ICMP protocol unreachable error messages can allow others to determine what protocols your network is running.</p> |
| ICMP-REDIRECT          | 5    | 0    | <p>ICMP redirect network error messages are sent by a J Series or an SRX Series device.</p> <p>We recommend denying these messages from the Internet to a trusted network.</p>                                                                                                                                                                                                              |
| ICMP-REDIRECT-HOST     | 5    | 1    | <p>ICMP redirect messages indicate datagrams destined for the specified host to be sent along another path.</p>                                                                                                                                                                                                                                                                             |
| ICMP-REDIRECT-TOS-HOST | 5    | 3    | <p>ICMP redirect type of service (TOS) and host error is a type of message.</p>                                                                                                                                                                                                                                                                                                             |
| ICMP-REDIRECT-TOS-NET  | 5    | 2    | <p>ICMP redirect TOS and network error is a type of message.</p>                                                                                                                                                                                                                                                                                                                            |

Table 18: ICMP Messages (*continued*)

| ICMP Message Name                                                              | Type | Code | Description                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------------------------------------------------|------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ICMP-SOURCE-QUENCH                                                             | 4    | 0    | <p>ICMP source quench error message indicates that a device does not have the buffer space available to accept, queue, and send the packets on to the next hop.</p> <p>Denying these messages will not help or impair internal network performance.</p> <p>Permitting these messages can allow others to know that a device is congested, making it a viable attack target.</p> |
| ICMP-SOURCE-ROUTE-FAIL                                                         | 3    | 5    | <p>ICMP source route failed error message</p> <p>We recommend denying these messages from the Internet (external).</p>                                                                                                                                                                                                                                                          |
| ICMP-TIME-EXCEEDED                                                             | 11   | 0    | <p>ICMP time-to-live (TTL) exceeded error message indicates that a packet's TTL setting reached zero before the packet reached its destination. This ensures that older packets are discarded before resent ones are processed.</p> <p>We recommend denying these messages from a trusted network out to the Internet.</p>                                                      |
| ICMP-TIMESTAMP                                                                 | 13   | 0    | <p>ICMP-TIMESTAMP query messages provide the mechanism to synchronize time and coordinate time distribution in a large, diverse network.</p>                                                                                                                                                                                                                                    |
| <ul style="list-style-type: none"> <li>• Request</li> <li>• Reply</li> </ul>   | 14   | 0    |                                                                                                                                                                                                                                                                                                                                                                                 |
| Ping (ICMP ECHO)                                                               | 8    | 0    | <p>Ping is a utility to determine whether a specific host is accessible by its IP address.</p> <p>Denying ping functionality removes your ability to check to see if a host is active.</p> <p>Permitting ping can allow others to execute a denial-of-service (DoS) or Smurf attack.</p>                                                                                        |
| ICMP-ECHO-FRAGMENT-ASSEMBLY-EXPIRE                                             | 11   | 1    | <p>ICMP fragment echo reassembly time expired error message indicates that the reassembly time was exceeded.</p> <p>We recommend denying these messages.</p>                                                                                                                                                                                                                    |
| Traceroute                                                                     | 30   | 0    | <p>Traceroute is a utility to indicate the path to access a specific host.</p> <p>We recommend denying this utility from the Internet (external) to your trusted network (internal).</p>                                                                                                                                                                                        |
| <ul style="list-style-type: none"> <li>• Forward</li> <li>• Discard</li> </ul> | 30   | 1    |                                                                                                                                                                                                                                                                                                                                                                                 |

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Security Policy Applications Overview on page 187
  - Default Behaviour of ICMP Unreachable Errors on page 202
  - Example: Configuring Applications and Application Sets on page 189

## Default Behaviour of ICMP Unreachable Errors

For different levels of security, the default behavior for ICMP unreachable errors from downstream Juniper Networks device is handled as follows:

- Sessions do not close for ICMP type-3 code-4 messages.  
ICMP messages pass through without dropping sessions. Packets are, however, dropped per session.
- Sessions do not close on receiving any kind of ICMP unreachable messages.
- Sessions store ICMP unreachable message, thereby restricting the number of messages flowing through to 1.  
One ICMP unreachable message is generated globally per router. The remaining ICMP unreachable errors are dropped.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Security Policy Applications Overview on page 187
  - Understanding the ICMP Predefined Policy Application on page 198
  - Example: Configuring Applications and Application Sets on page 189

## Understanding Internet-Related Predefined Policy Applications

When you create a policy, you can specify predefined Internet-related applications for the policy.

Table 19 on page 202 lists Internet-related predefined applications. Depending on your network requirements, you can choose to permit or deny any or all of these applications. Each entry lists the application name, default receiving port, and application description.

**Table 19: Predefined Applications**

| Application Name | Port(s)      | Application Description                                                                                     |
|------------------|--------------|-------------------------------------------------------------------------------------------------------------|
| AOL              | 5190-5193    | America Online Internet service provider (ISP) provides Internet, chat, and instant messaging applications. |
| DHCP relay       | 67 (default) | Dynamic Host Configuration Protocol.                                                                        |



Table 19: Predefined Applications (*continued*)

| Application Name         | Port(s)                | Application Description                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------------------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DHCP                     | 68 client<br>67 server | Dynamic Host Configuration Protocol allocates network addresses and delivers configuration parameters from server to hosts.                                                                                                                                                                                                                                                                                                                                            |
| DNS                      | 53                     | Domain Name System translates domain names into IP addresses.                                                                                                                                                                                                                                                                                                                                                                                                          |
| FTP                      | 20 data<br>21 control  | File Transfer Protocol (FTP) allows the sending and receiving of files between machines. You can choose to deny or permit ANY (GET or PUT) or to selectively permit or deny either GET or PUT. GET receives files from another machine and PUT sends files to another machine.<br><br>We recommend denying FTP applications from untrusted sources (Internet).                                                                                                         |
| Gopher                   | 70                     | Gopher organizes and displays Internet servers' contents as a hierarchically structured list of files.<br><br>We recommend denying Gopher access to avoid exposing your network structure.                                                                                                                                                                                                                                                                             |
| HTTP                     | 8080                   | HyperText Transfer Protocol is the underlying protocol used by the World Wide Web (WWW).<br><br>Denying HTTP application disables your users from viewing the Internet.<br><br>Permitting HTTP application allows your trusted hosts to view the Internet.                                                                                                                                                                                                             |
| HTTP-EXT                 | —                      | Hypertext Transfer Protocol with extended nonstandard ports                                                                                                                                                                                                                                                                                                                                                                                                            |
| HTTPS                    | 443                    | Hypertext Transfer Protocol with Secure Sockets Layer (SSL) is a protocol for transmitting private documents through the Internet.<br><br>Denying HTTPS disables your users from shopping on the Internet and from accessing certain online resources that require secure password exchange.<br><br>Permitting HTTPS allows your trusted hosts to participate in password exchange, shop online, and visit various protected online resources that require user login. |
| Internet Locator Service | —                      | Internet Locator Service includes LDAP, User Locator Service, and LDAP over TSL/SSL.                                                                                                                                                                                                                                                                                                                                                                                   |
| IRC                      | 6665-6669              | Internet Relay Chat (IRC) allows people connected to the Internet to join live discussions.                                                                                                                                                                                                                                                                                                                                                                            |
| LDAP                     | 389                    | Lightweight Directory Access Protocol is a set of protocols used to access information directories.                                                                                                                                                                                                                                                                                                                                                                    |
| PC-Anywhere              | —                      | PC-Anywhere is a remote control and file transfer software.                                                                                                                                                                                                                                                                                                                                                                                                            |
| TFTP                     | 69                     | Trivial File transfer Protocol (TFTP) is a protocol for simple file transfer.                                                                                                                                                                                                                                                                                                                                                                                          |

Table 19: Predefined Applications (*continued*)

| Application Name | Port(s) | Application Description                                                         |
|------------------|---------|---------------------------------------------------------------------------------|
| WAIS             | —       | Wide Area Information Server is a program that finds documents on the Internet. |

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policy Applications Overview on page 187
- Understanding Dynamic Routing Protocols Predefined Policy Applications on page 205
- Example: Configuring Applications and Application Sets on page 189

## Understanding Microsoft Predefined Policy Applications

When you create a policy, you can specify predefined Microsoft applications for the policy.

Table 20 on page 204 lists predefined Microsoft applications, parameters associated with each application, and a brief description of each application. Parameters include universal unique identifiers (UUIDs) and TCP/UDP source and destination ports. A UUID is a 128-bit unique number generated from a hardware address, a timestamp, and seed values.

Table 20: Predefined Microsoft Applications

| Application                       | Parameter/UUID                                                                                                       | Description                                                                                                                                                                                                                    |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Junos MS-RPC-EPM                  | 135<br>e1af8308-5d1f-11c9-91a4-08002b14a0fa                                                                          | Microsoft remote procedure call (RPC) Endpoint Mapper (EPM) Protocol.                                                                                                                                                          |
| Junos MS-RPC                      | —                                                                                                                    | Any Microsoft remote procedure call (RPC) applications.                                                                                                                                                                        |
| Junos MS-RPC-MSEXCHANGE           | 3 members                                                                                                            | Microsoft Exchange application group includes: <ul style="list-style-type: none"> <li>• Junos-MS-RPC-MSEXCHANGE-DATABASE</li> <li>• Junos-MS-RPC-MSEXCHANGE-DIRECTORY</li> <li>• Junos-MS-RPC-MSEXCHANGE-INFO-STORE</li> </ul> |
| Junos-MS-RPC-MSEXCHANGE-DATABASE  | 1a190310-bb9c-11cd-90f8-00aa00466520                                                                                 | Microsoft Exchange Database application.                                                                                                                                                                                       |
| Junos-MS-RPC-MSEXCHANGE-DIRECTORY | f5cc5a18-4264-101a-8c59-08002b2f8426<br>f5cc5a7c-4264-101a-8c59-08002b2f8426<br>f5cc59b4-4264-101a-8c59-08002b2f8426 | Microsoft Exchange Directory application.                                                                                                                                                                                      |

Table 20: Predefined Microsoft Applications (*continued*)

| Application                        | Parameter/UUID                                                                                                                                               | Description                                                |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| Junos-MS-RPC-MSEXCHANGE-INFO-STORE | 0e4a0156-dd5d-11d2-8c2f-00c04fb6bcde<br>1453c42c-0fa6-11d2-a910-00c04f990f3b<br>10f24e8e-0fa6-11d2-a910-00c04f990f3b<br>1544f5e0-613c-11d1-93df-00c04fd7bd09 | Microsoft Exchange Information Store application.          |
| Junos-MS-RPC-TCP                   | —                                                                                                                                                            | Microsoft Transmission Control Protocol (TCP) application. |
| Junos-MS-RPC-UDP                   | —                                                                                                                                                            | Microsoft User Datagram Protocol (UDP) application.        |
| Junos-MS-SQL                       | —                                                                                                                                                            | Microsoft Structured Query Language (SQL).                 |
| Junos-MSN                          | —                                                                                                                                                            | Microsoft Network Messenger application.                   |

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Security Policy Applications Overview on page 187
  - Example: Configuring Applications and Application Sets on page 189

## Understanding Dynamic Routing Protocols Predefined Policy Applications

When you create a policy, you can specify predefined dynamic routing protocol applications for the policy.

Depending on your network requirements, you can choose to permit or deny messages generated from these dynamic routing protocols and packets of these dynamic routing protocols. Table 21 on page 205 lists each supported dynamic routing protocol by name, port, and description.

Table 21: Dynamic Routing Protocols

| Dynamic Routing Protocol | Port | Description                                       |
|--------------------------|------|---------------------------------------------------|
| RIP                      | 520  | RIP is a common distance-vector routing protocol. |
| OSPF                     | 89   | OSPF is a common link-state routing protocol.     |
| BGP                      | 179  | BGP is an exterior/interdomain routing protocol.  |

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Security Policy Applications Overview on page 187

- Example: Configuring Applications and Application Sets on page 189

## Understanding Streaming Video Predefined Policy Applications

When you create a policy, you can specify predefined streaming video applications for the policy.

Table 22 on page 206 lists each supported streaming video application by name and includes the default port and description. Depending on your network requirements, you can choose to permit or deny any or all of these applications.

**Table 22: Supported Streaming Video Applications**

| Application | Port                                                                                                  | Description                                                                                                                                                  |
|-------------|-------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| H.323       | TCP source 1-65535; TCP destination 1720, 1503, 389, 522, 1731<br>UDP source 1-65535; UDP source 1719 | H.323 is a standard approved by the International Telecommunication Union (ITU) that defines how audiovisual conference data is transmitted across networks. |
| NetMeeting  | TCP source 1-65535; TCP destination 1720, 1503, 389, 522<br>UDP source 1719                           | Microsoft NetMeeting uses TCP to provide teleconferencing (video and audio) applications over the Internet.                                                  |
| Real media  | TCP source 1-65535; TCP destination 7070                                                              | Real Media is streaming video and audio technology.                                                                                                          |
| RTSP        | 554                                                                                                   | Real-Time Streaming Protocol (RTSP) is for streaming media applications                                                                                      |
| SIP         | 5056                                                                                                  | Session Initiation Protocol (SIP) is an Application-Layer control protocol for creating, modifying, and terminating sessions.                                |
| VDO Live    | TCP source 1-65535; TCP destination 7000-7010                                                         | VDOLive is a scalable, video streaming technology.                                                                                                           |

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policy Applications Overview on page 187
- Example: Configuring Applications and Application Sets on page 189

## Understanding Sun RPC Predefined Policy Applications

When you create a policy, you can specify predefined Sun RPC applications for the policy.

Table 23 on page 207 lists each Sun remote procedure call Application Layer Gateway (RPC ALG) application name, parameters, and full name.

Table 23: RPC ALG Applications

| Application              | Program Numbers  | Full Name                            |
|--------------------------|------------------|--------------------------------------|
| SUN-RPC-PORTMAPPER       | 111100000        | Sun RPC Portmapper protocol          |
| SUN-RPC-ANY              | ANY              | Any Sun RPC applications             |
| SUN-RPC-PROGRAM-MOUNTD   | 100005           | Sun RPC Mount Daemon                 |
| SUN-RPC-PROGRAM-NFS      | 100003<br>100227 | Sun RPC Network File System          |
| SUN-RPC-PROGRAM-NLOCKMGR | 100021           | Sun RPC Network Lock Manager         |
| SUN-RPC-PROGRAM-RQUOTAD  | 100011           | Sun RPC Remote Quota Daemon          |
| SUN-RPC-PROGRAM-RSTATD   | 100001           | Sun RPC Remote Status Daemon         |
| SUN-RPC-PROGRAM-RUSERD   | 100002           | Sun RPC Remote User Daemon           |
| SUN-RPC-PROGRAM-SADMIND  | 100232           | Sun RPC System Administration Daemon |
| SUN-RPC-PROGRAM-SPRAYD   | 100012           | Sun RPC Spray Daemon                 |
| SUN-RPC-PROGRAM-STATUS   | 100024           | Sun RPC Status                       |
| SUN-RPC-PROGRAM-WALLD    | 100008           | Sun RPC Wall Daemon                  |
| SUN-RPC-PROGRAM-YPBIND   | 100007           | SUN RPC Yellow Page Bind application |

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Security Policy Applications Overview on page 187
  - Example: Configuring Applications and Application Sets on page 189

## Understanding Security and Tunnel Predefined Policy Applications

When you create a policy, you can specify predefined security and tunnel applications for the policy.

Table 24 on page 208 lists each supported application and gives the default port(s) and a description of each entry.

Table 24: Supported Applications

| Application | Port                                                                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------|--------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IKE         | UDP source 1-65535; UDP destination 500<br>4500 (used for NAT traversal) | Internet Key protocol (IKE) is a protocol to obtain authenticated keying material for use with ISAKMP.<br><br>When configuring auto IKE, you can choose from three predefined Phase 1 or Phase 2 proposals: <ul style="list-style-type: none"> <li>• Standard: AES and 3DES</li> <li>• Basic: DES and two different types of authentication algorithms</li> <li>• Compatible: Four commonly used authentication and encryption algorithms</li> </ul> |
| L2TP        | 1723                                                                     | L2TP combines PPTP with Layer 2 Forwarding (L2F) for remote access.                                                                                                                                                                                                                                                                                                                                                                                  |
| PPTP        | —                                                                        | Point-to-Point Tunneling Protocol allows corporations to extend their own private network through private <i>tunnels</i> over the public Internet.                                                                                                                                                                                                                                                                                                   |

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policy Applications Overview on page 187
- Example: Configuring Applications and Application Sets on page 189

## Understanding IP-Related Predefined Policy Applications

When you create a policy, you can specify predefined IP-related applications for the policy.

Table 25 on page 208 lists the predefined IP-related applications. Each entry includes the default port and a description of the application.

Table 25: Predefined IP-Related Applications

| Application | Port    | Description                              |
|-------------|---------|------------------------------------------|
| Any         | —       | Any application                          |
| TCP-ANY     | 1-65535 | Any protocol using the TCP TCPMUX port 1 |
| UDP-ANY     | 137     | Any protocol using the UDP               |

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policy Applications Overview on page 187
- Example: Configuring Applications and Application Sets on page 189

## Understanding Instant Messaging Predefined Policy Applications

When you create a policy, you can specify predefined instant messaging applications for the policy.

Table 26 on page 209 lists predefined Internet-messaging applications. Each entry includes the name of the application, the default or assigned port, and a description of the application.

**Table 26: Predefined Internet-Messaging Applications**

| Application | Port           | Description                                                                                                                                   |
|-------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Gnutella    | 6346 (default) | Gnutella is a public domain file sharing protocol that operates over a distributed network. You can assign any port, but the default is 6346. |
| MSN         | 1863           | Microsoft Network Messenger is a utility that allows you to send instant messages and talk online.                                            |
| NNTP        | 119            | Network News Transport Protocol is a protocol used to post, distribute, and retrieve USENET messages.                                         |
| SMB         | 445            | Server Message Block (SMB) over IP is a protocol that allows you to read and write files to a server on a network.                            |
| YMSG        | 5010           | Yahoo! Messenger is a utility that allows you to check when others are online, send instant messages, and talk online.                        |

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Security Policy Applications Overview on page 187](#)
- [Understanding Management Predefined Policy Applications on page 209](#)
- [Example: Configuring Applications and Application Sets on page 189](#)

## Understanding Management Predefined Policy Applications

When you create a policy, you can specify predefined management applications for the policy.

Table 27 on page 209 lists the predefined management applications. Each entry includes the name of the application, the default or assigned port, and a description of the application.

**Table 27: Predefined Management Applications**

| Application | Port | Description                                                                      |
|-------------|------|----------------------------------------------------------------------------------|
| NBNAME      | 137  | NetBIOS Name application displays all NetBIOS name packets sent on UDP port 137. |

Table 27: Predefined Management Applications (*continued*)

| Application   | Port                    | Description                                                                                                                                                                                                                                                  |
|---------------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NDBDS         | 138                     | NetBIOS Datagram application, published by IBM, provides connectionless (datagram) applications to PCs connected with a broadcast medium to locate resources, initiate sessions, and terminate sessions. It is unreliable and the packets are not sequenced. |
| NFS           | —                       | Network File System uses UDP to allow network users to access shared files stored on computers of different types. SUN RPC is a building block of NFS.                                                                                                       |
| NS Global     | —                       | NS-Global is the central management protocol for Juniper Networks Firewall/VPN devices.                                                                                                                                                                      |
| NS Global PRO | —                       | NS Global-PRO is the scalable monitoring system for the Juniper Networks Firewall/VPN device family.                                                                                                                                                         |
| NSM           | —                       | Network and Security Manager                                                                                                                                                                                                                                 |
| NTP           | 123                     | Network Time Protocol provides a way for computers to synchronize to a time reference.                                                                                                                                                                       |
| RLOGIN        | 513                     | RLOGIN starts a terminal session on a remote host.                                                                                                                                                                                                           |
| RSH           | 514                     | RSH executes a shell command on a remote host.                                                                                                                                                                                                               |
| SNMP          | 161                     | Simple Network Management Protocol is a set of protocols for managing complex networks.                                                                                                                                                                      |
| SQL*Net V1    | 66                      | SQL*Net Version 1 is a database language that allows for the creation, access, modification, and protection of data.                                                                                                                                         |
| SQL*Net V2    | 66                      | SQL*Net Version 2 is a database language that allows for the creation, access, modification, and protection of data.                                                                                                                                         |
| MSSQL         | 1433 (default instance) | Microsoft SQL is a proprietary database server tool that allows for the creation, access, modification, and protection of data.                                                                                                                              |
| SSH           | 22                      | SSH is a program to log into another computer over a network through strong authentication and secure communications on an unsecure channel.                                                                                                                 |
| SYSLOG        | 514                     | Syslog is a UNIX program that sends messages to the system logger.                                                                                                                                                                                           |
| Talk          | 517-518                 | Talk is a visual communication program that copies lines from your terminal to that of another user.                                                                                                                                                         |
| Telnet        | 23                      | Telnet is a UNIX program that provides a standard method of interfacing terminal devices and terminal-oriented processes to each other.                                                                                                                      |
| WinFrame      | —                       | WinFrame is a technology that allows users on non-Windows machines to run Windows applications.                                                                                                                                                              |



Table 27: Predefined Management Applications (*continued*)

| Application | Port | Description                                                                          |
|-------------|------|--------------------------------------------------------------------------------------|
| X-Windows   | —    | X-Windows is the windowing and graphics system that Motif and OpenLook are based on. |

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policy Applications Overview on page 187
- Example: Configuring Applications and Application Sets on page 189

## Understanding Mail Predefined Policy Applications

When you create a policy, you can specify predefined mail applications for the policy.

Table 28 on page 211 lists the predefined mail applications. Each includes the name of the application, the default or assigned port number, and a description of the application.

Table 28: Predefined Mail Applications

| Application | Port | Description                                                             |
|-------------|------|-------------------------------------------------------------------------|
| IMAP        | 143  | Internet Message Access Protocol is used for retrieving messages.       |
| Mail (SMTP) | 25   | Simple Mail Transfer Protocol is used to send messages between servers. |
| POP3        | 110  | Post Office Protocol is used for retrieving e-mail.                     |

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policy Applications Overview on page 187
- Example: Configuring Applications and Application Sets on page 189

## Understanding UNIX Predefined Policy Applications

When you create a policy, you can specify predefined UNIX applications for the policy.

Table 29 on page 211 lists the predefined UNIX applications. Each entry includes the name of the application, the default or assigned port, and a description of the application.

Table 29: Predefined UNIX Applications

| Application | Port | Description                                                                                                                                     |
|-------------|------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| FINGER      | 79   | Finger is a UNIX program that provides information about the users.                                                                             |
| UUCP        | 117  | UNIX-to-UNIX Copy Protocol (UUCP) is a UNIX utility that enables file transfers between two computers over a direct serial or modem connection. |

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Security Policy Applications Overview on page 187
  - Example: Configuring Applications and Application Sets on page 189

## Understanding Miscellaneous Predefined Policy Applications

When you create a policy, you can specify miscellaneous predefined applications for the policy.

Table 30 on page 212 lists predefined miscellaneous applications. Each entry includes the application name, default or assigned port, and a description of the application.

**Table 30: Predefined Miscellaneous Applications**

| Application | Port                                               | Description                                                                                                                                      |
|-------------|----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| CHARGEN     | 19                                                 | Character Generator Protocol is a UDP- or TCP-based debugging and measurement tool.                                                              |
| DISCARD     | 9                                                  | Discard protocol is an Application Layer protocol that describes a process for discarding TCP or UDP data sent to port 9.                        |
| IDENT       | 113                                                | Identification protocol is a TCP/IP Application Layer protocol used for TCP client authentication.                                               |
| LPR         | 515 listen;<br>721-731 source range<br>(inclusive) | Line Printer Daemon protocol is a TCP-based protocol used for printing applications.                                                             |
| RADIUS      | 1812                                               | Remote Authentication Dial-In User Service application is a server program used for authentication and accounting purposes.                      |
| SQLMON      | 1434 (SQL Monitor Port)                            | SQL monitor (Microsoft)                                                                                                                          |
| VNC         | 5800                                               | Virtual Network Computing facilitates viewing and interacting with another computer or mobile Juniper Networks device connected to the Internet. |
| WHOIS       | 43                                                 | Network Directory Application Protocol is a way to look up domain names.                                                                         |
| IPsec-NAT   | —                                                  | IPSEC-NAT allows Network Address Translation for ISAKMP and ESP packets.                                                                         |
| SCCP        | 2000                                               | Cisco Station Call Control Protocol (SCCP) uses the signaling connection control port to provide high availability and flow control.             |
| VoIP        | —                                                  | Voice over IP application group provides voice applications over the Internet and includes H.323 and Session Initiation Protocol (SIP).          |

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policy Applications Overview on page 187
- Example: Configuring Applications and Application Sets on page 189



## PART 4

# Application Layer Gateways

- ALGs on page 217
- H.323 ALGs on page 223
- ALG for IKE and ESP on page 259
- SIP ALGs on page 269
- SCCP ALGs on page 325
- MGCP ALGs on page 347
- RPC ALGs on page 379



## CHAPTER 9

# ALGs

- ALG Overview on page 217
- Understanding ALG Types on page 218
- Understanding VoIP DSCP Rewrite Rules on page 220
- Example: Configuring VoIP DSCP Rewrite Rules on page 220

### ALG Overview

---

An *Application Layer Gateway (ALG)* is a software component that is designed to manage specific protocols such as Session Initiation Protocol (SIP) or FTP on Juniper Networks devices running Junos OS. The ALG module is responsible for Application-Layer aware packet processing.

ALG functionality can be triggered either by a service or application configured in the security policy:

- A *service* is an object that identifies an application protocol using Layer 4 information (such as standard and accepted TCP and UDP port numbers) for an application service (such as Telnet, FTP, SMTP, and HTTP).
- An *application* specifies the Layer 7 application that maps to a Layer 4 service.

A predefined service already has a mapping to a Layer 7 application. However, for custom services, you must link the service to an application explicitly, especially if you want the policy to apply an ALG.

ALGs for packets destined to well-known ports are triggered by service type. The ALG intercepts and analyzes the specified traffic, allocates resources, and defines dynamic policies to permit the traffic to pass securely through the device:

1. When a packet arrives at the device, the flow module forwards the packet according to the security rule set in the policy.
2. If a policy is found to permit the packet, the associated service type or application type is assigned and a session is created for this type of traffic.
3. If a session is found for the packet, no policy rule match is needed. The ALG module is triggered if that particular service or application type requires the supported ALG processing.

The ALG also inspects the packet for embedded IP address and port information in the packet payload, and performs Network Address Translation (NAT) processing if necessary. The ALG also opens a gate for the IP address and port number to permit data exchange for the session. The control session and data session can be coupled to have the same timeout value, or they can be independent.

ALGs are supported on chassis clusters. For information about chassis clusters, see “Chassis Cluster Overview” on page 1137.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding ALG Types on page 218](#)
- [Understanding H.323 ALGs on page 223](#)
- [Understanding SIP ALGs on page 269](#)
- [Understanding SCCP ALGs on page 325](#)
- [Understanding MGCP ALGs on page 347](#)
- [Understanding RPC ALGs on page 379](#)

---

## Understanding ALG Types

---

Junos OS supports voice-over-IP Application Layer Gateways (VoIP ALGs) and basic data ALGs. (Note that supported ALG types vary depending on which hardware device you are using.)

*VoIP ALGs* provide stateful Application Layer inspection and Network Address Translation (NAT) capabilities to VoIP signaling and media traffic. The ALG inspects the state of transactions, or calls, and forwards or drops packets based on those states.

Junos OS supports the following VoIP ALGs:

- **H.323**—The H.323 ALG provides support for the H.323 legacy VoIP protocol. The ALG lets you secure VoIP communication between terminal hosts, such as IP phones and multimedia devices. In such a telephony system, the gatekeeper device manages call registration, admission, and call status for VoIP calls. Gatekeepers can reside in the two different zones or in the same zone.
- **SIP**—The SIP ALG provides support for the Session Initiation Protocol (SIP). SIP is an Internet Engineering Task Force (IETF)–standard protocol for initiating, modifying, and terminating multimedia sessions over the Internet. Such sessions might include conferencing, telephony, or multimedia, with features such as instant messaging and application-level mobility in network environments.
- **SCCP**—The SCCP ALG provides support for Skinny Client Control Protocol (SCCP). SCCP is a Cisco proprietary protocol for call signaling. Skinny is based on a call-agent-based call-control architecture. The control protocol uses binary-coded



frames encoded on TCP frames sent to well-known TCP port number destinations to set up and tear down RTP media sessions.

- MGCP—The MGCP ALG provides support for Media Gateway Control Protocol (MGCP). MGCP is a text-based Application Layer protocol used for call setup and call control between the media gateway and the media gateway controller (MGC).

Junos OS also supports the following data ALGs:

- DNS—Provides an ALG for the Domain Name System. The DNS ALG monitors DNS query and reply packets and closes session if the DNS flag indicates the packet is a reply message.
- FTP—Provides an ALG for the File Transfer Protocol (FTP).The FTP ALG monitors PORT, PASV, and 227 commands. It performs NAT on the IP, port, or both in the message and gate opening on the device as necessary.
- TFTP—Provides an ALG for the Trivial File Transfer Protocol (TFTP). The TFTP ALG processes TFTP packets that initiate the request and opens a gate to allow return packets from the reverse direction to the port that sends the request.
- PPTP—Provides an ALG for the Point-to-Point Tunneling Protocol (PPTP). The PPTP is a Layer 2 protocol that tunnels PPP data across TCP/IP networks. The PPTP client is freely available on Windows systems and is widely deployed for building Virtual Private Networks (VPNs).
- REAL—Provides an ALG for the Real-Time Streaming Protocol.
- MSRPC—Provides an ALG for the Microsoft Remote Procedure Call.
- SUNRPC—Provides an ALG for the SUN Remote Procedure Call.
- RSH—Provides an ALG for the Remote Shell (RSH). The RSH ALG handles TCP packets destined for port 514 and processes the RSH port command. The RSH ALG performs NAT on the port in the port command and opens gates as necessary.
- SQL—Provides an ALG for the Structured Query Language (SQL). The SQLNET ALG processes SQL TNS response frame from the server side. It parses the packet and looks for the (HOST=ipaddress), (PORT=port) pattern and performs NAT and gate opening on the client side for the TCP data channel.
- TALK—Provides an ALG for the TALK Protocol. The TALK protocol uses UDP port 517 and port 518 for control channel connections. The talk program consists of a server and a client. The server handles client notifications and helps to establish talk sessions. There are two types of talk servers: ntalk and talkd. The TALK ALG processes packets of both ntalk and talkd formats. It also performs NAT and gate opening as necessary.

For information about enabling and configuring each of these ALGs through J-Web, select the **Configure>Security>ALG** page in the J-Web user interface and click **Help**.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- ALG Overview on page 217
- Understanding H.323 ALGs on page 223

- [Understanding SIP ALGs on page 269](#)
- [Understanding SCCP ALGs on page 325](#)
- [Understanding MGCP ALGs on page 347](#)
- [Understanding RPC ALGs on page 379](#)

## Understanding VoIP DSCP Rewrite Rules

---

This topic describes the voice over IP Application Layer Gateway (VoIP ALG) mechanism for modifying the Differentiated Services Code Point (DSCP) field of Real-Time Transport Protocol (RTP) packets. The VoIP ALG mechanism is applicable for the RTP session, which is recognized by the ALG.

DSCP is a modification of the type of service byte for class of service (CoS). Six bits of this byte are reallocated for use as the DSCP field, where each DSCP specifies a particular per-hop behavior that is applied to a packet.

To avoid VoIP quality degradation caused by network congestion, the RTP packets are required to mark the DSCP bit to ensure they get higher routing priority. A downstream router can put those packets in a higher priority queue for faster forwarding. To provide this functionality, there needs to be a per-VoIP mechanism for modifying the DSCP field of RTP packets according to the specific configuration. This will ensure that all RTP packets based on User Datagram Protocol/Transport Control Protocol (UDP/TCP) that encounter the ALG will be assigned a specific DSCP bit.

A rewrite rule modifies the appropriate CoS bits in an outgoing packet to meet the requirements of the targeted peer. Each rewrite rule reads the current CoS value that is configured at the VoIP ALG level. Every packet that hits the VoIP ALG is marked by this CoS value.

This feature supports ALG DSCP marking for H323, Session Initiation Protocol (SIP), Media Gateway Control Protocol (MGCP), and Skinny Client Control Protocol (SCCP). It provides a 6-bit DSCP value configuration for each of these. When the first RTP packet hits the ALG, this feature receives the 6-bit DSCP value from the configuration and sets it to the RTP session that the packet has created. This first RTP packet and the following RTP packets passing through the RTP session are marked according to the 6-bit DSCP value in the session.

### Related Documentation

- [Example: Configuring VoIP DSCP Rewrite Rules on page 220](#)

## Example: Configuring VoIP DSCP Rewrite Rules

---

This example shows how to configure VoIP DSCP.

- [Requirements on page 221](#)
- [Overview on page 221](#)

- Configuration on page 221
- Verification on page 221

## Requirements

This example uses an SRX210 device. The example assumes that the ALG has been enabled.

## Overview

This example shows how to configure four ALG DSCP markings; SIP, H323, MGCP, and SCCP. You set the 6-bit DSCP value configuration for each ALG DSCP.

## Configuration

### Step-by-Step Procedure

To configure VoIP DSCP rewrite rules:

1. Set the DSCP for each VoIP ALG.

```
[edit]
```

```
user@host# set security alg sip dscp-rewrite code-point 101010
user@host# set security alg h323 dscp-rewrite code-point 010101
user@host# set security alg mgcp dscp-rewrite code-point 111000
user@host# set security alg sccp dscp-rewrite code-point 000111
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
```

```
user@host# commit
```

## Verification

To verify that the configuration is working properly, enter the **show security alg** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding VoIP DSCP Rewrite Rules on page 220



## CHAPTER 10

# H.323 ALGs

- Understanding H.323 ALGs on page 223
- Understanding the Avaya H.323 ALG on page 225
- H.323 ALG Configuration Overview on page 227
- H.323 ALG Endpoint Registration Timeouts on page 227
- H.323 ALG Media Source Port Ranges on page 229
- H.323 ALG DoS Attack Protection on page 230
- H.323 ALG Unknown Message Types on page 232
- Example: Passing H.323 ALG Traffic to a Gatekeeper in the Private Zone on page 234
- Example: Passing H.323 ALG Traffic to a Gatekeeper in the External Zone on page 239
- Example: Using NAT with the H.323 ALG to Enable Incoming Calls on page 246
- Example: Using NAT with the H.323 ALG to Enable Outgoing Calls on page 252

## Understanding H.323 ALGs

---

The H.323 standard is a legacy voice-over-IP (VoIP) protocol defined by the International Telecommunication Union (ITU-T). H.323 consists of a suite of protocols (such as H.225.0 and H.245) that are used for call signaling and call control for VoIP.

H.323 uses the ASN.1 coding format. It sets up the dynamic links for data, video, and audio streams, following the protocols Q.931 (with port number 1720) and H.245. There are three major processes in H.323:

- Gatekeeper Discovery—An endpoint finds its gatekeeper through the gatekeeper discovery process, through broadcast or unicast (to a known IP and the well-known UDP port 1719). (Junos OS supports unicast only.)
- Endpoint Registration, Admission, and Status—An endpoint registers to a gatekeeper and asks for its management. Before making a call, an endpoint asks its gatekeeper for permission to place the call. In both registration and admission phases, the Registration, Admission, and Status (RAS) channel is used. The Transport Service Access Point (TSAP) can be either the well-known UDP port (1719) or a dynamically assigned port from the discovery or registration phase.
- Call Control and Call Setup—Calls can be established within a zone or across two zones, or even across multiple zones (multipoint conference). The call setup and tear

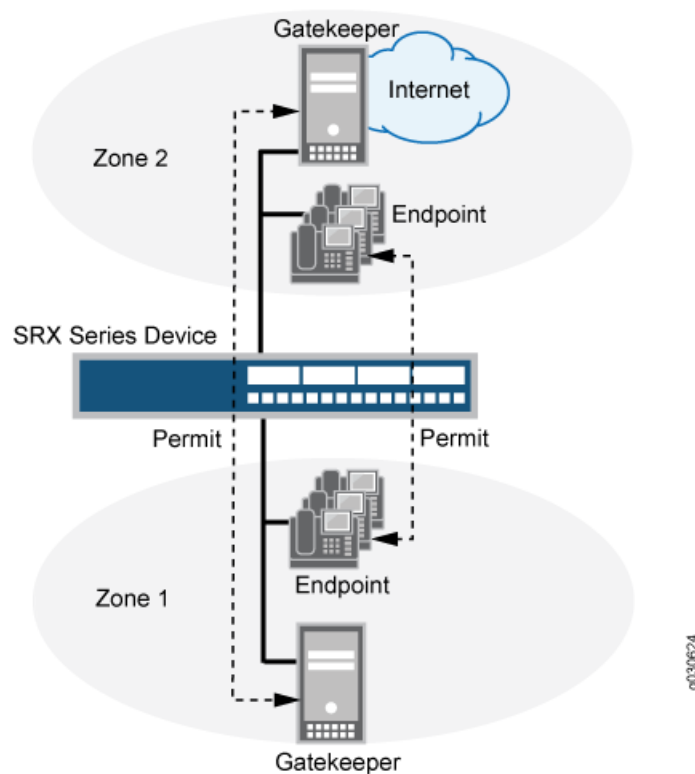
down is performed through the call signaling channel whose TSAP is the well-known TCP port (1720). The call control, including opening/closing media channels between two endpoints, is performed through the call control channel whose TSAP is dynamically assigned from the previous call signaling process. H.245 messages are used in the call control channel, and are encoded using ASN.1.



**NOTE:** Detailed information on H.323 can be found in ITU-T Recommendation H.323.

The H.323 Application Layer Gateway (ALG) lets you secure VoIP communication between terminal hosts, such as IP phones and multimedia devices. In such a telephony system, the gatekeeper device manages call registration, admission, and call status for VoIP calls. Gatekeepers can reside in the two different zones or in the same zone. (See Figure 13 on page 224.)

Figure 13: H.323 ALG for VoIP Calls



**NOTE:** The illustration uses IP phones for illustrative purposes, although it is possible to make configurations for other hosts that use VoIP, such as Microsoft NetMeeting multimedia devices.

Related  
Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

- ALG Overview on page 217
- Understanding the Avaya H.323 ALG on page 225
- H.323 ALG Configuration Overview on page 227

## Understanding the Avaya H.323 ALG

---

The H.323 standard is a legacy voice-over-IP (VoIP) protocol defined by the International Telecommunication Union (ITU-T). H.323 consists of a suite of protocols (such as H.225.0 and H.245) that are used for call signaling and call control for VoIP. The processes for configuring the H.323 standard Application Layer Gateway (ALG) and the proprietary Avaya H.323 ALG are the same.

However, Avaya H.323 ALG has some special features. To understand and configure the Avaya H.323-specific features listed here, see the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.

This topic contains the following sections:

- Avaya H.323 ALG-Specific Features on page 225
- Call Flow Details in the Avaya H.323 ALG on page 225

### Avaya H.323 ALG-Specific Features

Avaya H.323-specific features are as follows:

- H.323 Fast Connect
- H.323 asymmetric media
- Call waiting
- Call forwarding
- Voice mail
- Call identification
- Conference calling

### Call Flow Details in the Avaya H.323 ALG

- Connecting the Phone into the Network—Avaya performs the Q.931 Setup/Connect negotiation when the phone is wired into the network rather than when a call is being initiated.
- Making a call—When a call is made, because the PBX has already stored the capabilities for each phone when the phone is connected to the network, no further Q.931 and PBX negotiations are required to set up the call. It no longer exchanges Q.931 Setup and Connect messages with the PBX. The phone and the PBX exchange H.323 Facility messages to set up the call.

- Registering with a CM—When a call has been made, Avaya H.323 registers with the Avaya Communication Manager (CM). The registration process is similar to a generic H.323 standard registration process.



**NOTE:** The direct mode and tunnel mode are not defined by Avaya H.323 ALG.

For a call to work, the CM must be deployed with Avaya Endpoints. During the call, RAS and Q.931 messages are exchanged between the CM and the Avaya Endpoints.



**NOTE:** For Avaya H.323 with a source Network Address Translation (NAT) pool, the registration process allows only one IP address in the pool.

- Setting up Real-Time Transport Protocol (RTP)/Real-Time Control Protocol (RTCP) ports—The Q.931 Setup, Facility and Information messages are used to set up RTP/RTCP ports. The hierarchy for an Avaya H.323 session is Q.931, RTP/RTCP, Parent, and then Child.



**NOTE:** H.245 ports are not used in an Avaya call flow process.

- Using Avaya H.323 counters—The counters for calls and active calls are not applicable to the Avaya H.323 ALG. The call creation and tearing down is done by Facility messages afterward. When resources are allocated for a call, all counters for calls and active calls increment. If resources are allocated for a call multiple times, messages belonging to the same call that pass the firewall multiple times will trigger multiple increments of the counters. In other words, messages that belong to the same call and pass the firewall multiple times might trigger multiple increments of the counters if the resource for a call needs to be allocated multiple times.

For example, in the two-zone case, the setup and connect message pair allocates one call resource. The active call counter is increased once. Each time the setup and connect message pair passes the firewall, a different call resource with unique interfaces and NAT is allocated. Therefore, the counter increments twice in a three-zone scenario.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- ALG Overview on page 217
- Understanding H.323 ALGs on page 223
- H.323 ALG Configuration Overview on page 227



## H.323 ALG Configuration Overview

The H.323 Application Layer Gateway (ALG) is enabled by default on the device—no action is required to enable it. However, you might choose to fine-tune H.323 ALG operations by using the following instructions:

1. Specify how long an endpoint registration entry remains in the Network Address Translation (NAT) table. For instructions, see “Example: Setting H.323 ALG Endpoint Registration Timeouts” on page 228.
2. Enable media traffic on a narrow or wide range of ports. For instructions, see “Example: Setting H.323 ALG Media Source Port Ranges” on page 229.
3. Protect the H.323 gatekeeper from denial-of-service (DoS) flood attacks. For instructions, see “Example: Configuring H.323 ALG DoS Attack Protection” on page 231.
4. Enable unknown messages to pass when the session is in NAT mode and route mode. For instructions, see “Example: Allowing Unknown H.323 ALG Message Types” on page 233.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding H.323 ALGs on page 223](#)
- [Example: Passing H.323 ALG Traffic to a Gatekeeper in the Private Zone on page 234](#)
- [Example: Passing H.323 ALG Traffic to a Gatekeeper in the External Zone on page 239](#)
- [Example: Using NAT with the H.323 ALG to Enable Incoming Calls on page 246](#)
- [Example: Using NAT with the H.323 ALG to Enable Outgoing Calls on page 252](#)

## H.323 ALG Endpoint Registration Timeouts

- [Understanding H.323 ALG Endpoint Registration Timeouts on page 227](#)
- [Example: Setting H.323 ALG Endpoint Registration Timeouts on page 228](#)

### Understanding H.323 ALG Endpoint Registration Timeouts

In Network Address Translation (NAT) mode, when endpoints in the protected network behind the Juniper Networks device register with the H.323 gatekeeper, the device adds an entry to the NAT table containing a mapping of the public-to-private address for each endpoint. These entries make it possible for endpoints in the protected network to receive incoming calls.

You set an endpoint registration timeout to specify how long an endpoint registration entry remains in the NAT table. To ensure uninterrupted incoming call service, set the endpoint registration timeout to a value equal to or greater than the keepalive value the administrator configures on the gatekeeper. The range is 10 to 50,000 seconds, the default value is 3600 seconds.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding H.323 ALGs on page 223
  - H.323 ALG Configuration Overview on page 227
  - Example: Setting H.323 ALG Endpoint Registration Timeouts on page 228

## Example: Setting H.323 ALG Endpoint Registration Timeouts

This example shows how to specify the endpoint registration timeout.

- Requirements on page 228
- Overview on page 228
- Configuration on page 228
- Verification on page 229

### Requirements

---

Before you begin, understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.

### Overview

---

You set an endpoint registration timeout range to specify how long an endpoint registration entry remains in the NAT table. The range is 10 to 50,000 seconds, and the default value is 3600 seconds.

### Configuration

---

#### J-Web Quick Configuration

To specify the H.323 ALG endpoint registration timeout:

1. Select **Configure>Security>ALG**.
2. Select the **H323** tab.
3. In the Timeout for endpoints box, type **5000**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

#### Step-by-Step Procedure

To specify the H.323 ALG endpoint registration timeout:

1. Configure the H.323 ALG and set the endpoint registration timeout to 5000 seconds.  

```
[edit]  
user@host# set security alg h323 endpoint-registration-timeout 5000
```
2. If you are done configuring the device, commit the configuration.  

```
[edit]  
user@host# commit
```

---

### Verification

To verify the configuration is working properly, enter the **show security alg h323** command and the **show security alg h323 counters** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding H.323 ALG Endpoint Registration Timeouts on page 227](#)
- [H.323 ALG Configuration Overview on page 227](#)

---

## H.323 ALG Media Source Port Ranges

- [Understanding H.323 ALG Media Source Port Ranges on page 229](#)
- [Example: Setting H.323 ALG Media Source Port Ranges on page 229](#)

### Understanding H.323 ALG Media Source Port Ranges

The media source port feature enables you to configure the device to allow media traffic on a narrow or wide range of ports. By default, the device listens for H.323 traffic on a wide range of ports. If your endpoint equipment allows you to specify a sending port and a listening port, you might want to narrow the range of ports the device allows media traffic on. This enhances security by opening a smaller pinhole for H.323 traffic.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding H.323 ALGs on page 223](#)
- [H.323 ALG Configuration Overview on page 227](#)
- [Example: Setting H.323 ALG Media Source Port Ranges on page 229](#)

### Example: Setting H.323 ALG Media Source Port Ranges

This example shows how to enable the H.323 ALG media source port feature.

- [Requirements on page 229](#)
- [Overview on page 229](#)
- [Configuration on page 230](#)
- [Verification on page 230](#)

#### Requirements

Before you begin, understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.

#### Overview

The media source port feature enables you to configure the device to allow media traffic on a narrow or wide range of ports. By default, the device listens for H.323 traffic on a

narrow range of ports. This example shows how to configure the device to open a wide gate for media traffic by enabling the media source port feature.

### Configuration

---

#### J-Web Quick Configuration

To enable the H.323 ALG media source port feature:

1. Select **Configure>Security>ALG**.
2. Select the **H323** tab.
3. Select the **Enable Permit media from any source port** check box.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

#### Step-by-Step Procedure

To enable the H.323 ALG media source port feature:

1. Set a narrow gate for media traffic by disabling the media source port for the H.323 ALG.

[edit]

```
user@host# delete security alg h323 media-source-port-any
```

2. If you are done configuring the device, commit the configuration.

[edit]

```
user@host# commit
```

### Verification

---

To verify the configuration is working properly, enter the **show security alg h323** command and the **show security alg h323 counters** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding H.323 ALG Media Source Port Ranges on page 229](#)
- [H.323 ALG Configuration Overview on page 227](#)

## H.323 ALG DoS Attack Protection

---

- [Understanding H.323 ALG DoS Attack Protection on page 230](#)
- [Example: Configuring H.323 ALG DoS Attack Protection on page 231](#)

### Understanding H.323 ALG DoS Attack Protection

You can protect the H.323 gatekeeper from denial-of-service (DoS) flood attacks by limiting the number of Registration, Admission, and Status (RAS) messages per second it will attempt to process. Incoming RAS request messages exceeding the threshold you specify are dropped by H.323 Application Layer Gateway (ALG). The range is 2 to 50,000 messages per second, the default value is 1000.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding H.323 ALGs on page 223
  - H.323 ALG Configuration Overview on page 227
  - Example: Configuring H.323 ALG DoS Attack Protection on page 231

## Example: Configuring H.323 ALG DoS Attack Protection

This example shows how to configure the H.323 ALG DoS attack protection feature.

- Requirements on page 231
- Overview on page 231
- Configuration on page 231
- Verification on page 232

### Requirements

Before you begin, understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.

### Overview

You can protect the H.323 gatekeeper from DoS flood attacks by limiting the range of Registration, Admission, and Status (RAS) messages per second it will attempt to process. The range is 2 to 50,000 messages per second, and the default value is 1000. This example limits the number of incoming RAS request messages to 5000 messages per second.

### Configuration

#### J-Web Quick Configuration

To configure the H.323 ALG DoS attack protection feature:

1. Select **Configure>Security>ALG**.
2. Select the **H323** tab.
3. In the Message flood gatekeeper threshold box, type **5000**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

#### Step-by-Step Procedure

To configure the H.323 ALG DoS attack protection feature:

1. Configure the gatekeeper for the H.323 ALG and set the threshold.
 

```
[edit]
user@host# set security alg h323 application-screen message-flood gatekeeper
threshold 5000
```
2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security alg h323** command and the **show security alg h323 counters** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding H.323 ALG DoS Attack Protection on page 230
- H.323 ALG Configuration Overview on page 227

## H.323 ALG Unknown Message Types

- Understanding H.323 ALG Unknown Message Types on page 232
- Example: Allowing Unknown H.323 ALG Message Types on page 233

### Understanding H.323 ALG Unknown Message Types

This feature enables you to specify how unidentified H.323 messages are handled by the device. The default is to drop unknown (unsupported) messages.

You can protect the H.323 gatekeeper from denial-of-service (DoS) flood attacks by limiting the number of Registration, Admission, and Status (RAS) messages per second it will attempt to process. Incoming RAS request messages exceeding the threshold you specify are dropped by the H.323 Application Layer Gateway (ALG). The range is 2 to 50,000 messages per second, the default value is 1000.

We do not recommend permitting unknown messages because they can compromise security. However, in a secure test or production environment, this command can be useful for resolving interoperability issues with disparate vendor equipment. Permitting unknown H.323 messages can help you get your network operational, so that you can analyze your voice-over-IP (VoIP) traffic to determine why some messages were being dropped. The unknown H.323 message type feature enables you to configure the device to accept H.323 traffic containing unknown message types in both Network Address Translation (NAT) mode and route mode.



**NOTE:** This option applies only to received packets identified as supported VoIP packets. If a packet cannot be identified, it is always dropped. If a packet is identified as a supported protocol and you have configured the device to permit unknown message types, the message is forwarded without processing.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding H.323 ALGs on page 223
- H.323 ALG Configuration Overview on page 227

- Example: Allowing Unknown H.323 ALG Message Types on page 233

## Example: Allowing Unknown H.323 ALG Message Types

This example shows how to configure the device to allow unknown H.323 message types in both route and NAT modes.

- Requirements on page 233
- Overview on page 233
- Configuration on page 233
- Verification on page 234

### Requirements

Before you begin, understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.

### Overview

This feature enables you to specify how unidentified H.323 messages are handled by the device. The default is to drop unknown (unsupported) messages. The Enable Permit NAT applied option and the **permit-nat-applied** configuration statement specify that unknown messages be allowed to pass if the session is in NAT mode. The Enable Permit routed option and the **permit-routed** configuration statement specify that unknown messages be allowed to pass if the session is in route mode. (Sessions in transparent mode are treated as route mode.)

### Configuration

#### J-Web Quick Configuration

To configure the device to allow unknown H.323 message types in both route and NAT modes:

1. Select **Configure>Security>ALG**.
2. Select the **H323** tab.
3. Select the **Enable Permit NAT applied** check box.
4. Select the **Enable Permit routed** check box.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

#### Step-by-Step Procedure

To configure the device to allow unknown H.323 message types in both route and NAT modes:

1. Specify that unknown messages be allowed to pass if the session is in NAT mode.  

```
[edit]
user@host# set security alg h323 application-screen unknown-message
permit-nat-applied
```

2. Specify that unknown messages be allowed to pass if the session is in route mode.

```
[edit]
user@host# set security alg h323 application-screen unknown-message
permit-routed
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

---

To verify the configuration is working properly, enter the **show security alg h323** command and the **show security alg h323 counters** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding H.323 ALG Unknown Message Types on page 232](#)
- [H.323 ALG Configuration Overview on page 227](#)

## Example: Passing H.323 ALG Traffic to a Gatekeeper in the Private Zone

---

This example shows how to set up two policies that allow H.323 traffic to pass between IP phone hosts and a gatekeeper in the private zone, and an IP phone host (2.2.2.5/32) in the public zone.

- [Requirements on page 234](#)
- [Overview on page 234](#)
- [Configuration on page 235](#)
- [Verification on page 238](#)

### Requirements

Before you begin:

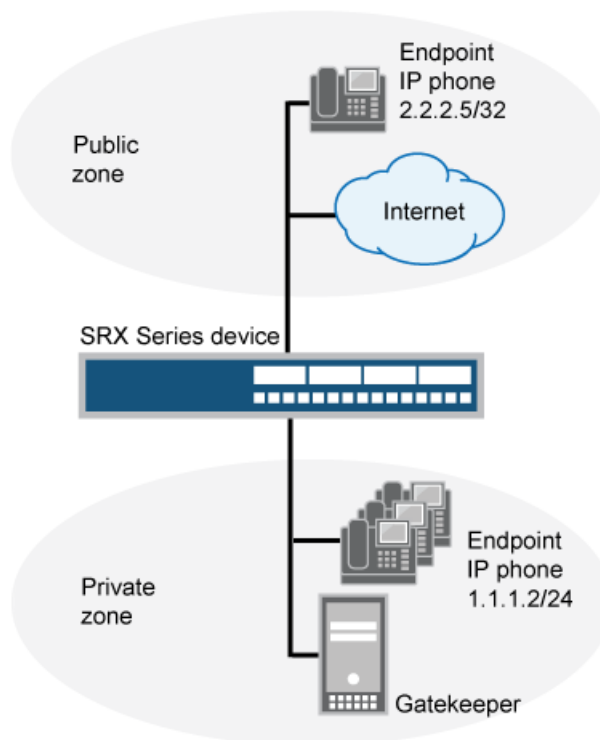
- Understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.
- Configure security zones. See “Understanding Security Zones” on page 113.

### Overview

This example shows how to set up two policies that allow H.323 traffic to pass between IP phone hosts and a gatekeeper in the private zone, and an IP phone host (2.2.2.5/32) in the public zone. The connection to the device can either be with or without NAT. See Figure 14 on page 235.



Figure 14: H.323 Gatekeeper in the Private Zone



8030625

## Configuration

**CLI Quick Configuration** To quickly configure the device to pass H.323 ALG traffic to a gatekeeper in the private zone, copy the following commands and paste them into the CLI:

[edit]

```
set security zones security-zone public address-book address ip_phone 2.2.2.5/32
```

```
set security zones security-zone private address-book address gateway 2.2.2.5/32
```

```
set security policies from-zone private to-zone public policy P1 match source-address any
```

```
set security policies from-zone private to-zone public policy P1 match destination-address IP_Phone
```

```
set security policies from-zone private to-zone public policy P1 match application junos-h323
```

```
set security policies from-zone private to-zone public policy P1 then permit
```

```
set security policies from-zone public to-zone private policy P2 match source-address any
```

```
set security policies from-zone public to-zone private policy P2 match destination-address gateway
```

```
set security policies from-zone public to-zone private policy P2 match application junos-h323
```

```
set security policies from-zone public to-zone private policy P2 then permit
```

**Step-by-Step Procedure**

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To configure the device to pass H.323 ALG traffic to a gatekeeper in the private zone:

1. Configure two address books.

```
[edit]
```

```
user@host# set security zones security-zone public address-book address ip_phone 2.2.2.5/32
```

```
set security zones security-zone private address-book address gateway 2.2.2.5/32
```

2. Configure policy P1 from the private zone to the public zone.

```
[edit]
```

```
user@host# set security policies from-zone private to-zone public policy P1 match source-address any
```

```
user@host# set security policies from-zone private to-zone public policy P1 match destination-address IP_Phone
```

```
user@host# set security policies from-zone private to-zone public policy P1 match application junos-h323
```

```
user@host# set security policies from-zone private to-zone public policy P1 then permit
```

3. Configure policy P2 from the public zone to the private zone.

```
[edit]
```

```
user@host# set security policies from-zone public to-zone private policy P2 match source-address any
```

```
user@host# set security policies from-zone public to-zone private policy P2 match destination-address gateway
```

```
user@host# set security policies from-zone public to-zone private policy P2 match application junos-h323
```

```
user@host# set security policies from-zone public to-zone private policy P2 then
permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this show output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
user@host# show security policies
...
from-zone trust to-zone trust {
  policy default-permit {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone trust to-zone untrust {
  policy default-permit {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone untrust to-zone trust {
  policy default-deny {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      deny;
    }
  }
}
from-zone private to-zone public {
  policy P1 {
    match {
      source-address any;
      destination-address IP_Phone;
      application junos-h323;
    }
    then {
      permit;
    }
  }
}
```

```

}
from-zone public to-zone private {
  policy P2 {
    match {
      source-address any;
      destination-address gateway;
      application junos-h323;
    }
    then {
      permit;
    }
  }
}
...

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform this task:

- Verifying H.323 ALG Configurations on page 238

### Verifying H.323 ALG Configurations

**Purpose** Display information about active calls.



**NOTE:** H.323 counters for calls and active calls in the output to this **show security** command do not apply to the proprietary Avaya implementation of H.323. This is because Q.931 setup and connect messages are exchanged right after the phone is powered up and call creation and tear down is done by Facility messages.

Counters for calls and active calls are increased when the resources allocated for calls are increased—that is, messages belonging to the same call and that pass the firewall multiple times increment the counters. This applies when resources for a call need to be allocated multiple times. For example, in a two-zone scenario the setup and connect message pair allocates one call resource, and the active call counter is increased by one. But in a three-zone scenario the setup and connect message pair passes the firewall twice, each time allocating different call resources. In this case, the counter is incremented.

**Action** From the J-Web interface, select **Monitor>ALGs>H323**. Alternatively, from the CLI, enter the **show security alg h323 counters** command.

Counters for H.245 messages received also will not be accurate in the case of H.245 tunneling. Because H.245 messages are encapsulated in Q.931 packets, the counter for H.245 messages received will remain zero even when there are H.245 messages. The **Other H245** counter will, however, reflect these packet transmissions.

[edit]

```

user@host> show security alg h323 counters
H.323 counters summary:
  Packets received      : 0
  Packets dropped       : 0
  RAS message received  : 0
  Q.931 message received : 0
  H.245 message received : 0
  Number of calls       : 0
  Number of active calls : 0
H.323 error counters:
  Decoding errors       : 0
  Message flood dropped  : 0
  NAT errors            : 0
  Resource manager errors : 0
H.323 message counters:
  RRQ      : 0
  RCF      : 0
  ARQ      : 0
  ACF      : 0
  URQ      : 0
  UCF      : 0
  DRQ      : 0
  DCF      : 0
  Oth RAS  : 0
  Setup    : 0
  Alert    : 0
  Connect  : 0
  CallProd : 0
  Info     : 0
  RelCmpl  : 0
  Facility : 0
  Empty    : 0
  OLC      : 0
  OLC-ACK  : 0
  Oth H245 : 0

```

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding H.323 ALGs on page 223](#)
  - [H.323 ALG Configuration Overview on page 227](#)

## Example: Passing H.323 ALG Traffic to a Gatekeeper in the External Zone

This example shows how to set up two policies to allow H.323 traffic to pass between IP phone hosts in the internal zone, and the IP phone at IP address 2.2.2.5/32 (and the gatekeeper) in the external zone.

- [Requirements on page 239](#)
- [Overview on page 240](#)
- [Configuration on page 240](#)
- [Verification on page 244](#)

## Requirements

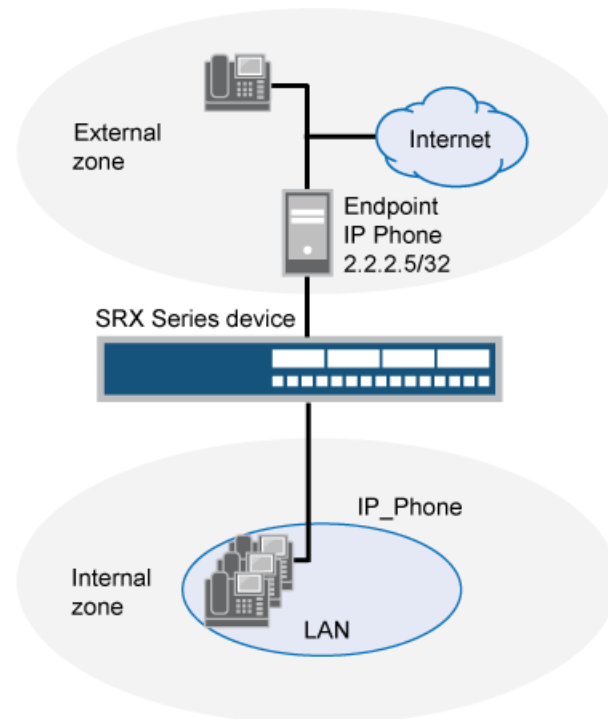
Before you begin:

- Understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.
- Configure security zones. See “Understanding Security Zones” on page 113.

## Overview

Because route mode does not require address mapping of any kind, a device configuration for a gatekeeper in the external, or public, zone is usually identical to the configuration for a gatekeeper in an internal, or private, zone. This example shows how to set up two policies to allow H.323 traffic to pass between IP phone hosts in the internal zone, and the IP phone at IP address 2.2.2.5/32 (and the gatekeeper) in the external zone. The device can be in transparent or route mode. See Figure 15 on page 240.

Figure 15: H.323 Gatekeeper in the External Zone



## Configuration

**CLI Quick Configuration** To quickly configure the device to pass H.323 ALG traffic to a gatekeeper in the external zone, copy the following commands and paste them into the CLI:

[edit]

```
set security zones security-zone external address-book address IP_Phone 2.2.2.5/32
```

```
set security zones security-zone internal address-book address gatekeeper 2.2.2.10/32
```

---

```
set security policies from-zone internal to-zone external policy P1 match source-address  
any
```

```
set security policies from-zone internal to-zone external policy P1 match  
destination-address IP_Phone
```

```
set security policies from-zone internal to-zone external policy P1 match application  
junos-h323
```

```
set security policies from-zone internal to-zone external policy P1 then permit
```

```
set security policies from-zone internal to-zone external policy P2 match source-address  
any
```

```
set security policies from-zone internal to-zone external policy P2 match  
destination-address gatekeeper
```

```
set security policies from-zone internal to-zone external policy P2 match application  
junos-h323
```

```
set security policies from-zone internal to-zone external policy P2 then permit
```

```
set security policies from-zone external to-zone internal policy P3 match source-address  
IP_Phone
```

```
set security policies from-zone external to-zone internal policy P3 match  
destination-address any
```

```
set security policies from-zone external to-zone internal policy P3 match application  
junos-h323
```

```
set security policies from-zone external to-zone internal policy P3 then permit
```

```
set security policies from-zone external to-zone internal policy P4 match source-address  
gatekeeper
```

```
set security policies from-zone external to-zone internal policy P4 match  
destination-address any
```

```
set security policies from-zone external to-zone internal policy P4 match application  
junos-h323
```

```
set security policies from-zone external to-zone internal policy P4 then permit
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To configure the device to pass H.323 ALG traffic to a gatekeeper in the external zone:

1. Configure two address books.

[edit]

```
user@host# set security zones security-zone external address-book address
IP_Phone 2.2.2.5/32
```

```
user@host# set security zones security-zone internal address-book address
gatekeeper 2.2.2.10/32
```

2. Configure policy P1 from the internal zone to the external zone.

[edit]

```
user@host# set security policies from-zone internal to-zone external policy P1 match
source-address any
```

```
user@host# set security policies from-zone internal to-zone external policy P1 match
destination-address IP_Phone
```

```
user@host# set security policies from-zone internal to-zone external policy P1 match
application junos-h323
```

```
user@host# set security policies from-zone internal to-zone external policy P1 then
permit
```

3. Configure policy P2 to allow traffic between the internal zone and the gatekeeper in the external zone.

[edit]

```
user@host# set security policies from-zone internal to-zone external policy P2 match
source-address any
```

```
user@host# set security policies from-zone internal to-zone external policy P2 match
destination-address gatekeeper
```

```
user@host# set security policies from-zone internal to-zone external policy P2 match
application junos-h323
```

```
user@host# set security policies from-zone internal to-zone external policy P2 then
permit
```

4. Configure policy P3 to allow traffic between phones in the internal zone and the external zone.

[edit]



```
user@host# set security policies from-zone external to-zone internal policy P3 match
source-address IP_Phone
```

```
user@host# set security policies from-zone external to-zone internal policy P3 match
destination-address any
```

```
user@host# set security policies from-zone external to-zone internal policy P3 match
application junos-h323
```

```
user@host# set security policies from-zone external to-zone internal policy P3 then
permit
```

5. Configure policy P4 to allow traffic between phones in the internal zone and the gatekeeper in the external zone.

```
[edit]
```

```
user@host# set security policies from-zone external to-zone internal policy P4
match source-address gatekeeper
```

```
user@host# set security policies from-zone external to-zone internal policy P4
match destination-address any
```

```
user@host# set security policies from-zone external to-zone internal policy P4
match application junos-h323
```

```
user@host# set security policies from-zone external to-zone internal policy P4 then
permit
```

**Results** From configuration mode, confirm your configuration by entering the `show security policies` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this show output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
```

```
user@host# show security policies
```

```
...
from-zone internal to-zone external {
  policy P1 {
    match {
      source-address any;
      destination-address IP_Phone;
      application junos-h323;
    }
    then {
      permit;
    }
  }
  policy P2 {
    match {
      source-address any;
      destination-address gatekeeper;
      application junos-h323;
    }
  }
}
```

```
        }
        then {
            permit;
        }
    }
}
from-zone external to-zone internal {
    policy P3 {
        match {
            source-address IP_Phone;
            destination-address any;
            application junos-h323;
        }
        then {
            permit;
        }
    }
    policy P4 {
        match {
            source-address gatekeeper;
            destination-address any;
            application junos-h323;
        }
        then {
            permit;
        }
    }
}
...

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform this task:

- Verifying H.323 ALG Configurations on page 244

### [Verifying H.323 ALG Configurations](#)

---

**Purpose** Display information about active calls.



**NOTE:** H.323 counters for calls and active calls in the output to this `show security` command do not apply to the proprietary Avaya implementation of H.323. This is because Q.931 setup and connect messages are exchanged right after the phone is powered up and call creation and tear down is done by Facility messages.

Counters for calls and active calls are increased when the resources allocated for calls are increased—that is, messages belonging to the same call and that pass the firewall multiple times increment the counters. This applies when resources for a call need to be allocated multiple times. For example, in a two-zone scenario the setup and connect message pair allocates one call resource, and the active call counter is increased by one. But in a three-zone scenario the setup and connect message pair passes the firewall twice, each time allocating different call resources. In this case, the counter is incremented.

**Action** From the J-Web interface, select **Monitor>ALGs>H323**. Alternatively, from the CLI, enter the `show security alg h323 counters` command.

Counters for H.245 messages received also will not be accurate in the case of H.245 tunneling. Because H.245 messages are encapsulated in Q.931 packets, the counter for H.245 messages received will remain zero even when there are H.245 messages. The **Other H245** counter will, however, reflect these packet transmissions.

```
[edit]
user@host> show security alg h323 counters
H.323 counters summary:
  Packets received      : 0
  Packets dropped      : 0
  RAS message received : 0
  Q.931 message received : 0
  H.245 message received : 0
  Number of calls      : 0
  Number of active calls : 0
H.323 error counters:
  Decoding errors      : 0
  Message flood dropped : 0
  NAT errors           : 0
  Resource manager errors : 0
H.323 message counters:
  RRQ      : 0
  RCF      : 0
  ARQ      : 0
  ACF      : 0
  URQ      : 0
  UCF      : 0
  DRQ      : 0
  DCF      : 0
  Oth RAS  : 0
  Setup    : 0
  Alert    : 0
  Connect  : 0
  CallProd : 0
```

```
Info      : 0
RelCmp1   : 0
Facility  : 0
Empty     : 0
OLC       : 0
OLC-ACK   : 0
Oth H245  : 0
```

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding H.323 ALGs on page 223](#)
- [H.323 ALG Configuration Overview on page 227](#)

## Example: Using NAT with the H.323 ALG to Enable Incoming Calls

---

This example shows how to configure NAT with the H.323 ALG to enable calls from a public to a private network.

- [Requirements on page 246](#)
- [Overview on page 246](#)
- [Configuration on page 247](#)
- [Verification on page 251](#)

### Requirements

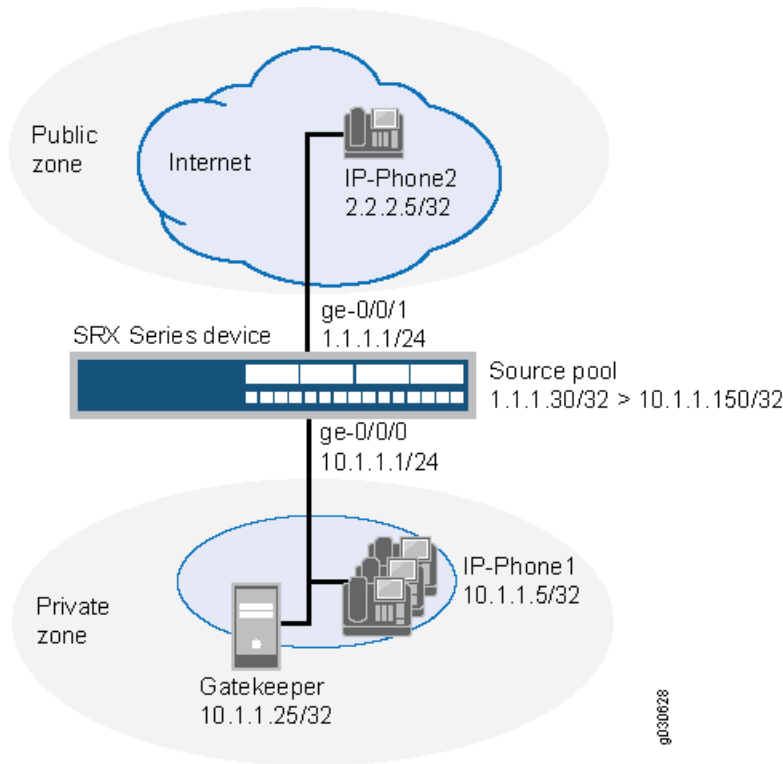
Before you begin, understand H.323 ALGs. See “Understanding H.323 ALGs” on page 223.

### Overview

In a two-zone scenario with a server in the private zone, you can use NAT for incoming calls by configuring a NAT pool on the interface to the public zone.

In this example (see Figure 16 on page 247), IP-Phone1 and a server called gatekeeper are in the private zone, and IP-Phone2 is in the public zone. You configure a static nat rule set and a source NAT pool to do NAT. You also create two policies, private-to-public and public-to-private, to permit ALG H.323 traffic from and to the private and public zones.

Figure 16: NAT with the H.323 ALG—Incoming Calls



In this example, you configure source NAT as follows:

- Create a static NAT rule set called gatekeeper with a rule called gatekeeper to match packets from the public zone with the destination address 1.1.1.25/32. For matching packets, the destination IP address is translated to the private address 10.1.1.25/32.
- Define a source NAT pool called h323-nat-pool to contain the IP address range from 1.1.1.30/32 through 1.1.1.150/32.
- Create a source NAT rule set called h323-nat with rule h323-r1 to match packets from the private zone to the public zone with the source IP address 10.1.1.0/24. For matching packets, the source address is translated to the IP address in h323-nat-pool.
- Configure proxy ARP for the addresses 1.1.1.30/32 through 1.1.1.150/32 on interface ge-0/0/1.0. This allows the system to respond to ARP requests received on the interface for these addresses.

## Configuration

**CLI Quick Configuration** To quickly configure NAT with the H.323 ALG to enable calls from a public to a private network, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces ge-0/0/1 unit 0 family inet address 1.1.1.1/24
set security zones security-zone private address-book address IP-Phone1 10.1.1.5/32
set security zones security-zone private address-book address gatekeeper 10.1.1.25/32
```

```

set security zones security-zone private interfaces ge-0/0/0.0
set security zones security-zone public address-book address IP-Phone2 2.2.2.5/32
set security zones security-zone public interfaces ge-0/0/1.0
set security nat source pool h323-nat-pool address 1.1.1.30/32 to 1.1.1.150/32
set security nat source address-persistent
set security nat source rule-set h323-nat from zone private
set security nat source rule-set h323-nat to zone public
set security nat source rule-set h323-nat rule h323-r1 match source-address 10.1.1.0/24
set security nat source rule-set h323-nat rule h323-r1 then source-nat pool h323-nat-pool
set security nat proxy-arp interface ge-0/0/1.0 address 1.1.1.30/32 to 1.1.1.150/32
set security policies from-zone private to-zone public policy private-to-public match
  source-address IP-Phone1
set security policies from-zone private to-zone public policy private-to-public match
  source-address gatekeeper
set security policies from-zone private to-zone public policy private-to-public match
  destination-address IP-Phone2
set security policies from-zone private to-zone public policy private-to-public match
  application junos-h323
set security policies from-zone private to-zone public policy private-to-public then permit
set security policies from-zone public to-zone private policy public-to-private match
  source-address IP-Phone2
set security policies from-zone public to-zone private policy public-to-private match
  destination-address IP-Phone1
set security policies from-zone public to-zone private policy public-to-private match
  destination-address gatekeeper
set security policies from-zone public to-zone private policy public-to-private match
  application junos-h323
set security policies from-zone public to-zone private policy public-to-private then permit

```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure NAT with H.323 ALG to enable calls from a public to a private network:

1. Configure interfaces.

```

[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 1.1.1.1/24

```

2. Configure zones and assign addresses to them.

```

[edit security zones]
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone private address-book address IP-Phone1 10.1.1.5/32
user@host# set security-zone private address-book address gatekeeper 10.1.1.25/32
user@host# set security-zone public interfaces ge-0/0/1.0
user@host# set security-zone public address-book address IP-Phone2 2.2.2.5/32

```

3. Create a static NAT rule set.

```

[edit security nat static rule-set ip-phones]
user@host# set from zone public
user@host# set match destination-address 1.1.1.25/32
user@host# set then static-nat prefix 10.1.1.25/32

```

4. Configure proxy ARP.
 

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/1.0 address 1.1.1.25/32
```
5. Configure a source NAT rule set.
 

```
[edit security nat]
set source pool h323-nat-pool address 1.1.1.30/32 to 1.1.1.150/32
set source address-persistent
set source rule-set h323-nat from zone private
set source rule-set h323-nat to zone public
set source rule-set h323-nat rule h323-r1 match source-address 10.1.1.0/24
set source rule-set h323-nat rule h323-r1 then source-nat pool h323-nat-pool
set proxy-arp interface ge-0/0/1.0 address 1.1.1.30/32 to 1.1.1.150/32
```
6. Configure policies for outgoing traffic.
 

```
[edit security policies from-zone private to-zone public policy private-to-public]
user@host# set match source-address IP-Phone1
user@host# set match source-address gatekeeper
user@host# set match destination-address IP-Phone2
user@host# set match application junos-h323
user@host# set then permit
```
7. Configure policies for incoming traffic.
 

```
[edit security policies from-zone public to-zone private policy public-to-private]
user@host# set match source-address IP-Phone2
user@host# set match destination-address IP-Phone1
user@host# set match destination-address gatekeeper
user@host# set match application junos-h323
user@host# set then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show security zones**, **show security nat**, and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 10.1.1.1/24;
    }
  }
}
ge-0/0/1 {
  unit 0 {
    family inet {
      address 1.1.1.1/24;
    }
  }
}
[edit]
user@host# show security zones
security-zone private {
```

```
address-book {
  address IP-Phone1 10.1.1.5/32;
  address gatekeeper 10.1.1.25/32;
}
interfaces {
  ge-0/0/0.0;
}
}
security-zone public {
  address-book {
    address IP-Phone2 2.2.2.5/32;
  }
  interfaces {
    ge-0/0/1.0;
  }
}
[edit]
user@host# show security nat
source {
  pool h323-nat-pool {
    address {
      1.1.1.30/32 to 1.1.1.150/32;
    }
  }
  address-persistent;
  rule-set h323-nat {
    from zone private;
    to zone public;
    rule h323-r1 {
      match {
        source-address 10.1.1.0/24;
      }
      then {
        source-nat {
          pool {
            h323-nat-pool;
          }
        }
      }
    }
  }
}
}
proxy-arp {
  interface ge-0/0/1.0 {
    address {
      1.1.1.30/32 to 1.1.1.150/32;
    }
  }
}
static {
  rule-set ip-phones {
    from zone public;
    rule gatekeeper {
      match {
        destination-address 1.1.1.25/32;
      }
    }
  }
}
```



```

        then {
            static-nat prefix 10.1.1.25/32;
        }
    }
}
proxy-arp {
    interface ge-0/0/1.0 {
        address {
            1.1.1.25/32;
        }
    }
}
[edit]
user@host# show security policies
from-zone private to-zone public {
    policy private-to-public {
        match {
            source-address [IP-Phone1 gatekeeper];
            destination-address IP-Phone2;
            application junos-h323;
        }
        then {
            permit;
        }
    }
}
from-zone public to-zone private {
    policy public-to-private {
        match {
            source-address IP-Phone2;
            destination-address [IP-Phone1 gatekeeper];
            application junos-h323;
        }
        then {
            permit;
        }
    }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying H.323 ALG Status on page 251
- Verifying Static NAT Configuration on page 252
- Verifying Source NAT Rule Usage on page 252

### Verifying H.323 ALG Status

**Purpose** Verify that H.323 ALG is enabled on your system.

**Action** From operational mode, enter the **show security alg h323 counters** command to display information about active calls.

### [Verifying Static NAT Configuration](#)

---

**Purpose** Verify that there is traffic matching the static NAT rule set.

**Action** From operational mode, enter the **show security nat static rule** command. View the Translation hits field to check for traffic that matches the rule.

### [Verifying Source NAT Rule Usage](#)

---

**Purpose** Verify that there is traffic matching the source NAT rule.

**Action** From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Verifying H.323 ALG Configurations](#)
- [H.323 ALG Configuration Overview on page 227](#)

## [Example: Using NAT with the H.323 ALG to Enable Outgoing Calls](#)

---

This example shows how to configure static NAT with H.323 ALG to enable calls from a private to a public network.

- [Requirements on page 252](#)
- [Overview on page 252](#)
- [Configuration on page 253](#)
- [Verification on page 257](#)

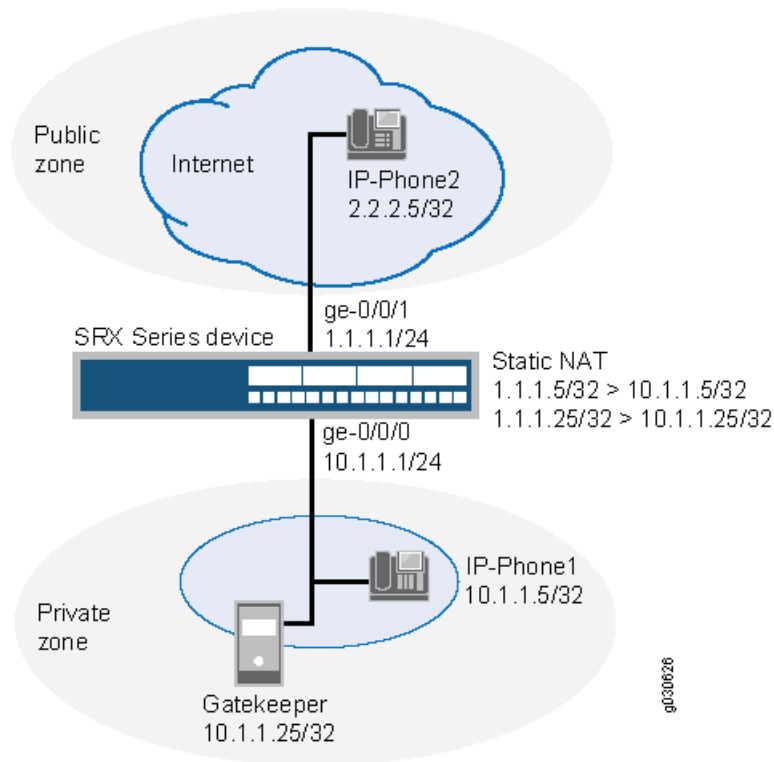
### Requirements

Before you begin, understand the H.323 ALG and its processes. See “Understanding H.323 ALGs” on page 223.

### Overview

In this example (see Figure 17 on page 253), IP-Phone1 and a server called gatekeeper are in the private zone and IP-Phone2 is in the public zone. You configure static NAT to enable IP-Phone1 and gatekeeper to call IP-Phone2 in the public zone. You then create a policy called public-to-private to allow ALG H.323 traffic from the public zone to the private zone and a policy called private-to-public to allow ALG H.323 traffic from the private zone to the public zone.

Figure 17: NAT with the H.323 ALG—Outgoing Calls



In this example, you configure static NAT as follows:

- Create a static NAT rule set called `ip-phones` with a rule called `phone1` to match packets from the public zone with the destination address `1.1.1.5/32`. For matching packets, the destination IP address is translated to the private address `10.1.1.5/32`.
- Define a second rule called `gatekeeper` to match packets from the public zone with the destination address `1.1.1.25/32`. For matching packets, the destination IP address is translated to the private address `10.1.1.25/32`.
- Create proxy ARP for the addresses `1.1.1.5/32` and `1.1.1.25/32` on interface `ge-0/0/1`. This allows the system to respond to ARP requests received on the specified interface for these addresses.

## Configuration

**CLI Quick Configuration** To quickly configure static NAT with the H.323 ALG to enable calls from a private to a public network, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces ge-0/0/1 unit 0 family inet address 1.1.1.1/24
set security zones security-zone private address-book address IP-Phone1 10.1.1.5/32
set security zones security-zone private address-book address gatekeeper 10.1.1.25/32
set security zones security-zone private interfaces ge-0/0/0.0
set security zones security-zone public address-book address IP-Phone2 2.2.2.5/32
set security zones security-zone public interfaces ge-0/0/1.0
```

```

set security nat static rule-set ip-phones from zone public
set security nat static rule-set ip-phones rule phone1 match destination-address 1.1.1.5/32
set security nat static rule-set ip-phones rule phone1 then static-nat prefix 10.1.1.5/32
set security nat static rule-set ip-phones rule gatekeeper match destination-address
  1.1.1.25/32
set security nat static rule-set ip-phones rule gatekeeper then static-nat prefix 10.1.1.25/32
set security nat proxy-arp interface ge-0/0/1.0 address 1.1.1.5/32
set security nat proxy-arp interface ge-0/0/1.0 address 1.1.1.25/32
set security policies from-zone public to-zone private policy public-to-private match
  source-address IP-Phone2
set security policies from-zone public to-zone private policy public-to-private match
  destination-address gatekeeper
set security policies from-zone public to-zone private policy public-to-private match
  application junos-h323
set security policies from-zone public to-zone private policy public-to-private then permit
set security policies from-zone private to-zone public policy private-to-public match
  source-address IP-Phone1
set security policies from-zone private to-zone public policy private-to-public match
  source-address gatekeeper
set security policies from-zone private to-zone public policy private-to-public match
  destination-address IP-Phone2
set security policies from-zone private to-zone public policy private-to-public match
  application junos-h323
set security policies from-zone private to-zone public policy private-to-public then permit

```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure static NAT with the H.323 ALG to enable calls from a private to a public network:

1. Configure interfaces.

```

user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 1.1.1.1/24

```

2. Create zones and assign addresses to them.

```

[edit security zones]
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone public interfaces ge-0/0/1.0
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone private address-book address IP-Phone1 10.1.1.5/32
user@host# set security-zone private address-book address gatekeeper 10.1.1.25/32
user@host# set security-zone public interfaces ge-0/0/1.0
user@host# set security-zone public address-book address IP-Phone2 2.2.2.5/32

```

3. Configure static NAT rule set with rules.

```

[edit security nat static rule-set ip-phones]
user@host# set from zone public
user@host# set rule phone1 match destination-address 1.1.1.5/32
user@host# set rule phone1 then static-nat prefix 10.1.1.5/32
user@host# set rule gatekeeper match destination-address 1.1.1.25/32
user@host# set rule gatekeeper then static-nat prefix 10.1.1.25/32

```

4. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/1 address 1.1.1.5/32
user@host# set proxy-arp interface ge-0/0/1 address 1.1.1.25/32
```

5. Configure a security policy for incoming traffic.

```
[edit security policies from-zone public to-zone private policy public-to-private]
user@host# set match source-address IP-Phone2
user@host# set match destination-address gatekeeper
user@host# set match application junos-h323
user@host# set then permit
```

6. Configure a security policy for outgoing traffic.

```
[edit security policies from-zone private to-zone public policy private-to-public]
user@host# set match source-address IP-Phone1
user@host# set match source-address gatekeeper
user@host# set match destination-address IP-Phone2
user@host# set match application junos-h323
user@host# set then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show security zones**, **show security nat**, and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 10.1.1.1/24;
    }
  }
}
ge-0/0/1 {
  unit 0 {
    family inet {
      address 1.1.1.1/24;
    }
  }
}
[edit]
user@host# show security zones
security-zone private {
  address-book {
    address IP-Phone1 10.1.1.5/32;
    address gatekeeper 10.1.1.25/32;
  }
  interfaces {
    ge-0/0/0.0;
  }
}
security-zone public {
  address-book {
```

```
        address IP-Phone2 2.2.2.5/32;
    }
    interfaces {
        ge-0/0/1.0;
    }
}
[edit]
user@host# show security nat
static {
    rule-set ip-phones {
        from zone public;
        rule phone1 {
            match {
                destination-address 1.1.1.5/32;
            }
            then {
                static-nat prefix 10.1.1.5/32;
            }
        }
        rule gatekeeper {
            match {
                destination-address 1.1.1.25/32;
            }
            then {
                static-nat prefix 10.1.1.25/32;
            }
        }
    }
}
proxy-arp {
    interface ge-0/0/1.0 {
        address {
            1.1.1.5/32;
            1.1.1.25/32;
        }
    }
}
[edit]
user@host# show security policies
from-zone public to-zone private {
    policy public-to-private {
        match {
            source-address IP-Phone2;
            destination-address gatekeeper;
            application junos-h323;
        }
        then {
            permit;
        }
    }
}
from-zone private to-zone public {
    policy private-to-public {
        match {
            source-address [ IP-Phone1 gatekeeper ];
            destination-address IP-Phone2;
```

```
        application junos-h323;
    }
    then {
        permit;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying H.323 ALG Status on page 257
- Verifying Static NAT Configuration on page 257

### Verifying H.323 ALG Status

---

**Purpose** Verify that H.323 ALG is enabled on your system.

**Action** From operational mode, enter the **show security alg h323 counters** command to display information about active calls.

### Verifying Static NAT Configuration

---

**Purpose** Verify that there is traffic matching the static NAT rule set.

**Action** From operational mode, enter the **show security nat static rule** command. View the Translation hits field to check for traffic that matches the rule.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Verifying H.323 ALG Configurations
- H.323 ALG Configuration Overview on page 227





# ALG for IKE and ESP

- Understanding ALG for IKE and ESP on page 259
- Understanding ALG for IKE and ESP Operation on page 260
- Example: Configuring the IKE and ESP ALG on page 261
- Example: Enabling IKE and ESP ALG and Setting Timeouts on page 266

## Understanding ALG for IKE and ESP

---

An SRX Series or J Series device can be used solely as a Network Address Translation (NAT) device when placed between VPN clients on the private side of the NAT gateway and the virtual private network (VPN) gateways on the public side.

Internet Key Exchange (IKE) and Encapsulating Security Payload (ESP) traffic is exchanged between the clients and the server. However, if the clients do not support NAT-Traversal (NAT-T) and if the device assigns the same NAT-generated IP address to two or more clients, the device will be unable to distinguish and route return traffic properly.



**NOTE:** If the user wants to support both NAT-T-capable and non-NAT-T-capable clients, then some additional configurations are required. If there are NAT-T capable clients, the user must enable the source NAT address persistence.

ALG for IKE and ESP monitors IKE traffic between the client and the server and permits only one IKE Phase 2 message exchange between any given client/server pair, not just one exchange between any client and any server.

ALG for IKE and ESP traffic has been created and NAT has been enhanced to implement the following:

- To enable the SRX Series and J Series devices to pass IKE and ESP traffic with a source NAT pool
- To allow the device to be configured to return the same NAT-generated IP address for the same IP address without NAT ("address-persistent NAT"). As a result, the device is able to associate a client's outgoing IKE traffic with its return traffic from the server, especially when the IKE session times out and needs to be reestablished.

- The resulting ESP traffic between the client and the server is also allowed, especially in the direction from the server to the client.
- The return ESP traffic matches the following:
  - The server IP address as source IP
  - The client IP address as destination IP

## Understanding ALG for IKE and ESP Operation

---

The proposed ALG for IKE and ESP traffic will have the following behavior:

- The ALG for IKE and ESP monitors IKE traffic between the client and the server, and permits only one IKE Phase 2 message exchange between the client and the server at any given time.
- When a Phase 2 message is seen:
  - If no Phase 2 exchange between the client and server is already taking place, the IKE ALG will open gates for the relevant ESP traffic in the client to server and server to client directions.
  - If the gates cannot be successfully opened, or if there is already a Phase 2 exchange taking place, the Phase 2 message will be dropped.
- When ESP traffic hits those gates, sessions will be created to capture subsequent ESP traffic, and perform the proper NATing (source IP address translation for client ->server traffic, and destination IP address translation for server->client traffic).
- If no traffic hits either or both of the gates, the gate(s) will naturally time out.
- Once the gates are collapsed or timed out, another IKE Phase 2 exchange will be permitted.
- IKE NAT-T traffic on floating port 4500 will not be processed in IKE ALG. To support mixture of NAT-T-capable and non-capable clients, users is required to enable source NAT address persistent.

### Related Documentation

- ALG Overview on page 217
- NAT Overview on page 1335
- Understanding ALG for IKE and ESP on page 259
- Example: Configuring the IKE and ESP ALG on page 261
- Example: Enabling IKE and ESP ALG and Setting Timeouts on page 266

## Example: Configuring the IKE and ESP ALG

This example shows how to configure the IKE and ESP ALG to pass through IKE and ESP traffic with a source NAT pool on Juniper Networks devices.

- Requirements on page 261
- Overview on page 261
- Configuration on page 261
- Verification on page 265

### Requirements

Before you begin:

- Configure proxy ARP for all IP addresses in the source NAT pool.
- Understand the concepts behind ALG for IKE and ESP. See “Understanding ALG for IKE and ESP Operation” on page 260.

### Overview

In this example, the ALG for IKE and ESP is configured to monitor and allow IKE and ESP traffic to be exchanged between the clients and the server located on opposite sides of a Juniper Networks device.

This example shows how to configure a source NAT pool and rule set, configure a custom application to support the IKE and ESP ALG, and associate this ALG to a policy.

If you want to support a mixture of NAT-traversal (NAT-T) capable clients and noncapable clients, you must enable persistent source NAT translation (so that once a particular source NAT is associated with a given IP address, subsequent source NAT translations use the same IP address). You also must configure a custom IKE NAT traversal application to support the encapsulation of IKE and ESP in UDP port 4500. This configuration enables IKE and ESP to pass through the NAT-enabled device.

### Configuration

- Configuring a NAT Source Pool and Rule Set on page 261
- Configuring a Custom Application and Associating it to a Policy on page 263
- Configuring IKE and ESP ALG Support for Both NAT-T Capable and Noncapable Clients on page 264

#### Configuring a NAT Source Pool and Rule Set

##### CLI Quick Configuration

To quickly configure a NAT source pool and rule set, copy the following commands and paste them into the CLI:

```
[edit]
set security nat source pool pool1 address 10.10.10.1/32 to 10.10.10.10/32
set security zones security-zone green address-book address sa1 1.1.1.0/24
set security zones security-zone red address-book address da1 2.2.2.0/24
set security nat source rule-set rs1 from zone green
```

```

set security nat source rule-set rs1 to zone red
set security nat source rule-set rs1 rule r1 match source-address 1.1.1.0/24
set security nat source rule-set rs1 rule r1 match destination-address 2.2.2.0/24
set security nat source rule-set rs1 rule r1 then source-nat pool pool1

```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a source NAT pool:

1. Create a NAT source pool.

```

[edit ]
user@host# set security nat source pool pool1 address 10.10.10.1/32 to 10.10.10.10/32

```

2. Configure security zone address book entries.

```

[edit]
user@host# set security zones security-zone green address-book address sa1
1.1.1.0/24
user@host# set security zones security-zone red address-book address da1 2.2.2.0/24

```

3. Create a NAT source rule set.

```

[edit security nat source rule-set rs1]
user@host# set from zone green
user@host# set to zone red
user@host# set rule r1 match source-address 1.1.1.0/24
user@host# set rule r1 match destination-address 2.2.2.0/24
user@host# set rule r1 then source-nat pool pool1

```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

user@host# show security nat
source {
  pool pool1 {
    address {
      10.10.10.1/32 to 10.10.10.10/32;
    }
  }
}
rule-set rs1 {
  from zone green;
  to zone red;
  rule r1 {
    match {
      source-address 1.1.1.0/24;
      destination-address 2.2.2.0/24;
    }
    then {
      source-nat {
        pool {
          pool1;
        }
      }
    }
  }
}

```

```

    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring a Custom Application and Associating it to a Policy

**CLI Quick Configuration** To quickly configure a custom application and associate it to a policy, copy the following commands and paste them into the CLI:

```

[edit]
set applications application custom-ike-alg source-port 500 destination-port 500 protocol
  udp application-protocol ike-esp-nat
set security policies from-zone green to-zone red policy pol1 match destination-address
  da1
set security policies from-zone green to-zone red policy pol1 match application
  custom-ike-alg
set security policies from-zone green to-zone red policy pol1 then permit

```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a custom application and associate it to a policy:

1. Configure a custom application.

```

[edit]
user@host# set applications application custom-ike-alg source-port 500
  destination-port 500 protocol udp application-protocol ike-esp-nat

```

2. Associate the custom application to a policy.

```

[edit security policies from-zone green to-zone red policy pol1]
user@host# set match source-address sa1
user@host# set match destination-address da1
user@host# set match application custom-ike-alg
user@host# set then permit

```

**Results** From configuration mode, confirm your configuration by entering the **show applications** and **show security zones** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show applications
application custom-ike-alg {
  application-protocol ike-esp-nat;
  protocol udp;
  source-port 500;
  destination-port 500;
}

[edit]
user@host# show security zones

```

```

security-zone Trust {
  host-inbound-traffic {
    system-services {
      all;
    }
    protocols {
      all;
    }
  }
  interfaces {
    ge-0/0/1.0;
  }
}
security-zone green {
  address-book {
    address sa1 1.1.1.0/24;
  }
}
security-zone red {
  address-book {
    address da1 2.2.2.0/24;
  }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring IKE and ESP ALG Support for Both NAT-T Capable and Noncapable Clients

#### CLI Quick Configuration

To quickly configure IKE and ESP ALG support for both NAT-T and noncapable clients, copy the following commands and paste them into the CLI.

```

[edit]
set security nat source address-persistent
set applications application custom-ike-natt protocol udp source-port 4500
  destination-port 4500
set security policies from-zone green to-zone red policy pol1 match source-address sa1
set security policies from-zone green to-zone red policy pol1 match destination-address
  da1
set security policies from-zone green to-zone red policy pol1 match application
  custom-ike-natt
set security policies from-zone green to-zone red policy pol1 then permit

```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure IKE and ESP ALG support for both NAT-T capable and noncapable clients:

1. Globally enable persistent source NAT translation.

```

[edit]
user@host# set security nat source address-persistent

```

2. Configure the IKE NAT-T application.

```
[edit]
user@host# set applications application custom-ike-natt protocol udp source-port
4500 destination-port 4500
```

3. Associate the NAT-T application using a policy.

```
[edit security policies from-zone green to-zone red policy pol1]
user@host# set match source-address sa1
user@host# set match destination-address da1
user@host# set match application custom-ike-natt
user@host# set then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
source {
  address-persistent;
}

[edit]
user@host# show security policies
from-zone green to-zone red {
  policy pol1 {
    match {
      source-address sa1;
      destination-address da1;
      application [ custom-ike-alg custom-ike-natt ];
    }
    then {
      permit;
    }
  }
}
default-policy {
  permit-all;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying IKE and ESP ALG Custom Applications on page 265
- Verifying the NAT Source Pool and Rule Set on page 266

### [Verifying IKE and ESP ALG Custom Applications](#)

**Purpose** Verify that the custom applications to support the IKE and ESP ALG are enabled or not.

**Action** From operational mode, enter the **show applications** command.

### Verifying the NAT Source Pool and Rule Set

---

- Purpose** Verify that the NAT source pool and rule set used to support the IKE and ESP ALG are working properly.
- Action** From operational mode, enter the **show security nat** command.
- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [ALG Overview on page 217](#)
  - [Understanding ALG for IKE and ESP on page 259](#)
  - [Example: Enabling IKE and ESP ALG and Setting Timeouts on page 266](#)

### Example: Enabling IKE and ESP ALG and Setting Timeouts

---

This example shows how to enable the IKE and ESP ALG and set the timeout values to allow time for the ALG to process ALG state information, ESP gates, and ESP sessions.

- [Requirements on page 266](#)
- [Overview on page 266](#)
- [Configuration on page 266](#)
- [Verification on page 267](#)

#### Requirements

Understand the concepts behind ALG for IKE and ESP. See “Understanding ALG for IKE and ESP Operation” on page 260.

#### Overview

The IKE and ESP ALG processes all traffic specified in any policy to which the ALG is attached. In this example, you configure the **set security alg ike-esp-nat enable** statement so the current default IPsec pass-through behavior is disabled for all IPsec pass-through traffic, regardless of policy.

You then set the timeout values to allow time for the IKE and ESP ALG to process ALG state information, ESP gates, and ESP sessions. In this example, you set the timeout of ALG state information. The timeout range is 180 through 86400 seconds. The default timeout is 14400 seconds. You then set the timeout of the ESP gates created after an IKE Phase 2 exchange has completed. The timeout range is 2 through 30 seconds. The default timeout is 5 seconds. Finally, you set the idle timeout of the ESP sessions created from the IPsec gates. If no traffic hits the session, it is aged out after this period of time. The timeout range is 60 through 2400 seconds. The default timeout is 1800 seconds.

#### Configuration

- CLI Quick Configuration** To quickly enable the IKE and ESP ALG and set timeout values, copy the following commands and paste them into the CLI:



```
[edit]
set security alg ike-esp-nat enable
set security alg ike-esp-nat esp-gate-timeout 20
set security alg ike-esp-nat esp-session-timeout 2400
set security alg ike-esp-nat state-timeout 360
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To enable the IKE and ESP ALG and set the timeout values:

1. Enable the IKE and ESP ALG.

```
[edit]
user@host# set security alg ike-esp-nat enable
```

2. Set the timeout for the ALG state information.

```
[edit security alg ike-esp-nat]
user@host# set state-timeout 360
```

3. Set the timeout for the ESP gates created after an IKE Phase 2 exchange has completed.

```
[edit security alg ike-esp-nat]
user@host# set esp-gate-timeout 20
```

4. Set the idle timeout for the ESP sessions created from the IPsec gates.

```
[edit security alg ike-esp-nat]
user@host# set esp-session-timeout 2400
```

**Results** From configuration mode, confirm your configuration by entering the **show security alg** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security alg
ike-esp-nat {
  enable;
  state-timeout 360;
  esp-gate-timeout 20;
  esp-session-timeout 2400;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying the ALG for IKE and ESP and Timeout Settings on page 268

### [Verifying the ALG for IKE and ESP and Timeout Settings](#)

---

**Purpose** Verify that the ALG for IKE and ESP is enabled and the timeout settings for this feature are correct.

**Action** From operational mode, enter the **show security alg ike-esp-nat** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [ALG Overview on page 217](#)
  - [NAT Overview on page 1335](#)
  - [Understanding ALG for IKE and ESP on page 259](#)
  - [Understanding ALG for IKE and ESP Operation on page 260](#)
  - [Example: Configuring the IKE and ESP ALG on page 261](#)

## CHAPTER 12

# SIP ALGs

- Understanding SIP ALGs on page 269
- Understanding SIP ALG Request Methods on page 274
- SIP ALG Configuration Overview on page 275
- SIP ALG Call Duration and Timeouts on page 275
- SIP ALG DoS Attack Protection on page 278
- SIP ALG Unknown Message Types on page 279
- SIP ALG Hold Resources on page 281
- SIP ALGs and NAT on page 282
- Verifying SIP ALG Configurations on page 321

### Understanding SIP ALGs

---

Session Initiation Protocol (SIP) is an Internet Engineering Task Force (IETF)-standard protocol for initiating, modifying, and terminating multimedia sessions over the Internet. Such sessions might include conferencing, telephony, or multimedia, with features such as instant messaging and application-level mobility in network environments.

Junos OS supports SIP as a service, allowing and denying it based on a policy that you configure. SIP is a predefined service in Junos OS and uses port 5060 as the destination port.

One of SIP's functions is to distribute session-description information and, during the session, to negotiate and modify the parameters of the session. SIP is also used to terminate a multimedia session, signal a call establishment, provide failure indication, and provide methods for endpoint to register.

Session-description information is included in INVITE and ACK messages and indicates the multimedia type of the session; for example, whether it is voice or video. Although SIP can use different description protocols to describe the session, the Juniper Networks SIP Application Layer Gateway (ALG) supports only the Session Description Protocol (SDP).

SDP provides information that a system can use to join a multimedia session. SDP might include information such as IP addresses, port numbers, times, and dates. Note that the IP address and port number in the SDP header (the c= and m= fields, respectively) are

the address and port where the client wants to receive the media streams and not the IP address and port number from which the SIP request originates (although they can be the same).

SIP messages consist of requests from a client to a server and responses to the requests from a server to a client with the purpose of establishing a session (or a call). A User Agent (UA) is an application that runs at the endpoints of the call and consists of two parts:

- User Agent Client (UAC), which sends SIP requests on behalf of the user
- User Agent Server (UAS), which listens to the responses and notifies the user when they arrive

UAC and UAS are defined in relation to the role a particular agent is playing in a negotiation.

Examples of UAs are SIP proxy servers and phones.

This topic contains the following sections:

- SIP ALG Operation on page 270
- SDP Session Descriptions on page 271
- Pinhole Creation on page 272

## SIP ALG Operation

There are two types of SIP traffic, the signaling and the media stream. SIP signaling traffic consists of request and response messages between client and server and uses transport protocols such as UDP or TCP. The media stream carries the data (audio data, for example) using transport protocols.

By default, Junos OS supports SIP signaling messages on port 5060. You can configure the port. You can simply create a policy that permits SIP service, and the software filters SIP signaling traffic like any other type of traffic, permitting or denying it. The media stream, however, uses dynamically assigned port numbers that can change several times during the course of a call. Without fixed ports, it is insecure to create a static policy to control media traffic. In this case, the device invokes the SIP ALG. The device transport ports used for the media sessions are not known in advance, however, the ports used for the SIP negotiation are well-known (or predefined). The ALG registers interest in packets from the control session which can be easily distinguished from the other packets and inspects the negotiation looking for the transport information used for the media session (both IP addresses and ports).



**NOTE:** Pinholes are created when a matching port, transport address, and protocol is determined (whatever information is known at the time the pinhole is opened).

---

The SIP ALG monitors SIP transactions and dynamically creates and manages pinholes based on the information it extracts from these transactions. The Juniper Networks SIP

ALG supports all SIP methods and responses. You can allow SIP transactions to traverse the Juniper Networks firewall by creating a static policy that permits SIP service. If the policy is configured to inspect SIP traffic (or, more appropriately, if the policy sends some traffic to the SIP ALG for inspection) the allowed actions are to permit the traffic (in which case the appropriate pinholes are opened) or to deny the traffic.

The SIP ALG intercepts SIP messages that contain SDP and, using a parser, extracts the information it requires to create pinholes. The SIP ALG examines the SDP portion of the packet, and a parser extracts information such as IP addresses and port numbers, which the SIP ALG records in a pinhole table. The SIP ALG uses the IP addresses and port numbers recorded in the pinhole table to open pinholes and allow media streams to traverse the device.



**NOTE:** When the device is performing NAT, the transport addresses that the user agents employ are incorrect. ALG modifies the transport addresses based on the translated ports and addresses allocated by the NAT-ing device. When SDP is encrypted, the device cannot either extract nor modify the contents of the message and therefore cannot correct the transport addresses. To provide a workaround, the STUN protocol has been deployed (which requires NAT devices to do some form of cone-NAT) which allows the clients to determine the translated addresses and use those newly discovered addresses in the SDP messages.

NEC SIP products are conditionally supported.

## SDP Session Descriptions

An SDP session description is a well defined format for conveying sufficient information to discover and participate in a multimedia session. A session is described by a series of attribute/value pairs, one per line. The attribute names are single characters, followed by '=', and a value. Optional values are specified with '=\*'. Values are either an ASCII string, or a sequence of specific types separated by spaces. Attribute names are only unique within the associated syntactic construct, such as within the session, time, or media only.



**NOTE:** In the SDP session description, the media-level information begins with the m= field.

Of the many fields in the SDP description, two are particularly useful to the SIP ALG because they contain Transport Layer information.

- **c=** for connection information

This field can appear at the session or media level. It displays in this format:

```
c=<network-type><address-type><connection-address>
```

Currently, Junos OS supports only “IN” (for Internet) as the network type, “IP4” as the address type, and a unicast IP address or domain name as the destination (connection) IP address.

If the destination IP address is a unicast IP address, the SIP ALG creates pinholes using the IP address and port numbers specified in the media description field `m=`.

- `m=` for media announcement

This field appears at the media level and contains the description of the media. It displays in this format:

```
m=<media><port><transport><fmt list>
```

Currently, the Junos OS supports only “audio” as the media and “RTP” as the Application Layer transport protocol. The port number indicates the destination port of the media stream (the origin is allocated by the remote user agent). The format list (fmt list) provides information on the Application Layer protocol that the media uses.

The software opens ports only for RTP and Real-Time Control Protocol (RTCP). Every RTP session has a corresponding RTCP session. Therefore, whenever a media stream uses RTP, the SIP ALG must reserve ports (create pinholes) for both RTP and RTCP traffic. By default, the port number for RTCP is one higher than the RTP port number.

## Pinhole Creation

Both pinholes for the RTP and RTCP traffic share the same destination IP address. The IP address comes from the `c=` field in the SDP session description. Because the `c=` field can appear in either the session-level or media-level portion of the SDP session description, the parser determines the IP address based on the following rules (in accordance with SDP conventions):

- First, the SIP ALG parser verifies if there is a `c=` field containing an IP address in the media level. If there is one, the parser extracts that IP address, and the SIP ALG uses it to create a pinhole for the media.
- If there is no `c=` field in the media level, the SIP ALG parser extracts the IP address from the `c=` field in the session level, and the SIP ALG uses it to create a pinhole for the media. If the session description does not contain a `c=` field in either level, this indicates an error in the protocol stack, and the device drops the packet and logs the event.

The SIP ALG needs the following information to create a pinhole. This information comes from the SDP session description and parameters on the device:

- Protocol—UDP.
- Source IP—Unknown.
- Source port—Unknown.
- Destination IP—The parser extracts the destination IP address from the `c=` field in the media or session level.
- Destination port—The parser extracts the destination port number for RTP from the `m=` field in the media level and calculates the destination port number for RTCP using the following formula:

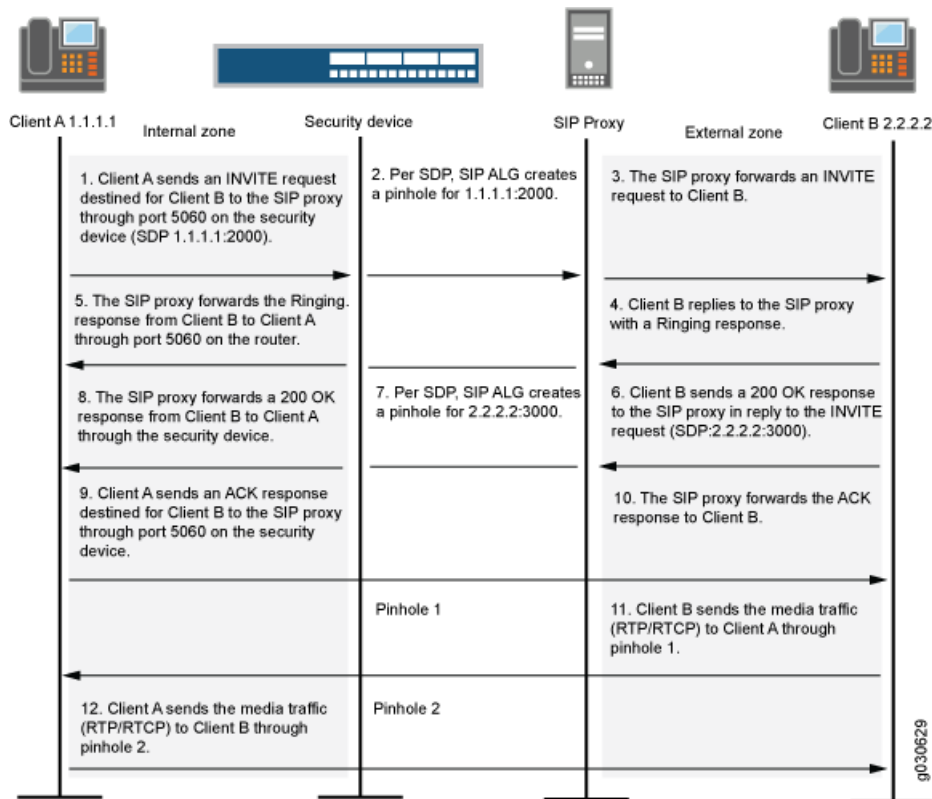
RTP port number + one

- Lifetime—This value indicates the length of time (in seconds) during which a pinhole is open to allow a packet through. A packet must go through the pinhole before the lifetime expires. When the lifetime expires, the SIP ALG removes the pinhole.

When a packet goes through the pinhole within the lifetime period, immediately afterwards the SIP ALG removes the pinhole for the direction from which the packet came.

Figure 18 on page 273 describes a call setup between two SIP clients and how the SIP ALG creates pinholes to allow RTP and RTCP traffic. The illustration assumes that the device has a policy that permits SIP, thus opening port 5060 for SIP signaling messages.

Figure 18: SIP ALG Call Setup



**NOTE:** The SIP ALG does not create pinholes for RTP and RTCP traffic when the destination IP address is 0.0.0.0, which indicates that the session is on hold. To put a session on hold during a telephone communication, for example, user A sends user B a SIP message in which the destination IP address is 0.0.0.0. Doing so indicates to user B that it should not send any media until further notice. If user B sends media anyway, the device drops the packets.

**Related Documentation** • [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

- ALG Overview on page 217
- Understanding SIP ALG Request Methods on page 274
- Understanding SIP ALGs and NAT on page 283
- SIP ALG Configuration Overview on page 275

## Understanding SIP ALG Request Methods

---

The Session Initiation Protocol (SIP) transaction model includes a number of request and response messages, each of which contains a *method* field that denotes the purpose of the message.

Junos OS supports the following method types and response codes:

- INVITE—A user sends an INVITE request to invite another user to participate in a session. The body of an INVITE request can contain the description of the session.
- ACK—The user from whom the INVITE originated sends an ACK request to confirm reception of the final response to the INVITE request. If the original INVITE request did not contain the session description, the ACK request must include it.
- OPTIONS—The User Agent (UA) obtains information about the capabilities of the SIP proxy. A server responds with information about what methods, session description protocols, and message encoding it supports.
- BYE—A user sends a BYE request to abandon a session. A BYE request from either user automatically terminates the session.
- CANCEL—A user sends a CANCEL request to cancel a pending INVITE request. A CANCEL request has no effect if the SIP server processing the INVITE had sent a final response for the INVITE before it received the CANCEL.
- REGISTER—A user sends a REGISTER request to a SIP registrar server to inform it of the current location of the user. A SIP registrar server records all the information it receives in REGISTER requests and makes this information available to any SIP server attempting to locate a user.
- Info—Used to communicate mid-session signaling information along the signaling path for the call.
- Subscribe—Used to request current state and state updates from a remote node.
- Notify—Sent to inform subscribers of changes in state to which the subscriber has a subscription.
- Refer—Used to refer the recipient (identified by the Request-URI) to a third party by the contact information provided in the request.

For example, if user A in a private network refers user B, in a public network, to user C, who is also in the private network, the SIP Application Layer Gateway (ALG) allocates a new IP address and port number for user C so that user C can be contacted by user B. If user C is registered with a registrar, however, its port mapping is stored in the ALG Network Address Translation (NAT) table and is reused to perform the translation.



- Update—Used to open pinhole for new or updated SDP information. The Via:, From:, To:, Call-ID:, Contact:, Route:, and Record-Route: header fields are modified.
- 1xx, 202, 2xx, 3xx, 4xx, 5xx, 6xx Response Codes—Used to indicate the status of a transaction. Header fields are modified.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [ALG Overview on page 217](#)
- [Understanding SIP ALGs on page 269](#)

## SIP ALG Configuration Overview

The Session Initiation Protocol Application Layer Gateway (SIP ALG) is enabled by default on the device—no action is required to enable it. However, you might choose to fine-tune SIP ALG operations by using the following instructions:

1. Control SIP call activity. For instructions, see “Example: Setting SIP ALG Call Duration and Timeouts” on page 276.
2. Protect the SIP proxy server from denial-of-service (DoS) flood attacks. For instructions, see “Example: Configuring SIP ALG DoS Attack Protection” on page 278.
3. Enable unknown messages to pass when the session is in Network Address Translation (NAT) mode and route mode. For instructions, see “Example: Allowing Unknown SIP ALG Message Types” on page 280.
4. Accommodate proprietary SIP call flows. For instructions, see:
  - [Retaining SIP ALG Hold Resources \(J-Web Procedure\) on page 282](#)
  - [Retaining SIP ALG Hold Resources \(CLI Procedure\) on page 282](#)

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding SIP ALGs on page 269](#)
- [Understanding SIP ALGs and NAT on page 283](#)
- [Verifying SIP ALG Configurations on page 321](#)

## SIP ALG Call Duration and Timeouts

- [Understanding SIP ALG Call Duration and Timeouts on page 275](#)
- [Example: Setting SIP ALG Call Duration and Timeouts on page 276](#)

### Understanding SIP ALG Call Duration and Timeouts

The call duration and timeout features give you control over Session Initiation Protocol (SIP) call activity and help you to manage network resources.

Typically a call ends when one of the clients sends a BYE or CANCEL request. The SIP Application Layer Gateway (ALG) intercepts the BYE or CANCEL request and removes all media sessions for that call. There could be reasons or problems preventing clients in a call from sending BYE or CANCEL requests, for example, a power failure. In this case, the call might go on indefinitely, consuming resources on the device.

A call can have one or more voice channels. Each voice channel has two sessions (or two media streams), one for Real-Time Transport Protocol (RTP) traffic and one for Real-Time Control Protocol (RTCP) signaling. When managing the sessions, the device considers the sessions in each voice channel as one group. Timeouts and call duration settings apply to a group as opposed to each session.

The following parameters govern SIP call activity:

- **inactive-media-timeout**—This parameter indicates the maximum length of time (in seconds) a call can remain active without any media (RTP or RTCP) traffic within a group. Each time an RTP or RTCP packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the temporary openings (pinholes) in the firewall the SIP ALG opened for media are closed. The default setting is 120 seconds, and the range is 10 through 2550 seconds. Note that upon timeout, while resources for media (sessions and pinholes) are removed, the call is not terminated.
- **maximum-call-duration**—This parameter sets the absolute maximum length of a call. When a call exceeds this parameter setting, the SIP ALG tears down the call and releases the media sessions. The default setting is 43200 seconds, and the range is 180 through 432000 seconds.
- **t1-interval**—This parameter specifies the roundtrip time estimate, in seconds, of a transaction between endpoints. The default is 500 milliseconds. Because many SIP timers scale with the t1-interval (as described in RFC 3261), when you change the value of the t1-interval timer, those SIP timers also are adjusted.
- **t4-interval**—This parameter specifies the maximum time a message remains in the network. The default is 5 seconds and the range is 5 through 10 seconds. Because many SIP timers scale with the t4-interval (as described in RFC 3261), when you change the value of the t4-interval timer, those SIP timers also are adjusted.
- **c-timeout**—This parameter specifies the INVITE transaction timeout at the proxy, in minutes; the default is 3. Because the SIP ALG is in the middle, instead of using the INVITE transaction timer value B (which is  $(64 * T1) = 32$  seconds), the SIP ALG gets its timer value from the proxy.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding SIP ALGs on page 269](#)
- [SIP ALG Configuration Overview on page 275](#)
- [Example: Setting SIP ALG Call Duration and Timeouts on page 276](#)

### Example: Setting SIP ALG Call Duration and Timeouts

This example shows how to set the call duration and the media inactivity timeout.

## Requirements

Before you begin, review the call duration and timeout features used to control SIP call activity. See “Understanding SIP ALG Call Duration and Timeouts” on page 275.

## Overview

The call duration and inactivity media timeout features help you to conserve network resources and maximize throughput.

The **maximum-call-duration** parameter sets the maximum allowable length of time a call can be active. When the duration is exceeded, the SIP ALG tears down the call and releases the media sessions. This setting also frees up bandwidth in cases where calls fail to properly terminate.

The **inactive-media-timeout** parameter indicates the maximum length of time (in seconds) a call can remain active without any media (RTP or RTPC) traffic within a group. Each time an RTP or RTCP packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the SIP ALG temporary openings (pinholes) for media in the firewall are closed. The default setting is 120 seconds, and the range is 10 through 2550 seconds. Upon timeout, while resources for media (sessions and pinholes) are removed, the call is not terminated.

In this example, the call duration is set to 180000 seconds and the media inactivity timeout is set to 90 seconds.

## Configuration

### J-Web Quick Configuration

To set the SIP ALG call duration and the media inactivity timeout:

1. Select **Configure>Security>ALG**.
2. Select the **SIP** tab.
3. In the Maximum call duration field, type **3000**.
4. In the Inactive media timeout field, enter **90**.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

### Step-by-Step Procedure

To set the SIP ALG call duration and the media inactivity timeout:

1. Configure the SIP ALG call duration.
 

```
[edit]
user@host# set security alg sip maximum-call-duration 3000
```
2. Configure the SIP ALG inactivity media timeout.
 

```
[edit]
user@host# set security alg sip inactive-media-timeout 90
```
3. If you are done configuring the device, commit the configuration.
 

```
[edit]
```

```
user@host# commit
```

### Verification

---

To verify the configuration is working properly, enter the **show security alg sip** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- SIP ALG Configuration Overview on page 275
- Verifying SIP ALG Configurations on page 321

## SIP ALG DoS Attack Protection

---

- Understanding SIP ALG DoS Attack Protection on page 278
- Example: Configuring SIP ALG DoS Attack Protection on page 278

### Understanding SIP ALG DoS Attack Protection

The ability of the Session Initiation Protocol (SIP) proxy server to process calls can be impacted by repeat SIP INVITE requests—requests that it initially denied. The denial-of-service (DoS) protection feature enables you to configure the device to monitor INVITE requests and proxy server replies to them. If a reply contains a 3xx, 4xx, or 5xx response code (see “Classes of SIP Responses” on page 290), the ALG stores the source IP address of the request and the IP address of the proxy server in a table. Subsequently, the device checks all INVITE requests against this table and, for a configurable number of seconds (the default is 3), discards any packets that match entries in the table. You can configure the device to monitor and deny repeat INVITE requests to all proxy servers, or you can protect a specific proxy server by specifying the destination IP address. SIP attack protection is configured globally.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding SIP ALGs on page 269
- SIP ALG Configuration Overview on page 275
- Example: Configuring SIP ALG DoS Attack Protection on page 278

### Example: Configuring SIP ALG DoS Attack Protection

This example shows how to configure the DoS attack protection feature.

#### Requirements

---

Before you begin, review the DoS attack protection feature used to control SIP call activity. See “Understanding SIP ALG DoS Attack Protection” on page 278.

#### Overview

---

The ability of the SIP proxy server to process calls can be impacted by repeat SIP INVITE requests—requests that the server initially denied. The DoS protection feature enables you to configure the device to monitor INVITE requests and proxy server replies to them.

In this example, the device is configured to protect a single SIP proxy server (1.1.1.3) from repeat INVITE requests to which it has already been denied service. Packets are dropped for a period of 5 seconds, after which the device resumes forwarding INVITE requests from those sources.

### Configuration

---

#### J-Web Quick Configuration

To configure SIP ALG DoS attack protection:

1. Select **Configure>Security>ALG**.
2. Select the **SIP** tab.
3. In the Enable attack protection area, click the **Selected servers** option.
4. In the Destination IP box, enter **1.1.1.3** and click **Add**.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

#### Step-by-Step Procedure

To configure SIP ALG DoS attack protection:

1. Configure the device to protect a single SIP proxy server.
 

```
[edit]
user@host# set security alg sip application-screen protect deny destination-ip 1.1.1.3
```
2. Configure the device for the deny timeout period.
 

```
[edit]
user@host# set security alg sip application-screen protect deny timeout 5
```
3. If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

### Verification

---

To verify the configuration is working properly, enter the **show security alg sip** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [SIP ALG Configuration Overview on page 275](#)
- [Verifying SIP ALG Configurations on page 321](#)

## SIP ALG Unknown Message Types

---

- [Understanding SIP ALG Unknown Message Types on page 280](#)
- [Example: Allowing Unknown SIP ALG Message Types on page 280](#)

## Understanding SIP ALG Unknown Message Types

This feature enables you to specify how unidentified Session Initiation Protocol (SIP) messages are handled by the device. The default is to drop unknown (unsupported) messages.

We do not recommend permitting unknown messages because they can compromise security. However, in a secure test or production environment, this command can be useful for resolving interoperability issues with disparate vendor equipment. Permitting unknown SIP messages can help you get your network operational so you can later analyze your voice-over-IP (VoIP) traffic to determine why some messages were being dropped. The unknown SIP message type feature enables you to configure the device to accept SIP traffic containing unknown message types in both Network Address Translation (NAT) mode and route mode.



**NOTE:** This option applies only to received packets identified as supported VoIP packets. If a packet cannot be identified, it is always dropped. If a packet is identified as a supported protocol and you have configured the device to permit unknown message types, the message is forwarded without processing.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding SIP ALGs on page 269](#)
- [SIP ALG Configuration Overview on page 275](#)
- [Example: Allowing Unknown SIP ALG Message Types on page 280](#)

## Example: Allowing Unknown SIP ALG Message Types

This example shows how to allow unknown message types.

### Requirements

Before you begin, review how unidentified SIP messages are handled by the device. See “Understanding SIP ALG Unknown Message Types” on page 280.

### Overview

In this example, you configure the device to allow unknown message types in SIP traffic in both NAT mode and route mode. The default is to drop unknown (unsupported) messages.

### Configuration

#### J-Web Quick Configuration

To allow unknown SIP ALG message types:

1. Select **Configure>Security>ALG**.
2. Select the **SIP** tab.

3. Select the **Enable Permit NAT applied** check box.
4. Select the **Enable Permit routed** check box.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

### Step-by-Step Procedure

To allow unknown SIP ALG message types:

1. Configure the device to allow unknown message types in SIP traffic.
 

```
[edit]
user@host# set security alg sip application-screen unknown-message
permit-nat-applied permit-routed
```
2. If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security alg sip** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [SIP ALG Configuration Overview on page 275](#)
- [Verifying SIP ALG Configurations on page 321](#)

## SIP ALG Hold Resources

- [Understanding SIP ALG Hold Resources on page 281](#)
- [Retaining SIP ALG Hold Resources \(J-Web Procedure\) on page 282](#)
- [Retaining SIP ALG Hold Resources \(CLI Procedure\) on page 282](#)

## Understanding SIP ALG Hold Resources

When a user puts a call on hold, the Session Initiation Protocol Application Layer Gateway (SIP ALG) releases Session Description Protocol (SDP) media resources, such as pinholes and translation contexts. When the user resumes the call, an INVITE request message negotiates a new SDP offer and answer and the SIP ALG reallocates resources for the media stream. This can result in new translated IP address and port numbers for the media description even when the media description is the same as the previous description. This is compliant with *RFC 3264 An Offer/Answer Model with the Session Description Protocol (SDP)*.

Some proprietary SIP implementations have designed call flows so that the User Agent (UA) module ignores the new SDP INVITE offer and continues to use the SDP offer of the previous negotiation. To accommodate this functionality, you must configure the device to retain SDP media resources when a call is put on hold for reuse when the call is resumed.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding SIP ALGs on page 269
  - SIP ALG Configuration Overview on page 275
  - Retaining SIP ALG Hold Resources (J-Web Procedure) on page 282
  - Retaining SIP ALG Hold Resources (CLI Procedure) on page 282

### Retaining SIP ALG Hold Resources (J-Web Procedure)

To accommodate proprietary SIP call flows:

1. Select **Configure>Security>ALG**.
2. Select the **SIP** tab.
3. Select the **Enable retail hold resource** check box.
4. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding SIP ALG Hold Resources on page 281
  - SIP ALG Configuration Overview on page 275
  - Retaining SIP ALG Hold Resources (CLI Procedure) on page 282
  - Verifying SIP ALG Configurations on page 321

### Retaining SIP ALG Hold Resources (CLI Procedure)

To accommodate proprietary SIP call flows:

```
user@host# set security alg sip retain-hold-resource
```

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding SIP ALG Hold Resources on page 281
  - SIP ALG Configuration Overview on page 275
  - Retaining SIP ALG Hold Resources (J-Web Procedure) on page 282
  - Verifying SIP ALG Configurations on page 321

## SIP ALGs and NAT

---

- Understanding SIP ALGs and NAT on page 283
- Understanding Incoming SIP ALG Call Support Using the SIP Registrar and NAT on page 292
- Example: Configuring Interface Source NAT for Incoming SIP Calls on page 293



- Example: Configuring a Source NAT Pool for Incoming SIP Calls on page 298
- Example: Configuring Static NAT for Incoming SIP Calls on page 304
- Example: Configuring the SIP Proxy in the Private Zone and NAT in the Public Zone on page 309
- Example: Configuring a Three-Zone SIP ALG and NAT Scenario on page 314

## Understanding SIP ALGs and NAT

The Network Address Translation (NAT) protocol enables multiple hosts in a private subnet to share a single public IP address to access the Internet. For outgoing traffic, NAT replaces the private IP address of the host in the private subnet with the public IP address. For incoming traffic, the public IP address is converted back into the private address, and the message is routed to the appropriate host in the private subnet.

Using NAT with the Session Initiation Protocol (SIP) service is more complicated because SIP messages contain IP addresses in the SIP headers as well as in the SIP body. When using NAT with the SIP service, the SIP headers contain information about the caller and the receiver, and the device translates this information to hide it from the outside network. The SIP body contains the Session Description Protocol (SDP) information, which includes IP addresses and port numbers for transmission of the media. The device translates SDP information for allocating resources to send and receive the media.

How IP addresses and port numbers in SIP messages are replaced depends on the direction of the message. For an outgoing message, the private IP address and port number of the client are replaced with the public IP address and port number of the Juniper Networks firewall. For an incoming message, the public address of the firewall is replaced with the private address of the client.

When an INVITE message is sent out across the firewall, the SIP Application Layer Gateway (ALG) collects information from the message header into a call table, which it uses to forward subsequent messages to the correct endpoint. When a new message arrives, for example an ACK or 200 OK, the ALG compares the “From:”, “To:”, and “Call-ID:” fields against the call table to identify the call context of the message. If a new INVITE message arrives that matches the existing call, the ALG processes it as a REINVITE.

When a message containing SDP information arrives, the ALG allocates ports and creates a NAT mapping between them and the ports in the SDP. Because the SDP requires sequential ports for the Real-Time Transport Protocol (RTP) and Real-Time Control Protocol (RTCP) channels, the ALG provides consecutive even-odd ports. If it is unable to find a pair of ports, it discards the SIP message.

This topic contains the following sections:

- Outgoing Calls on page 284
- Incoming Calls on page 284
- Forwarded Calls on page 285
- Call Termination on page 285
- Call Re-INVITE Messages on page 285
- Call Session Timers on page 285

- Call Cancellation on page 285
- Forking on page 286
- SIP Messages on page 286
- SIP Headers on page 286
- SIP Body on page 288
- SIP NAT Scenario on page 288
- Classes of SIP Responses on page 290

### Outgoing Calls

---

When a SIP call is initiated with a SIP request message from the internal to the external network, NAT replaces the IP addresses and port numbers in the SDP and binds the IP addresses and port numbers to the Juniper Networks firewall. Via, Contact, Route, and Record-Route SIP header fields, if present, are also bound to the firewall IP address. The ALG stores these mappings for use in retransmissions and for SIP response messages.

The SIP ALG then opens pinholes in the firewall to allow media through the device on the dynamically assigned ports negotiated based on information in the SDP and the Via, Contact, and Record-Route header fields. The pinholes also allow incoming packets to reach the Contact, Via, and Record-Route IP addresses and ports. When processing return traffic, the ALG inserts the original Contact, Via, Route, and Record-Route SIP fields back into packets.

### Incoming Calls

---

Incoming calls are initiated from the public network to public static NAT addresses or to interface IP addresses on the device. Static NATs are statically configured IP addresses that point to internal hosts; interface IP addresses are dynamically recorded by the ALG as it monitors REGISTER messages sent by internal hosts to the SIP registrar. When the device receives an incoming SIP packet, it sets up a session and forwards the payload of the packet to the SIP ALG.

The ALG examines the SIP request message (initially an INVITE) and, based on information in the SDP, opens gates for outgoing media. When a 200 OK response message arrives, the SIP ALG performs NAT on the IP addresses and ports and opens pinholes in the outbound direction. (The opened gates have a short time-to-live, and they time out if a 200 OK response message is not received quickly.)

When a 200 OK response arrives, the SIP proxy examines the SDP information and reads the IP addresses and port numbers for each media session. The SIP ALG on the device performs NAT on the addresses and port numbers, opens pinholes for outbound traffic, and refreshes the timeout for gates in the inbound direction.

When the ACK arrives for the 200 OK, it also passes through the SIP ALG. If the message contains SDP information, the SIP ALG ensures that the IP addresses and port numbers are not changed from the previous INVITE—if they are, the ALG deletes old pinholes and creates new pinholes to allow media to pass through. The ALG also monitors the Via, Contact, and Record-Route SIP fields and opens new pinholes if it determines that these fields have changed.

---

### Forwarded Calls

A forwarded call is when, for example, user A outside the network calls user B inside the network, and user B forwards the call to user C outside the network. The SIP ALG processes the INVITE from user A as a normal incoming call. But when the ALG examines the forwarded call from B to C outside the network and notices that B and C are reached using the same interface, it does not open pinholes in the firewall, because media will flow directly between user A and user C.

---

### Call Termination

The BYE message terminates a call. When the device receives a BYE message, it translates the header fields just as it does for any other message. But because a BYE message must be acknowledged by the receiver with a 200 OK, the ALG delays call teardown for five seconds to allow time for transmission of the 200 OK.

---

### Call Re-INVITE Messages

Re-INVITE messages add new media sessions to a call and remove existing media sessions. When new media sessions are added to a call, new pinholes are opened in the firewall and new address bindings are created. The process is identical to the original call setup. When one or more media sessions are removed from a call, pinholes are closed and bindings released just as with a BYE message.

---

### Call Session Timers

The SIP ALG uses the Session-Expires value to time out a session if a Re-INVITE or UPDATE message is not received. The ALG gets the Session-Expires value, if present, from the 200 OK response to the INVITE and uses this value for signaling timeout. If the ALG receives another INVITE before the session times out, it resets all timeout values to this new INVITE or to default values, and the process is repeated.

As a precautionary measure, the SIP ALG uses hard timeout values to set the maximum amount of time a call can exist. This ensures that the device is protected should one of the following events occur:

- End systems crash during a call and a BYE message is not received.
- Malicious users never send a BYE in an attempt to attack a SIP ALG.
- Poor implementations of SIP proxy fail to process Record-Route and never send a BYE message.
- Network failures prevent a BYE message from being received.

---

### Call Cancellation

Either party can cancel a call by sending a CANCEL message. Upon receiving a CANCEL message, the SIP ALG closes pinholes through the firewall—if any have been opened—and releases address bindings. Before releasing the resources, the ALG delays the control channel age-out for approximately five seconds to allow time for the final 200 OK to pass through. The call is terminated when the five second timeout expires, regardless of whether a 487 or non-200 response arrives.

## Forking

---

Forking enables a SIP proxy to send a single INVITE message to multiple destinations simultaneously. When the multiple 200 OK response messages arrive for the single call, the SIP ALG parses but updates call information with the first 200 OK messages it receives.

## SIP Messages

---

The SIP message format consists of a SIP header section and the SIP body. In request messages, the first line of the header section is the request line, which includes the method type, request-URI, and protocol version. In response messages, the first line is the status line, which contains a status code. SIP headers contain IP addresses and port numbers used for signaling. The SIP body, separated from the header section by a blank line, is reserved for session description information, which is optional. Junos OS currently supports the SDP only. The SIP body contains IP addresses and port numbers used to transport the media.

## SIP Headers

---

In the following sample SIP request message, NAT replaces the IP addresses in the header fields to hide them from the outside network.

```
INVITE bob@10.150.20.5 SIP/2.0
Via: SIP/2.0/UDP 10.150.20.3:5434
From: alice@10.150.20.3
To: bob@10.150.20.5
Call-ID: a12abcde@10.150.20.3
Contact: alice@10.150.20.3:5434
Route: <sip:netscreen@10.150.20.3:5060>
Record-Route: <sip:netscreen@10.150.20.3:5060>
```

How IP address translation is performed depends on the type and direction of the message. A message can be any of the following:

- Inbound request
- Outbound response
- Outbound request
- Inbound response

Table 31 on page 287 shows how NAT is performed in each of these cases. Note that for several of the header fields the ALG determine more than just whether the messages comes from inside or outside the network. It must also determine what client initiated the call, and whether the message is a request or response.

Table 31: Requesting Messages with NAT Table

|                                               |               |                                        |
|-----------------------------------------------|---------------|----------------------------------------|
| Inbound Request<br>(from public to private)   | To:           | Replace domain with local address      |
|                                               | From:         | None                                   |
|                                               | Call-ID:      | None                                   |
|                                               | Via:          | None                                   |
|                                               | Request-URI:  | Replace ALG address with local address |
|                                               | Contact:      | None                                   |
|                                               | Record-Route: | None                                   |
|                                               | Route:        | None                                   |
| Outbound Response<br>(from private to public) | To:           | Replace ALG address with local address |
|                                               | From:         | None                                   |
|                                               | Call-ID:      | None                                   |
|                                               | Via:          | None                                   |
|                                               | Request-URI:  | N/A                                    |
|                                               | Contact:      | Replace local address with ALG address |
|                                               | Record-Route: | Replace local address with ALG address |
|                                               | Route:        | None                                   |
| Outbound Request<br>(from private to public)  | To:           | None                                   |
|                                               | From:         | Replace local address with ALG address |
|                                               | Call-ID:      | Replace local address with ALG address |
|                                               | Via:          | Replace local address with ALG address |
|                                               | Request-URI:  | None                                   |
|                                               | Contact:      | Replace local address with ALG address |
|                                               | Record-Route: | Replace local address with ALG address |
|                                               | Route:        | Replace ALG address with local address |

Table 31: Requesting Messages with NAT Table (*continued*)

|                                               |               |                                        |
|-----------------------------------------------|---------------|----------------------------------------|
| Outbound Response<br>(from public to private) | To:           | None                                   |
|                                               | From:         | Replace ALG address with local address |
|                                               | Call-ID:      | Replace ALG address with local address |
|                                               | Via:          | Replace ALG address with local address |
|                                               | Request-URI:  | N/A                                    |
|                                               | Contact:      | None                                   |
|                                               | Record-Route: | Replace ALG address with local address |
|                                               | Route:        | Replace ALG address with local address |

### SIP Body

The SDP information in the SIP body includes IP addresses the ALG uses to create channels for the media stream. Translation of the SDP section also allocates resources, that is, port numbers to send and receive the media.

The following excerpt from a sample SDP section shows the fields that are translated for resource allocation.

```
o=user 2344234 55234434 IN IP4 10.150.20.3
c=IN IP4 10.150.20.3
m=audio 43249 RTP/AVP 0
```

SIP messages can contain more than one media stream. The concept is similar to attaching multiple files to an e-mail message. For example, an INVITE message sent from a SIP client to a SIP server might have the following fields:

```
c=IN IP4 10.123.33.4
m=audio 33445 RTP/AVP 0
c=IN IP4 10.123.33.4
m=audio 33447 RTP/AVP 0
c=IN IP4 10.123.33.4
m=audio 33449 RTP/AVP 0
```

Junos OS supports up to 6 SDP channels negotiated for each direction, for a total of 12 channels per call. For more information, see “SDP Session Descriptions” on page 271.

### SIP NAT Scenario

Figure 19 on page 289 and Figure 20 on page 290 show a SIP call INVITE and 200 OK. In Figure 19 on page 289, ph1 sends a SIP INVITE message to ph2. Note how the IP addresses in the header fields—shown in bold font—are translated by the device.

The SDP section of the INVITE message indicates where the caller is willing to receive media. Note that the Media Pinhole contains two port numbers, 52002 and 52003, for RTCP and RTP. The Via/Contact Pinhole provides port number 5060 for SIP signaling.

Observe how, in the 200 OK response message in Figure 20 on page 290, the translations performed in the INVITE message are reversed. The IP addresses in this message, being public, are not translated, but gates are opened to allow the media stream access to the private network.

Figure 19: SIP NAT Scenario 1

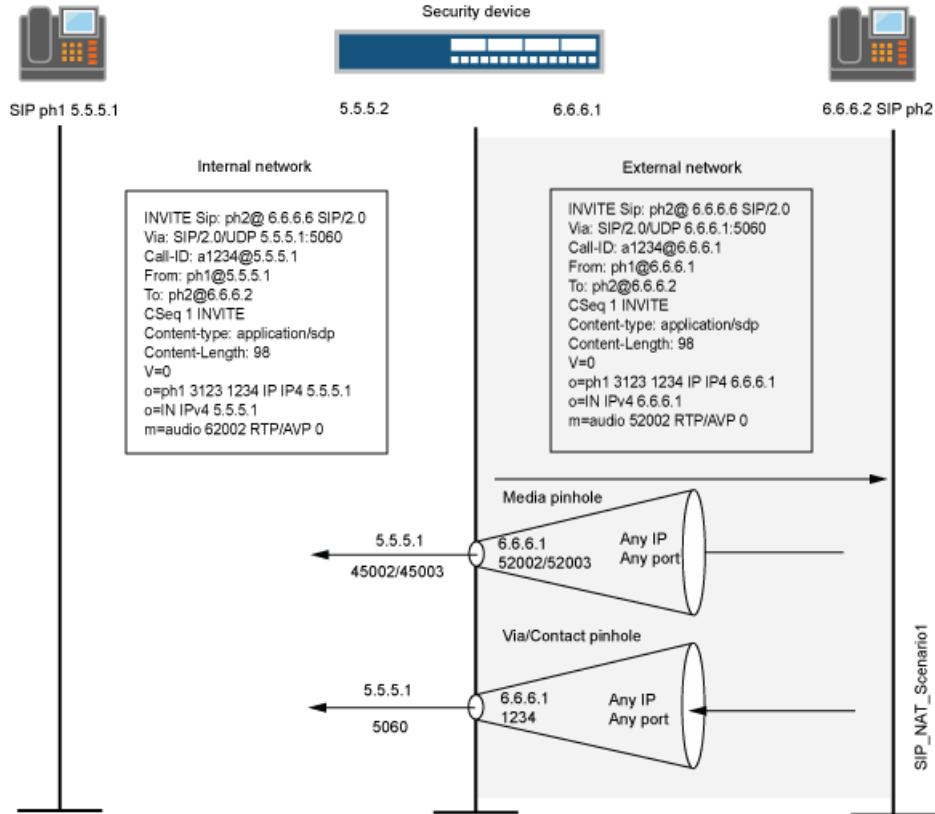
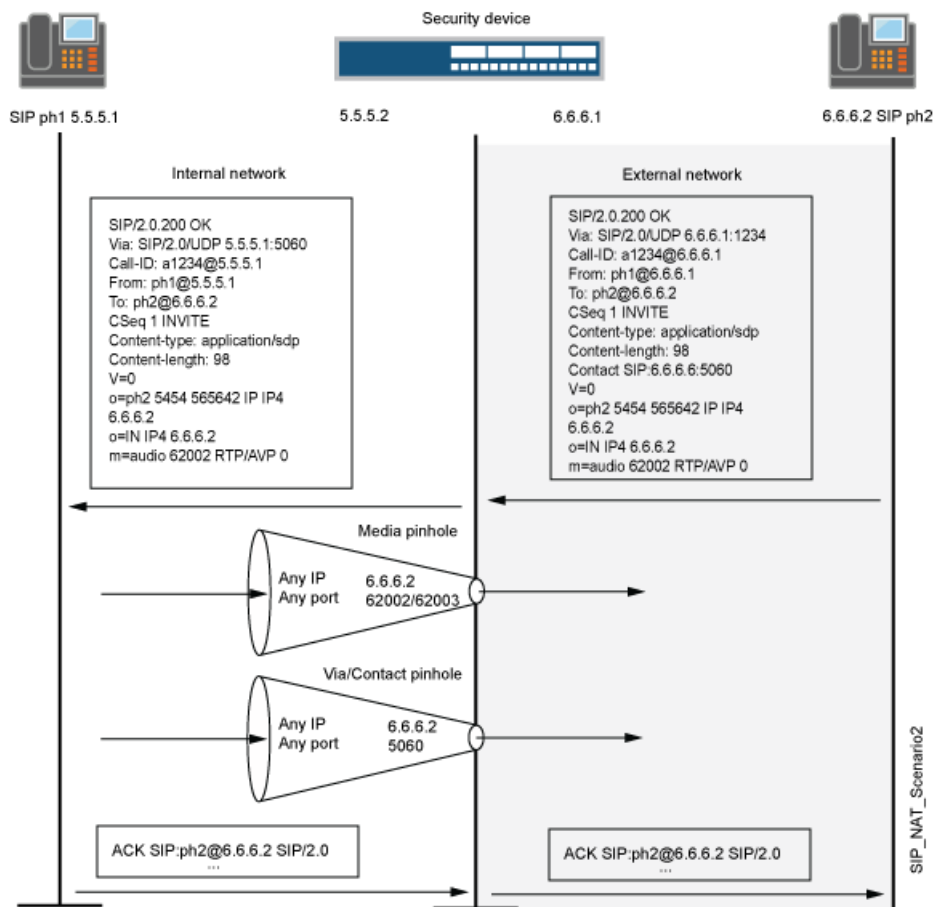


Figure 20: SIP NAT Scenario 2



### Classes of SIP Responses

SIP responses provide status information about SIP transactions and include a response code and a reason phrase. SIP responses are grouped into the following classes:

- Informational (100 to 199)—Request received, continuing to process the request.
- Success (200 to 299)—Action successfully received, understood, and accepted.
- Redirection (300 to 399)—Further action required to complete the request.
- Client Error (400 to 499)—Request contains bad syntax or cannot be fulfilled at this server.
- Server Error (500 to 599)—Server failed to fulfill an apparently valid request.
- Global Failure (600 to 699)—Request cannot be fulfilled at any server.

Table 32 on page 291 provides a complete list of current SIP responses.



Table 32: SIP Responses

|                         |                              |                                   |                                         |
|-------------------------|------------------------------|-----------------------------------|-----------------------------------------|
| Informational           | 100 Trying                   | 180 Ringing                       | 181 Call is being forwarded             |
|                         | 182 Queued                   | 183 Session progress              |                                         |
| Success                 | 200 OK                       | 202 Accepted                      |                                         |
| Redirection             | 300 Multiple choices         | 301 Moved permanently             | 302 Moved temporarily                   |
|                         | 305 Use proxy                | 380 Alternative service           |                                         |
| Client Error            | 400 Bad request              | 401 Unauthorized                  | 402 Payment required                    |
|                         | 403 Forbidden                | 404 Not found                     | 405 Method not allowed                  |
|                         | 406 Not acceptable           | 407 Proxy authentication required | 408 Request time-out                    |
|                         | 409 Conflict                 | 410 Gone                          | 411 Length required                     |
|                         | 413 Request entity too large | 414 Request URL too large         | 415 Unsupported media type              |
|                         | 420 Bad extension            | 480 Temporarily not available     | 481 Call leg/transaction does not exist |
|                         | 482 Loop detected            | 483 Too many hops                 | 484 Address incomplete                  |
|                         | 485 Ambiguous                | 486 Busy here                     | 487 Request canceled                    |
|                         | 488 Not acceptable here      |                                   |                                         |
|                         | Server Error                 | 500 Server internal error         | 501 Not implemented                     |
| 502 Service unavailable |                              | 504 Gateway time-out              | 505 SIP version not supported           |
| Global Failure          | 600 Busy everywhere          | 603 Decline                       | 604 Does not exist anywhere             |
|                         | 606 Not acceptable           |                                   |                                         |

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding SIP ALGs on page 269
- Understanding Incoming SIP ALG Call Support Using the SIP Registrar and NAT on page 292
- Example: Configuring Interface Source NAT for Incoming SIP Calls on page 293
- Example: Configuring a Source NAT Pool for Incoming SIP Calls on page 298
- Example: Configuring Static NAT for Incoming SIP Calls on page 304

- Example: Configuring the SIP Proxy in the Private Zone and NAT in the Public Zone on page 309
- Example: Configuring a Three-Zone SIP ALG and NAT Scenario on page 314

## Understanding Incoming SIP ALG Call Support Using the SIP Registrar and NAT

Session Initiation Protocol (SIP) registration provides a discovery capability by which SIP proxies and location servers can identify the location or locations where users want to be contacted. A user registers one or more contact locations by sending a REGISTER message to the registrar. The To and Contact fields in the REGISTER message contain the address-of-record Uniform Resource Identifier (URI) and one or more contact URIs, as shown in Figure 21 on page 293. Registration creates bindings in a location service that associates the address-of-record with the contact address or addresses.

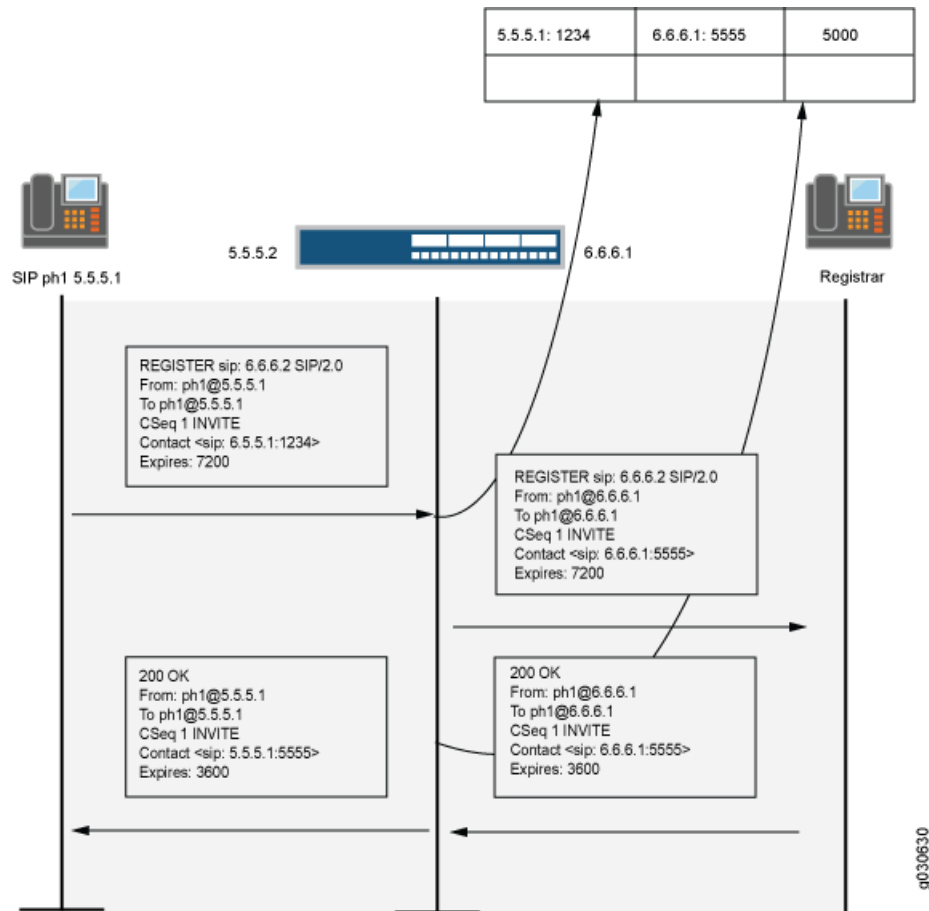
The device monitors outgoing REGISTER messages, performs Network Address Translation (NAT) on these addresses, and stores the information in an Incoming NAT table. Then, when an INVITE message is received from outside the network, the device uses the Incoming NAT table to identify which internal host to route the INVITE message to. You can take advantage of SIP proxy registration service to allow incoming calls by configuring interface source NAT or NAT pools on the egress interface of the device. Interface source NAT is adequate for handling incoming calls in a small office, whereas we recommend setting up source NAT pools for larger networks or an enterprise environment.



**NOTE:** Incoming call support using interface source NAT or a source NAT pool is supported for SIP and H.323 services only. For incoming calls, Junos OS currently supports UDP and TCP only. Domain name resolution is also currently not supported; therefore, URIs must contain IP addresses, as shown in Figure 21 on page 293.

---

Figure 21: Using the SIP Registrar

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [ALG Overview](#) on page 217
- [Understanding SIP ALGs and NAT](#) on page 283
- [SIP ALG Configuration Overview](#) on page 275

**Example: Configuring Interface Source NAT for Incoming SIP Calls**

This example shows how to configure a source NAT rule on a public zone interface allowing NAT to be used for incoming SIP calls.

- [Requirements](#) on page 294
- [Overview](#) on page 294
- [Configuration](#) on page 295
- [Verification](#) on page 298

## Requirements

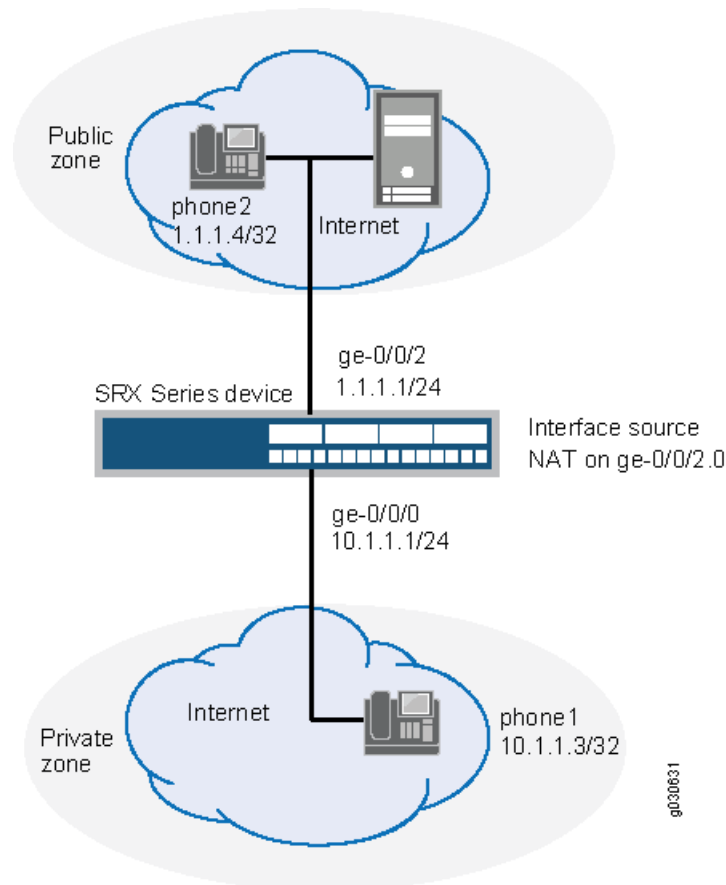
Before you begin, understand how NAT works with the SIP ALG. See “Understanding SIP ALGs and NAT” on page 283.

## Overview

In a two-zone scenario with the SIP proxy server in an external zone, you can use NAT for incoming calls by configuring a source NAT rule on the interface in the public or external zone.

In this example (see Figure 22 on page 294), phone1 is on the ge-0/0/0 interface in the private zone, and phone2 and the proxy server are on the ge-0/0/2 interface in the public zone. You configure a source NAT rule on the public interface ge-0/0/2.0.

**Figure 22: Source NAT for Incoming SIP Calls**



In this example, after creating zones called private and public and assigning them to interfaces, you configure address books to be used in the source NAT rule set. Then you configure source NAT by defining a rule set called sip-phones and a rule called phone1 that matches any packets from the source address 10.1.1.3/32.

Finally, you create security policies to allow all SIP traffic between the private and public zones.

## Configuration

### CLI Quick Configuration

To quickly configure a source NAT rule on a public zone interface, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces ge-0/0/2 unit 0 family inet address 1.1.1.1/24
set security zones security-zone private address-book address phone1 10.1.1.3/32
set security zones security-zone private interfaces ge-0/0/0.0
set security zones security-zone public address-book address proxy 1.1.1.3/32
set security zones security-zone public address-book address phone2 1.1.1.4/32
set security zones security-zone public interfaces ge-0/0/2.0
set security nat source rule-set sip-phones from zone private
set security nat source rule-set sip-phones to zone public
set security nat source rule-set sip-phones rule phone1 match source-address 10.1.1.3/32
set security nat source rule-set sip-phones rule phone1 then source-nat interface
set security policies from-zone private to-zone public policy outgoing match
  source-address phone1
set security policies from-zone private to-zone public policy outgoing match
  destination-address phone2
set security policies from-zone private to-zone public policy outgoing match
  destination-address proxy
set security policies from-zone private to-zone public policy outgoing match application
  junos-sip
set security policies from-zone private to-zone public policy outgoing then permit
set security policies from-zone public to-zone private policy incoming match
  source-address phone2
set security policies from-zone public to-zone private policy incoming match
  destination-address phone1
set security policies from-zone public to-zone private policy incoming match
  destination-address proxy
set security policies from-zone public to-zone private policy incoming match application
  junos-sip
set security policies from-zone public to-zone private policy incoming then permit
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a source NAT rule on a public zone interface:

1. Configure interfaces.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 1.1.1.1/24
```

2. Configure zones and assign them to the interfaces.

```
[edit security zones]
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone public interfaces ge-0/0/2.0
```

3. Configure address books and create addresses.

```
[edit security zones]
```

```

user@host# set security-zone private address-book address phone1 10.1.1.3/32
user@host# set security-zone public address-book address proxy 1.1.1.3/32
user@host# set security-zone public address-book address phone2 1.1.1.4/32

```

4. Configure a source NAT rule set.

```

[edit security nat source]
user@host# set rule-set sip-phones from zone private
user@host# set rule-set sip-phones to zone public
user@host# set rule-set sip-phones rule phone1 match source-address 10.1.1.3/32
user@host# set rule-set sip-phones rule phone1 then source-nat interface

```

5. Enable persistent source NAT translation.

```

[edit security nat source]
user@host# set address-persistent

```

6. Configure a security policy to allow outgoing SIP traffic.

```

[edit security policies from-zone private to-zone public policy outgoing]
user@host# set match source-address phone1
user@host# set match destination-address phone2
user@host# set match destination-address proxy
user@host# set match application junos-sip
user@host# set then permit

```

7. Configure a security policy to allow incoming SIP traffic.

```

[edit security policies from-zone public to-zone private policy incoming]
user@host# set match source-address phone2
user@host# set match destination-address phone1
user@host# set match destination-address proxy
user@host# set match application junos-sip
user@host# set then permit

```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show security zones**, **show security policies**, and **show security nat** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 10.1.1.1/24;
    }
  }
}
ge-0/0/2 {
  unit 0 {
    family inet {
      address 1.1.1.1/24;
    }
  }
}
[edit]

```

```
user@host# show security zones
security-zone private {
  address-book {
    address phone1 10.1.1.3/32;
  }
  interfaces {
    ge-0/0/0.0;
  }
}
security-zone public {
  address-book {
    address proxy 1.1.1.3/32;
    address phone2 2.2.2.4/32;
  }
  interfaces {
    ge-0/0/2.0;
  }
}
[edit]
user@host# show security nat
source {
  rule-set sip-phones {
    from zone private;
    to zone public;
    rule phone1 {
      match {
        source-address 10.1.1.3/32;
      }
      then {
        source-nat {
          interface;
        }
      }
    }
  }
}
[edit]
user@host# show security policies
from-zone private to-zone public {
  policy outgoing {
    match {
      source-address phone1;
      destination-address [ phone2 proxy ];
      application junos-sip;
    }
    then {
      permit;
    }
  }
}
from-zone public to-zone private {
  policy incoming {
    match {
      source-address phone2;
      destination-address [ phone1 proxy ];
      application junos-sip;
    }
  }
}
```

```

    }
    then {
      permit;
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

---

To confirm that the configuration is working properly, perform these tasks:

- Verifying Source NAT Rule Usage on page 298
- Verifying SIP ALG Status on page 298

#### *Verifying Source NAT Rule Usage*

**Purpose** Verify that there is traffic matching the source NAT rule.

**Action** From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

#### *Verifying SIP ALG Status*

**Purpose** Verify that SIP ALG is enabled on your system.

**Action** From operational mode, enter the **show security alg status** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Verifying SIP ALG Configurations on page 321

## Example: Configuring a Source NAT Pool for Incoming SIP Calls

This example shows how to configure a source NAT pool on an external interface to enable NAT for incoming SIP calls.

- Requirements on page 298
- Overview on page 298
- Configuration on page 300
- Verification on page 303

### Requirements

---

Before you begin, understand how NAT works with the SIP ALG. See “Understanding SIP ALGs and NAT” on page 283.

### Overview

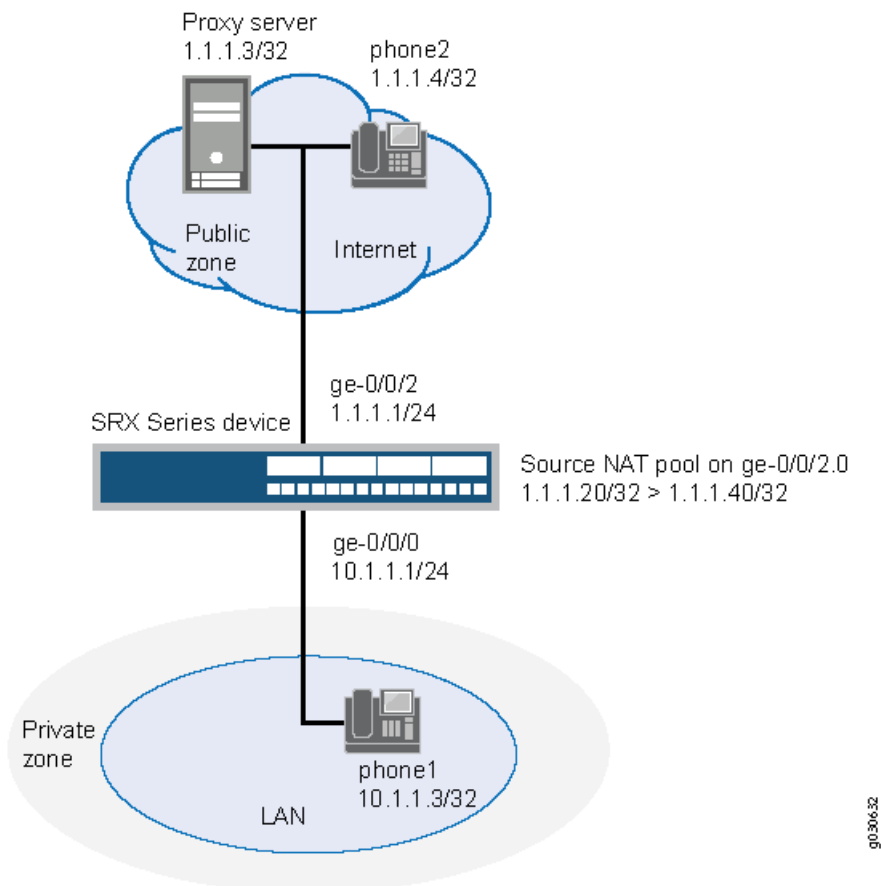
---

In a two-zone scenario with the SIP proxy server in an external or public zone, you can use NAT for incoming calls by configuring a NAT pool on the interface to the public zone.



In this example (see Figure 23 on page 299), phone1 is in the private zone, and phone2 and the proxy server are in the public zone. You configure a source NAT pool to do NAT. You also create a policy that permits SIP traffic from the private to the public zone. This enables phone1 in the private zone to register with the proxy server in the public zone, and it also enables incoming calls from the public zone to the private zone.

Figure 23: Source NAT Pool for Incoming SIP Calls



In this example, you configure source NAT as follows:

- Define source NAT pool called sip-nat-pool to contain the IP address range from 1.1.1.20/32 through 1.1.1.40/32.
- Create a source NAT rule set called sip-nat with a rule sip-r1 to match packets from the private zone to the public zone with the source IP address 10.1.1.3/24. For matching packets, the source address is translated to one of the IP address in sip-nat-pool.
- Configure proxy ARP for the addresses 1.1.1.20/32 through 1.1.1.40/32 on interface ge-0/0/2.0. This allows the system to respond to ARP requests received on the interface for these addresses.

## Configuration

**CLI Quick Configuration** To quickly configure a source NAT pool for incoming SIP calls, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces ge-0/0/2 unit 0 family inet address 1.1.1.1/24
set security zones security-zone private address-book address phone1 10.1.1.3/32
set security zones security-zone private interfaces ge-0/0/0.0
set security zones security-zone public address-book address proxy 1.1.1.3/32
set security zones security-zone public address-book address phone2 1.1.1.4/32
set security zones security-zone public interfaces ge-0/0/2.0
set security nat source pool sip-nat-pool address 1.1.1.20/32 to 1.1.1.40/32
set security nat source address-persistent
set security nat source rule-set sip-nat from zone private
set security nat source rule-set sip-nat to zone public
set security nat source rule-set sip-nat rule sip-r1 match source-address 10.1.1.3/24
set security nat source rule-set sip-nat rule sip-r1 then source-nat pool sip-nat-pool
set security nat proxy-arp interface ge-0/0/2.0 address 1.1.1.20/32 to 1.1.1.40/32
set security policies from-zone private to-zone public policy outgoing match
  source-address phone1
set security policies from-zone private to-zone public policy outgoing match
  destination-address any
set security policies from-zone private to-zone public policy outgoing match application
  junos-sip
set security policies from-zone private to-zone public policy outgoing then permit
set security policies from-zone public to-zone private policy incoming match
  source-address phone2
set security policies from-zone public to-zone private policy incoming match
  destination-address phone1
set security policies from-zone public to-zone private policy incoming match application
  junos-sip
set security policies from-zone public to-zone private policy incoming then permit
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a source NAT pool for incoming calls:

1. Configure interfaces.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 1.1.1.1/24
```

2. Configure zones and assign interfaces to them.

```
[edit security zones]
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone public interfaces ge-0/0/2.0
```

3. Configure address books.

```
[edit security zones]
user@host# set security-zone private address-book address phone1 10.1.1.3/32
```

- ```

user@host# set security-zone public address-book address proxy 1.1.1.3/32
user@host# set security-zone public address-book address phone2 1.1.1.4/32

```
4. Configure a source NAT pool.

```

[edit security nat]
user@host# set source pool sip-nat-pool address 1.1.1.20/32 to 1.1.1.40/32

```
  5. Configure a source NAT rule set with a rule.

```

[edit security nat source rule-set sip-nat]
user@host# set from zone private
user@host# set to zone public
user@host# set rule sip-r1 match source-address 10.1.1.3/24
user@host# set rule sip-r1 then source-nat pool sip-nat-pool

```
  6. Enable persistent NAT.

```

[edit security nat]
user@host# set source address-persistent

```
  7. Configure proxy ARP.

```

[edit security nat]
user@host# set proxy-arp interface ge-0/0/2.0 address 1.1.1.20/32 to 1.1.1.40/32

```
  8. Configure a security policy to allow outgoing SIP traffic.

```

[edit security policies from-zone private to-zone public policy outgoing]
set match source-address phone1
set match destination-address any
set match application junos-sip
set then permit

```
  9. Configure a security policy to allow incoming SIP traffic.

```

[edit security policies from-zone public to-zone private policy incoming]
set match source-address phone2
set match destination-address phone1
set match application junos-sip
set then permit

```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show security zones**, **show security nat**, and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 10.1.1.1/24;
    }
  }
}
ge-0/0/2 {
  unit 0 {
    family inet {

```

```
        address 1.1.1.1/24;
    }
}
[edit]
user@host# show security zones
security-zone private {
  address-book {
    address phone1 10.1.1.3/32;
  }
  interfaces {
    ge-0/0/0.0;
  }
}
security-zone public {
  address-book {
    address proxy 1.1.1.3/32;
    address phone2 1.1.1.4/32;
  }
  interfaces {
    ge-0/0/2.0;
  }
}
user@host# show security nat
source {
  pool sip-nat-pool {
    address {
      1.1.1.20/32 to 1.1.1.40/32;
    }
  }
  address-persistent;
  rule-set sip-nat {
    from zone private;
    to zone public;
    rule sip-r1 {
      match {
        source-address 10.1.1.3/24;
      }
      then {
        source-nat {
          pool {
            sip-nat-pool;
          }
        }
      }
    }
  }
}
proxy-arp {
  interface ge-0/0/2.0 {
    address {
      1.1.1.20/32 to 1.1.1.40/32;
    }
  }
}
[edit]
```

```

user@host# show security policies
from-zone private to-zone public {
  policy outgoing {
    match {
      source-address phone1;
      destination-address any;
      application junos-sip;
    }
    then {
      permit;
    }
  }
}
from-zone public to-zone private {
  policy incoming {
    match {
      source-address phone2;
      destination-address phone1;
      application junos-sip;
    }
    then {
      permit;
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Source NAT Pool Usage on page 303
- Verifying Source NAT Rule Usage on page 303
- Verifying SIP ALG Status on page 303

#### *Verifying Source NAT Pool Usage*

**Purpose** Verify that there is traffic using IP addresses from the source NAT pool.

**Action** From operational mode, enter the **show security nat source pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

#### *Verifying Source NAT Rule Usage*

**Purpose** Verify that there is traffic matching the source NAT rule.

**Action** From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

#### *Verifying SIP ALG Status*

**Purpose** Verify that SIP ALG is enabled on your system.

**Action** From operational mode, enter the **show security alg status** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Verifying SIP ALG Configurations on page 321](#)

## Example: Configuring Static NAT for Incoming SIP Calls

This example shows how to configure a static NAT mapping that allows callers in the private zone to register with the proxy server in the public zone.

- [Requirements on page 304](#)
- [Overview on page 304](#)
- [Configuration on page 305](#)
- [Verification on page 308](#)

### Requirements

---

Before you begin, understand how NAT works with the SIP ALG. See “Understanding SIP ALGs and NAT” on page 283.

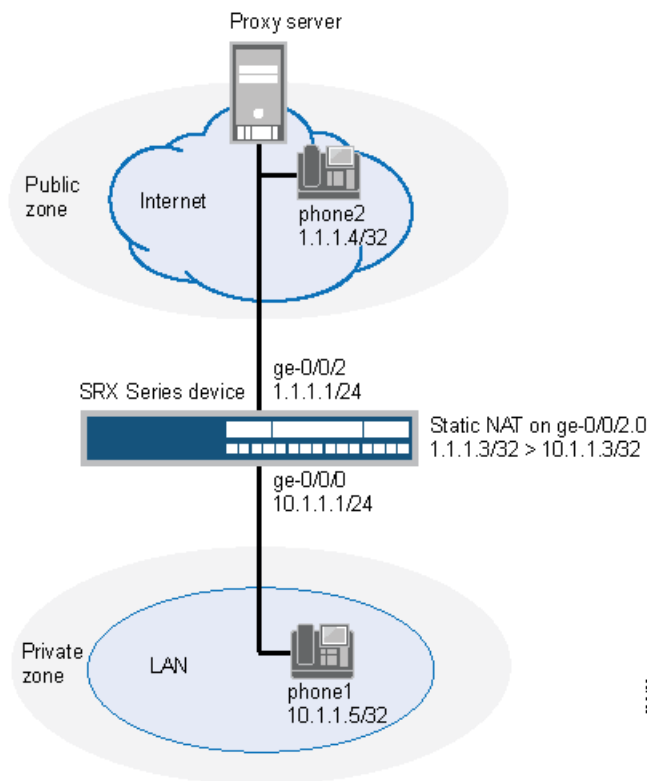
### Overview

---

When a SIP proxy server is located in an external or public zone, you can configure static NAT on the public interface to enable callers in the private zone to register with the proxy server.

In this example (see Figure 24 on page 305), phone1 is on the ge-0/0/0 interface in the private zone, and phone2 and the proxy server are on the ge-0/0/2 interface in the public zone. You create a static NAT rule set called incoming-sip with a rule called phone1 to match packets from the public zone with the destination address 1.1.1.3/32. For matching packets, the destination IP address is translated to the private address 10.1.1.3/32. You also create proxy ARP for the address 1.1.1.3/32 on interface ge-0/0/2.0. This allows the system to respond to ARP requests received on the interface for these addresses. Finally, you create a security policy called incoming that allows SIP traffic from the public zone to the private zone.

Figure 24: Static NAT for Incoming Calls



**NOTE:** When configuring static NAT for incoming SIP calls, make sure to configure one public address for each private address in the private zone.

### Configuration

#### CLI Quick Configuration

To quickly configure a static NAT mapping for incoming calls, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces ge-0/0/2 unit 0 family inet address 1.1.1.1/24
set security zones security-zone private interfaces ge-0/0/0.0
set security zones security-zone private address-book address phone1 10.1.1.5/32
set security zones security-zone public interfaces ge-0/0/2.0
set security zones security-zone public address-book address proxy 1.1.1.3/32
set security zones security-zone public address-book address phone2 1.1.1.4/32
set security nat static rule-set incoming-sip from zone public
set security nat static rule-set incoming-sip rule phone1 match destination-address
  1.1.1.3/32
set security nat static rule-set incoming-sip rule phone1 then static-nat prefix 10.1.1.3/32
set security nat proxy-arp interface ge-0/0/2.0 address 1.1.1.3/32
set security policies from-zone public to-zone private policy incoming match
  source-address phone2
set security policies from-zone public to-zone private policy incoming match
  source-address proxy
```

```

set security policies from-zone public to-zone private policy incoming match
  destination-address phone1
set security policies from-zone public to-zone private policy incoming match application
  junos-sip
set security policies from-zone public to-zone private policy incoming then permit
set security policies from-zone private to-zone public policy outgoing match
  source-address phone1
set security policies from-zone private to-zone public policy outgoing match
  destination-address phone2
set security policies from-zone private to-zone public policy outgoing match
  destination-address proxy
set security policies from-zone private to-zone public policy outgoing match application
  junos-sip
set security policies from-zone private to-zone public policy outgoing then permit

```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure static NAT for incoming calls:

1. Configure interfaces.

```

[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 1.1.1/24

```

2. Create security zones.

```

[edit security zones]
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone public interfaces ge-0/0/2.0

```

3. Assign addresses to the security zones.

```

[edit security zones]
user@host# set security-zone private address-book address phone1 10.1.1.5/32
user@host# set security-zone public address-book address proxy 1.1.1.3/32
user@host# set security-zone public address-book address phone2 1.1.1.4/32

```

4. Create a static NAT rule set with a rule.

```

[edit security nat static rule-set incoming-sip]
user@host# set from zone public
user@host# set rule phone1 match destination-address 1.1.1.3/32
user@host# set rule phone1 then static-nat prefix 10.1.1.3/32

```

5. Configure proxy ARP.

```

[edit security nat]
user@host# set proxy-arp interface ge-0/0/2.0 address 1.1.1.3/32

```

6. Define a security policy to allow incoming SIP traffic.

```

[edit security policies from-zone public to-zone private policy incoming]
user@host# set match source-address phone2
user@host# set match source-address proxy
user@host# set match destination-address phone1
user@host# set match application junos-sip
user@host# set then permit

```



7. Define a security policy to allow outgoing SIP traffic.

```
[edit security policies from-zone private to-zone public policy outgoing]
user@host# set match source-address phone1
user@host# set match destination-address phone2
user@host# set match destination-address proxy
user@host# set match application junos-sip
user@host# set then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show security zones**, **show security nat**, and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 10.1.1.1/24;
    }
  }
}
ge-0/0/2 {
  unit 0 {
    family inet {
      address 1.1.1.1/24;
    }
  }
}

[edit]
user@host# show security zones
security-zone private {
  address-book {
    address phone1 10.1.1.5/32;
  }
  interfaces {
    ge-0/0/0.0;
  }
}
security-zone public {
  address-book {
    address proxy 1.1.1.3/32;
    address phone2 1.1.1.4/32;
  }
  interfaces {
    ge-0/0/2.0;
  }
}

[edit]
user@host# show security nat
static {
  rule-set incoming-sip {
    from zone public;
```

```

rule phone1 {
  match {
    destination-address 1.1.1.3/32;
  }
  then {
    static-nat prefix 10.1.1.3/32;
  }
}
}
}
proxy-arp {
  interface ge-0/0/2.0 {
    address {
      1.1.1.3/32;
    }
  }
}
}
[edit]
user@host# show security policies
from-zone public to-zone private {
  policy incoming {
    match {
      source-address phone2;
      destination-address phone1;
      application junos-sip;
    }
    then {
      permit;
    }
  }
}
from-zone private to-zone public {
  policy outgoing {
    match {
      source-address phone1;
      destination-address [phone2 proxy];
      application junos-sip;
    }
    then {
      permit;
    }
  }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Static NAT Configuration on page 308
- Verifying SIP ALG Status on page 309

### *Verifying Static NAT Configuration*

**Purpose** Verify that there is traffic matching the static NAT rule set.

**Action** From operational mode, enter the **show security nat static rule** command. View the Translation hits field to check for traffic that matches the rule.

#### **Verifying SIP ALG Status**

**Purpose** Verify that SIP ALG is enabled on your system.

**Action** From operational mode, enter the **show security alg status** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Verifying SIP ALG Configurations on page 321

### Example: Configuring the SIP Proxy in the Private Zone and NAT in the Public Zone

This example shows how to configure a SIP proxy server in a private zone and static NAT in a public zone to allow callers in the public zone to register with the proxy server.

- Requirements on page 309
- Overview on page 309
- Configuration on page 310
- Verification on page 314

#### **Requirements**

---

Before you begin, understand how NAT works with the SIP ALG. See “Understanding SIP ALGs and NAT” on page 283.

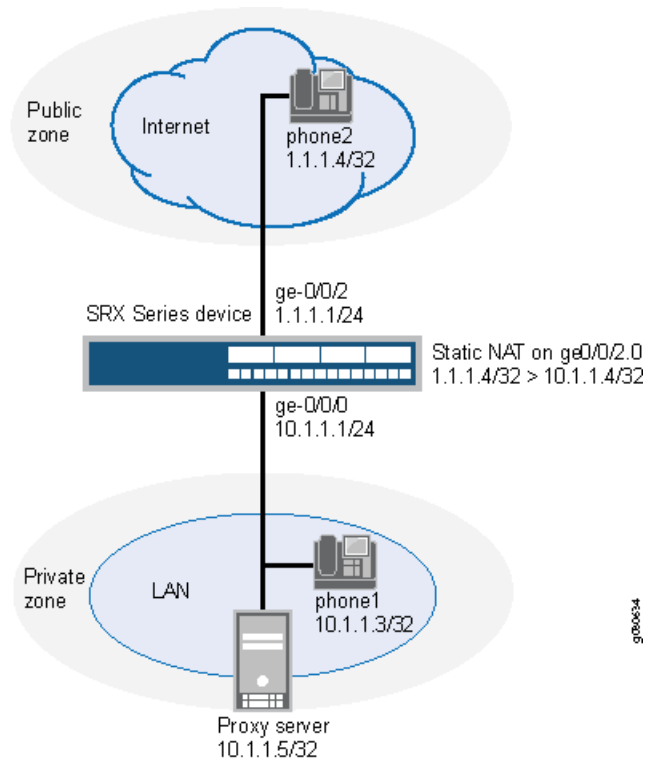
#### **Overview**

---

With the SIP proxy server in the private zone, you can configure static NAT on the external, or public, interface to allow callers in the public zone to register with the proxy server.

In this example (see Figure 25 on page 310), phone1 and the SIP proxy server are on the ge-0/0/0 interface in the private zone, and phone2 is on the ge-0/0/2 interface in the public zone. You configure a static NAT rule for the proxy server to allow phone2 to register with the proxy server, and then create a policy called outgoing that allows SIP traffic from the public to the private zone to enable callers in the public zone to register with the proxy server. You also configure a policy called incoming from the private to the public zone to allow phone1 to call out.

Figure 25: Configuring SIP Proxy in the Private Zone and NAT in Public Zone



In this example, you configure NAT as follows:

- Configure static NAT on the ge-0/0/2 interface to the proxy server with a rule set called incoming-sip with a rule called proxy to match packets from the public zone with the destination address 1.1.1.4/32. For matching packets, the destination IP address is translated to the private address 10.1.1.4/32.
- Configure a second rule set called sip-phones with a rule called phone1 to enable interface NAT for communication from phone1 to phone2.

### Configuration

#### CLI Quick Configuration

To quickly configure a SIP proxy server in a private zone and static NAT in a public zone and allow callers in the public zone to register with the proxy server, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces ge-0/0/2 unit 0 family inet address 1.1.1.1/24
set security zones security-zone private address-book address phone1 10.1.1.3/32
set security zones security-zone private address-book address proxy 10.1.1.5/32
set security zones security-zone private interfaces ge-0/0/0.0
set security zones security-zone public address-book address phone2 1.1.1.4/32
set security zones security-zone public interfaces ge-0/0/2.0
set security nat source rule-set sip-phones from zone private
set security nat source rule-set sip-phones to zone public
```

```

set security nat source rule-set sip-phones rule phone1 match source-address 10.1.1.3/32
set security nat source rule-set sip-phones rule phone1 then source-nat interface
set security nat static rule-set incoming-sip from zone public
set security nat static rule-set incoming-sip rule proxy match destination-address 1.1.1.4/32
set security nat static rule-set incoming-sip rule proxy then static-nat prefix 10.1.1.4/32
set security policies from-zone private to-zone public policy outgoing match
    source-address any
set security policies from-zone private to-zone public policy outgoing match
    destination-address phone2
set security policies from-zone private to-zone public policy outgoing match application
    junos-sip
set security policies from-zone private to-zone public policy outgoing then permit
set security policies from-zone public to-zone private policy incoming match
    source-address phone2
set security policies from-zone public to-zone private policy incoming match
    destination-address proxy
set security policies from-zone public to-zone private policy incoming match application
    junos-sip
set security policies from-zone public to-zone private policy incoming then permit

```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure static NAT for incoming calls:

1. Configure interfaces.

```

[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 1.1.1.1/24

```

2. Configure security zones.

```

[edit security zones]
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone public interfaces ge-0/0/2.0

```

3. Assign addresses to the security zones.

```

[edit security zones]
user@host# set security-zone private address-book address phone1 10.1.1.3/32
user@host# set security-zone private address-book address proxy 10.1.1.5/32
user@host# set security-zone public address-book address phone2 1.1.1.4/32

```

4. Create a rule set for static NAT and assign a rule to it.

```

[edit security nat static rule-set incoming-sip]
user@host# set from zone public
user@host# set rule proxy match destination-address 1.1.1.4/32
user@host# set rule proxy then static-nat prefix 10.1.1.4/32

```

5. Configure the second rule set and assign a rule to it.

```

[edit security nat source rule-set sip-phones]
user@host# set from zone private
user@host# set to zone public
user@host# set rule phone1 match source-address 10.1.1.3/32
user@host# set rule phone1 then source-nat interface

```

- Configure a security policy for outgoing traffic.

```
[edit security policies from-zone private to-zone public policy outgoing]
user@host#set match source-address any
user@host#set match destination-address phone2
user@host#set match application junos-sip
user@host#set then permit
```

- Configure a security policy for incoming traffic.

```
[edit security policies from-zone public to-zone private policy incoming]
user@host#set match source-address phone2
user@host#set match destination-address proxy
user@host#set match application junos-sip
user@host#set then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show security zones**, **show security nat**, and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 10.1.1.1/24;
    }
  }
}
ge-0/0/2 {
  unit 0 {
    family inet {
      address 1.1.1.1/24;
    }
  }
}
[edit]
user@host# show security zones
security-zone private {
  address-book {
    address phone1 10.1.1.3/32;
    address proxy 10.1.1.5/32;
  }
  interfaces {
    ge-0/0/0.0;
  }
}
security-zone public {
  address-book {
    address phone2 1.1.1.4/32;
  }
  interfaces {
    ge-0/0/2.0;
  }
}
```

```
[edit]
user@host# show security nat
source {
  rule-set sip-phones {
    from zone private;
    to zone public;
    rule phone1 {
      match {
        source-address 10.1.1.3/32;
      }
      then {
        source-nat {
          interface;
        }
      }
    }
  }
}
static {
  rule-set incoming-sip {
    from zone public;
    rule proxy {
      match {
        destination-address 1.1.1.4/32;
      }
      then {
        static-nat prefix 10.1.1.4/32;
      }
    }
  }
}
[edit]
user@host# show security policies
from-zone private to-zone public {
  policy outgoing {
    match {
      source-address any;
      destination-address phone2;
      application junos-sip;
    }
    then {
      permit;
    }
  }
}
from-zone public to-zone private {
  policy incoming {
    match {
      source-address phone2;
      destination-address proxy;
      application junos-sip;
    }
    then {
      permit;
    }
  }
}
```

```
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

---

To confirm that the configuration is working properly, perform these tasks:

- Verifying Static NAT Configuration on page 314
- Verifying SIP ALG Status on page 314

#### **Verifying Static NAT Configuration**

**Purpose** Verify that there is traffic matching the static NAT rule set.

**Action** From operational mode, enter the **show security nat static rule** command. View the Translation hits field to check for traffic that matches the rule.

#### **Verifying SIP ALG Status**

**Purpose** Verify that SIP ALG is enabled on your system.

**Action** From operational mode, enter the **show security alg status** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Verifying SIP ALG Configurations on page 321

## Example: Configuring a Three-Zone SIP ALG and NAT Scenario

This example shows how to configure a SIP proxy server in a private zone and static NAT in a public zone to allow callers in the public zone to register with the proxy server.

- Requirements on page 314
- Overview on page 314
- Configuration on page 315
- Verification on page 320

### Requirements

---

Before you begin, understand how NAT works with the SIP ALG. See “Understanding SIP ALGs and NAT” on page 283.

### Overview

---

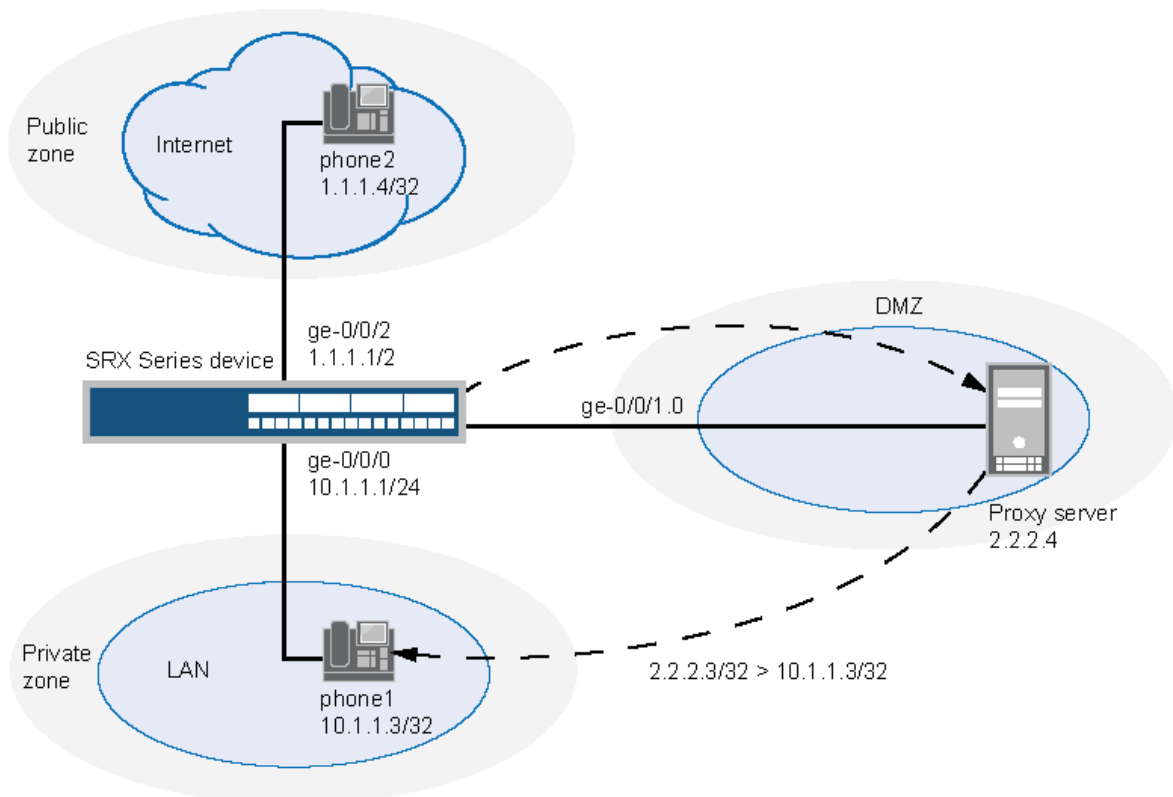
In a three-zone SIP configuration, the SIP proxy server is typically in a different zone from the calling and called systems. Such a scenario requires additional address and zone configuration, and policies to ensure that all systems have access to each other and to the proxy server.

In this example, phone1 is on the ge-0/0/0.0 interface in the private zone, phone2 is on the ge-0/0/2.0 interface in the public zone, and the proxy server is on the ge-0/0/1.0



interface in the DMZ. You configure static NAT rule for phone1 in the private zone. You then create policies for traffic traversing from the private zone to the DMZ and from the DMZ to the private zone, from the public zone to the DMZ and from the DMZ to the public zone, and from the private zone to the public zone. The arrows in Figure 26 on page 315 show the flow of SIP signaling traffic when phone2 in the public zone places a call to phone1 in the private zone. After the session is initiated, the data flows directly between phone1 and phone2.

Figure 26: Three-Zone SIP Configuration with Proxy in the DMZ



g030636

In this example, you configure NAT as follows:

- Configure a static NAT rule set called incoming-sip with a rule phone1 to match packets from the public zone with the destination address 2.2.2.3/32. For matching packets, the destination IP address is translated to the private address 10.1.1.3/32.
- Configure proxy ARP for the address 2.2.2.3/32 on interface ge-0/0/1.0 allowing the system to respond to ARP requests received on the interface for this address.
- Configure a second rule set called sip-phones with a rule r1 to enable interface NAT for communication from phone1 to the proxy server and from phone1 to phone2.

### Configuration

**CLI Quick Configuration** To quickly configure a SIP proxy server in a private zone and static NAT in a public zone, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces ge-0/0/1 unit 0 family inet address 2.2.2.2/24
set interfaces ge-0/0/2 unit 0 family inet address 1.1.1.1/24
set security zones security-zone private address-book address phone1 10.1.1.3/32
set security zones security-zone private interfaces ge-0/0/0.0
set security zones security-zone public address-book address phone2 1.1.1.4/32
set security zones security-zone public interfaces ge-0/0/2.0
set security zones security-zone dmz address-book address proxy 2.2.2.4/32
set security zones security-zone dmz interfaces ge-0/0/1.0
set security nat source rule-set sip-phones from zone private
set security nat source rule-set sip-phones to zone dmz
set security nat source rule-set sip-phones rule r1 match source-address 10.1.1.3/32
set security nat source rule-set sip-phones rule r1 then source-nat interface
set security policies from-zone private to-zone dmz policy private-to-proxy match
  source-address phone1
set security policies from-zone private to-zone dmz policy private-to-proxy match
  destination-address proxy
set security policies from-zone private to-zone dmz policy private-to-proxy match
  application junos-sip
set security policies from-zone private to-zone dmz policy private-to-proxy then permit
set security policies from-zone public to-zone dmz policy public-to-proxy match
  source-address phone2
set security policies from-zone public to-zone dmz policy public-to-proxy match
  destination-address proxy
set security policies from-zone public to-zone dmz policy public-to-proxy match application
  junos-sip
set security policies from-zone public to-zone dmz policy public-to-proxy then permit
set security policies from-zone private to-zone public policy private-to-public match
  source-address phone1
set security policies from-zone private to-zone public policy private-to-public match
  destination-address phone2
set security policies from-zone private to-zone public policy private-to-public match
  application junos-sip
set security policies from-zone private to-zone public policy private-to-public then permit
set security policies from-zone dmz to-zone private policy proxy-to-private match
  source-address proxy
set security policies from-zone dmz to-zone private policy proxy-to-private match
  destination-address phone1
set security policies from-zone dmz to-zone private policy proxy-to-private match
  application junos-sip
set security policies from-zone dmz to-zone private policy proxy-to-private then permit
set security policies from-zone dmz to-zone public policy proxy-to-public match
  source-address proxy
set security policies from-zone dmz to-zone public policy proxy-to-public match
  destination-address phone2
set security policies from-zone dmz to-zone public policy proxy-to-public match application
  junos-sip
set security policies from-zone dmz to-zone public policy proxy-to-public then permit
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a SIP proxy server in a private zone and static NAT in a public zone:

1. Configure interfaces.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 2.2.2.2/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 1.1.1.1/24
```

2. Configure security zones.

```
[edit security zones]
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone public interfaces ge-0/0/2.0
user@host# set security-zone dmz interfaces ge-0/0/1.0
```

3. Assign addresses to the security zones.

```
[edit security zones]
user@host# set security-zone private address-book address phone1 10.1.1.3/32
user@host# set security-zone public address-book address phone2 1.1.1.4/32
user@host# set security-zone dmz address-book address proxy 2.2.2.4/32
```

4. Configure interface NAT for communication from phone1 to proxy.

```
[edit security nat source rule-set sip-phones]
user@host# set from zone private
user@host# set to zone dmz
user@host# set rule r1 match source-address 10.1.1.3/32
user@host# set rule r1 then source-nat interface
```

5. Configure a security policy to allow traffic from zone private to zone DMZ.

```
[edit security policies from-zone private to-zone dmz policy private-to-proxy]
user@host# set match source-address phone1
user@host# set match destination-address proxy
user@host# set match application junos-sip
user@host# set then permit
```

6. Configure a security policy to allow traffic from zone public to zone DMZ.

```
[edit security policies from-zone public to-zone dmz policy public-to-proxy]
user@host# set match source-address phone2
user@host# set match destination-address proxy
user@host# set match application junos-sip
user@host# set then permit
```

7. Configure a security policy to allow traffic from zone private to zone public.

```
[edit security policies from-zone private to-zone public policy private-to-public]
user@host# set match source-address phone1
user@host# set match destination-address phone2
user@host# set match application junos-sip
user@host# set then permit
```

8. Configure a security policy to allow traffic from zone DMZ to zone private.

```
[edit security policies from-zone dmz to-zone private policy proxy-to-private]
user@host# set match source-address proxy
user@host# set match destination-address phone1
user@host# set match application junos-sip
user@host# set then permit
```

- Configure a security policy to allow traffic from zone DMZ to zone public.

```
[edit security policies from-zone dmz to-zone public policy proxy-to-public]
user@host# set match source-address proxy
user@host# set match destination-address phone2
user@host# set match application junos-sip
user@host# set then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show security zones**, **show security nat**, and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 10.1.1.1/24;
    }
  }
}
ge-0/0/1 {
  unit 0 {
    family inet {
      address 2.2.2.2/24;
    }
  }
}
ge-0/0/2 {
  unit 0 {
    family inet {
      address 1.1.1.1/24;
    }
  }
}

[edit]
user@host# show security zones
security-zone private {
  address-book {
    address phone1 10.1.1.3/32;
  }
  interfaces {
    ge-0/0/0.0;
  }
}
security-zone public {
  address-book {
    address phone2 1.1.1.4/32;
  }
}
```

```
    }
    interfaces {
        ge-0/0/2.0;
    }
}
security-zone dmz {
    address-book {
        address proxy 2.2.2.4/32;
    }
    interfaces {
        ge-0/0/1.0;
    }
}
[edit]
user@host# show security nat
source {
    rule-set sip-phones {
        from zone private;
        to zone dmz;
        rule r1 {
            match {
                source-address 10.1.1.3/32;
            }
            then {
                source-nat {
                    interface;
                }
            }
        }
    }
}
proxy-arp {
    interface ge-0/0/1.0 {
        address {
            2.2.2.3/32;
        }
    }
}
[edit]
user@host# show security policies
from-zone private to-zone dmz {
    policy private-to-proxy {
        match {
            source-address phone1;
            destination-address proxy;
            application junos-sip;
        }
        then {
            permit;
        }
    }
}
from-zone public to-zone dmz {
    policy public-to-proxy {
        match {
```

```

        source-address phone2;
        destination-address proxy;
        application junos-sip;
    }
    then {
        permit;
    }
}
}
from-zone private to-zone public {
    policy private-to-public {
        match {
            source-address phone1;
            destination-address phone2;
            application junos-sip;
        }
        then {
            permit;
        }
    }
}
from-zone dmz to-zone private {
    policy proxy-to-private {
        match {
            source-address proxy;
            destination-address phone2;
            application junos-sip;
        }
        then {
            permit;
        }
    }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Source NAT Rule Usage on page 320
- Verifying Static NAT Configuration on page 320
- Verifying SIP ALG Status on page 321

#### *Verifying Source NAT Rule Usage*

**Purpose** Verify that there is traffic matching the source NAT rule.

**Action** From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

#### *Verifying Static NAT Configuration*

**Purpose** Verify that there is traffic matching the static NAT rule set.

**Action** From operational mode, enter the **show security nat static rule** command. View the Translation hits field to check for traffic that matches the rule.

#### *Verifying SIP ALG Status*

**Purpose** Verify that SIP ALG is enabled on your system.

**Action** From operational mode, enter the **show security alg status** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Verifying SIP ALG Configurations on page 321

## Verifying SIP ALG Configurations

- Verifying SIP ALGs on page 321
- Verifying SIP ALG Calls on page 321
- Verifying SIP ALG Call Details on page 322
- Verifying SIP ALG Counters on page 322
- Verifying the Rate of SIP ALG Messages on page 323

### Verifying SIP ALGs

**Purpose** Verify SIP ALG verification options.

**Action** From the CLI, enter the **show security alg sip ?** command.

```
user@host> show security alg sip ?
Possible completions:
  calls           Show SIP calls
  counters        Show SIP counters
  rate            Show SIP rate
```

**Meaning** The output shows a list of all SIP verification parameters. Verify the following information:

- Calls—Lists all SIP calls.
- Counters—Provides counters of response codes for each SIP request method and error type.
- Rate—Provides speed and periodicity of SIP signaling messages.

### Verifying SIP ALG Calls

**Purpose** Display information about active calls.

**Action** From the J-Web interface, select **Monitor>ALGs>SIP>Calls**. Alternatively, from the CLI, enter the **show security alg sip calls** command.

```
user@host> show security alg sip calls
Total number of calls: 1
  Call ID: 47090a32@30.2.20.5
```

Method: INVITE

**Meaning** The output shows a list of all active SIP calls. Verify the User Agent Server (UAS) call ID and local and remote tags, and the state of the call.

## Verifying SIP ALG Call Details

**Purpose** Display address and SDP about active calls.

**Action** From the J-Web interface, select **Monitor>ALGs>SIP>Details**. Alternatively, from the CLI, enter the **show security alg sip calls detail** command.

```
user@host> show security alg sip calls detail
Total number of calls: 1
  Call ID      : 47090a32@30.2.20.5
Method       : INVITE
State        : SETUP
Group ID     : 24575
```

**Meaning** The output provides details about all active SIP calls. Verify the following information:

- The total number of calls, their ID and tag information, and state
- Remote group ID
- The IP addresses and port numbers and SDP connection and media details

## Verifying SIP ALG Counters

**Purpose** Display information about SIP counters.

**Action** From the J-Web interface, select **Monitor>ALGs>SIP>Counters**. Alternatively, from the CLI, enter the **show security alg sip counters** command.

```
user@host> show security alg sip counters
Method      T      1xx      2xx      3xx      4xx      5xx      6xx
            RT      RT      RT      RT      RT      RT      RT
INVITE      4      4      3      0      0      0      0
            0      0      0      0      0      0      0
CANCEL      0      0      0      0      0      0      0
            0      0      0      0      0      0      0
ACK         3      0      0      0      0      0      0
            0      0      0      0      0      0      0
BYE         3      0      3      0      0      0      0
            0      0      0      0      0      0      0
REGISTER    7      0      7      0      0      0      0
            0      0      0      0      0      0      0
OPTIONS     0      0      0      0      0      0      0
            0      0      0      0      0      0      0
INFO        0      0      0      0      0      0      0
            0      0      0      0      0      0      0
MESSAGE     0      0      0      0      0      0      0
            0      0      0      0      0      0      0
NOTIFY      0      0      0      0      0      0      0
            0      0      0      0      0      0      0
```



PRACK	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
PUBLISH	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
REFER	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
SUBSCRIBE	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
UPDATE	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
BENOTIFY	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
SERVICE	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
OTHER	0	0	0	0	0	0	0
	0	0	0	0	0	0	0

## SIP Error Counters

```

-----
Total Pkt-in                               :34
Total Pkt dropped on error                 :0
  Call error                               :0
IP resolve error                           :0
NAT error                                  :0
Resource manager error                     :0
RR header exceeded max                     :0
Contact header exceeded max                :0
Call Dropped due to limit                  :0
SIP stack error                            : 0
SIP decode error                           : 0
SIP unknown method error                   : 0
RTO message sent                           : 0
RTO message received                       : 0
RTO buffer allocation failure               : 0
RTO buffer transmit failure                : 0
RTO send processing error                   : 0
RTO receive processing error                : 0
RTO receive invalid length                  : 0
RTO receive call process error              : 0
RTO receive call allocation error           : 0
RTO receive call register error            : 0
RTO receive invalid status error           : 0

```

**Meaning** The output provides a count of all SIP response codes transmitted and received, and of SIP errors. Verify the following information:

- A count of transmissions of response codes for each SIP request method
- A count of all possible error types

## Verifying the Rate of SIP ALG Messages

**Purpose** Display information about SIP message rate.

**Action** From the J-Web interface, select **Monitor>ALGs>SIP>Rate**. Alternatively, from the CLI, enter the **show security alg sip rate** command.

```

user@host> show security alg sip rate
CPU ticks per microseconds is 3735928559
Time taken for the last message is 0 microseconds

```

Total time taken for 0 messages is 0 microseconds(in less than 10 minutes)  
Rate: 3735928559 messages/second

**Meaning** The output provides information about CPU usage for messages, and speed and periodicity of SIP signaling messages. Verify the following information:

- CPU ticks per US
- Passage time for last message, for all messages, and the rate at which messages transit the network

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- SIP ALG Configuration Overview on page 275
- Example: Configuring Interface Source NAT for Incoming SIP Calls on page 293
- Example: Configuring a Source NAT Pool for Incoming SIP Calls on page 298
- Example: Configuring Static NAT for Incoming SIP Calls on page 304
- Example: Configuring the SIP Proxy in the Private Zone and NAT in the Public Zone on page 309
- Example: Configuring a Three-Zone SIP ALG and NAT Scenario on page 314

# SCCP ALGs

- Understanding SCCP ALGs on page 325
- SCCP ALG Configuration Overview on page 330
- SCCP ALG Inactive Media Timeout on page 331
- SCCP ALG Unknown Message Types on page 332
- SCCP ALG DoS Attack Protection on page 334
- Example: Configuring the SCCP ALG Call Manager or TFTP Server in the Private Zone on page 336
- Verifying SCCP ALG Configurations on page 342

## Understanding SCCP ALGs

---

The Skinny Client Control Protocol (SCCP) is a Cisco proprietary protocol for call signaling. Skinny is based on a call-agent-based call-control architecture. The control protocol uses binary-coded frames encoded on TCP frames sent to well-known TCP port number destinations to set up and tear down RTP media sessions.

The SCCP protocol just as other call control protocols, negotiates media endpoint parameters—specifically the Real-Time Transport Protocol (RTP) port number and the IP address of media termination—by embedding information in the control packets. The SCCP Application Layer Gateway (ALG) parses these control packets and facilitates media and control packets to flow through the system.

The SCCP ALG also implements rate limiting of calls and helps protect critical resources from overloading and denial-of-service (DoS) attacks.

The following functions are implemented by the SCCP ALG in Junos OS:

- Validation of SCCP protocol data units
- Translation of embedded IP address and port numbers
- Allocation of firewall resources (pinholes and gates) to pass media
- Aging out idle calls
- Configuration API for SCCP ALG parameters
- Operational mode API for displaying counters, status and statistics

In the SCCP architecture, a proxy, known as the CallManager, does most of the processing. IP phones, also called End Stations, run the SCCP client and connect to a primary (and, if available, a secondary) CallManager over TCP on port 2000 and register with the primary CallManager. This connection is then used to establish calls coming to or from the client.

The SCCP ALG supports the following:

- Call flow from a SCCP client, through the CallManager, to another SCCP client.
- Seamless failover—Switches over all calls in process to the standby firewall during failure of the primary.
- Voice-over-IP (VoIP) signaling payload inspection—Fully inspects the payload of incoming VoIP signaling packets. Any malformed packet attack is blocked by the ALG.
- SCCP signaling payload inspection—Fully inspects the payload of incoming SCCP signaling packets. Any malformed-packet attack is blocked by the ALG.
- Stateful processing—Invokes the corresponding VoIP-based state machines to process the parsed information. Any out-of-state or out-of-transaction packet is identified and properly handled.
- Network Address Translation (NAT)—Translates any embedded IP address and port information in the payload, based on the existing routing information and network topology, with the translated IP address and port number, if necessary.
- Pinhole creation and management for VoIP traffic—Identifies IP address and port information used for media or signaling and dynamically opens (and closes) pinholes to securely stream the media.

This topic includes the following sections:

- SCCP Security on page 326
- SCCP Components on page 327
- SCCP Transactions on page 327
- SCCP Control Messages and RTP Flow on page 328
- SCCP Messages on page 329

## SCCP Security

The SCCP ALG includes the following security features:

- Stateful inspection of SCCP control messages over TCP and validation of the message format, and message validity for the current call state. Invalid messages are dropped.
- Security policy enforcement between Cisco IP phones and Cisco CallManager.
- Protect against call flooding by rate limiting the number of calls processed by the ALG.
- Seamless failover of calls, including the ones in progress in case of device failure in a clustered deployment.

---

## SCCP Components

The principal components of the SCCP VoIP architecture include the following:

- SCCP Client on page 327
- CallManager on page 327
- Cluster on page 327

### SCCP Client

---

The SCCP client runs on an IP phone, also called an *End Station*, which uses SCCP for signaling and for making calls. For an SCCP client to make a call, it must first register with a Primary CallManager (and a secondary, if available). The connection between the client and the CallManager is over TCP on port 2000. This connection is then used to establish calls to or from the client. Transmission of media is over RTP, UDP, and IP.

### CallManager

---

The CallManager implements SCCP call control server software and has overall control of all devices and communication in the SCCP VoIP network. Its functions include defining, monitoring and controlling SCCP groups, regions of numbers, and route plans; providing initialization, admission, and registration of devices on the network; providing a redundant database that contains addresses, phone numbers, and number formats; and initiating contact with called devices or their agents to establish logical sessions in which voice communication can flow.

### Cluster

---

A *cluster* is a collection of SCCP clients and a CallManager. The CallManager in the cluster detects all SCCP clients in the cluster. There can be more than one CallManager for backup in a cluster. CallManager behavior varies in each of the following cluster scenarios:

- Intra-Cluster, in which the CallManager detects each SCCP client, and the call is between SCCP clients of the same cluster.
- Inter-Cluster, in which the CallManager needs to communicate with another CallManager using H.323 for call setup.
- Inter-Cluster calls using the gatekeeper for admission control and address resolution.

CallManager behavior also varies with calls between an SCCP client and a phone in a public switched telephone network (PSTN), and with calls between an SCCP client and a phone in another administrative domain that is using H.323.

## SCCP Transactions

SCCP transactions are the processes that need to take place in order for an SCCP call to proceed. SCCP transactions include the following processes:

- Client Initialization on page 328
- Client Registration on page 328

- Call Setup on page 328
- Media Setup on page 328

### Client Initialization

---

To initialize, the SCCP client needs to determine the IP address of the CallManager, its own IP address, and other information about the IP gateway and DNS servers. Initialization takes place on the local LAN. The client sends a Dynamic Host Control Protocol (DHCP) request to get an IP address, the DNS server address, and the TFTP server name and address. The client needs the TFTP server name to download the configuration file called *sepmacaddr.cnf*. If the TFTP name is not given, the client uses the default filename in the IP phone. The client then downloads the .cnf (xml) configuration file from TFTP server. CNF files contain the IP address or addresses of the primary and secondary Cisco CallManager. With this information, the client contacts the CallManager to register.

### Client Registration

---

The SCCP client, after initialization, registers with the CallManager over a TCP connection on well-known default port 2000. The client registers by providing the CallManager with its IP address, the MAC address of the phone, and other information, such as protocol and version. The client cannot initiate or receive calls until it is registered. Keepalive messages keep this TCP connection open between the client and CallManager so that the client can initiate or receive calls at any time, provided that a policy on the device allows this.

### Call Setup

---

IP phone-to-IP phone call setup using SCCP is always handled by the CallManager. Messages for call setup are sent to the CallManager, which returns messages appropriate to the status of the call. If call setup is successful, and a policy on the device allows the call, the CallManager sends the media setup messages to the client.

### Media Setup

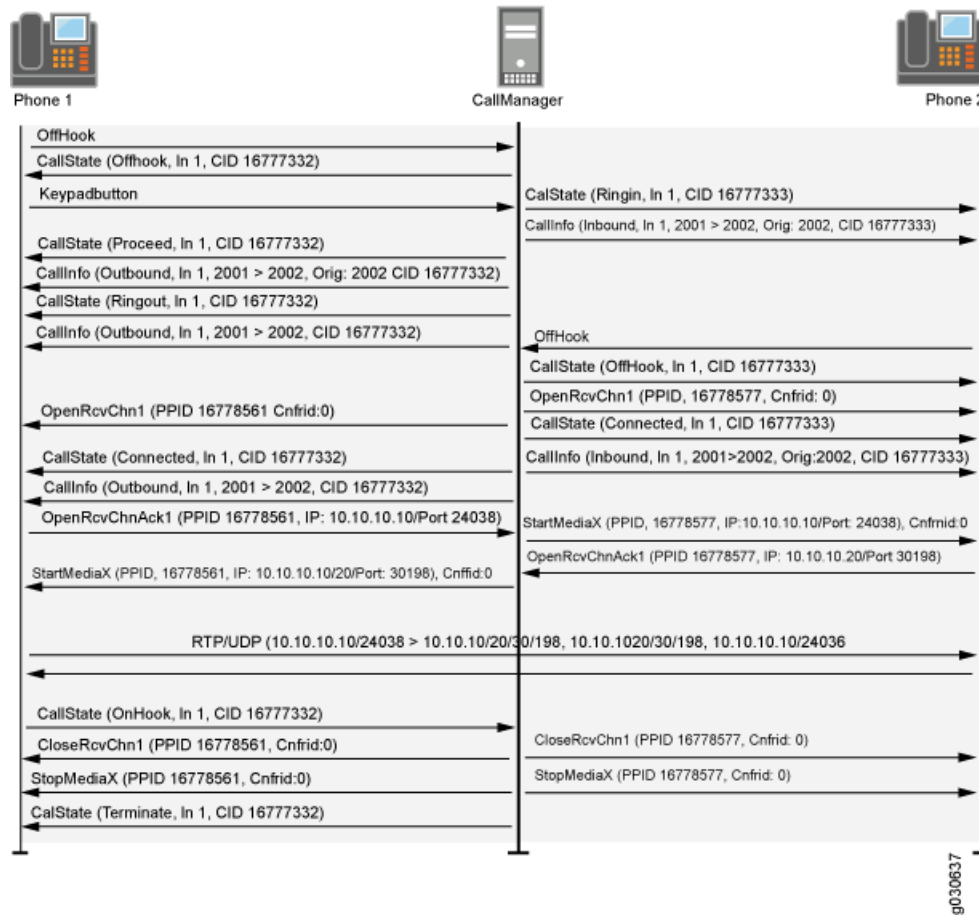
---

The CallManager sends the IP address and port number of the called party to the calling party. The CallManager also sends the media IP address and port number of the calling party to the called party. After media setup, media is transmitted directly between clients. When the call ends, the CallManager is informed and terminates the media streams. At no time during this process does the CallManager hand over call-setup function to the client. Media is streamed directly between clients through RTP/UDP/IP.

## SCCP Control Messages and RTP Flow

Figure 27 on page 329 shows the SCCP control messages used to set up and tear down a simple call between Phone 1 and Phone 2. Except for the OffHook message initiating the call from Phone1 and the OnHook message signaling the end of the call, all aspects of the call are controlled by the CallManager.

Figure 27: Call Setup and Teardown



## SCCP Messages

Table 33 on page 329, Table 34 on page 329, Table 35 on page 330, and Table 36 on page 330 list the SCCP call message IDs in the four intervals allowed by the device.

**Table 33: Station to CallManager Messages**

#define STATION_REGISTER_MESSAGE	0x00000001
#define STATION_IP_PORT_MESSAGE	0x00000002
#define STATION_ALARM_MESSAGE	0x00000020
#define STATION_OPEN_RECEIVE_CHANNEL_ACK	0x00000022

**Table 34: CallManager to Station Messages**

#define STATION_START_MEDIA_TRANSMISSION	0x00000001
#define STATION_STOP_MEDIA_TRANSMISSION	0x00000002

**Table 34: CallManager to Station Messages (continued)**

#define STATION_CALL_INFO_MESSAGE	0x00000020
#define STATION_OPEN_RECEIVE_CHANNEL_ACK	0x00000022
#define STATION_CLOSE_RECEIVE_CHANNEL	0x00000106

**Table 35: CallManager 4.0 Messages and Post Sccp 6.2**

#define STATION_REGISTER_TOKEN_REQ_MESSAGE	0x00000029
#define STATION_MEDIA_TRANSMISSION_FAILURE	0x0000002A
#define STATION_OPEN_MULTIMEDIA_RECEIVE_CHANNEL_ACK	0x00000031

**Table 36: CallManager to Station**

#define STATION_OPEN_MULTIMEDIA_RECEIVE_CHANNEL	0x00000131
#define STATION_START_MULTIMEDIA_TRANSMISSION	0x00000132
#define STATION_STOP_MULTIMEDIA_TRANSMISSION	0x00000133
#define STATION_CLOSE_MULTIMEDIA_RECEIVE_CHANNEL	0x00000136

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [ALG Overview](#) on page 217
- [SCCP ALG Configuration Overview](#) on page 330
- [Example: Configuring the SCCP ALG Call Manager or TFTP Server in the Private Zone](#) on page 336

## SCCP ALG Configuration Overview

The Skinny Client Control Protocol Application Layer Gateway (SCCP ALG) is enabled by default on the device—no action is required to enable it. However, you might choose to fine-tune SCCP ALG operations by using the following instructions:

1. Conserve network resources and maximize throughput. For instructions, see “Example: Setting SCCP ALG Inactive Media Timeouts” on page 331.
2. Enable unknown messages to pass when the session is in Network Address Translation (NAT) mode and route mode. For instructions, see “Example: Allowing Unknown SCCP ALG Message Types” on page 333.
3. Protect the SCCP clients from denial-of-service (DoS) flood attacks. For instructions, see “Example: Configuring SCCP ALG DoS Attack Protection” on page 335.



- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding SCCP ALGs on page 325](#)
  - [Example: Configuring the SCCP ALG Call Manager or TFTP Server in the Private Zone on page 336](#)
  - [Verifying SCCP ALG Configurations on page 342](#)

---

## SCCP ALG Inactive Media Timeout

---

- [Understanding SCCP ALG Inactive Media Timeouts on page 331](#)
- [Example: Setting SCCP ALG Inactive Media Timeouts on page 331](#)

### Understanding SCCP ALG Inactive Media Timeouts

The inactive media timeout feature helps you to conserve network resources and maximize throughput.

This parameter indicates the maximum length of time (in seconds) a call can remain active without any media traffic within a group. Each time a Real-Time Transport Protocol (RTP) or Real-Time Control Protocol (RTCP) packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the gates the Skinny Client Control Protocol (SCCP) opened for media are closed. The default setting is 120 seconds, and the range is from 10 to 2550 seconds. Note that upon timeout, while resources for media (sessions and pinholes) are removed, the call is not terminated.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding SCCP ALGs on page 325](#)
  - [SCCP ALG Configuration Overview on page 330](#)
  - [Example: Setting SCCP ALG Inactive Media Timeouts on page 331](#)

### Example: Setting SCCP ALG Inactive Media Timeouts

This example shows how to set the inactive media timeout value for the SCCP ALG.

- [Requirements on page 331](#)
- [Overview on page 332](#)
- [Configuration on page 332](#)
- [Verification on page 332](#)

#### Requirements

---

Before you begin, review the parameter used to indicate the maximum length of time (in seconds) a call can remain active without any media traffic within a group. See “Understanding SCCP ALG Inactive Media Timeouts” on page 331.

### Overview

---

Each time an RTP or RTCP packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the gates the SCCP opened for media are closed. This example sets the media inactivity timeout to 90 seconds.

### Configuration

---

#### J-Web Quick Configuration

To set the inactive media timeout for the SCCP ALG:

1. Select **Configure>Security>ALG**.
2. Select the **SCCP** tab.
3. In the Inactive Media Timeout box, enter **90**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

#### Step-by-Step Procedure

To set the inactive media timeout for the SCCP ALG:

1. Configure the SCCP ALG inactive media timeout value.  

```
[edit]  
user@host# set security alg sccp inactive-media-timeout 90
```
2. If you are done configuring the device, commit the configuration.  

```
[edit]  
user@host# commit
```

### Verification

---

To verify the configuration is working properly, enter the **show security alg sccp** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding SCCP ALG Inactive Media Timeouts on page 331](#)
- [SCCP ALG Configuration Overview on page 330](#)
- [Verifying SCCP ALG Configurations on page 342](#)

## SCCP ALG Unknown Message Types

---

- [Understanding SCCP ALG Unknown Message Types on page 332](#)
- [Example: Allowing Unknown SCCP ALG Message Types on page 333](#)

### Understanding SCCP ALG Unknown Message Types

To accommodate on-going development of the Skinny Client Control Protocol (SCCP), you might want to allow traffic containing new SCCP message types. The unknown SCCP message type feature enables you to configure the device to accept SCCP traffic containing unknown message types in both Network Address Translation (NAT) mode and route mode.

This feature enables you to specify how unidentified SCCP messages are handled by the device. The default is to drop unknown (unsupported) messages. We do not recommend permitting unknown messages because they can compromise security. However, in a secure test or production environment, this command can be useful for resolving interoperability issues with disparate vendor equipment. Permitting unknown SCCP messages can help you get your network operational so that you can later analyze your voice-over-IP (VoIP) traffic to determine why some messages were being dropped.

Note that this command applies only to received packets identified as supported VoIP packets. If a packet cannot be identified, it is always dropped. If a packet is identified as a supported protocol and you have configured the device to permit unknown message types, the message is forwarded without processing.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding SCCP ALGs on page 325](#)
- [SCCP ALG Configuration Overview on page 330](#)
- [Example: Allowing Unknown SCCP ALG Message Types on page 333](#)

### Example: Allowing Unknown SCCP ALG Message Types

This example shows how to configure the SCCP ALG to allow unknown SCCP message types in both NAT mode and route mode.

- [Requirements on page 333](#)
- [Overview on page 333](#)
- [Configuration on page 333](#)
- [Verification on page 334](#)

#### Requirements

Before you begin, determine whether to accommodate new and unknown SCCP message types for the device. See “Understanding SCCP ALG Unknown Message Types” on page 332.

#### Overview

This feature enables you to specify how unidentified SCCP messages are handled by a Juniper Networks device. The default is to drop unknown (unsupported) messages because unknown messages can compromise security. However, in a secure test or production environment, this command can be useful for resolving interoperability issues with disparate vendor equipment.

#### Configuration

#### J-Web Quick Configuration

To configure the SCCP ALG to allow unknown message types:

1. Select **Configure>Security>ALG**.
2. Select the **SCCP** tab.
3. Select the **Enable Permit NAT applied** check box.
4. Select the **Enable Permit routed** check box.

5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options**>**Commit**.

**Step-by-Step Procedure**

To configure the SCCP ALG to allow unknown message types:

1. Allow unknown message types to pass if the session is in either NAT mode or in route mode.

[edit]

```
user@host# set security alg sccp application-screen unknown-message permit-nat-applied permit-routed
```

2. If you are done configuring the device, commit the configuration.

[edit]

```
user@host# commit
```

---

**Verification**

To verify the configuration is working properly, enter the **show security alg sccp** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding SCCP ALG Unknown Message Types on page 332](#)
- [SCCP ALG Configuration Overview on page 330](#)
- [Verifying SCCP ALG Configurations on page 342](#)

---

## SCCP ALG DoS Attack Protection

---

- [Understanding SCCP ALG DoS Attack Protection on page 334](#)
- [Example: Configuring SCCP ALG DoS Attack Protection on page 335](#)

### Understanding SCCP ALG DoS Attack Protection

You can protect Skinny Client Control Protocol Application Layer Gateway (SCCP ALG) clients from denial-of-service (DoS) flood attacks by limiting the number of calls they attempt to process.

When you configure SCCP call flood protection, the SCCP ALG drops any calls exceeding the threshold you set. The range is 2 to 1000 calls per second per client, the default is 20.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding SCCP ALGs on page 325](#)
- [SCCP ALG Configuration Overview on page 330](#)
- [Example: Configuring SCCP ALG DoS Attack Protection on page 335](#)

## Example: Configuring SCCP ALG DoS Attack Protection

This example shows how to configure connection flood protection for the SCCP ALG.

- Requirements on page 335
- Overview on page 335
- Configuration on page 335
- Verification on page 335

### Requirements

Before you begin, determine whether to protect the SCCP media gateway from DoS flood attacks. See “Understanding SCCP ALG DoS Attack Protection” on page 334.

### Overview

In this example, the device is configured to drop any calls exceeding 500 per second per client.

### Configuration

#### J-Web Quick Configuration

To configure call flood protection for the SCCP ALG:

1. Select **Configure>Security>ALG**.
2. Select the **SCCP** tab.
3. In the Call flood threshold box, type **500**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

#### Step-by-Step Procedure

To configure call flood protection for the SCCP ALG:

1. Configure the DoS attack protection:
 

```
[edit]
user@host# set security alg sccp application-screen call-flood threshold 500
```
2. If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security alg sccp** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding SCCP ALG DoS Attack Protection on page 334
- SCCP ALG Configuration Overview on page 330
- Verifying SCCP ALG Configurations on page 342

## Example: Configuring the SCCP ALG Call Manager or TFTP Server in the Private Zone

This example shows how to configure static NAT on the outgoing interface of a Juniper Networks device to allow callers in a public zone to register with an SCCP ALG Call Manager or a TFTP server located in a private zone.

- Requirements on page 336
- Overview on page 336
- Configuration on page 337
- Verification on page 342

### Requirements

Before you begin, understand NAT support with SCCP ALG. See “Understanding SCCP ALGs” on page 325.

### Overview

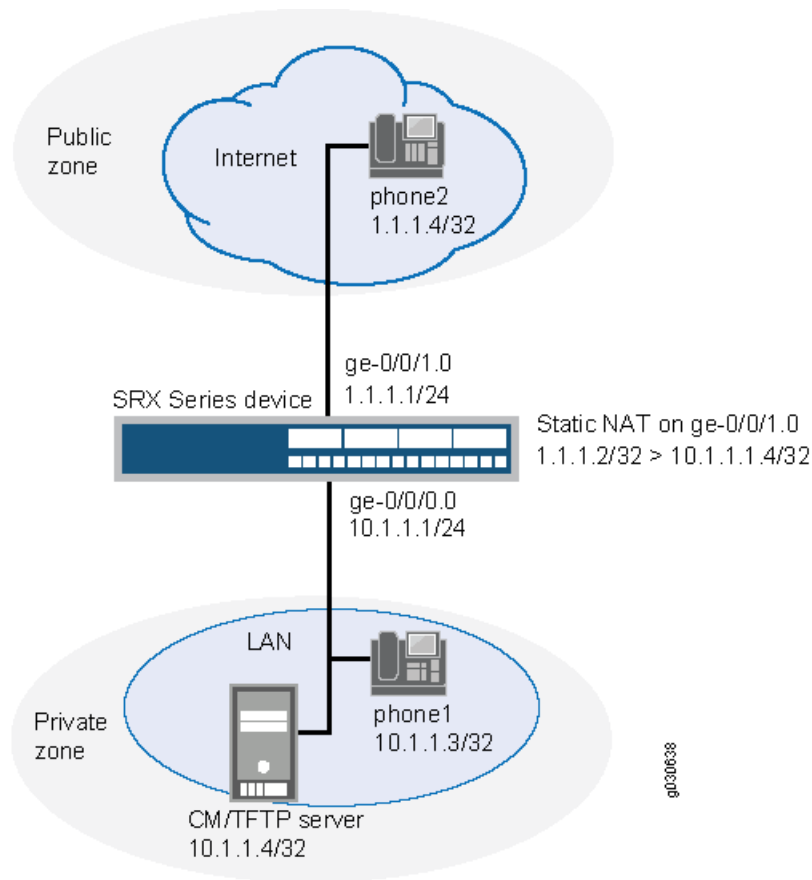
In this example (see Figure 28 on page 337), a single device is serving as a Call Manager or a TFTP server. The Call Manager or TFTP server and phone1 are on the ge-0/0/0.0 interface in the private zone, and phone2 is on the ge-0/0/1.0 interface in the public zone. You configure a static NAT rule set for the Call Manager or TFTP server so that when phone2 boots up it contacts the TFTP server and obtains the IP address of the Call Manager. You then create a policy called in-pol to allow SCCP traffic from the public to the private zone and a policy called out-pol to allow phone1 to call out.



**NOTE:** We recommend that you change the IP address of the Call Manager, which resides in the TFTP server configuration file (`sep <mac_addr>.cnf`), to the NAT IP address of the Call Manager.

---

Figure 28: Call Manager or TFTP Server in the Private Zone



In this example, you configure NAT as follows:

- Create a static NAT rule set called to-proxy with a rule called phone2 to match packets from the public zone with the destination address 1.1.1.2/32. For matching packets, the destination IP address is translated to the private address 10.1.1.4/32.
- Configure proxy ARP for the address 1.1.1.2/32 on interface ge-0/0/1.0. This allows the system to respond to ARP requests received on the interface for these addresses.
- Configure a second rule set called phones with a rule called phone1 to enable interface NAT for communication from phone1 to phone2.

## Configuration

**CLI Quick Configuration** To quickly configure NAT for an SCCP ALG Call Manager or a TFTP server located in a private zone, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces ge-0/0/1 unit 0 family inet address 1.1.1.1/24
set security zones security-zone private address-book address phone1 10.1.1.3/32
set security zones security-zone private address-book address cm-tftp_server 10.1.1.4/32
set security zones security-zone private interfaces ge-0/0/0.0
```

```

set security zones security-zone public address-book address phone2 1.1.1.4/32
set security zones security-zone public interfaces ge-0/0/1.0
set security nat source rule-set phones from zone private
set security nat source rule-set phones to zone public
set security nat source rule-set phones rule phone1 match source-address 10.1.1.3/32
set security nat source rule-set phones rule phone1 then source-nat interface
set security nat static rule-set to-proxy from zone public
set security nat static rule-set to-proxy rule phone2 match destination-address 1.1.1.2/32
set security nat static rule-set to-proxy rule phone2 then static-nat prefix 10.1.1.4/32
set security nat proxy-arp interface ge-0/0/1.0 address 1.1.1.2/32
set security policies from-zone public to-zone private policy in-pol match source-address
  phone2
set security policies from-zone public to-zone private policy in-pol match
  destination-address cm-tftp_server
set security policies from-zone public to-zone private policy in-pol match
  destination-address phone1
set security policies from-zone public to-zone private policy in-pol match application
  junos-sccp
set security policies from-zone public to-zone private policy in-pol then permit
set security policies from-zone private to-zone public policy out-pol match source-address
  any
set security policies from-zone private to-zone public policy out-pol match
  destination-address phone2
set security policies from-zone private to-zone public policy out-pol match application
  junos-sccp
set security policies from-zone private to-zone public policy out-pol then permit
set security policies from-zone private to-zone private policy tftp-pol match
  source-address any
set security policies from-zone private to-zone private policy tftp-pol match
  destination-address any
set security policies from-zone private to-zone private policy tftp-pol match application
  junos-tftp
set security policies from-zone private to-zone private policy tftp-pol then permit

```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure NAT for an SCCP ALG Call Manager or a TFTP server located in a private zone:

1. Configure interfaces.

```

[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 1.1.1.1/24

```

2. Create zones and assign addresses to them.

```

[edit security zones]
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone private address-book address phone1 10.1.1.3/32
user@host# set security-zone private address-book address cm-tftp_server
  10.1.1.4/32
user@host# set security-zone public interfaces ge-0/0/1.0
user@host# set security-zone public address-book address phone2 1.1.1.4/32

```



3. Create a rule set for static NAT and assign a rule to it.
 

```
[edit security nat static rule-set to-proxy]
user@host# set from zone public
user@host#set rule phone2 match destination-address 1.1.1.2/32
user@host#set rule phone2 then static-nat prefix 10.1.1.4/32
```
4. Configure proxy ARP.
 

```
[edit security nat]
user@host#set proxy-arp interface ge-0/0/1.0 address 1.1.1.2/32
```
5. Configure interface NAT for communication from phone1 to phone2.
 

```
[edit security nat source rule-set phones]
user@host#set from zone private
user@host#set to zone public
user@host#set rule phone1 match source-address 10.1.1.3/32
user@host#set rule phone1 then source-nat interface
```
6. Configure a policy to allow traffic from the public zone to the private zone.
 

```
[edit security policies from-zone public to-zone private policy in-pol]
user@host# set match source-address phone2
user@host# set match destination-address cm-tftp_server
user@host# set match destination-address phone1
user@host# set match application junos-sccc
user@host# set then permit
```
7. Configure a policy to allow traffic from the private zone to the public zone.
 

```
[edit security policies from-zone private to-zone public policy out-pol]
user@host# set match source-address any
user@host# set match destination-address phone2
user@host# set match application junos-sccc
user@host# set then permit
```
8. Configure a policy to allow traffic from phone1 to the CM/TFTP server.
 

```
[edit security policies from-zone private to-zone private policy tftp-pol]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application junos-tftp
user@host# set then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show security zones**, **show security nat**, and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 10.1.1.1/24;
    }
  }
}
```

```
ge-0/0/1 {
  unit 0 {
    family inet {
      address 1.1.1/24;
    }
  }
}
[edit]
user@host# show security zones
security-zone private {
  address-book {
    address phone1 10.1.1.3/32;
    address cm-tftp_server 10.1.1.4/32;
  }
  interfaces {
    ge-0/0/0.0;
  }
}
security-zone public {
  address-book {
    address phone2 1.1.1.4/32;
  }
  interfaces {
    ge-0/0/1.0;
  }
}
[edit]
user@host# show security nat
source {
  rule-set phones {
    from zone private;
    to zone public;
    rule phone1 {
      match {
        source-address 10.1.1.3/32;
      }
      then {
        source-nat {
          interface;
        }
      }
    }
  }
}
static {
  rule-set to-proxy {
    from zone public;
    rule phone2 {
      match {
        destination-address 1.1.1.2/32;
      }
      then {
        static-nat prefix 10.1.1.4/32;
      }
    }
  }
}
```

```

}
proxy-arp {
  interface ge-0/0/1.0 {
    address {
      1.1.1.2/32;
    }
  }
}
[edit]
user@host# show security policies
from-zone public to-zone private {
  policy in-pol {
    match {
      source-address phone2;
      destination-address cm-tftp_server;
      destination-address phone1;
      application junos-sccp;
    }
    then {
      permit;
    }
  }
}
from-zone private to-zone public {
  policy out-pol {
    match {
      source-address any;
      destination-address phone2;
      application junos-sccp;
    }
    then {
      permit;
    }
  }
}
from-zone private to-zone private {
  policy tftp-pol {
    match {
      source-address any;
      destination-address any;
      application junos-tftp;
    }
    then {
      permit;
    }
  }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Source NAT Rule Usage on page 342
- Verifying Static NAT Configuration on page 342
- Verifying SCCP ALG on page 342

---

### Verifying Source NAT Rule Usage

**Purpose** Verify that there is traffic matching the source NAT rule.

**Action** From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

---

### Verifying Static NAT Configuration

**Purpose** Verify that there is traffic matching the static NAT rule set.

**Action** From operational mode, enter the **show security nat static rule** command. View the Translation hits field to check for traffic that matches the rule.

---

### Verifying SCCP ALG

**Purpose** Verify that the SCCP ALG is enabled.

**Action** From operational mode, enter the **show security alg status | match sccp** command.

**Related Documentation** • [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

---

## Verifying SCCP ALG Configurations

- Verifying SCCP ALGs on page 342
- Verifying SCCP Calls on page 343
- Verifying SCCP Call Details on page 343
- Verifying SCCP Counters on page 344

## Verifying SCCP ALGs

**Purpose** Display SCCP verification options.

**Action** From the CLI, enter the **show security alg sccp** command.

```
user@host> show security alg sccp ?
Possible completions:
  calls           Show SCCP calls
  counters        Show SCCP counters
```

**Meaning** The output shows a list of all SCCP verification parameters. Verify the following information:

- All SCCP calls
- Counters for all SCCP calls

## Verifying SCCP Calls

**Purpose** Display a list of all SCCP calls

**Action** From the CLI, enter the **show security alg sccp calls** command.

```
user@host> show security alg sccp calls
Possible completions:
  calls          Show SCCP calls
  counters       Show SCCP counters
  endpoints      Show SCCP endpoints
```

**Meaning** The output shows a list of all SCCP verification parameters. Verify the following information:

- All SCCP calls
- Counters for all SCCP calls
- Information about all SCCP endpoints

## Verifying SCCP Call Details

**Purpose** Display details about all SCCP calls.

**Action** From the CLI, enter the **show security alg sccp calls detail** command.

```
user@host> show security alg sccp calls detail
Client IP address: 11.0.102.91
Client zone: 7
CallManager IP: 13.0.99.226
Conference ID: 16789504
Resource manager group: 2048
SCCP channel information:
Media transmit channel address (IP address/Port): 0.0.0.0:0
Media transmit channel translated address (IP address/Port): 0.0.0.0:0
Media transmit channel pass-through party ID (PPID): 0
Media transmit channel resource ID: 0
Media receive channel address (IP address/Port): 11.0.102.91:20060
Media receive channel translated address (IP address/Port): 25.0.0.1:1032
Media receive channel pass-through party ID (PPID): 16934451
Media receive channel resource ID: 8185
Multimedia transmit channel address (IP address/Port): 0.0.0.0:0
Multimedia transmit channel translated address (IP address/Port): 0.0.0.0:0
Multimedia transmit channel pass-through party ID (PPID): 0
Multimedia transmit channel resource ID: 0
Multimedia receive channel address (IP address/Port): 0.0.0.0:0
Multimedia receive channel translated address (IP address/Port): 0.0.0.0:0
Multimedia receive channel pass-through party ID (PPID): 0
```

```

Multimedia receive channel resource ID: 0
Total number of calls = 1

```

**Meaning** The output shows a list of all SCCP verification parameters. Verify the following information:

- Client zone
- CallManager IP address: 13.0.99.226
- Conference ID
- Resource manager group
- SCCP channel information
- Total number of calls

## Verifying SCCP Counters

**Purpose** Display a list of all SCCP counters

**Action** From the J-Web interface, select **Monitor>ALGs>SCCP>Counters**. Alternatively, from the CLI, enter the **show security alg sccp counters** command.

```
user@host> show security alg sccp counters
```

```
SCCP call statistics:
```

```

Active client sessions      : 0
Active calls                : 0
Total calls                 : 0
Packets received           : 0
PDUs processed              : 0
Current call rate           : 0

```

```
Error counters:
```

```

Packets dropped              : 0
Decode errors               : 0
Protocol errors              : 0
Address translation errors   : 0
Policy lookup errors        : 0
Unknown PDUs                : 0
Maximum calls exceeded      : 0
Maximum call rate exceeded   : 0
Initialization errors       : 0
Internal errors              : 0
Nonspecific error           : 0
No active calls to delete    : 0
No active client sessions to delete : 0
Session cookie create errors : 0
Invalid NAT cookie detected  : 0

```

**Meaning** The output shows a list of all SCCP verification parameters. Verify the following information:

- SCCP call statistics
- Error counters

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- SCCP ALG Configuration Overview on page 330
- Example: Configuring the SCCP ALG Call Manager or TFTP Server in the Private Zone on page 336





## CHAPTER 14

# MGCP ALGs

- Understanding MGCP ALGs on page 347
- MGCP ALG Configuration Overview on page 353
- MGCP ALG Call Duration and Timeouts on page 353
- MGCP ALG DoS Attack Protection on page 358
- MGCP ALG Unknown Message Types on page 359
- Example: Configuring Media Gateways in Subscriber Homes Using MGCP ALGs on page 361
- Example: Configuring Three-Zone ISP-Hosted Service Using MGCP ALG and NAT on page 368

### Understanding MGCP ALGs

---

The Media Gateway Control Protocol (MGCP) is a text-based Application Layer protocol used for call setup and call control between the media gateway and the media gateway controller (MGC).

The protocol is based on a master/slave call control architecture: the MGC (call agent) maintains call control intelligence, and media gateways carry out the instructions from the call agent. Both signaling packets and media packets are transmitted over UDP. Junos OS supports MGCP in route mode and Network Address Translation (NAT) mode.

The MGCP Application Layer Gateway (ALG) performs the following procedures:

- Conducts voice-over-IP (VoIP) signaling payload inspection. The payload of the incoming VoIP signaling packet is fully inspected based on related RFCs and proprietary standards. Any malformed packet attack is blocked by the ALG.
- Conducts MGCP signaling payload inspection. The payload of the incoming MGCP signaling packet is fully inspected in accordance with RFC 3435. Any malformed-packet attack is blocked by the ALG.
- Provides stateful processing. The corresponding VoIP-based state machines are invoked to process the parsed information. Any out-of-state or out-of-transaction packet is identified and properly handled.

- Performs NAT. Any embedded IP address and port information in the payload is properly translated based on the existing routing information and network topology, and is then replaced with the translated IP address and port number, if necessary.
- Manages pinholes for VoIP traffic. To keep the VoIP network secure, the IP address and port information used for media or signaling is identified by the ALG, and any needed pinhole is dynamically created and closed during call setup.

This topic contains the following sections:

- MGCP Security on page 348
- Entities in MGCP on page 348
- Commands on page 350
- Response Codes on page 352

## MGCP Security

The MGCP ALG includes the following security features:

- Denial-of-service (DoS) attack protection. The ALG performs stateful inspection at the UDP packet level, the transaction level, and the call level. MGCP packets matching the RFC 3435 message format, transaction state, and call state, are processed. All other messages are dropped.
- Security policy enforcement between gateway and gateway controller (signaling policy).
- Security policy enforcement between gateways (media policy).
- Per-gateway MGCP message flooding control. Any malfunctioning or hacked gateway will not disrupt the whole VoIP network. Combined with per-gateway flooding control, damage is contained within the impacted gateway.
- Per-gateway MGCP connection flooding control.
- Seamless switchover/failover if calls, including calls in progress, are switched to the standby firewall in case of system failure.

## Entities in MGCP

There are four basic entities in MGCP:

- Endpoint on page 348
- Connection on page 349
- Call on page 349
- Call Agent on page 349

### Endpoint

---

A media gateway is a collection of endpoints. An endpoint can be an analog line, trunk, or any other access point. An endpoint contains the following elements:

`local-endpoint-name@domain-name`

The following examples are some valid endpoint IDs:

```
group1/Trk8@mynetwork.net
group2/Trk1/*@[192.168.10.8] (wild-carding)
$@voiptel.net (any endpoint within the media gateway)
*@voiptel.net (all endpoints within the media gateway)
```

### Connection

---

Connections are created on each endpoint by an MG during call setup. A typical VoIP call involves two connections. A complex call, for example a three-party call or conference call, might require more connections. The MGC can instruct media gateways to create, modify, delete, and audit a connection.

A connection is identified by its connection ID, which is created by the MG when it is requested to create a connection. Connection ID is presented as a hexadecimal string, and its maximum length is 32 characters.

### Call

---

A call is identified by its call ID, which is created by the MGC when establishing a new call. Call ID is a hexadecimal string with a maximum length of 32 characters. Call ID is unique within the MGC. Two or more connections can have the same call ID if they belong to the same call.

### Call Agent

---

One or more call agents (also called media gateway controllers) are supported in MGCP to enhance reliability in the VoIP network. The following two examples are of call agent names:

```
CallAgent@voipCA.mynetwork.com
voipCA.mynetwork.com
```

Several network addresses can be associated under one domain name in the Domain Name System (DNS). By keeping track of the time to live (TTL) of DNS query/response data and implementing retransmission using other alternative network addresses, switchover and failover is achieved in MGCP.

The concept of a *notified entity* is essential in MGCP. The notified entity for an endpoint is the call agent currently controlling that endpoint. An endpoint should send any MGCP command to its notified entity. However, different call agents might send MGCP commands to this endpoint.

The notified entity is set to a provisioned value upon startup, but can be changed by a call agent through the use of the **NotifiedEntity** parameter contained in an MGCP message. If the notified entity for an endpoint is empty or has not been set explicitly, its value defaults to the source address of the last successful non-audit MGCP command received for that endpoint.

## Commands

The MGCP protocol defines nine commands for controlling endpoints and connections. All commands are composed of a command header, optionally followed by Session Description Protocol (SDP) information. A command header has the following elements:

- A command line: command verb + transaction ID + endpointId + MGCP version.
- Zero or more parameter lines, composed of a parameter name followed by a parameter value.

Table 37 on page 350 lists supported MGCP commands and includes a description of each, the command syntax, and examples. Refer to RFC 2234 for a complete explanation of command syntax.

**Table 37: MGCP Commands**

Command	Description	Command Syntax	Example
EPCF	EndpointConfiguration—Used by a call agent to inform a gateway of coding characteristics (a-law or mu-law) expected by the line side of the endpoint.	ReturnCode [PackageList] EndpointConfiguration (EndpointId,[BearerInformation])	EPCF 2012 wxx/T2@mynet.com MGCP 1.0B: e:mu
CRCX	CreateConnection—Used by a call agent to instruct the gateway to create a connection with, and endpoint inside, the gateway.	ReturnCode, [ConnectionId,] [SpecificEndPointId,] [LocalConnectionDescriptor,] [SecondEndPointId,] [SecondConnectionId,] [PackageList] CreateConnection (CallId, EndpointId, [NotifiedEntity,] [LocalConnectionOption,] Mode, [RemoteConnectionDescriptor   SecondEndPointId,] [encapsulated RQNT,] [encapsulated EPCF])	CRCX 1205 aaln/1@gw-25.att.net MGCP 1.0C: A3C47F21456789F0L: p:10, a:PCMUM: sendrecvX: 0123456789ADR: L/hdS: L/rgv=0o=- 25678 753849 IN IP4 128.96.41.1s=-c=IN IP4 128.96.41.1t=0 Om=audio 3456 RTP/AVP 0
MDCX	ModifyConnection—Used by a call agent to instruct a gateway to change the parameters for an existing connection.	ReturnCode, [LocalConnectionDescriptor,] [PackageList] ModifyConnection (CallId, EndpointId, ConnectionId, [NotifiedEntity,] [LocalConnectionOption,] [Mode,]  [RemoteConnectionDescriptor,] [encapsulated RQNT,] [encapsulated EPCF])	MDCX 1210 aaln/1@rgw-25.att.net MGCP 1.0C: A3C47F21456789F0I: FDE234C8M: recvonlyX: 0123456789AER: L/huS: G/rtv=0o=- 4723891 7428910 IN IP4 128.96.63.25s=-c=IN IP4 128.96.63.25t=0 Om=audio 3456 RTP/AVP 0

Table 37: MGCP Commands (continued)

Command	Description	Command Syntax	Example
DLCX	<p>DeleteConnection—Used by a call agent to instruct a gateway to delete an existing connection.</p> <p>DeleteConnection can also be used by a gateway to release a connection that can no longer be sustained.</p>	<b>ReturnCode, ConnectionParameters, [PackageList] DeleteConnection (CallId, EndpointId, ConnectionId, [NotifiedEntity,] [encapsulated RQNT,] [encapsulated EPCF])</b>	<p>Example 1: MGC -&gt; MG</p> <p>DLCX 9210 aaln/1@rgw-25.att.net MGCP 1.0C: A3C47F21456789F0I: FDE234C8</p> <p>Example 2: MG -&gt; MGC</p> <p>DLCX 9310 aaln/1@rgw-25.att.net MGCP 1.0C: A3C47F21456789F0I: FDE234C8E: 900 - Hardware errorP: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48</p>
RQNT	<p>NotificationRequest command—Used by a call agent to instruct an MG to monitor for certain event(s) or signal(s) for a specific endpoint.</p>	<b>ReturnCode, [PackageList] NotificationRequest([EndpointId, [NotifiedEntity,] [RequestedEvents,] RequestIdentifier, [DigitMap,] [SignalRequests,] [QuarantineHandling,] [DetectEvents,] [encapsulated EPCF])</b>	<p>RQNT 1205 aaln/1@rgw-25.att.net MGCP 1.0N: ca-new@callagent-ca.att.netX: 0123456789AAR: L/hd(A, E(S(L/d),R(L/oc,L/hu,D/[0-9#*T](D))))D: (0T 00T xx 9 xxxxxxxxxx 90 ix.T)S:T: G/ft</p>
NTFY	<p>Notify—Used by a gateway to inform the call agent when requested event(s) or signal(s) occur.</p>	<b>ReturnCode, [PackageList] Notify (EndpointID, [NotifiedEntity,] RequestIdentifier, ObservedEvents)</b>	<p>NTFY 2002 aaln/1@rgw-25.att.net MGCP 1.0N: ca@ca1.att.net:5678X: 0123456789ACO: L/hd,D/9,D/1,D/2,D/0,D/1,D/8,D/2,D/9,D/4, D/2,D/6,D/6</p>
AUEP	<p>AuditEndpoint—Used by a call agent to audit the status of the endpoint.</p>	<b>ReturnCode, EndPointIdList,   { [RequestedEvents,] [QuarantineHandling,] [DigitMap,] [SignalRequests,] [RequestedIdentifier,] [NotifiedEntity,] [ConnectionIdentifier,] [DetectEvents,] [ObservedEvents,] [EventStats,] [BearerInformation,] [BearerMethod,] [RestartDelay,] [ReasonCode,] [MaxMGCPDatagram,] [Capabilities]} [PackageList] AuditEndpoint (EndpointId, [RequestedInfo])</b>	<p>Example 1:</p> <p>AUEP 1201 aaln/1@rgw-25.att.net MGCP 1.0F: A, R,D,S,X,N,I,T,O</p> <p>Example 2:</p> <p>AUEP 1200 *@rgw-25.att.net MGCP 1.0</p>
AUCX	<p>AuditConnection—Used by a call agent to collect the parameters applied to a connection.</p>	<b>ReturnCode, [CallId,] [NotifiedEntity,] [LocalConnectionOptions,] [Mode,] [RemoteConnectionDescriptor,] [LocalConnectionDescriptor,] [ConnectionParameters,] [PackageList] AuditConnection (EndpointId, ConnectionId, RequestedInfo)</b>	<p>AUCX 3003 aaln/1@rgw-25.att.net MGCP 1.0I: 32F345E2F: C,N,L,M,LC,P</p>
RSIP	<p>RestareInProgress—Used by a gateway to notify a call agent that one or more endpoints are being taken out of service or placed back in service.</p>	<b>ReturnCode, [NotifiedEntity,] [PackageList] RestartInProgress (EndpointId, RestartMethod, [RestartDelay,] [ReasonCode])</b>	<p>RSIP 5200 aaln/1@rg2-25.att.net MGCP 1.0RM: gracefulRD: 300</p>

## Response Codes

Every command sent by the calling agent or gateway, whether successful or not, requires a response code. The response code is in the header of the response message, and optionally is followed by session description information.

The response header is composed of a response line, followed by zero or more parameter lines, each containing a parameter name letter followed by its value. The response header is composed of a three-digit response code, transaction ID, and optionally followed by commentary. The response header in the following response message shows response code 200 (successful completion), followed by ID 1204 and the comment:OK.

```
200 1204 OK
I: FDE234C8
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 96
a=rtpmap:96 G726-32/8000
```

The ranges of response codes are defined as follows:

- 000 — 099 indicate a response acknowledgement.
- 100 — 199—indicate a provisional response.
- 200 — 299 indicate a successful completion (final response).
- 400 — 499 indicate a transient error (final response).
- 500 — 599 indicate a permanent error (final response).

Refer to RFC 3661 for detailed information about response codes.

A response to a command is sent to the source address of the command, not to the current notified entity. A media gateway can receive MGCP commands from various network addresses simultaneously, and send back responses to corresponding network addresses. However, it sends all MGCP commands to its current notified entity.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [ALG Overview on page 217](#)
- [MGCP ALG Configuration Overview on page 353](#)
- [Example: Configuring Media Gateways in Subscriber Homes Using MGCP ALGs on page 361](#)
- [Example: Configuring Three-Zone ISP-Hosted Service Using MGCP ALG and NAT on page 368](#)

## MGCP ALG Configuration Overview

The Media Gateway Control Protocol (MGCP ALG) is enabled by default on the device—no action is required to enable it. However, you might choose to fine-tune MGCP ALG operations by using the following instructions:

1. Free up bandwidth when calls fail to properly terminate. See “Example: Setting MGCP ALG Call Duration” on page 354.
2. Control how long a call can remain active without any media traffic. See “Example: Setting MGCP ALG Inactive Media Timeout” on page 355.
3. Track and clear signaling traffic when it times out. See “Example: Setting MGCP ALG Transaction Timeout” on page 357.
4. Protect the media gateway from denial-of-service (DoS) flood attacks. See “Example: Configuring MGCP ALG DoS Attack Protection” on page 358.
5. Enable unknown messages to pass when the session is in Network Address Translation (NAT) mode and route mode. See “Example: Allowing Unknown MGCP ALG Message Types” on page 360.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding MGCP ALGs on page 347](#)
- [Example: Configuring Media Gateways in Subscriber Homes Using MGCP ALGs on page 361](#)
- [Example: Configuring Three-Zone ISP-Hosted Service Using MGCP ALG and NAT on page 368](#)

## MGCP ALG Call Duration and Timeouts

- [Understanding MGCP ALG Call Duration and Timeouts on page 353](#)
- [Example: Setting MGCP ALG Call Duration on page 354](#)
- [Example: Setting MGCP ALG Inactive Media Timeout on page 355](#)
- [Example: Setting MGCP ALG Transaction Timeout on page 357](#)

### Understanding MGCP ALG Call Duration and Timeouts

The call duration feature gives you control over Media Gateway Control Protocol (MGCP) call activity and helps you to manage network resources.

Typically a Delete Connection (DLCX) message will be sent out to delete a connection. The MGCP Application Layer Gateway (ALG) intercepts it and removes all media sessions for that connection.

A call can have one or more voice channels. Each voice channel has two sessions (or two media streams), one for Real-Time Transport Protocol (RTP) traffic and one for Real-Time

Control Protocol (RTCP) signaling. When managing the sessions, the device considers the sessions in each voice channel as one group. Timeouts and call duration settings apply to a group as opposed to each session.

The following parameters govern MGCP call activity:

- **maximum-call-duration**—This parameter sets the absolute maximum length of a call. When a call exceeds this parameter setting, the MGCP ALG tears down the call and releases the media sessions. The default setting is 43200 seconds, and the range is from 180 through 432000 seconds. This setting also frees up bandwidth in cases where calls fail to properly terminate.
- **inactive-media-timeout**—This parameter indicates the maximum length of time (in seconds) a call can remain active without any media (RTP or RTCP) traffic within a group. Each time an RTP or RTCP packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the MGCP ALG gates opened for media are closed. The default setting is 120 seconds, and the range is 10 through 2550 seconds. Note that upon timeout, while resources for media (sessions and pinholes) are removed, the call is not terminated.
- **transaction-timeout**—A transaction is a signaling message, for example, an NTFY from the gateway to the call agent or a 200 OK from the call agent to the gateway. The Juniper Networks device tracks these transactions and clears them when they time out. The timeout range for MGCP transactions is 3 through 50 seconds and the default is 30 seconds.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding MGCP ALGs on page 347](#)
- [MGCP ALG Configuration Overview on page 353](#)
- [Example: Setting MGCP ALG Call Duration on page 354](#)
- [Example: Setting MGCP ALG Inactive Media Timeout on page 355](#)
- [Example: Setting MGCP ALG Transaction Timeout on page 357](#)

### Example: Setting MGCP ALG Call Duration

This example shows how to set call duration for the MGCP ALG.

- [Requirements on page 354](#)
- [Overview on page 355](#)
- [Configuration on page 355](#)
- [Verification on page 355](#)

#### Requirements

---

Before you begin, determine the type of parameter used to control the MGCP call activity and manage its network resources. See “Understanding MGCP ALG Call Duration and Timeouts” on page 353.



## Overview

The **maximum-call-duration** parameter governs MGCP call activity and sets the absolute maximum length of a call. When a call exceeds this parameter setting, the MGCP ALG tears down the call and releases the media sessions. The default setting is 432000 seconds, and the range is 180 through 432000 seconds. This setting also frees up bandwidth in cases where calls fail to properly terminate. In this example, the call duration is set to 180000 seconds.

## Configuration

### J-Web Quick Configuration

To set call duration for the MGCP ALG:

1. Select **Configure>Security>ALG**.
2. Select the **MGCP** tab.
3. In the Maximum call duration box, enter **3000**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

### Step-by-Step Procedure

To set call duration for the MGCP ALG:

1. Configure the MGCP ALG call duration.
 

```
[edit]
user@host# set security alg mgcp maximum-call-duration 3000
```
2. If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

## Verification

To verify the configuration is working properly, enter the **show security alg mgcp** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding MGCP ALG Call Duration and Timeouts on page 353](#)
- [MGCP ALG Configuration Overview on page 353](#)

## Example: Setting MGCP ALG Inactive Media Timeout

This example shows how to set the inactive media timeout value for the MGCP ALG.

- [Requirements on page 356](#)
- [Overview on page 356](#)
- [Configuration on page 356](#)
- [Verification on page 356](#)

## Requirements

---

Before you begin, determine the type of parameter used to control the MGCP call activity and manage its network resources. See “Understanding MGCP ALG Call Duration and Timeouts” on page 353.

## Overview

---

The **inactive-media-timeout** parameter governs MGCP call activity and indicates the maximum length of time (in seconds) a call can remain active without any media (RTP or RTCP) traffic within a group. Each time an RTP or RTCP packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the MGCP ALG gates opened for media are closed. The default setting is 120 seconds, and the range is from 10 to 2550 seconds. Note that upon timeout, while resources for media (sessions and pinholes) are removed, the call is not terminated. In this example, the inactive media timeout is set to 90 seconds.

## Configuration

---

### J-Web Quick Configuration

To set the inactive media timeout for the MGCP ALG:

1. Select **Configure>Security>ALG**.
2. Select the **MGCP** tab.
3. In the Inactive Media Timeout box, enter **90**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

### Step-by-Step Procedure

To set the inactive media timeout for the MGCP ALG:

1. Configure the MGCP ALG inactive media timeout value.  

```
[edit]  
user@host# set security alg mgcp inactive-media-timeout 90
```
2. If you are done configuring the device, commit the configuration.  

```
[edit]  
user@host# commit
```

## Verification

---

To verify the configuration is working properly, enter the **show security alg mgcp** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding MGCP ALG Call Duration and Timeouts on page 353](#)
- [MGCP ALG Configuration Overview on page 353](#)

## Example: Setting MGCP ALG Transaction Timeout

This example shows how to set the transaction timeout for the MGCP ALG.

- Requirements on page 357
- Overview on page 357
- Configuration on page 357
- Verification on page 357

### Requirements

Before you begin, determine the type of parameter used to control the MGCP call activity and manage its network resources. See “Understanding MGCP ALG Call Duration and Timeouts” on page 353.

### Overview

The **transaction-timeout** parameter governs MGCP call activity and is a signaling message; for example, a NTFY from the gateway to the call agent or a 200 OK from the call agent to the gateway. The Juniper Networks device tracks these transactions, and clears them when they time out. The timeout range for MGCP transactions is from 3 to 50 seconds, and the default is 30 seconds. In this example, the transaction timeout is set to 20 seconds.

### Configuration

#### J-Web Quick Configuration

To set the transaction timeout for the MGCP ALG:

1. Select **Configure>Security>ALG**.
2. Select the **MGCP** tab.
3. In the Transaction Timeout box, enter **20**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

#### Step-by-Step Procedure

To set the transaction timeout for the MGCP ALG:

1. Configure the MGCP ALG transaction timeout value.
 

```
[edit]
user@host# set security alg mgcp transaction-timeout 20
```
2. If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security alg mgcp** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding MGCP ALG Call Duration and Timeouts on page 353](#)
  - [MGCP ALG Configuration Overview on page 353](#)

## MGCP ALG DoS Attack Protection

---

- [Understanding MGCP ALG DoS Attack Protection on page 358](#)
- [Example: Configuring MGCP ALG DoS Attack Protection on page 358](#)

### Understanding MGCP ALG DoS Attack Protection

You can protect the Media Gateway Control Protocol (MGCP) media gateway from denial-of-service (DoS) flood attacks by limiting the number of remote access service (RAS) messages and connections per second it will attempt to process.

When you configure MGCP message flood protection, the MGCP Application Layer Gateway (ALG) drops any messages exceeding the threshold you set. The range is 2 to 50,000 messages per second per media gateway, and the default is 1000 messages per second per media gateway.

When you configure MGCP connection flood protection, the MGCP ALG drops any connection request exceeding the threshold you set. This limits the rate of processing of **CreateConnection (CRCX)** commands, thereby indirectly limiting pinhole creation. The range is 2 to 10,000 connection requests per second per media gateway, the default is 200.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding MGCP ALGs on page 347](#)
  - [MGCP ALG Configuration Overview on page 353](#)
  - [Example: Configuring MGCP ALG DoS Attack Protection on page 358](#)

### Example: Configuring MGCP ALG DoS Attack Protection

This example shows how to configure connection flood protection for the MGCP ALG.

- [Requirements on page 358](#)
- [Overview on page 359](#)
- [Configuration on page 359](#)
- [Verification on page 359](#)

#### Requirements

---

Before you begin, determine whether to protect the MGCP media gateway from DoS flood attacks. See “Understanding MGCP ALG DoS Attack Protection” on page 358.

---

## Overview

In this example, you configure the MGCP ALG to drop any message requests exceeding 10,000 requests per second and to drop any connection requests exceeding 4000 per second.

---

## Configuration

### J-Web Quick Configuration

To configure connection flood protection for the MGCP ALG:

1. Select **Configure>Security>ALG**.
2. Select the **MGCP** tab.
3. In the Message flood gatekeeper threshold box, type **10000**.
4. In the Connection flood threshold box, type **4000**.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

### Step-by-Step Procedure

To configure connection flood protection for the MGCP ALG:

1. Configure the connection flood threshold value.

```
[edit]
user@host# set security alg mgcp application-screen message-flood threshold
10000
user@host# set security alg mgcp application-screen connection-flood threshold
4000
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

---

## Verification

To verify the configuration is working properly, enter the **show security alg mgcp** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding MGCP ALG DoS Attack Protection on page 358](#)
- [MGCP ALG Configuration Overview on page 353](#)

---

## MGCP ALG Unknown Message Types

- [Understanding MGCP ALG Unknown Message Types on page 359](#)
- [Example: Allowing Unknown MGCP ALG Message Types on page 360](#)

### Understanding MGCP ALG Unknown Message Types

To accommodate on-going development of the Media Gateway Control Protocol (MGCP), you might want to allow traffic containing new MGCP message types. The unknown

MGCP message type feature enables you to configure the Juniper Networks device to accept MGCP traffic containing unknown message types in both Network Address Translation (NAT) mode and route mode.

This feature enables you to specify how unidentified MGCP messages are handled by the Juniper Networks device. The default is to drop unknown (unsupported) messages. Unknown messages can compromise security. However, in a secure test or production environment, this command can be useful for resolving interoperability issues with disparate vendor equipment. Permitting unknown MGCP messages can help you get your network operational so that you can later analyze your voice-over-IP (VoIP) traffic to determine why some messages were being dropped.

Note that this command applies only to received packets identified as supported VoIP packets. If a packet cannot be identified, it is always dropped. If a packet is identified as a supported protocol and you have configured the device to permit unknown message types, the message is forwarded without processing.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding MGCP ALGs on page 347](#)
- [MGCP ALG Configuration Overview on page 353](#)
- [Example: Allowing Unknown MGCP ALG Message Types on page 360](#)

## Example: Allowing Unknown MGCP ALG Message Types

This example shows how to configure the MGCP ALG to allow unknown MGCP message types in both NAT mode and route mode.

- [Requirements on page 360](#)
- [Overview on page 360](#)
- [Configuration on page 361](#)
- [Verification on page 361](#)

### Requirements

---

Before you begin, determine whether to accommodate new and unknown MGCP message types for the device. See “Understanding MGCP ALG Unknown Message Types” on page 359.

### Overview

---

This feature enables you to specify how unidentified MGCP messages are handled by a Juniper Networks device. The default is to drop unknown (unsupported) messages, because unknown messages can compromise security. However, in a secure test or production environment, this command can be useful for resolving interoperability issues with disparate vendor equipment.

## Configuration

---

### J-Web Quick Configuration

To configure the MGCP ALG to allow unknown message types:

1. Select **Configure>Security>ALG**.
2. Select the **MGCP** tab.
3. Select the **Enable Permit NAT applied** check box.
4. Select the **Enable Permit routed** check box.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

### Step-by-Step Procedure

To configure the MGCP ALG to allow unknown message types:

1. Allow unknown message types to pass if the session is in either NAT mode or in route mode.

[edit]

```
user@host# set security alg mgcp application-screen unknown-message
permit-nat-applied permit-routed
```

2. If you are done configuring the device, commit the configuration.

[edit]

```
user@host# commit
```

## Verification

---

To verify the configuration is working properly, enter the **show security alg mgcp** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding MGCP ALG Unknown Message Types on page 359](#)
- [MGCP ALG Configuration Overview on page 353](#)

## Example: Configuring Media Gateways in Subscriber Homes Using MGCP ALGs

---

This example shows how to configure media gateways in subscriber homes using MGCP ALGs.

- [Requirements on page 361](#)
- [Overview on page 362](#)
- [Configuration on page 363](#)
- [Verification on page 366](#)

## Requirements

Before you begin:

- Configure zones. See “Example: Creating Security Zones” on page 114.

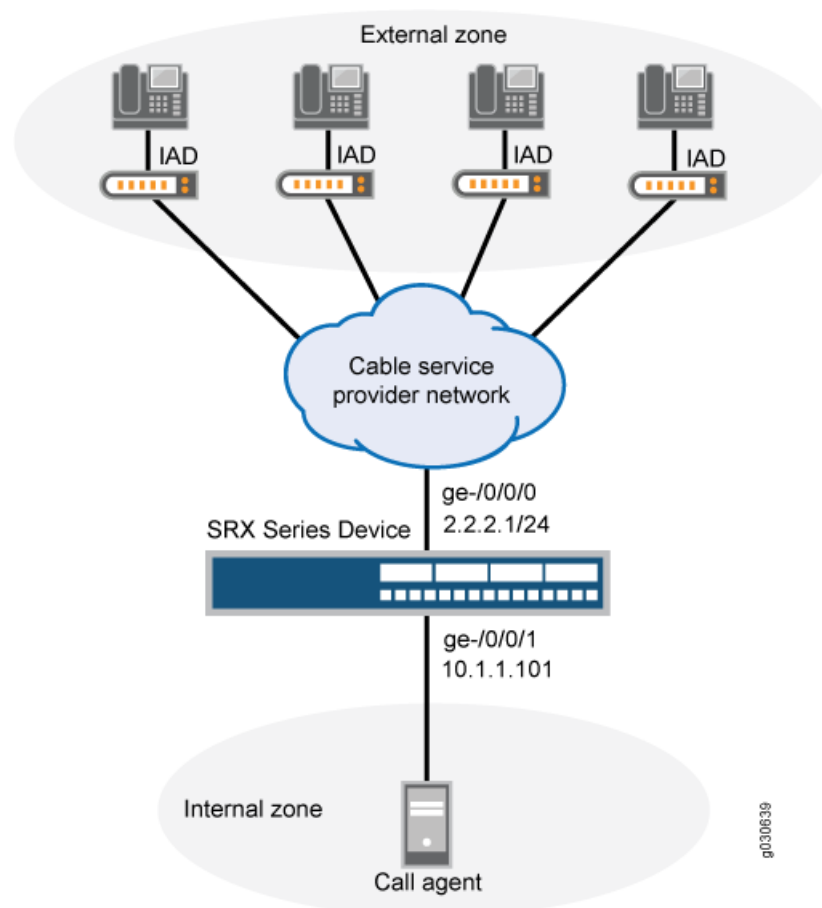
- Configure addresses and interfaces. See “Example: Configuring Address Books and Address Sets” on page 139.
- Configure security policies. See “Security Policies Configuration Overview” on page 151.

## Overview

When a cable service provider offers MGCP services to residential subscribers, they locate the Juniper Networks device and call agent on their premises and install a set-top box, in each subscriber's home. The set-top boxes act as gateways for the residences.

After creating zones—`external_subscriber` for the customer and `internal_ca` for the service provider—you configure addresses, then interfaces, and finally policies to allow signaling between endpoints. Note that although gateways frequently reside in different zones, requiring policies for media traffic, in this example both gateways are in the same subnet. Note also that because RTP traffic between the gateways never passes through the device, no policy is needed for the media. See Figure 29 on page 362.

Figure 29: Media Gateway in Subscriber Homes





## Configuration

**CLI Quick Configuration** To quickly configure media gateways in subscriber homes using MGCP ALGs, copy the following commands and paste them into the CLI:

```
[edit]
set security zones security-zone external-subscriber host-inbound-traffic system-services
  all
set security zones security-zone external-subscriber host-inbound-traffic protocols all
set security zones security-zone internal-ca host-inbound-traffic system-services all
set security zones security-zone internal-ca host-inbound-traffic protocols all
set security zones security-zone internal-ca address-book address ca-agent-110.1.1.101/32
set security zones security-zone external-subscriber address-book address
  subscriber-subnet 2.2.2.1/24
set security zones security-zone external-subscriber interfaces ge-0/0/0
set interfaces ge-0/0/0 unit 0 family inet
set security zones security-zone internal-ca interfaces ge-0/0/1
set interfaces ge-0/0/1 unit 0 family inet address 2.2.2.1/24
set security policies from-zone internal-ca to-zone external-subscriber policy
  ca-to-subscribers match source-address ca-agent-1
set security policies from-zone internal-ca to-zone external-subscriber policy
  ca-to-subscribers match destination-address subscriber-subnet
set security policies from-zone internal-ca to-zone external-subscriber policy
  ca-to-subscribers match application junos-mgcp
set security policies from-zone internal-ca to-zone external-subscriber policy
  ca-to-subscribers then permit
set security policies from-zone external-subscriber to-zone internal-ca policy
  subscriber-to-ca match source-address subscriber-subnet
set security policies from-zone external-subscriber to-zone internal-ca policy
  subscriber-to-ca match destination-address ca-agent-1
set security policies from-zone external-subscriber to-zone internal-ca policy
  subscriber-to-ca match application junos-mgcp
set security policies from-zone external-subscriber to-zone internal-ca policy
  subscriber-to-ca then permit
set security policies from-zone internal-ca to-zone internal-ca policy intra-ca match
  source-address any
set security policies from-zone internal-ca to-zone internal-ca policy intra-ca match
  destination-address any
set security policies from-zone internal-ca to-zone internal-ca policy intra-ca match
  application any
set security policies from-zone internal-ca to-zone internal-ca policy intra-ca then permit
set security policies from-zone external-subscriber to-zone external-subscriber policy
  intra-subscriber match source-address any
set security policies from-zone external-subscriber to-zone external-subscriber policy
  intra-subscriber match destination-address any
set security policies from-zone external-subscriber to-zone external-subscriber policy
  intra-subscriber match application any
set security policies from-zone external-subscriber to-zone external-subscriber policy
  intra-subscriber then permit
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To configure media gateways in subscriber homes using MGCP ALGs:

1. Create security zones for the customer and for the service provider.

```
[edit security zones security-zone external-subscriber]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
```

```
[edit security zones security-zone internal-ca]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
```

2. Configure addresses for the zones.

```
[edit]
user@host# edit security zones security-zone internal-ca address-book address
ca-agent-1 10.1.1.101/32
user@host# set security zones security-zone external-subscriber address-book
address subscriber-subnet 2.2.2.1/24
```

3. Configure interfaces for the zones.

```
[edit]
user@host# edit security zones security-zone external-subscriber interfaces
ge-0/0/0
user@host# set interfaces ge-0/0/0 unit 0 family inet
user@host# set security zones security-zone internal-ca interfaces ge-0/0/1
user@host# set interfaces ge-0/0/1 unit 0 family inet address 2.2.2.1/24
```

4. Configure policies for traffic from the internal to the external zone.

```
[edit security policies from-zone internal-ca to-zone external-subscriber policy
ca-to-subscribers]
user@host# edit match source-address ca-agent-1
user@host# set match destination-address subscriber-subnet
user@host# set match application junos-mgcp
user@host# set then permit
```

5. Configure policies for traffic from the external to the internal zone.

```
[edit security policies from-zone external-subscriber to-zone internal-ca policy
subscriber-to-ca]
user@host# edit match source-address subscriber-subnet
user@host# set match destination-address ca-agent-1
user@host# set match application junos-mgcp
user@host# set then permit
```

6. Configure policies for traffic between two internal zones.

```
[edit security policies from-zone internal-ca to-zone internal-ca policy intra-ca]
user@host# edit match source-address any
user@host# set match destination-address any
user@host# set match application any
user@host# set then permit
```

7. Configure policies for traffic between two external zones.

```
[edit security policies from-zone external-subscriber to-zone external-subscriber
policy intra-subscriber]
user@host# edit match source-address any
user@host# set match destination-address any
user@host# set match application any
user@host# set then permit
```

**Results** From configuration mode, confirm your configuration by entering the `show security policies` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
from-zone internal-ca to-zone external-subscriber {
  policy ca-to-subscribers {
    match {
      source-address ca-agent-1;
      destination-address subscriber-subnet;
      application junos-mgcp;
    }
    then {
      permit;
    }
  }
}
from-zone external-subscriber to-zone internal-ca {
  policy subscriber-to-ca {
    match {
      source-address subscriber-subnet;
      destination-address ca-agent-1;
      application junos-mgcp;
    }
    then {
      permit;
    }
  }
}
from-zone internal-ca to-zone internal-ca {
  policy intra-ca {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone external-subscriber to-zone external-subscriber {
  policy intra-subscriber {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
```

```

        permit;
    }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform this task:

- Verifying MGCP ALGs on page 366
- Verifying MGCP ALG Calls on page 366
- Verifying MGCP ALG Endpoints on page 367
- Verifying MGCP ALG Counters on page 367

### Verifying MGCP ALGs

**Purpose** Verify the MGCP ALG verification options.

**Action** From operational mode, enter the **show security alg mgcp ?** command.

```

user@host> show security alg mgcp ?
Possible completions:
  calls           Show MGCP calls
  counters        Show MGCP counters
  endpoints       Show MGCP endpoints

```

**Meaning** The output shows a list of all MGCP verification parameters. Verify the following information:

- All MGCP calls
- Counters for all MGCP calls
- Information about all MGCP endpoints

### Verifying MGCP ALG Calls

**Purpose** Verify information about active MGCP calls.

**Action** From operational mode, enter the **show security alg mgcp calls** command.

```

user@host> show security alg mgcp calls
Endpoint@GW      Zone      Call ID      RM Group
d001@101.50.10.1 Trust     10d55b81140e0f76  512
  Connection Id> 0
  Local SDP>  o: 101.50.10.1      x_o: 101.50.10.1
                c: 101.50.10.1/32206  x_c: 101.50.10.1/32206
  Remote SDP> c: 3.3.3.5/16928  x_c: 3.3.3.5/16928
Endpoint@GW      Zone      Call ID      RM Group
d001@3.3.3.5    Untrust   3a104e9b41a7c4c9  511
  Connection Id> 0
  Local SDP>  o: 3.3.3.5          x_o: 3.3.3.5

```

```

c: 3.3.3.5/16928          x_c: 3.3.3.5/16928
Remote SDP> c: 101.50.10.1/32206  x_c: 101.50.10.1/32206

```

**Meaning** The output displays information about all MGCP calls. Verify the following information:

- Endpoint
- Zone
- Call identifier
- Resource Manager group

### Verifying MGCP ALG Endpoints

---

**Purpose** Verify information about MGCP endpoints.

**Action** From operational mode, enter the **show security alg mgcp endpoints** command.

```

user@host> show security alg mgcp endpoints
Gateway: 101.50.10.1 Zone: Trust IP: 101.50.10.1 -> 101.50.10.1
  Endpoint      Trans #  Call #  Notified Entity
  d001          1       1       0.0.0.0/0->0.0.0.0/0
Gateway: 3.3.3.5 Zone: Untrust IP: 3.3.3.5 -> 3.3.3.5
  Endpoint      Trans #  Call #  Notified Entity
  d001          1       1       0.0.0.0/0->0.0.0.0/0

```

**Meaning** The output displays information about all MGCP endpoints. Verify the following information:

- Gateway IP address and zone of both endpoints
- Endpoint identifier, transaction number, call number, and notified entity for each gateway

### Verifying MGCP ALG Counters

---

**Purpose** Verify information about MGCP counters.

**Action** From operational mode, enter the **show security alg mgcp counters** command.

```

user@host> show security alg mgcp counters
MGCP counters summary:
Packets received      :284
Packets dropped       :0
Message received     :284
Number of connections :4
Number of active connections :3
Number of calls       :4
Number of active calls :3
Number of transactions :121
Number of active transactions:52
Number of re-transmission :68
MGCP Error Counters:
Unknown-method       :0
Decoding error       :0
Transaction error    :0
Call error           :0

```

```

Connection error          :0
Connection flood drop    :0
Message flood drop       :0
IP resolve error         :0
NAT error                 :0
Resource manager error   :0
MGCP Packet Counters:
CRCX      :4      MDCX      :9      DLCX      :2
AUPEP     :1      AUCX      :0      NTFY      :43
RSIP      :79     EPCF      :0      RQNT      :51
000-199   :0      200-299  :95     300-999  :0

```

**Meaning** The output displays information about all MGCP counters. Verify the following information:

- Summary of MGCP counters
- MGCP error counters
- MGCP packet counters

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding MGCP ALGs on page 347](#)
- [MGCP ALG Configuration Overview on page 353](#)

## Example: Configuring Three-Zone ISP-Hosted Service Using MGCP ALG and NAT

This example shows how to configure a three-zone configuration using MGCP ALG and NAT.

- [Requirements on page 368](#)
- [Overview on page 368](#)
- [Configuration on page 369](#)
- [Verification on page 377](#)

### Requirements

Before you begin, understand NAT support with MGCP ALG. See “Understanding MGCP ALGs” on page 347.

### Overview

Typically, a three-zone configuration is used when an ISP in one geographical location provides service to two networks in different geographical locations.

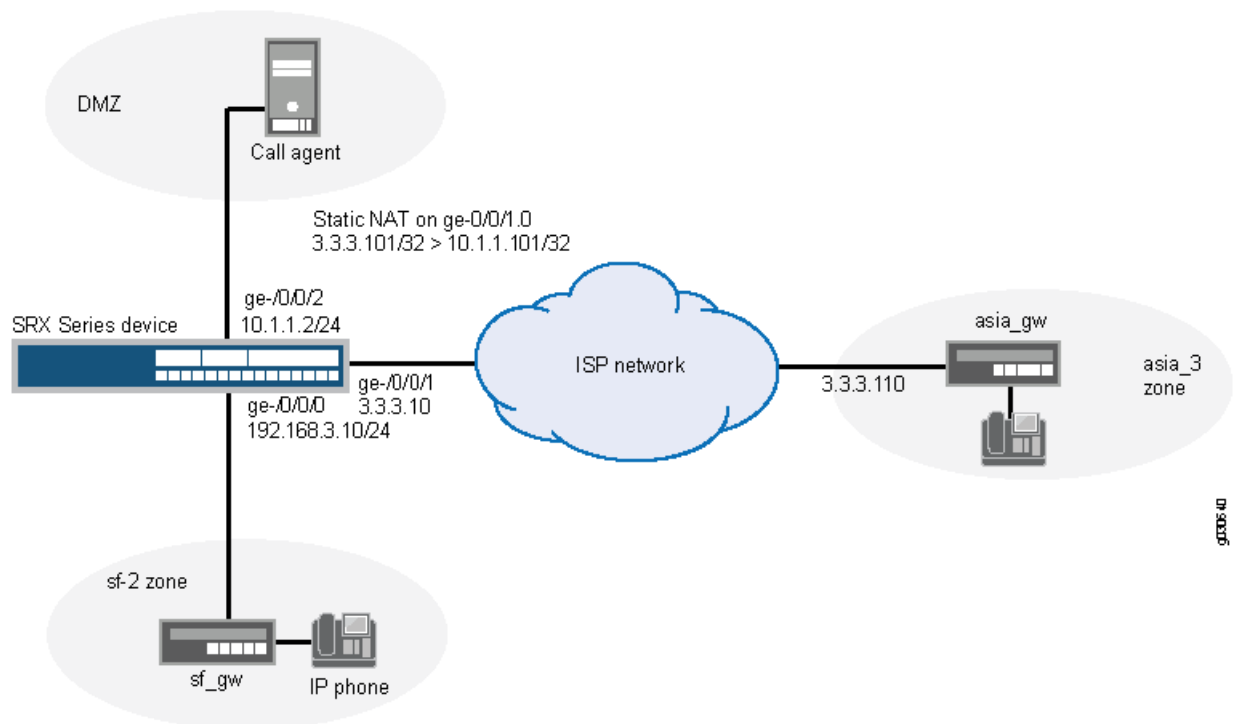
In this example (see Figure 30 on page 369), an ISP located on the USA West Coast provides MGCP service to customers in separate networks in Asia and San Francisco. Asia customers are in the asia-3 zone and are supported by the asia-gw gateway; San Francisco customers are in the sf-2 zone and are supported by the sf-gw gateway. A call agent, west-ca, is in the DMZ. The gateways and the call agent are listed in Table 38 on page 369, showing the corresponding IP address, interface, and zone.

Table 38: Three-Zone ISP-Host Service

Gateway	IP Address	Interface	Zone
sf-gw	192.168.3.201	ge-0/0/0	sf-2
asia-gw	3.3.3.101	ge-0/0/1	asia-3
west-ca	10.1.1.101	ge-0/0/2	DMZ

In this example, after creating zones and setting addresses for the gateways and the call agent, you associate the zones and addresses to interfaces, and then configure static NAT to the call agent and source NAT for communication from an IP phone in the sf-2 zone to phones in the asia-3 zone. You also configure a policy between the zones to allow the communication.

Figure 30: Three-Zone ISP-Hosted Service



## Configuration

**CLI Quick Configuration** To quickly configure a three-zone configuration using MGCP ALG and NAT, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet address 192.168.3.10/24
set interfaces ge-0/0/1 unit 0 family inet address 3.3.3.10/24
set interfaces ge-0/0/2 unit 0 family inet address 10.1.1.2/24
set security zones security-zone sf-2 address-book address sf-gw 192.168.3.201/32
set security zones security-zone sf-2 interfaces ge-0/0/0.0
```

```
set security zones security-zone asia-3 address-book address asia-gw 3.3.3.101/32
set security zones security-zone asia-3 interfaces ge-0/0/1.0
set security zones security-zone dmz address-book address west-ca 10.1.1.101/32
set security zones security-zone dmz interfaces ge-0/0/2.0
set security nat source pool ip-phone-pool address 3.3.3.20/32
set security nat source rule-set phones from zone sf-2
set security nat source rule-set phones to zone asia-3
set security nat source rule-set phones rule phone1 match source-address 192.168.3.10/32
set security nat source rule-set phones rule phone1 match destination 3.3.3.101/32
set security nat source rule-set phones rule phone1 then source-nat pool ip-phone-pool
set security nat static rule-set to-callagent from zone asia-3
set security nat static rule-set to-callagent rule phone1 match destination-address
  3.3.3.101/32
set security nat static rule-set to-callagent rule phone1 then static-nat prefix 10.1.1.101/32
set security nat proxy-arp interface ge-0/0/1.0 address 3.3.3.101/32
set security nat proxy-arp interface ge-0/0/1.0 address 3.3.3.20/32
set security policies from-zone dmz to-zone asia-3 policy pol-dmz-to-asia-3 match
  source-address west-ca
set security policies from-zone dmz to-zone asia-3 policy pol-dmz-to-asia-3 match
  destination-address asia-gw
set security policies from-zone dmz to-zone asia-3 policy pol-dmz-to-asia-3 match
  application junos-mgcp
set security policies from-zone dmz to-zone asia-3 policy pol-dmz-to-asia-3 then permit
set security policies from-zone asia-3 to-zone dmz policy pol-asia-3-to-dmz match
  source-address asia-gw
set security policies from-zone asia-3 to-zone dmz policy pol-asia-3-to-dmz match
  destination-address west-ca
set security policies from-zone asia-3 to-zone dmz policy pol-asia-3-to-dmz match
  application junos-mgcp
set security policies from-zone asia-3 to-zone dmz policy pol-asia-3-to-dmz then permit
set security policies from-zone sf-2 to-zone dmz policy pol-sf-2-to-dmz match
  source-address sf-gw
set security policies from-zone sf-2 to-zone dmz policy pol-sf-2-to-dmz match
  destination-address west-ca
set security policies from-zone sf-2 to-zone dmz policy pol-sf-2-to-dmz match application
  junos-mgcp
set security policies from-zone sf-2 to-zone dmz policy pol-sf-2-to-dmz then permit
set security policies from-zone dmz to-zone sf-2 policy pol-dmz-to-sf-2 match
  source-address west-ca
set security policies from-zone dmz to-zone sf-2 policy pol-dmz-to-sf-2 match
  destination-address sf-gw
set security policies from-zone dmz to-zone sf-2 policy pol-dmz-to-sf-2 match application
  junos-mgcp
set security policies from-zone dmz to-zone sf-2 policy pol-dmz-to-sf-2 then permit
set security policies from-zone sf-2 to-zone asia-3 policy pol-sf-2-to-asia-3 match
  source-address sf-gw
set security policies from-zone sf-2 to-zone asia-3 policy pol-sf-2-to-asia-3 match
  destination-address asia-gw
set security policies from-zone sf-2 to-zone asia-3 policy pol-sf-2-to-asia-3 match
  application junos-mgcp
set security policies from-zone sf-2 to-zone asia-3 policy pol-sf-2-to-asia-3 then permit
set security policies from-zone asia-3 to-zone sf-2 policy pol-asia-3-to-sf-2 match
  source-address asia-gw
set security policies from-zone asia-3 to-zone sf-2 policy pol-asia-3-to-sf-2 match
  destination sf-gw
```



```

set security policies from-zone asia-3 to-zone sf-2 policy pol-asia-3-to-sf-2 match
  application junos-mgcp
set security policies from-zone asia-3 to-zone sf-2 policy pol-asia-3-to-sf-2 then permit
set security policies from-zone sf-2 to-zone sf-2 policy pol-intra-sf-2 match source-address
  any
set security policies from-zone sf-2 to-zone sf-2 policy pol-intra-sf-2 match
  destination-address any
set security policies from-zone sf-2 to-zone sf-2 policy pol-intra-sf-2 match application
  any
set security policies from-zone sf-2 to-zone sf-2 policy pol-intra-sf-2 then permit
set security policies from-zone asia-3 to-zone asia-3 policy pol-intra-asia-3 match
  source-address any
set security policies from-zone asia-3 to-zone asia-3 policy pol-intra-asia-3 match
  destination-address any
set security policies from-zone asia-3 to-zone asia-3 policy pol-intra-asia-3 match
  application any
set security policies from-zone asia-3 to-zone asia-3 policy pol-intra-asia-3 then permit

```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a three-zone configuration using MGCP ALG and NAT:

1. Configure interfaces.

```

[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 192.168.3.10/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 3.3.3.10/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 10.1.1.2/24

```

2. Create zones and assign addresses to them.

```

[edit security zones]
user@host# set security-zone sf-2 address-book address sf-gw 192.168.3.201/32
user@host# set security-zone asia-3 address-book address asia-gw 3.3.3.101/32
user@host# set security-zone dmz address-book address west-ca 10.1.1.101/32

```

3. Associate the zones to the interfaces.

```

[edit security zones]
user@host# set security-zone sf-2 interfaces ge-0/0/0
user@host# set security-zone asia-3 interfaces ge-0/0/1
user@host# set security-zone dmz interfaces ge-0/0/2

```

4. Create a static NAT rule set and set the match conditions and actions for it.

```

[edit security nat static rule-set to-callagent]
user@host# set from zone asia-3
user@host# set rule phone1 match destination-address 3.3.3.101/32
user@host# set rule phone1 then static-nat prefix 10.1.1.101/32

```

5. Configure proxy ARP for address 3.3.3.101/32 on interface ge-0/0/1.0.

```

[edit security nat ]
user@host# set proxy-arp interface ge-0/0/1.0 address 3.3.3.101/32

```

6. Create a source NAT pool.

- ```
[edit security nat]
user@host# set source pool ip-phone-pool address 3.3.3.20/32
```
7. Create a source NAT rule set and set the match conditions and actions for it.

```
[edit security nat source rule-set phones]
user@host# set from zone sf-2
user@host# set to zone asia-3
user@host# set rule phone1 match source-address 192.168.3.10/32
user@host# set rule phone1 match destination-address 3.3.3.101/32
user@host# set rule phone1 then source-nat pool ip-phone-pool
```
  8. Configure proxy ARP for address 3.3.3.20/32 on interface ge-0/0/1.0.

```
[edit security nat ]
user@host# set proxy-arp interface ge-0/0/1.0 address 3.3.3.20/32
```
  9. Configure a policy to allow traffic from DMZ to Asia.

```
[edit security policies from-zone dmz to-zone asia-3 policy pol-dmz-to-asia-3]
user@host# set match source-address west-ca
user@host# set match destination-address asia-gw
user@host# set match application junos-mgcp
user@host# set then permit
```
  10. Configure a policy to allow traffic from Asia to DMZ.

```
[edit security policies from-zone asia-3 to-zone dmz policy pol-asia-3-to-dmz]
user@host# set match source-address asia-gw
user@host# set match destination-address west-ca
user@host# set match application junos-mgcp
user@host# set then permit
```
  11. Configure a policy to allow traffic from San Francisco to DMZ.

```
[edit security policies from-zone sf-2 to-zone dmz policy pol-sf-2-to-dmz]
user@host# set match source-address sf-gw
user@host# set match destination-address west-ca
user@host# set match application junos-mgcp
user@host# set then permit
```
  12. Configure a policy to allow traffic from DMZ to San Francisco.

```
[edit security policies from-zone dmz to-zone sf-2 policy pol-dmz-to-sf-2]
user@host# set match source-address west-ca
user@host# set match destination-address sf-gw
user@host# set match application junos-mgcp
user@host# set then permit
```
  13. Configure a policy to allow traffic from San Francisco to Asia.

```
[edit security policies from-zone sf-2 to-zone asia-3 policy pol-sf-2-to-asia-3]
user@host# set match source-address sf-gw
user@host# set match destination-address asia-gw
user@host# set match application junos-mgcp
user@host# set then permit
```
  14. Configure a policy to allow traffic from Asia to San Francisco.

```
[edit security policies from-zone asia-3 to-zone sf-2 policy pol-asia-3-to-sf-2]
user@host# set match source-address asia-gw
user@host# set match destination-address sf-gw
```

```

user@host# set match application junos-mgcp
user@host# set then permit

```

15. Configure a policy to allow traffic on devices within San Francisco.

```

[edit security policies from-zone sf-2 to-zone sf-2 policy pol-intra-sf-2]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application any
user@host# set then permit

```

16. Configure a policy to allow traffic on devices within Asia.

```

[edit security policies from-zone asia-3 to-zone asia-3 policy pol-intra-asia-3]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application any
user@host# set then permit

```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show security zones**, **show security nat**, and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 192.168.3.10/24;
    }
  }
}
ge-0/0/1 {
  unit 0 {
    family inet {
      address 3.3.3.10/24;
    }
  }
}
ge-0/0/2 {
  unit 0 {
    family inet {
      address 10.1.1.2/24;
    }
  }
}
[edit]
user@host# show security zones
security-zone sf-2 {
  address-book {
    address sf-gw 192.168.3.201/32;
  }
  interfaces {
    ge-0/0/0.0;
  }
}

```

```
}
security-zone asia-3 {
  address-book {
    address asia-gw 3.3.3.101/32;
  }
  interfaces {
    ge-0/0/1.0;
  }
}
security-zone dmz {
  address-book {
    address west-ca 10.1.1.101/32;
  }
  interfaces {
    ge-0/0/2.0;
  }
}
[edit]
user@host# show security nat
source {
  pool ip-phone-pool {
    address {
      3.3.3.20/32;
    }
  }
  rule-set phones {
    from zone sf-2;
    to zone asia-3;
    rule phone1 {
      match {
        source-address 192.168.3.10/32;
        destination-address 3.3.3.101/32;
      }
      then {
        source-nat {
          pool {
            ip-phone-pool;
          }
        }
      }
    }
  }
}
static {
  rule-set to-callagent {
    from zone asia-3;
    rule phone1 {
      match {
        destination-address 3.3.3.101/32;
      }
      then {
        static-nat prefix 10.1.1.101/32;
      }
    }
  }
}
```

```

proxy-arp {
  interface ge-0/0/1.0 {
    address {
      3.3.3.101/32;
      3.3.3.20/32;
    }
  }
}
[edit]
user@host# show security policies
from-zone dmz to-zone asia-3 {
  policy pol-dmz-to-asia-3 {
    match {
      source-address west-ca;
      destination-address asia-gw;
      application junos-mgcp;
    }
    then {
      permit;
    }
  }
}
from-zone asia-3 to-zone dmz {
  policy pol-asia-3-to-dmz {
    match {
      source-address asia-gw;
      destination-address west-ca;
      application junos-mgcp;
    }
    then {
      permit;
    }
  }
}
from-zone sf-2 to-zone dmz {
  policy pol-sf-2-to-dmz {
    match {
      source-address sf-gw;
      destination-address west-ca;
      application junos-mgcp;
    }
    then {
      permit;
    }
  }
}
from-zone dmz to-zone sf-2 {
  policy pol-dmz-to-sf-2 {
    match {
      source-address west-ca;
      destination-address sf-gw;
      application junos-mgcp;
    }
    then {
      permit;
    }
  }
}

```

```
    }
  }
  from-zone sf-2 to-zone asia-3 {
    policy pol-sf-2-to-asia-3 {
      match {
        source-address sf-gw;
        destination-address asia-gw;
        application junos-mgcp;
      }
      then {
        permit;
      }
    }
  }
  from-zone asia-3 to-zone sf-2 {
    policy pol-asia-3-to-sf-2 {
      match {
        source-address asia-gw;
        destination-address sf-gw;
        application junos-mgcp;
      }
      then {
        permit;
      }
    }
  }
  from-zone sf-2 to-zone sf-2 {
    policy pol-intra-sf-2 {
      match {
        source-address any;
        destination-address any;
        application any;
      }
      then {
        permit;
      }
    }
  }
  from-zone asia-3 to-zone asia-3 {
    policy pol-intra-asia-3 {
      match {
        source-address any;
        destination-address any;
        application any;
      }
      then {
        permit;
      }
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform these tasks:

- [Verifying MGCP ALG on page 377](#)
- [Verifying MGCP Calls on page 377](#)
- [Verifying MGCP ALG Statistics on page 377](#)
- [Verifying MGCP Endpoints on page 377](#)

---

### Verifying MGCP ALG

**Purpose** Verify if the MGCP ALG is enabled.

**Action** From operational mode, enter the `show security alg status | match mgcp` command.

---

### Verifying MGCP Calls

**Purpose** Verify the MGCP calls that are currently active.

**Action** From operational mode, enter the `show security alg mgcp calls` command.

---

### Verifying MGCP ALG Statistics

**Purpose** Verify the MGCP ALG statistics.

**Action** From operational mode, enter the `show security alg mgcp counters` command.

---

### Verifying MGCP Endpoints

**Purpose** Verify the MGCP endpoints.

**Action** From operational mode, enter the `show security alg mgcp endpoints` command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Static NAT Configuration Overview on page 1340](#)
- [Understanding Source NAT on page 1369](#)





## CHAPTER 15

# RPC ALGs

- Understanding RPC ALGs on page 379
- Sun RPC ALGs on page 380
- Microsoft RPC ALGs on page 383

### Understanding RPC ALGs

---

Junos OS supports basic Remote Procedure Call Application Layer Gateway (RPC ALG) services. RPC is a protocol that allows an application running in one address space to access the resources of applications running in another address space as if the resources were local to the first address space. The RPC ALG is responsible for RPC packet processing.

The RPC ALG in Junos OS supports the following services and features:

- Sun Microsystems RPC Open Network Computing (ONC)
- Microsoft RPC Distributed Computing Environment (DCE)
- Dynamic port negotiation
- Ability to allow and deny specific RPC services
- Static Network Address Translation (NAT) and source NAT (with no port translation)
- RPC applications in security policies

Use the RPC ALG if you need to run RPC-based applications such as NFS or Microsoft Outlook. The RPC ALG functionality is enabled by default.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- ALG Overview on page 217
- Understanding Sun RPC ALGs on page 380
- Understanding Microsoft RPC ALGs on page 383

## Sun RPC ALGs

---

- Understanding Sun RPC ALGs on page 380
- Enabling Sun RPC ALGs (J-Web Procedure) on page 381
- Enabling Sun RPC ALGs (CLI Procedure) on page 381
- Sun RPC Services and Applications on page 381

### Understanding Sun RPC ALGs

Sun Microsystems Remote Procedure Call (Sun RPC)—also known as Open Network Computing Remote Procedure Call (ONC RPC)—provides a way for a program running on one host to call procedures in a program running on another host. Because of the large number of RPC services and the need to broadcast, the transport address of an RPC service is dynamically negotiated based on the service's program number and version number. Several binding protocols are defined for mapping the RPC program number and version number to a transport address.

Junos OS supports the Sun RPC as a predefined service and allows and denies traffic based on a security policy you configure. The Application Layer Gateway (ALG) provides the functionality for Juniper Networks devices to handle the dynamic transport address negotiation mechanism of the Sun RPC and to ensure program number-based security policy enforcement. You can define a security policy to permit or deny all RPC requests, or to permit or deny by specific program number. The ALG also supports route mode and Network Address Translation (NAT) mode for incoming and outgoing requests.

When an application or a PC client calls a remote service, it needs to find the transport address of the service. In the case of TCP/UDP, the address is a port number. A typical procedure for this case is as follows:

1. The client sends the GETPORT message to the RPCBIND service on the remote machine. The GETPORT message contains the program number, and version and procedure number of the remote service it is attempting to call.
2. The RPCBIND service replies with a port number.
3. The client calls the remote service using the port number returned.
4. The remote service replies to the client.

A client also can use the CALLIT message to call the remote service directly, without determining the port number of the service. In this case, the procedure is as follows:

1. The client sends a CALLIT message to the RPCBIND service on the remote machine. The CALLIT message contains the program number and the version and procedure number of the remote service it attempting to call.
2. RPCBIND calls the service for the client.
3. RCPBIND replies to the client if the call has been successful. The reply contains the call result and the services's port number.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding RPC ALGs on page 379
  - Enabling Sun RPC ALGs (J-Web Procedure) on page 381
  - Enabling Sun RPC ALGs (CLI Procedure) on page 381
  - Understanding Sun RPC Services on page 382
  - Understanding Microsoft RPC ALGs on page 383

### Enabling Sun RPC ALGs (J-Web Procedure)

The Sun RPC ALG is enabled by default and requires no configuration.

To disable or re-enable the RPC ALG:

1. Select **Configure>Security>ALG**.
2. Select the **Enable SUNRPC** check box.
3. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding Sun RPC ALGs on page 380
  - Enabling Sun RPC ALGs (CLI Procedure) on page 381

### Enabling Sun RPC ALGs (CLI Procedure)

The Sun RPC ALG is enabled by default and requires no configuration.

To disable the Sun RPC ALG, enter the following command:

```
user@host# set security alg sunrpc disable
```

To re-enable the Sun RPC ALG, enter the following command:

```
user@host# delete security alg sunrpc
```

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding Sun RPC ALGs on page 380
  - Enabling Sun RPC ALGs (J-Web Procedure) on page 381

### Sun RPC Services and Applications

- Understanding Sun RPC Services on page 382
- Customizing Sun RPC Applications (CLI Procedure) on page 382

## Understanding Sun RPC Services

---

Predefined Sun RPC services include:

- `junos-sun-rpc-portmap-tcp`
- `junos-sun-rpc-portmap`
- `junos-sun-rpc-portmap-udp`

The Sun RPC ALG can be applied by using the following methods:

- ALG default application—Use one of the following predefined application sets for control and data connections in your policy:
  - `application-set junos-sun-rpc` (for control sessions)
  - `application-set junos-sun-rpc-portmap` (for data sessions)
- Default control application—Use the predefined control via `junos-sun-rpc`:
  - Create an application for data (`USER_DEFINED_DATA`). You can make a set of your own data (for example, `my_rpc_application_set`) and use it in the policy.
  - Use the predefined application set for control and customized data application in the policy:
    - `junos-sun-rpc`
    - `USER_DEFINED_DATA`
- Custom control and custom data application—Use a customized application:
  - Create an application for control (`USER_DEFINED_CONTROL`) and data (`USER_DEFINED_DATA`).
  - In the policy, use the user-defined application set for a control and customized data application:
    - `USER_DEFINED_CONTROL`
    - `USER_DEFINED_DATA`

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Sun RPC ALGs on page 380
- Customizing Sun RPC Applications (CLI Procedure) on page 382
- Understanding Microsoft RPC Services on page 385

## Customizing Sun RPC Applications (CLI Procedure)

---

All Sun RPC applications can be customized by using a predefined application set.

For example, an application can be customized to open the control session only and not allow any data sessions:

```
application-set junos-sun-rpc {
  application junos-sun-rpc-tcp;
  application junos-sun-rpc-udp;
}
```

In the following example, the predefined application set allows data sessions only. It will not work without the control session:

```
application-set junos-sun-rpc-portmap {
  application junos-sun-rpc-portmap-tcp;
  application junos-sun-rpc-portmap-udp;
}
```

To customize all Sun RPC applications with predefined application sets, use both application sets in the policy:

```
application-set [junos-sun-rpc junos-sun-rpc-portmap]
```



**NOTE:** MS RPC applications are customized in the same way as SUN RPC applications.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Sun RPC ALGs on page 380
- Customizing Microsoft RPC Applications (CLI Procedure) on page 385

## Microsoft RPC ALGs

- Understanding Microsoft RPC ALGs on page 383
- Enabling Microsoft RPC ALGs (J-Web Procedure) on page 384
- Enabling Microsoft RPC ALGs (CLI Procedure) on page 384
- Microsoft RPC Services and Applications on page 385
- Verifying the Microsoft RPC ALG Tables on page 386

### Understanding Microsoft RPC ALGs

Microsoft Remote Procedure Call (MS RPC) is the Microsoft implementation of the Distributed Computing Environment (DCE) RPC. Like the Sun RPC, MS RPC provides a way for a program running on one host to call procedures in a program running on another host. Because of the large number of RPC services and the need to broadcast, the transport address of an RPC service is dynamically negotiated based on the service program's Universal Unique Identifier (UUID). The specific UUID is mapped to a transport address.

Junos OS devices running Junos OS support MS RPC as a predefined service and allow and deny traffic based on a policy you configure. The Application Layer Gateway (ALG) provides the functionality for Juniper Networks devices to handle the dynamic transport

address negotiation mechanism of the MS RPC, and to ensure UUID-based security policy enforcement. You can define a security policy to permit or deny all RPC requests, or to permit or deny by specific UUID number. The ALG also supports route mode and Network Address Translation (NAT) mode for incoming and outgoing requests.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding RPC ALGs on page 379
- Enabling Microsoft RPC ALGs (J-Web Procedure) on page 384
- Enabling Microsoft RPC ALGs (CLI Procedure) on page 384
- Understanding Microsoft RPC Services on page 385
- Understanding Sun RPC ALGs on page 380
- Verifying the Microsoft RPC ALG Tables on page 386

### Enabling Microsoft RPC ALGs (J-Web Procedure)

The MS RPC ALG is enabled by default and requires no configuration.

To disable or re-enable the Microsoft ALG:

1. Select **Configure>Security>ALG**.
2. Select the **Enable MSRPC** check box.
3. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Microsoft RPC ALGs on page 383
- Enabling Microsoft RPC ALGs (CLI Procedure) on page 384
- Verifying the Microsoft RPC ALG Tables on page 386

### Enabling Microsoft RPC ALGs (CLI Procedure)

The MS RPC ALG is enabled by default and requires no configuration.

To disable the Microsoft RPC ALG, enter the following command:

```
user@host# set security alg msrpc disable
```

To re-enable the Microsoft RPC ALG, enter the following command:

```
user@host# delete security alg msrpc
```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Microsoft RPC ALGs on page 383
- Enabling Microsoft RPC ALGs (J-Web Procedure) on page 384
- Verifying the Microsoft RPC ALG Tables on page 386

## Microsoft RPC Services and Applications

- Understanding Microsoft RPC Services on page 385
- Customizing Microsoft RPC Applications (CLI Procedure) on page 385

### Understanding Microsoft RPC Services

---

Predefined MS RPC services include:

- `junos-ms-rpc-portmap`
- `junos-ms-rpc-portmap-tcp`
- `junos-ms-rpc-portmap-udp`

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Microsoft RPC ALGs on page 383
- Customizing Microsoft RPC Applications (CLI Procedure) on page 385
- Understanding Sun RPC Services on page 382

### Customizing Microsoft RPC Applications (CLI Procedure)

---

MS RPC applications are customized in the same way as SUN RPC applications.

MS RPC services in security policies are:

- `0e4a0156-dd5d-11d2-8c2f-00c04fb6bcde`
- `1453c42c-0fa6-11d2-a910-00c04f990f3b`
- `10f24e8e-0fa6-11d2-a910-00c04f990f3b`
- `1544f5e0-613c-11d1-93df-00c04fd7bd09`

The corresponding TCP/UDP ports are dynamic. To permit them, you use the following statement for each number:

```
set applications application-name term term-name uuid hex-number
```

The ALG maps the program numbers into dynamically negotiated TCP/UDP ports based on these four UUIDs and permits or denies the service based on a policy you configure.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Microsoft RPC Services on page 385
- Customizing Sun RPC Applications (CLI Procedure) on page 382
- Verifying the Microsoft RPC ALG Tables on page 386

## Verifying the Microsoft RPC ALG Tables

**Purpose** To verify the Microsoft RPC ALG, display the Microsoft Universal Unique Identifier to Object ID (UUID-to-OID) mapping table. The Microsoft RPC ALG monitors packets on TCP port 135.

**Action** From the CLI, enter the `show security alg msrpc object-id-map` command.

```
user@host> show security alg msrpc object-id-map
UUID                               OID
1be617c0-31a5-11cf-a7d8-00805f48a135 0x80000020
e3514235-4b06-11d1-ab04-00c04fc2dcd2 0x80000002
67df7c70-0f04-11ce-b13f-00aa003bac6c 0x80000014
```

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Enabling Microsoft RPC ALGs \(J-Web Procedure\) on page 384](#)
  - [Enabling Microsoft RPC ALGs \(CLI Procedure\) on page 384](#)
  - [Customizing Microsoft RPC Applications \(CLI Procedure\) on page 385](#)



## PART 5

# User Authentication

- Firewall User Authentication on page 389
- Infranet Authentication on page 421



# Firewall User Authentication

- Firewall User Authentication Overview on page 389
- Pass-Through Authentication on page 390
- Web Authentication on page 397
- External Authentication on page 405
- Client Groups for Firewall Authentication on page 415
- Firewall Authentication Banner Customization on page 418

## Firewall User Authentication Overview

---

A firewall user is a network user who must provide a username and password for authentication when initiating a connection across the firewall. Junos OS enables administrators to restrict and permit firewall users to access protected resources (different zones) behind a firewall based on their source IP address and other credentials.



**NOTE:** Junos OS also supports the administrator and Point-to-Point Protocol (PPP) user types. For more information, see the *Junos OS Administration Guide for Security Devices*.

After you define firewall users, you can create a policy that requires the users to authenticate themselves through one of two authentication schemes:

- Pass-Through Authentication—A host or a user from one zone tries to access resources on another zone. You must use an FTP, a Telnet, or an HTTP client to access the IP address of the protected resource and to get authenticated by the firewall. The device uses FTP, Telnet, or HTTP to collect username and password information, and subsequent traffic from the user or host is allowed or denied based on the result of this authentication.
- Web Authentication—Users try to connect, using HTTP, to an IP address on the device that is enabled for Web authentication; in this scenario, you do not use HTTP to get to the IP address of the protected resource. You are prompted for the username and password that are verified by the device. Subsequent traffic from the user or host to the protected resource is allowed or denied based on the result of this authentication.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding Pass-Through Authentication on page 390](#)
  - [Understanding Web Authentication on page 397](#)
  - [Understanding External Authentication Servers on page 406](#)
  - [Understanding Client Groups for Firewall Authentication on page 415](#)
  - [Understanding Firewall Authentication Banner Customization on page 418](#)

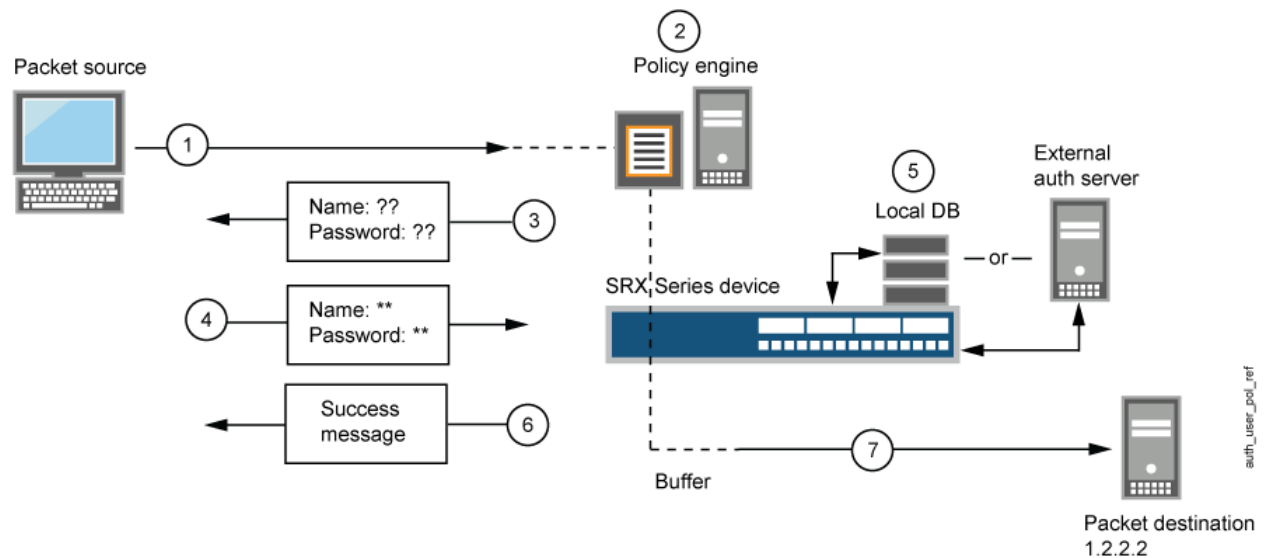
## Pass-Through Authentication

- [Understanding Pass-Through Authentication on page 390](#)
- [Example: Configuring Pass-Through Authentication on page 391](#)

### Understanding Pass-Through Authentication

With pass-through user authentication, when a user attempts to initiate an HTTP, an FTP, or a Telnet connection request that has a policy requiring authentication, the device intercepts the request and prompts the user to enter a username and password. Before granting permission, the device validates the username and password by checking them against those stored in the local database or on an external authentication server, as shown in Figure 31 on page 390.

Figure 31: Policy Lookup for a User



The steps in Figure 31 on page 390 are as follows:

1. A client user sends an FTP, an HTTP, or a Telnet packet to 1.2.2.2.
2. The device intercepts the packet, notes that its policy requires authentication from either the local database or an external authentication server, and buffers the packet.
3. The device prompts the user for login information through FTP, HTTP, or Telnet.

4. The user replies with a username and password.
5. The device either checks for an authentication user account on its local database or it sends the login information to the external authentication server as specified in the policy.
6. Finding a valid match (or receiving notice of such a match from the external authentication server), the device informs the user that the login has been successful.
7. The device forwards the packet from its buffer to its destination IP address 1.2.2.2.

After the device authenticates a user at a particular source IP address, it subsequently permits traffic—as specified in the policy requiring authentication through pass through—from any other user at that same address. This might be the case if the user originates traffic from behind a NAT device that changes all original source addresses to a single translated address.



**NOTE:** The pass-through user authentication method is recommended in situations when security has a higher priority than convenience. This authentication method applies only to the session and child sessions matching the policy that triggered it. You can apply this method on Internet-facing links, if used with caution.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Firewall User Authentication Overview on page 389](#)
- [Understanding Web Authentication on page 397](#)
- [Example: Configuring Pass-Through Authentication on page 391](#)

### Example: Configuring Pass-Through Authentication

This example shows how to configure pass-through authentication for a firewall.

- [Requirements on page 391](#)
- [Overview on page 391](#)
- [Configuration on page 392](#)
- [Verification on page 396](#)

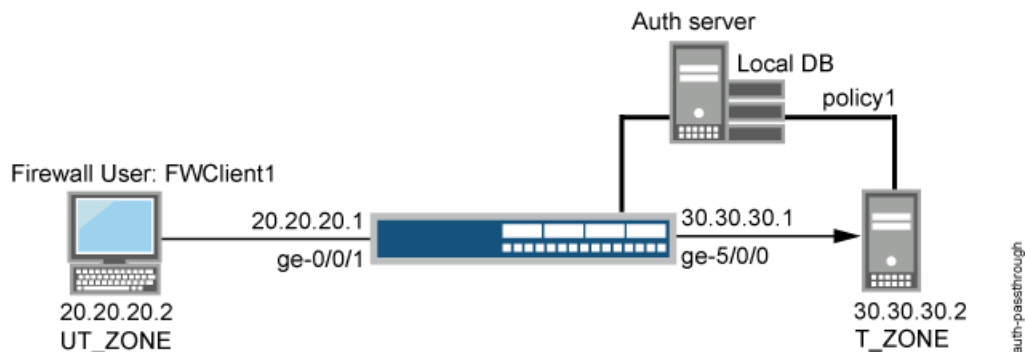
#### Requirements

Before you begin, define firewall users. See “Firewall User Authentication Overview” on page 389.

#### Overview

Pass-through firewall user authentication occurs when the client is trying to access a destination on another zone using FTP, Telnet, or HTTP. After authenticating successfully, the firewall acts as a proxy for an FTP, a Telnet, or an HTTP server so that it can first authenticate the user before allowing access to the actual FTP, Telnet, or HTTP server behind the firewall. Figure 32 on page 392 shows the topology used in this example.

Figure 32: Configuring Pass-Through Firewall Authentication



### Configuration

**CLI Quick Configuration** To quickly configure pass-through authentication, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/1 unit 0 family inet address 20.20.20.1/24
set interfaces ge-5/0/0 unit 0 family inet address 30.30.30.1/24
set access profile FWAUTH client FWClient1 firewall-user password pwd
set access firewall-authentication pass-through default-profile FWAUTH
set access firewall-authentication pass-through telnet banner success "WELCOME TO JUNIPER
TELNET SESSION"
set security zones security-zone UT-ZONE host-inbound-traffic system-services all
set security zones security-zone UT-ZONE interfaces ge-0/0/1.0 host-inbound-traffic protocols
all
set security zones security-zone T-ZONE host-inbound-traffic system-services all
set security zones security-zone T-ZONE interfaces ge-5/0/0.0 host-inbound-traffic protocols
all
set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match source-address any
set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match destination-address
any
set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match application junos-telnet
set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 then permit
firewall-authentication pass-through client-match FWClient1
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To configure pass-through authentication:

1. Configure two interfaces and assign IP addresses to them.



**NOTE:** For this example, it is optional to assign two addresses to the interfaces.

```
[edit]
```

```
user@host# set interfaces ge-0/0/1 unit 0 family inet address 20.20.20.1/24
```

```
user@host# set interfaces ge-5/0/0 unit 0 family inet address 30.30.30.1/24
```

2. Create the FWAUTH access profile for the FWClient1 user, specify the user's password, and define a success banner for Telnet sessions.

```
[edit access]
```

```
user@host# set access profile FWAUTH client FWClient1 firewall-user password
pwd
```

```
user@host# set firewall-authentication pass-through default-profile FWAUTH
```

```
user@host# set firewall-authentication pass-through telnet banner success
"WELCOME TO JUNIPER TELNET SESSION"
```

3. Configure security zones.



**NOTE:** For this example, it is optional to configure a second interface for a security zone.

```
[edit security zones]
```

```
user@host# set security-zone UT-ZONE host-inbound-traffic system-services all
```

```
user@host# set security-zone UT-ZONE interfaces ge-0/0/1.0 host-inbound-traffic
protocols all
```

```
user@host# set security-zone T-ZONE host-inbound-traffic system-services all
```

```
user@host# set security-zone T-ZONE interfaces ge-5/0/0.0 host-inbound-traffic
protocols all
```

4. Assign security policy P1 to the security zones.

```
[edit security policies]
```

```
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match
source-address any
```

```
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match
destination-address any
```

```
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match application
junos-telnet
```

```
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 then permit
firewall-authentication pass-through client-match FWClient1
```

5. Use Telnet to authenticate the FWClient1 firewall user to host2.

```
regress@FWClient1# run telnet 30.30.30.2
Trying 30.30.30.2...
```

```

Connected to 30.30.30.2.
Escape character is '^]'.
Firewall User Authentication
Username: FWClient1
Password:***
WELCOME TO JUNIPER TELNET SESSION
Host1 (tty0)
login: regress
Password:
--- JUNOS 10.1R1.1 built 2009-10-12 13:30:18 UTC
%
```

**Results** From configuration mode, confirm your configuration by entering these commands:

- **show interfaces**
- **show access**
- **show security zones**
- **show security policies**

If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```

user@host# show interfaces
...
}
ge-0/0/1 {
  unit 0 {
    family inet {
      address 20.20.20.1/24;
    }
  }
}
ge-5/0/0 {
  unit 0 {
    family inet {
      address 30.30.30.1/24;
    }
  }
}
...

user@host# show access
profile FWAUTH {
  authentication-order password;
  client FWClient1 {
    firewall-user {
      password "$9$XHhXVYGdkf5F"; ## SECRET-DATA
    }
  }
}
firewall-authentication {
  pass-through {
    default-profile FWAUTH;
  }
}
```



```

        telnet {
            banner {
                success "WELCOME TO JUNIPER TELNET SESSION";
            }
        }
    }
}

```

```
user@host# show security zones
```

```

...
}
security-zone UT-ZONE {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0 {
            host-inbound-traffic {
                protocols {
                    all;
                }
            }
        }
    }
}
security-zone T-ZONE {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-5/0/0.0 {
            host-inbound-traffic {
                protocols {
                    all;
                }
            }
        }
    }
}
}

```

```
user@host# show security policies
```

```

...
from-zone UT-ZONE to-zone T-ZONE {
    policy P1 {
        match {
            source-address any;
            destination-address any;
            application junos-telnet;
        }
        then {
            permit {
                firewall-authentication {
                    pass-through {
                        client-match FWClient1;
                    }
                }
            }
        }
    }
}

```

```

    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Firewall User Authentication and Monitoring Users and IP Addresses in the Authentication Table on page 396

### *Verifying Firewall User Authentication and Monitoring Users and IP Addresses in the Authentication Table*

**Purpose** Display firewall authentication user history and verify the number of firewall users who successfully authenticated and firewall users who failed to log in.

**Action** From operational mode, enter these **show** commands:

```

user@host> show security firewall-authentication history
user@host> show security firewall-authentication history identifier 1
user@host> show security firewall-authentication users
user@host> show security firewall-authentication users identifier 3

```

For more information, see the *Junos OS CLI Reference*.

```

user@host> show security firewall-authentication history
History of firewall authentication data:
Authentications: 2
Id Source Ip Date Time Duration Status User
1 20.20.20.2 2010-10-12 21:24:02 0:00:24 Failed FWClient1
2 20.20.20.2 2010-10-12 21:24:48 0:00:22 Success FWClient1

```

```

user@host> show security firewall-authentication history identifier 1
Username: FWClient1
Source IP: 20.20.20.2
Authentication state: Success
Authentication method: Pass-through using Telnet
Access start date: 2010-10-12
Access start time: 21:24:02
Duration of user access: 0:00:24
Source zone: UT-ZONE
Destination zone: T-ZONE
Access profile: FWAUTH
Bytes sent by this user: 0
Bytes received by this user: 2660

```

```

user@host> show security firewall-authentication users
Firewall authentication data:
Total users in table: 1
Id Source Ip Src zone Dst zone Profile Age Status User
4 20.20.20.2 UT-ZONE T-ZONE FWAUTH 1 Success FWClient1

```

```

user@host> show security firewall-authentication users identifier 3
Username: FWClient1
Source IP: 20.20.20.2
Authentication state: Success

```

Authentication method: Pass-through using Telnet  
Age: 3  
Access time remaining: 9  
Source zone: UT-ZONE  
Destination zone: T-ZONE  
Access profile: FWAUTH  
Interface Name: ge-0/0/1.0  
Bytes sent by this user: 0  
Bytes received by this user: 1521

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Firewall User Authentication Overview on page 389](#)
- [Understanding Pass-Through Authentication on page 390](#)

## Web Authentication

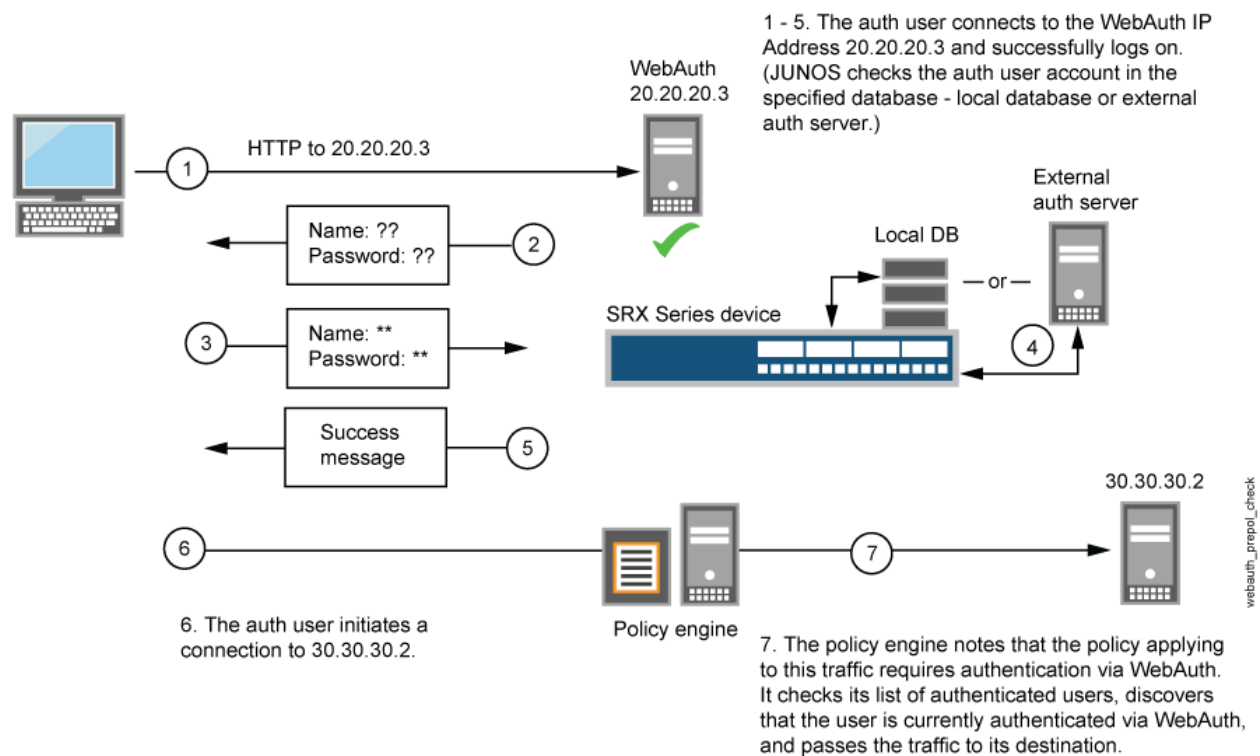
---

- [Understanding Web Authentication on page 397](#)
- [Example: Configuring Web Authentication on page 399](#)

### Understanding Web Authentication

Web authentication is an alternative to pass-through user authentication. Instead of pointing to the resource that you want to connect to from your client browser, you point the browser to an IP address on the device that is enabled for Web authentication. This initiates an HTTP session to the IP address hosting the Web authentication feature on the device. The device then prompts you for your username and password and caches the result in the device. Later, when traffic encounters a Web authentication policy, you are allowed or denied access based on the prior Web authentication results, as shown in Figure 33 on page 398.

Figure 33: Web Authentication Example



Follow these Web authentication guidelines:

- You can leave the default Web authentication server as the local database or you can choose an external authentication server for the role. The default Web authentication profile determines if the user authenticates using the local database or the external authentication server. An access profile stores usernames and passwords of users or points to external authentication servers where such information is stored.
- The Web authentication address must be in the same subnet as the interface that you want to use to host it. For example, if you want authentication users to connect using Web authentication through ethernet3, which has IP address 1.1.1/24, then you can assign Web authentication an IP address in the 1.1.1.0/24 subnet.
- You can put a Web authentication address in the same subnet as the IP address of any physical interface or virtual security interface (VSI). (For information about different types of interfaces, see “Security Zones and Interfaces Overview” on page 111.)
- You can put Web authentication addresses on multiple interfaces.
- After a device authenticates a user at a particular source IP address, it subsequently permits traffic—as specified in the policy requiring authentication through Web authentication—from any other user at that same address. This might be the case if the user originates traffic from behind a NAT device that changes all original source addresses to a single translated address.
- With Web authentication enabled, any HTTP traffic to the IP address will get the Web authentication login page instead of the administrator login page. Disabling this option

will show the administrator login page (assuming that `[system services web-management HTTP]` is enabled).

- We recommend that you have a separate primary or preferred IP address, if an address is used for Web authentication.



**NOTE:** The Web authentication method is recommended in situations when the client devices are immediately adjacent to the security gateway and there is high assurance that the client devices are not multiuser hosts. This authentication method is best applied to wireless links and DMZ, or conference room links.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Firewall User Authentication Overview on page 389](#)
- [Understanding Pass-Through Authentication on page 390](#)
- [Example: Configuring Web Authentication on page 399](#)

## Example: Configuring Web Authentication

This example shows how to enable Web authentication and set up a policy that allows access to a user when traffic encounters a policy that has Web authentication enabled.

- [Requirements on page 399](#)
- [Overview on page 399](#)
- [Configuration on page 400](#)
- [Verification on page 404](#)

### Requirements

Before you begin:

- Define firewall users. See “Firewall User Authentication Overview” on page 389.
- Add the Web authentication HTTP flag under the interface's address hierarchy to enable Web authentication.

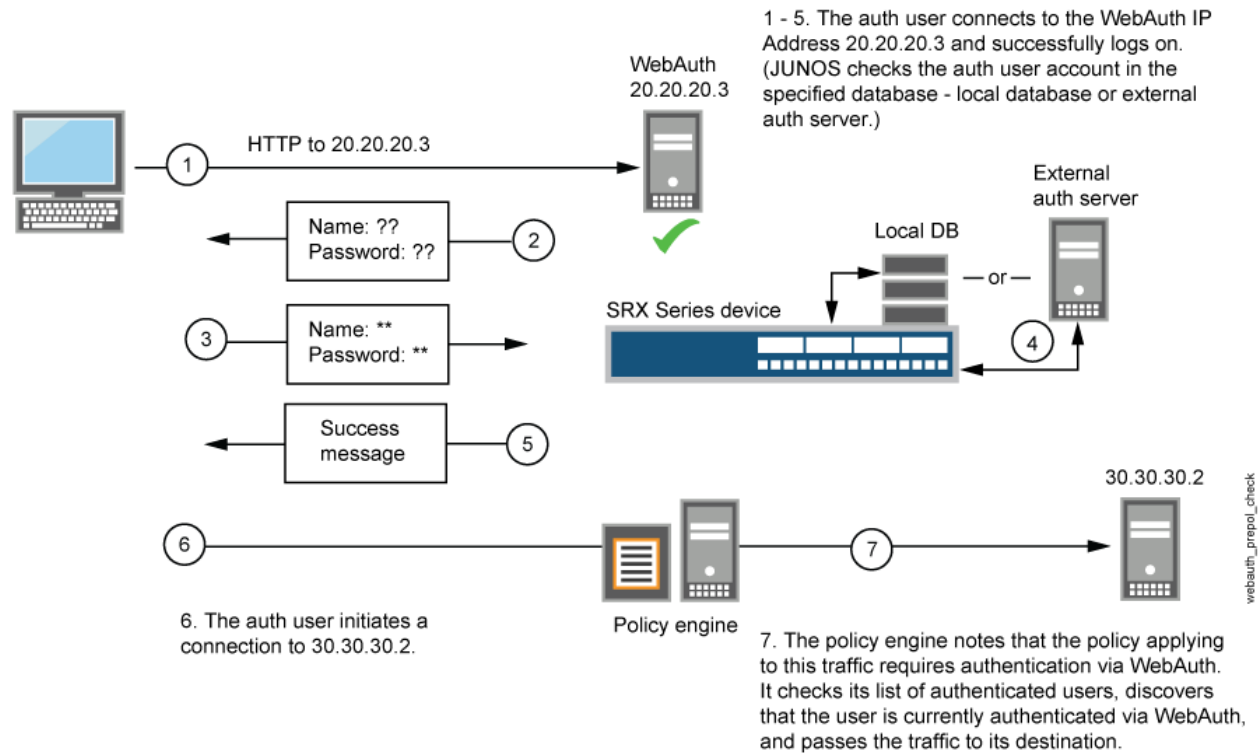
### Overview

To enable Web authentication, you must specify the IP address of the device hosting the HTTP session. These settings are used if the firewall user accessing a protected resource wants to be authenticated by directly accessing the webserver or Web authentication. The following instructions show how to set up a policy that allows access to the FWClient1 user when traffic encounters a policy that has Web authentication enabled (Policy-W). (See Figure 34 on page 400.) In this example, FWClient1 has already authenticated through the Web authentication login page.

The FWClient1 firewall user does the following to get authenticated:

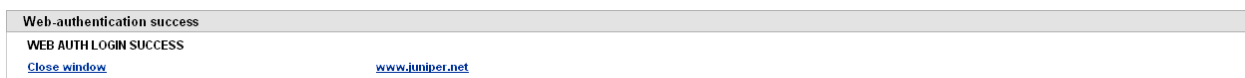
- a. Points the browser to the Web authentication IP (20.20.20.1) to get authenticated first
- b. Starts traffic to access resources specified by the policy-W policy

Figure 34: Web Authentication Example



When you configure the device as described in these instructions and the user successfully authenticates, the screen illustrated in Figure 35 on page 400 appears.

Figure 35: Web Authentication Success Banner



### Configuration

**CLI Quick Configuration** To quickly configure Web authentication as illustrated in Figure 34 on page 400, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/1 unit 0 family inet address 20.20.20.1/24
set interfaces ge-0/0/1 unit 0 family inet address 20.20.20.3/24 web-authentication http
set interfaces fe-5/0/0 unit 0 family inet address 30.30.30.1/24
set access profile WEBAUTH client FWclient1 firewall-user password pwd
set access firewall-authentication web-authentication default-profile WEBAUTH
set access firewall-authentication web-authentication banner success "WEB AUTH LOGIN SUCCESS"
set security zones security-zone UT-ZONE host-inbound-traffic system-services all
set security zones security-zone UT-ZONE interfaces ge-0/0/1.0 host-inbound-traffic protocols all
```

```

set security zones security-zone T-ZONE host-inbound-traffic system-services all
set security zones security-zone T-ZONE interfaces ge-5/0/0.0 host-inbound-traffic protocols
all
set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match source-address any
set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match destination-address
any
set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match application any
set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 then permit
firewall-authentication web-authentication client-match FWClient1
set system services web-management http interface ge-0/0/1.0

```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To configure Web authentication:

1. Configure two interfaces and assign IP addresses to them.



**NOTE:** For this example, it is optional to assign two addresses to the interfaces.

[edit]

```
user@host# set interfaces ge-0/0/1 unit 0 family inet address 20.20.20.1/24
```

```
user@host# set interfaces ge-0/0/1 unit 0 family inet address 20.20.20.3/24
web-authentication http
```

```
user@host# set interfaces fe-5/0/0 unit 0 family inet address 30.30.30.1/24
```

2. Create the WEBAUTH access profile for the FWClient1 user, specify the user’s password, and define a success banner.

[edit access]

```
user@host# set interfaces ge-0/0/1 unit 0 family inet address 20.20.20.1/24
```

```
user@host# set firewall-authentication web-authentication default-profile
WEBAUTH
```

```
user@host# set firewall-authentication web-authentication banner success "WEB
AUTH LOGIN SUCCESS"
```

3. Configure security zones.



**NOTE:** For this example, it is optional to configure a second interface for a security zone.

[edit security zones]

```
user@host# set security-zone UT-ZONE host-inbound-traffic system-services all
```

```
user@host# set security-zone UT-ZONE interfaces ge-0/0/1.0 host-inbound-traffic
  protocols all
```

```
user@host# set security-zone T-ZONE host-inbound-traffic system-services all
```

```
user@host# set security-zone T-ZONE interfaces ge-5/0/0.0 host-inbound-traffic
  protocols all
```

4. Assign security policy P1 to the security zones.

```
[edit security policies]
```

```
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match
  source-address any
```

```
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match
  destination-address any
```

```
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match application
  any
```

```
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 then permit
  firewall-authentication web-authentication client-match FWClient1
```

5. Activate the HTTP daemon on your device.

```
[edit]
```

```
user@host# set system services web-management http interface ge-0/0/1.0
```

**Results** From configuration mode, confirm your configuration by entering these commands:

- **show interfaces**
- **show access**
- **show security zones**
- **show security policies**
- **show system services**

If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
user@host# show interfaces
...
}
ge-0/0/1{
  unit 0 {
    family inet {
```



```

        address 20.20.20.1/24 {
        address 20.20.20.3/24 {
            web-authentication http;
        }
    }
}
fe-5/0/0 {
    unit 0 {
        family inet {
            address 30.30.30.1/24;
        }
    }
}
...

user@host# show access
profile WEBAUTH {
    client FWClient1 {
        firewall-user {
            password "$9$XHhxVYGdkf5F"; ## SECRET-DATA
        }
    }
}
firewall-authentication {
    web-authentication {
        default-profile WEBAUTH;
        banner {
            success "WEB AUTH LOGIN SUCCESS";
        }
    }
}

user@host# show security zones
...
}
security-zone UT-ZONE {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0 {
            host-inbound-traffic {
                protocols {
                    all;
                }
            }
        }
    }
}
security-zone T-ZONE {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-5/0/0.0 {

```

```

        host-inbound-traffic {
            protocols {
                all;
            }
        }
    }
}

user@host# show security policies
...
from-zone UT-ZONE to-zone T-ZONE {
    policy P1 {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit {
                firewall-authentication {
                    web-authentication {
                        client-match FWClient1;
                    }
                }
            }
        }
    }
}

user@host# show system services
...
ftp;
ssh;
telnet;
web-management {
    http {
        interface g-0/0/1.0;
    }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Firewall User Authentication and Monitoring Users and IP Addresses in the Authentication Table on page 404

### *Verifying Firewall User Authentication and Monitoring Users and IP Addresses in the Authentication Table*

**Purpose** Display firewall authentication user history and verify the number of firewall users who successfully authenticated and firewall users who failed to log in.

**Action** From operational mode, enter these **show** commands:

```
user@host> show security firewall-authentication history
```

```

user@host> show security firewall-authentication history identifier 1
user@host> show security firewall-authentication users
user@host> show security firewall-authentication users identifier 3

user@host> show security firewall-authentication history
History of firewall authentication data:
Authentications: 1
Id Source Ip Date Time Duration Status User
5 20.20.20.2      2010-04-24 01:08:57 0:10:30    Success  FWClient1

user@host> show security firewall-authentication history identifier 1
Username: FWClient1
Source IP: 20.20.20.2
Authentication state: Success
Authentication method: Web-authentication
Access start date: 2010-10-12
Access start time: 21:24:02
Duration of user access: 0:00:24
Source zone: N/A
Destination zone: N/A
Access profile: WEBAUTH
Bytes sent by this user: 0
Bytes received by this user: 2660

user@host> show security firewall-authentication users
Firewall authentication data:
Total users in table: 1
Id Source Ip Src zone Dst zone Profile Age Status User
4 20.20.20.2      N/A  N/A  WEBAUTH      1 Success  FWClient1

user@host> show security firewall-authentication users identifier 3
Username: FWClient1
Source IP: 20.20.20.2
Authentication state: Success
Authentication method: Web-authentication
Age: 3
Access time remaining: 9
Source zone: N/A
Destination zone: N/A
Access profile: WEBAUTH
Interface Name: ge-0/0/1.0
Bytes sent by this user: 0
Bytes received by this user: 1521

```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Web Authentication on page 397](#)
- [Understanding Firewall Authentication Banner Customization on page 418](#)
- [Security Zones and Interfaces Overview on page 111](#)

## External Authentication

- [Understanding External Authentication Servers on page 406](#)
- [Example: Configuring RADIUS and LDAP User Authentication on page 407](#)

- Example: Configuring SecurID User Authentication on page 411
- Example: Deleting the SecurID Node Secret File on page 414

## Understanding External Authentication Servers

AAA provides an extra level of protection and control for user access in the following ways:

- Authentication determines the firewall user.
- Authorization determines what the firewall user can do.
- Accounting determines what the firewall user did on the network.

You can use authentication alone or with authorization and accounting. Authorization always requires a user to be authenticated first. You can use accounting alone, or with authentication and authorization.

Once the user's credentials are collected, they are processed using firewall user authentication, which supports the following types of servers:

- Local authentication and authorization
- RADIUS authentication and authorization (compatible with Juniper Steel-Belted Radius server)
- LDAP authentication only (supports LDAP version 3 and compatible with Windows AD)
- SecurID authentication only (using an RSA SecurID external authentication server)



**NOTE:** Junos OS also supports administrative authentication using local, RADIUS, and TACACS+ servers. For more information on administrative authentication, see the *Junos OS Administration Guide for Security Devices*.

---

This topic includes the following sections:

- Understanding SecurID User Authentication on page 406

### Understanding SecurID User Authentication

---

SecurID is an authentication method that allows users to enter either static or dynamic passwords as their credentials. A dynamic password is a combination of a user's PIN and a randomly generated token that is valid for a short period of time, approximately one minute. A static password is set for the user on the SecurID server. For example, the SecurID server administrator might set a temporary static password for a user who lost his or her SecurID token.

When a user attempts to access a resource protected by a policy and SecurID is configured in the profile **authentication-order** parameter as either the only authentication mode or the first one to be used, the device forwards the user's credentials to the SecurID server for authentication. If the user enters valid values, the user is allowed access to the requested resource.



**NOTE:** The SecurID server includes a feature that presents a user with a challenge if the user provides wrong credentials repeatedly. However, Junos OS does not support the challenge feature. Instead, the SecurID server administrator must resynchronize the RSA token for the user.

For SecurID, you configure information about the Juniper Networks device on the SecurID server and this information is exported to a file called **sdconf.rec**.

To install the **sdconf.rec** file on the device, you must use an out-of-band method such as FTP. Install the file in a directory whose files are not deleted regularly. Do not put it in a temporary directory. For example, you might install it in **/var/db/secureid/server1/sdconf.rec**.

The **sdconf.rec** file contains information that provides the Juniper Networks device with the address of the SecurID server. You do not need to configure this information explicitly when you configure the SecurID server to be used as the external authentication server.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Firewall User Authentication Overview on page 389](#)
- [Example: Configuring RADIUS and LDAP User Authentication on page 407](#)
- [Example: Configuring SecurID User Authentication on page 411](#)
- [Example: Deleting the SecurID Node Secret File on page 414](#)

### Example: Configuring RADIUS and LDAP User Authentication

This example shows how to configure a device for external authentication.

- [Requirements on page 407](#)
- [Overview on page 407](#)
- [Configuration on page 408](#)
- [Verification on page 411](#)

#### Requirements

Before you begin, create an authentication user group.

#### Overview

You can put several user accounts together to form a user group, which you can store on the local database or on a RADIUS, an LDAP, or a SecurID server. When you reference an authentication user group and an external authentication server in a policy, the traffic matching the policy provokes an authentication check.

This example shows how access profile Profile-1 is configured for external authentication. Two RADIUS servers and one LDAP server are configured in the access profile. However, the order of authentication specifies RADIUS server only, so if the RADIUS server

authentication fails, then the firewall user fails to authenticate. The local database is not accessed.



**NOTE:** If the firewall clients are authenticated by the RADIUS server, then the group-membership VSA returned by the RADIUS server should contain alpha, beta, or gamma client groups in the RADIUS server configuration or in the access profile, Profile-1. Access profiles store usernames and passwords of users or point to external authentication servers where such information is stored.

## Configuration

### CLI Quick Configuration

To quickly configure a device for external authentication, copy the following commands and paste them into the CLI:

```
[edit]
set access profile Profile-1 authentication-order radius
set access profile Profile-1 client Client-1 client-group alpha
set access profile Profile-1 client Client-1 client-group beta
set access profile Profile-1 client Client-1 client-group gamma
set access profile Profile-1 client Client-1 firewall-user password pwd
set access profile Profile-1 client Client-2 client-group alpha
set access profile Profile-1 client Client-2 client-group beta
set access profile Profile-1 client Client-2 firewall-user password pwd
set access profile Profile-1 client Client-3 firewall-user password pwd
set access profile Profile-1 client Client-4 firewall-user password pwd
set access profile Profile-1 session-options client-group alpha
set access profile Profile-1 session-options client-group beta
set access profile Profile-1 session-options client-group gamma
set access profile Profile-1 session-options client-idle-timeout 255
set access profile Profile-1 session-options client-session-timeout 4
set access profile Profile-1 ldap-options base-distinguished-name
CN=users,DC=junos,DC=juniper,DC=net
set access profile Profile-1 ldap-options search search-filter sAMAccountName=
cn=admin,dc=junos,dc=juniper,dc=net
set access profile Profile-1 ldap-options search admin-search distinguished-name
cn=admin,dc=junos,dc=juniper,dc=net
set access profile Profile-1 ldap-options search admin-search password juniper
set access profile Profile-1 ldap-server 3.3.3.3
set access profile Profile-1 radius-server 4.4.4.4 secret juniper
set access profile Profile-1 radius-server 4.4.4.4 retry 10
set access profile Profile-1 radius-server 5.5.5.5 secret juniper
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a device for external authentication:

1. Specify the RADIUS server for external authentication order.

```
[edit]
```

```
user@host# set access profile Profile-1 authentication-order radius
```

2. Configure Client1-4 firewall users and assign the Client-1 firewall user and Client-2 firewall user to client groups.

```
[edit access profile Profile-1]
```

```
user@host# set client Client-1 client-group alpha
```

```
user@host# set client Client-1 client-group beta
```

```
user@host# set client Client-1 client-group gamma
```

```
user@host# set client Client-1 firewall-user password pwd
```

```
user@host# set client Client-2 client-group alpha
```

```
user@host# set client Client-2 client-group beta
```

```
user@host# set client Client-2 firewall-user password pwd
```

```
user@host# set client Client-3 firewall-user password pwd
```

```
user@host# set client Client-4 firewall-user password pwd
```

3. Configure client groups in the session options.

```
[edit access profile Profile-1]
```

```
user@host# set session-options client-group alpha
```

```
user@host# set session-options client-group beta
```

```
user@host# set session-options client-group gamma
```

```
user@host# set session-options client-idle-timeout 255
```

```
user@host# set session-options client-session-timeout 4
```

4. Configure the IP address for the LDAP server and server options.

```
[edit access profile Profile-1]
```

```
user@host# set ldap-options base-distinguished-name  
CN=users,DC=junos,DC=juniper,DC=net
```

```
user@host# set ldap-options search search-filter sAMAccountName=
```

```
user@host# set ldap-options search admin-search password juniper
```

```
user@host# set ldap-options search admin-search distinguished-name  
cn=administrator,cn=users,dc=junos,dc=juniper,dc=net
```

```
user@host# set ldap-server 3.3.3.3
```

5. Configure the IP addresses for the two RADIUS servers.

```
[edit access profile Profile-1]
```

```
user@host# set radius-server 4.4.4.4 secret juniper
```

```
user@host# set radius-server 4.4.4.4 retry 10
```

```
user@host# set radius-server 5.5.5.5 secret juniper
```

**Results** From configuration mode, confirm your configuration by entering the **show access profile Profile-1** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show access profile Profile-1
authentication-order radius;
client Client-1 {
  client-group [ alpha beta gamma ];
  firewall-user {
    password "$9$jpmT9A0REyn6y1"; ## SECRET-DATA
  }
}
client Client-2 {
  client-group [ alpha beta ];
  firewall-user {
    password "$9$IMVRyK7-w4oG-d"; ## SECRET-DATA
  }
}
client Client-3 {
  firewall-user {
    password "$9$GFukPn/tB1h9C"; ## SECRET-DATA
  }
}
client Client-4 {
  firewall-user {
    password "$9$JuZi.FnC00R/9"; ## SECRET-DATA
  }
}
session-options {
  client-group [ alpha beta gamma ];
  client-idle-timeout 255;
  client-session-timeout 4;
}
ldap-options {
  base-distinguished-name CN=users,DC=junos,DC=juniper,DC=net;
  search {
    search-filter sAMAccountName=;
    admin-search {
      distinguished-name
      cn=administrator,cn=users,dc=junos,dc=juniper,dc=net;
      password "$9$PFF/01h1eWB1X7"; ## SECRET-DATA
    }
  }
}
ldap-server {
  3.3.3.3;
}
```



```

radius-server {
  4.4.4.4 {
    secret "$9$Q5WMF3/At0IRc"; ## SECRET-DATA
    retry 10;
  }
  5.5.5.5 {
    secret "$9$YUg4JUDHmPT"; ## SECRET-DATA
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Troubleshooting with Logs on page 411

#### *Troubleshooting with Logs*

**Purpose** Use these logs to identify any issues.

**Action** From operational mode, enter the **show log messages** command and the **show log dcd** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding External Authentication Servers on page 406

## Example: Configuring SecurID User Authentication

This example shows how to configure SecurID as the external authentication server.

- Requirements on page 411
- Overview on page 411
- Configuration on page 412
- Verification on page 414

### Requirements

Before you begin, create an authentication user group.

### Overview

SecurID is an authentication method that allows users to enter either static or dynamic passwords as their credentials. A dynamic password is a combination of a user's PIN and a randomly generated token that is valid for a short period of time, approximately one minute. A static password is set for the user on the SecurID server. For example, the SecurID server administrator might set a temporary static password for a user who lost his or her SecurID token.

When a user attempts to access a resource protected by a policy and SecurID is configured in the profile **authentication-order** parameter as either the only authentication mode or the first one to be used, the device forwards the user's credentials to the SecurID server

for authentication. If the user enters valid values, the user is allowed access to the requested resource.

Specify that Server-1 is to be used as the SecurID server and that its configuration file resides on the device in the `/var/db/secuid/Server-1/sdconf.rec` file. From configuration mode, enter this command:

```
user@host# set access secuid-server Server-1 configuration-file
"/var/db/secuid/Server-1/sdconf.rec"
```

### Configuration

#### CLI Quick Configuration

To quickly configure SecurID as the external authentication server, copy the following commands and paste them into the CLI:

```
[edit]
set access profile Profile-2 authentication-order secuid
set access profile Profile-2 client Client-1 client-group alpha
set access profile Profile-2 client Client-1 client-group beta
set access profile Profile-2 client Client-1 client-group gamma
set access profile Profile-2 client Client-1 firewall-user password pwd
set access profile Profile-2 client Client-2 client-group alpha
set access profile Profile-2 client Client-2 client-group beta
set access profile Profile-2 client Client-2 firewall-user password pwd
set access profile Profile-2 client Client-3 firewall-user password pwd
set access profile Profile-2 client Client-4 firewall-user password pwd
set access profile Profile-2 session-options client-group alpha
set access profile Profile-2 session-options client-group beta
set access profile Profile-2 session-options client-group gamma
set access profile Profile-2 session-options client-idle-timeout 255
set access profile Profile-2 session-options client-session-timeout 4
```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To configure SecurID as the external authentication server:

1. For the Profile-2 profile, configure SecurID as the server to be used for external authentication.

```
[edit]
```

```
user@host# set access profile Profile-2 authentication-order secuid
```

To share a single SecurID server across multiple profiles, for each profile set the **authentication-order** parameter to include **secuid** as the authentication mode.

2. Configure Client1-4 firewall users and assign the Client-1 firewall user and the Client-2 firewall user to client groups.

```
[edit access profile Profile-2]
```

```
user@host# set client Client-1 client-group alpha
```

```
user@host# set client Client-1 client-group beta
```

```

user@host# set client Client-1 client-group gamma

user@host# set client Client-1 firewall-user password pwd

user@host# set client Client-2 client-group alpha

user@host# set client Client-2 client-group beta

user@host# set client Client-2 firewall-user password pwd

user@host# set client Client-3 firewall-user password pwd

user@host# set client Client-4 firewall-user password pwd

```

3. Configure client groups in the session options.

```

[edit access profile Profile-2]

user@host# set session-options client-group alpha

user@host# set session-options client-group beta

user@host# set session-options client-group gamma

user@host# set session-options client-idle-timeout 255

user@host# set session-options client-session-timeout 4

```

**Results** From configuration mode, confirm your configuration by entering the **show access profile Profile-2** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show access profile Profile-2
authentication-order securid;
client Client-1 {
  client-group [ alpha beta gamma ];
  firewall-user {
    password "$9$jpmT9A0REyn6y1"; ## SECRET-DATA
  }
}
client Client-2 {
  client-group [ alpha beta ];
  firewall-user {
    password "$9$IMVRyK7-w4oG-d"; ## SECRET-DATA
  }
}
client Client-3 {
  firewall-user {
    password "$9$GfukPn/tB1h9C"; ## SECRET-DATA
  }
}
client Client-4 {
  firewall-user {

```

```
        password "$9$JuZi.FnCOOR/9"; ## SECRET-DATA
    }
}
session-options {
    client-group [ alpha beta gamma ];
    client-idle-timeout 255;
    client-session-timeout 4;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

---

To confirm that the configuration is working properly, perform this task:

- [Troubleshooting with Logs](#) on page 414

#### ***Troubleshooting with Logs***

**Purpose** Use these logs to identify any issues.

**Action** From operational mode, enter the **show log messages** command and the **show log dcd** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding External Authentication Servers](#) on page 406
- [Example: Deleting the SecurID Node Secret File](#) on page 414

### Example: Deleting the SecurID Node Secret File

This example shows how to delete the node secret file.

- [Requirements](#) on page 414
- [Overview](#) on page 414
- [Configuration](#) on page 415
- [Verification](#) on page 415

### Requirements

---

Before you begin, confirm that it is necessary to delete the SecurID node secret file.

### Overview

---

When the Juniper Networks device initially communicates successfully with the SecurID server, a node secret file is created for it automatically. The file is created as a result of the handshake between the Juniper Networks device and the SecurID server after the software authenticates the first user successfully. All subsequent communication between the Juniper Networks device and the SecurID server relies on this secret as a representation of trust between the two nodes instead of repeating the handshake with each authentication request.

Under normal circumstances you should not delete the node secret file. In the rare case that you must do so, for example, to debug a serious problem, you can use the **clear** command to remove the file.



**WARNING:** If you delete the file, you must deselect a box on the SecurID server to indicate that the node secret file for the Juniper Networks device and the SecurID server no longer exists. Otherwise, authentication attempts will fail.

---

### Configuration

#### Step-by-Step Procedure

To delete the node secret file:

1. Use the **clear** command to remove the node secret file. During subsequent user authentication, the device reestablishes a shared secret with the SecurID server and re-creates the node secret file. From operational mode, enter the **clear network-access** command to clear the **securid-node-secret-file** for the Juniper Networks device.

```
user@host> clear network-access securid-node-secret-file
```

2. From operational mode, confirm your deletion by entering the **show network-access securid-node-secret-file** command. If the output does not display, repeat the instructions in this example to correct it.

```
user@host> show network-access securid-node-secret-file
```

---

### Verification

Verify the deletion by entering the **show network-access securid-node-secret-file** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding External Authentication Servers on page 406](#)
- [Example: Configuring SecurID User Authentication on page 411](#)

---

## Client Groups for Firewall Authentication

- [Understanding Client Groups for Firewall Authentication on page 415](#)
- [Example: Configuring Local Users for Client Groups on page 416](#)

### Understanding Client Groups for Firewall Authentication

To manage a number of firewall users, you can create user or client groups and store the information either on the local Juniper Networks device or on an external RADIUS or LDAP server.

A client group is a list of groups to which the client belongs. As with client-idle timeout, a client group is used only if the external authentication server does not return a value in its response (for example, LDAP servers do not return such information).

The RADIUS server sends the client's group information to the Juniper Networks device using Juniper VSA (46). The client-match portion of the policy accepts a string that can either be the username or groupname the client belongs to.

The reason to have a single database for different types of clients (except admins) is based on the assumption that a single client can be of multiple types. For example, a firewall user client can also be an L2TP client.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Firewall User Authentication Overview on page 389](#)
- [Example: Configuring Local Users for Client Groups on page 416](#)

## Example: Configuring Local Users for Client Groups

This example shows how to configure a local user for client groups in a profile.

- [Requirements on page 416](#)
- [Overview on page 416](#)
- [Configuration on page 416](#)
- [Verification on page 417](#)

### Requirements

---

Before you begin, create an access profile. See [Example: Configuring the Access Profile](#).

### Overview

---

A client group is a list of groups to which the client belongs. As with client-idle timeout, a client group is used only if the external authentication server does not return a value in its response (for example, LDAP servers do not return such information).

This example shows how to configure a local user called Client-1 for client groups G1, G2, and G3 in a profile called Managers. Within this example, client groups are configured for a client. If a client group is not defined for the client, then the client group under the **access profile session-options** hierarchy is used.

### Configuration

---

**CLI Quick Configuration**

To quickly configure a local user for client groups in a profile, copy the following commands and paste them into the CLI:

```
[edit]
set access profile Managers client Client-1 client-group G1
set access profile Managers client Client-1 client-group G2
set access profile Managers client Client-1 client-group G3
set access profile Managers client Client-1 firewall-user password pwd
set access profile Managers session-options client-group G1
set access profile Managers session-options client-group G2
set access profile Managers session-options client-group G3
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To configure a local user for client groups in a profile:

1. Configure the firewall user and assign client groups to it.

```
[edit access profile Managers]
```

```
user@host# set client Client-1 client-group G1
```

```
user@host# set client Client-1 client-group G2
```

```
user@host# set client Client-1 client-group G3
```

```
user@host# set client Client-1 firewall-user password pwd
```

2. Configure client groups in the session options.

```
[edit access profile Managers]
```

```
user@host# set session-options client-group G1
```

```
user@host# set session-options client-group G2
```

```
user@host# set session-options client-group G3
```

**Results** Confirm your configuration by entering the **show access profile Managers** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
```

```
user@host# show access profile Managers
```

```
client Client-1 {
  client-group [ G1 G2 G3 ];
  firewall-user {
    password "$9$jpmT9A0REyn6y1"; ## SECRET-DATA
  }
}
session-options {
  client-group [ G1 G2 G3 ];
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Troubleshooting with Logs on page 418

**Troubleshooting with Logs**

**Purpose** Use these logs to identify any issues.

**Action** From operational mode, enter the **show log messages** command and the **show log dcd** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Client Groups for Firewall Authentication on page 415

## Firewall Authentication Banner Customization

- Understanding Firewall Authentication Banner Customization on page 418
- Example: Customizing a Firewall Authentication Banner on page 418

### Understanding Firewall Authentication Banner Customization

A banner is a message that appears on a monitor in different places depending on the type of login. (See Figure 36 on page 418.)

**Figure 36: Banner Customization**



- At the top of a browser screen after a user has successfully logged into a Web authentication address as shown Figure 36 on page 418
- Before or after a Telnet, an FTP, or an HTTP login prompt, success message, and fail message for users

All of the banners, except for the one for a console login, already have default messages. You can customize the messages that appear on the banners to better suit the network environment in which you use the device.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Firewall User Authentication Overview on page 389
- Example: Customizing a Firewall Authentication Banner on page 418

### Example: Customizing a Firewall Authentication Banner

This example shows how to customize the banner text that appears in the browser.

- Requirements on page 419
- Overview on page 419
- Configuration on page 419
- Verification on page 420



## Requirements

Before you begin, create an access profile. See Example: Configuring the Access Profile.

## Overview

A banner is a message that appears on a monitor in different places depending on the type of login. This example shows how to change the banner that appears in the browser to indicate that a user has successfully authenticated after successfully logging in through Web authentication. The new message is “Web authentication is successful.” If the authentication fails, then the new message reads “Authentication failed.”

## Configuration

### CLI Quick Configuration

To quickly customize the banner text that appears in the browser, copy the following commands and paste them into the CLI:

```
[edit]
set access firewall-authentication pass-through default-profile Profile-1
set access firewall-authentication pass-through ftp banner fail " Authentication failed"
set access firewall-authentication web-authentication default-profile Profile-1
set access firewall-authentication web-authentication banner success " Web authentication is successful"
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To customize the banner text that appears in the browser:

1. Specify the banner text for failed pass-through authentication through FTP.

```
[edit]
```

```
user@host# set access firewall-authentication pass-through default-profile Profile-1
```

```
user@host# set access firewall-authentication pass-through ftp banner fail "
Authentication failed"
```

2. Specify the banner text for successful Web authentication.

```
[edit]
```

```
user@host# set access web-authentication default-profile Profile-1
```

```
user@host# set access web-authentication banner success " Web authentication
is successful"
```

### Results

From configuration mode, confirm your configuration by entering the **show access firewall-authentication** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show access firewall-authentication
pass-through {
    default-profile Profile-1;
```

```
ftp {
  banner {
    fail "Authentication failed";
  }
}
web-authentication {
  default-profile Profile-1;
  banner {
    success "Web authentication is successful";
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

---

To confirm that the configuration is working properly, perform this task:

- [Troubleshooting with Logs on page 420](#)

#### ***Troubleshooting with Logs***

**Purpose** Use these logs to identify any issues.

**Action** From operational mode, enter the **show log messages** command and the **show log dcd** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Firewall Authentication Banner Customization on page 418](#)

# Infranet Authentication

- UAC and Junos OS on page 421
- Junos OS Enforcer and Infranet Controller Communications on page 424
- Junos OS Enforcer Policy Enforcement on page 426
- Junos OS Enforcer and IPsec on page 429
- Junos OS Enforcer and Infranet Agent Endpoint Security on page 437
- Junos OS Enforcer and Captive Portal on page 438
- Junos OS Enforcer and Infranet Controller Cluster Failover on page 447

## UAC and Junos OS

---

- Understanding UAC in a Junos OS Environment on page 421
- Enabling UAC in a Junos OS Environment (CLI Procedure) on page 423

### Understanding UAC in a Junos OS Environment

A Unified Access Control (UAC) deployment uses the following components to secure a network and ensure that only qualified end users can access protected resources:

- Infranet Controllers—An Infranet Controller is a policy decision point in the network. It uses authentication information and policy rules to determine whether or not to provide access to individual resources on the network. You can deploy one or more Infranet Controllers in your network.



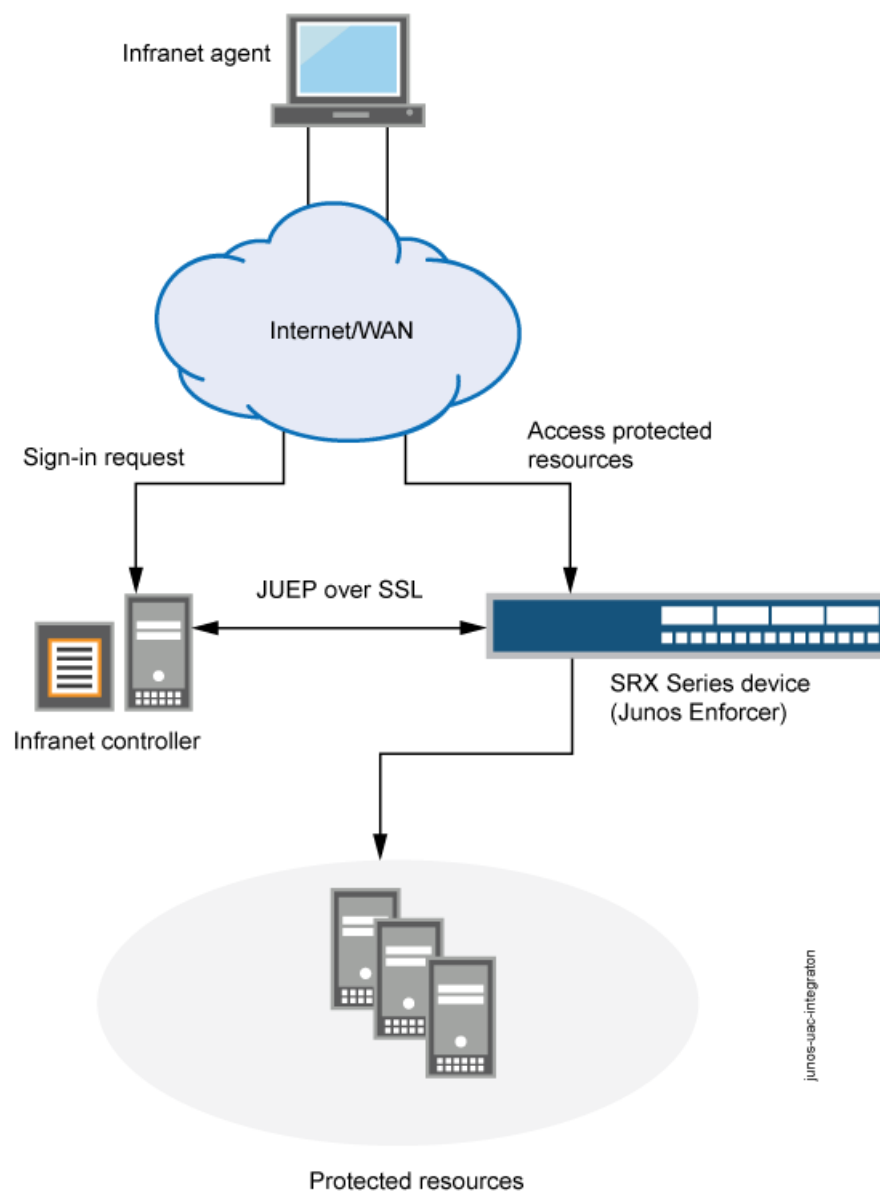
**NOTE:** Any change in the Unified Access Control's (UAC) contact interval and timeout values in the SRX Series or J Series device will be effective only after the next reconnection of the SRX Series or J Series device with the Infranet Controller.

- Infranet Enforcers—An Infranet Enforcer is a policy enforcement point in the network. It receives policies from the Infranet Controller and uses the rules defined in those policies to determine whether or not to allow an endpoint access to a resource. You deploy the Infranet Enforcers in front of the servers and resources that you want to protect.

- Infranet agents—An Infranet agent is a client-side component that runs directly on network endpoints (such as users' computers). The agent checks that the endpoint complies to the security criteria specified in Host Checker policies and relays that compliance information to the Infranet Enforcer. The Infranet Enforcer then allows or denies the endpoint access based on the compliance results.

An SRX Series or J Series device can act as an Infranet Enforcer in a UAC network. Specifically, it acts as a Layer 3 enforcement point, controlling access by using IP-based policies pushed down from the Infranet Controller. When deployed in a UAC network, an SRX Series or J Series device is called a *Junos OS Enforcer*. See Figure 37 on page 422.

**Figure 37: Integrating a Junos Security Device into a Unified Access Control Network**





**NOTE:** You can use the Junos OS Enforcer with the Infranet Controller and Secure Access devices in an *IF-MAP Federation* network. In a federated network, multiple Infranet Controllers and Secure Access devices that are not directly connected to the Junos OS Enforcer can access resources protected by the security device. There are no configuration tasks for IF-MAP Federation on the Junos OS Enforcer. You configure policies on Infranet Controllers that can dynamically create authentication table entries on the Junos OS Enforcer. See the *Unified Access Control Administration Guide*.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Unified Access Control Administration Guide](#)
- Enabling UAC in a Junos OS Environment (CLI Procedure) on page 423

### Enabling UAC in a Junos OS Environment (CLI Procedure)

Junos OS security policies enforce rules for transit traffic, defining what traffic can pass through the Juniper Networks device. The policies control traffic that enters from one zone (from-zone) and exits another (to-zone). To enable an SRX Series or J Series device as a Junos OS Enforcer in a UAC deployment, you must:

- Identify the source and destination zones through which UAC traffic will travel. It also needs the list of interfaces, including which zones they are in. The Infranet Controller uses the destination zone to match its own IPsec routing policies configured on Infranet Controller.
- Identify Junos OS security policies that encompass those zones, and enable UAC for those policies.

Before you begin:

1. Set up the interfaces through which UAC traffic should enter the SRX Series or J Series device. See [Junos OS Interfaces Configuration Guide for Security Devices](#).
2. Group interfaces with identical security requirements into zones. See “Example: Creating Security Zones” on page 114.
3. Create security policies to control the traffic that passes through the security zones. See “Example: Configuring a Security Policy to Permit or Deny All Traffic” on page 152.

To configure UAC through a Junos OS security policy, enter the following configuration statement:

```
user@host# set security policies from-zone zone-name to-zone zone-name policy match
then permit application-services uac-policy
```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding UAC in a Junos OS Environment on page 421

## Junos OS Enforcer and Infranet Controller Communications

---

- [Understanding Communications Between the Junos OS Enforcer and the Infranet Controller](#) on page 424
- [Configuring Communications Between the Junos OS Enforcer and the Infranet Controller \(CLI Procedure\)](#) on page 424

### Understanding Communications Between the Junos OS Enforcer and the Infranet Controller

When you configure an SRX Series or J Series device to connect to an Infranet Controller, the SRX Series or J Series device and the Infranet Controller establish secure communications as follows:

1. The Infranet Controller presents its server certificate to the SRX Series or J Series device. If configured to do so, the SRX Series or J Series device verifies the certificate. (Server certificate verification is not required; however, as an extra security measure you can verify the certificate to implement an additional layer of trust.)
2. The SRX Series or J Series device and the Infranet Controller perform mutual authentication using the proprietary challenge-response authentication. For security reasons, the password is not included in the message sent to the Infranet Controller.
3. After successfully authenticating the SRX Series or J Series device, the Infranet Controller sends it user authentication and resource access policy information. The SRX Series and J Series devices use this information to act as the Junos OS Enforcer in the UAC network.
4. Thereafter, the Infranet Controller and the Junos OS Enforcer can communicate freely with one another over the SSL connection. The communications are controlled by a proprietary protocol called *Junos UAC Enforcer Protocol (JUEP)*.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Unified Access Control Administration Guide](#)
- [Understanding UAC in a Junos OS Environment](#) on page 421
- [Configuring Communications Between the Junos OS Enforcer and the IC Series UAC Appliance \(CLI Procedure\)](#) on page 424

### Configuring Communications Between the Junos OS Enforcer and the Infranet Controller (CLI Procedure)

To configure an SRX Series or J Series device to act as a Junos OS Enforcer in a UAC deployment, and therefore to enforce Infranet Controller policies, you must specify an Infranet Controller to which the SRX Series or J Series device should connect.

Before you begin:

1. Enable UAC through the relevant Junos OS security policies. See “Enabling UAC in a Junos OS Environment (CLI Procedure)” on page 423.
2. (Optional) Import the Infranet Controller’s server certificate onto the SRX Series or J Series device and create a profile for the certificate authority (CA) that signed the certificate. See “Example: Loading CA and Local Certificates Manually” on page 585.
3. Configure user authentication and authorization by setting up user roles, authentication and authorization servers, and authentication realms on the Infranet Controller. See the *Unified Access Control Administration Guide*.
4. Configure resource access policies on the Infranet Controller to specify which endpoints are allowed or denied access to protected resources. See the *Unified Access Control Administration Guide*.

To configure an SRX Series or J Series device to act as a Junos OS Enforcer:

1. Specify the Infranet Controller(s) to which the SRX Series or J Series device should connect.

- To specify the Infranet Controller’s hostname:

```
user@host# set services unified-access-control infranet-controller hostname
```

- To specify the Infranet Controller’s IP address:

```
user@host# set services unified-access-control infranet-controller hostname address ip-address
```



**NOTE:** When configuring access to multiple Infranet Controllers, you must define each separately. For example:

```
user@host# set services unified-access-control infranet-controller IC1
user@host# set services unified-access-control infranet-controller IC2
user@host# set services unified-access-control infranet-controller IC3

user@host# set services unified-access-control infranet-controller IC1
address 10.10.10.1
user@host# set services unified-access-control infranet-controller IC2
address 10.10.10.2
user@host# set services unified-access-control infranet-controller IC3
address 10.10.10.3
```

Make sure that all of the Infranet Controllers are members of the same cluster.



**NOTE:** By default, the Infranet Controller should select port 11123. To determine if this default has changed, see the *Unified Access Control Administration Guide*.

- Specify the Junos OS interface to which the Infranet Controller should connect:

```
user@host# set services unified-access-control infranet-controller hostname interface interface-name
```

- Specify the password that the SRX Series or J Series device should use to initiate secure communications with the Infranet Controller:



**NOTE:** Any change in the Unified Access Control's (UAC) contact interval and timeout values in the SRX Series or J Series device will be effective only after the next reconnection of the SRX Series or J Series device with the Infranet Controller.

```
user@host# set services unified-access-control infranet-controller hostname password password
```

- (Optional) Specify information about the certificate that the device should use for SSL communications with the Infranet Controller.

- To specify the certificate that the device should use:

```
user@host# set services unified-access-control infranet-controller hostname server-certificate-subject certificate-name
```

- To specify the CA profile associated with the certificate:

```
user@host# set services unified-access-control infranet-controller hostname ca-profile ca-profile
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Communications Between the Junos OS Enforcer and the IC Series UAC Appliance on page 424

## Junos OS Enforcer Policy Enforcement

- Understanding Junos OS Enforcer Policy Enforcement on page 427
- Testing Junos OS Enforcer Policy Access Decisions Using Test-Only Mode (CLI Procedure) on page 428
- Verifying Junos OS Enforcer Policy Enforcement on page 429



## Understanding Junos OS Enforcer Policy Enforcement

Once the SRX Series or J Series device has successfully established itself as the Junos OS Enforcer, it secures traffic as follows:

1. First, the Junos OS Enforcer uses the appropriate Junos OS security policy to process the traffic. A *security policy* uses criteria such as the traffic's source IP address or the time of day that the traffic was received to determine whether or not the traffic should be allowed to pass.
2. Once it determines that the traffic may pass based on the Junos OS security policy, the Junos OS Enforcer maps the traffic flow to an authentication table entry. The Junos OS Enforcer uses the source IP address of the first packet in the flow to create the mapping.

An *authentication table entry* contains the source IP address and user role(s) of a user who has already successfully established a UAC session. A *user role* identifies a group of users based on criteria such as type (for instance, "Engineering" or "Marketing") or status (for instance, "Antivirus Running"). The Junos OS Enforcer determines whether to allow or deny the traffic to pass based on the authentication results stored in the appropriate authentication table entry.

The Infranet Controller pushes authentication table entries to the Junos OS Enforcer when the devices first connect to one another and, as necessary, throughout the session. For example, the Infranet Controller might push updated authentication table entries to the Junos OS Enforcer when the user's computer becomes noncompliant with endpoint security policies, when you change the configuration of a user's role, or when you disable all user accounts on the Infranet Controller in response to a security problem such as a virus on the network.

If the Junos OS Enforcer drops a packet due to a missing authentication table entry, the device sends a message to the Infranet Controller, which in turn may provision a new authentication table entry and send it to the Junos OS Enforcer. This process is called *dynamic authentication table provisioning*.

3. Once it determines that the traffic may pass based on the authentication table entries, the Junos OS Enforcer maps the flow to a resource. The Junos OS Enforcer uses the destination IP address specified in the flow to create the mapping. Then the device uses that resource as well as the user role specified in the authentication table entry to map the flow to a resource access policy.

A *resource access policy* specifies a particular resource to which you want to control access based on user role. For instance, you might create a resource access policy that allows only users who are members of the Engineering and Antivirus Running user roles access to the Engineering-Only server. Or you might create a resource access policy that allows members of the No Antivirus Running user role access to the Remediation server on which antivirus software is available for download.

The Infranet Controller pushes resource access policies to the Junos OS Enforcer when the devices first connect to one another and when you modify your resource access policy configurations on the Infranet Controller.

If the Junos OS Enforcer drops the packet because of a “deny” policy, the Junos OS Enforcer sends a message to the Infranet Controller, which in turn sends a message to the endpoint’s Odyssey Access Client (if available). (The Infranet Controller does not send “deny” messages to the agentless client.)

4. Once it determines that the traffic may pass based on the resource access policies, the Junos OS Enforcer processes the traffic using the remaining application services defined in the Junos OS policy. The Junos OS Enforcer runs the remaining services in the following order: Intrusion Detection and Prevention (IDP), URL filtering, and Application Layer Gateways (ALGs).

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Unified Access Control Administration Guide](#)
- Understanding Communications Between the Junos OS Enforcer and the IC Series UAC Appliance on page 424
- Security Policies Overview on page 145
- Testing Junos OS Enforcer Policy Access Decisions Using Test-Only Mode (CLI Procedure) on page 428
- Verifying Junos OS Enforcer Policy Enforcement on page 429

### Testing Junos OS Enforcer Policy Access Decisions Using Test-Only Mode (CLI Procedure)

When configured in test-only mode, the SRX Series or J Series device enables all UAC traffic to go through regardless of the UAC policy settings. The device logs the UAC policy’s access decisions without enforcing them so you can test the implementation without impeding traffic.

Before you begin:

1. Enable UAC through the relevant Junos OS security policies. See “Enabling UAC in a Junos OS Environment (CLI Procedure)” on page 423
2. Configure the SRX Series and J Series devices as a Junos OS Enforcer. See “Configuring Communications Between the Junos OS Enforcer and the Infranet Controller (CLI Procedure)” on page 424.
3. If you are connecting to a cluster of Infranet Controllers, enable failover options. See “Configuring Junos OS Enforcer Failover Options (CLI Procedure)” on page 447.

To activate or deactivate test-only mode, enter the following configuration statement:

```
user@host# set services unified-access-control test-only-mode (true | false)
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Junos OS Enforcer Policy Enforcement on page 427
- Verifying Junos OS Enforcer Policy Enforcement on page 429

## Verifying Junos OS Enforcer Policy Enforcement

- Displaying Infranet Controller Authentication Table Entries from the Junos OS Enforcer on page 429
- Displaying Infranet Controller Resource Access Policies from the Junos OS Enforcer on page 429

### Displaying Infranet Controller Authentication Table Entries from the Junos OS Enforcer

**Purpose** Display a summary of the authentication table entries configured from the Infranet Controller.

**Action** Enter the `show services unified-access-control authentication-table` CLI command.

### Displaying Infranet Controller Resource Access Policies from the Junos OS Enforcer

**Purpose** Display a summary of UAC resource access policies configured from the Infranet Controller.

**Action** Enter the `show services unified-access-control policies` CLI command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Junos OS Enforcer Policy Enforcement on page 427](#)
- [Testing Junos OS Enforcer Policy Access Decisions Using Test-Only Mode \(CLI Procedure\) on page 428](#)
- [Junos OS CLI Reference](#)

## Junos OS Enforcer and IPsec

- [Understanding Junos OS Enforcer Implementations Using IPsec on page 429](#)
- [Example: Configuring the Device as a Junos OS Enforcer Using IPsec \(CLI\) on page 431](#)

### Understanding Junos OS Enforcer Implementations Using IPsec

To configure an SRX Series or J Series device to act as a Junos OS Enforcer using IPsec, you must:

- Include the identity configured under the security IKE gateway. The identity is a string such as "gateway1.juniper.net", where gateway1.juniper.net distinguishes between IKE gateways. (The identities specify for which tunnel traffic is intended.)
- Include the preshared seed. This generates the preshared key from the full identity of the remote user for Phase 1 credentials.
- Include the RADIUS shared secret. This allows the Infranet Controller to accept RADIUS packets for extended authentication (XAuth) from the Junos OS Infranet Enforcer.

When configuring IPsec between the Infranet Controller, the Odyssey Access Client, and the SRX or J Series device, you should note that the following are IKE (or Phase 1) proposal methods or protocol configurations that are supported from the Infranet Controller to the Odyssey Access Client:

- IKE proposal: **authentication-method pre-shared-keys** (you must specify **pre-shared-keys**)
- IKE policy:
  - **mode aggressive** (you must use aggressive mode)
  - **pre-shared-key ascii-text key** (only ASCII text preshared-keys are supported)
- IKE gateway: dynamic
  - **hostname *identity*** (you must specify a unique identity among gateways)
  - **ike-user-type group-ike-id** (you must specify **group-ike-id**)
  - **xauth access-profile *profile*** (you must specify **xauth**)

The following are IPsec (or Phase 2) proposal methods or protocol configurations that are supported from the Infranet Controller to the Odyssey Access Client.

- IPsec proposal: **protocol esp** (you must specify **esp**)
- IPsec VPN: **establish-tunnels immediately** (you must specify **establish-tunnels immediately**)

**NOTE:**

- Only one IPsec VPN tunnel is supported per from-zone to to-zone security policy. This is a limitation on the Infranet Controller.
  - Junos OS security policies enable you to define multiple policies differentiated by different source addresses, destination addresses, or both. The Infranet Controller, however, cannot differentiate such configurations. If you enable multiple policies in this manner, the Infranet Controller could potentially identify the incorrect IKE gateway.
- 

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Unified Access Control Administration Guide](#)
- [Understanding Junos OS Enforcer Policy Enforcement on page 427](#)
- [VPN Overview on page 451](#)
- [Security Policies Overview on page 145](#)
- [Example: Configuring the Device as a Junos OS Enforcer Using IPsec \(CLI\) on page 431](#)

## Example: Configuring the Device as a Junos OS Enforcer Using IPsec (CLI)

To configure an SRX Series or J Series device to act as a Junos OS Enforcer using IPsec:

1. Set system and syslog information using the following configuration statements:

```

system {
  host-name test_host;
  domain-name test.juniper.net;
  host-name test_host;
  root-authentication {
    encrypted-password "$1$uhqXoD0T$6h26f0xXExOqkPHQLvaTF0";
  }
  services {
    ftp;
    ssh;
    telnet;
    web-management {
      http {
        interface ge-0/0/0.0;
      }
    }
  }
}
syslog {
  user * {
    any emergency;
  }
  file messages {
    any critical;
    authorization info;
  }
  file interactive-commands {
    interactive-commands error;
  }
}
max-configurations-on-flash 5;
max-configuration-rollback 5;
license {
  autoupdate {
    url https://ae1.juniper.net/junos/key_retrieval;
  }
}
ntp {
  boot-server 1.2.3.4;
  server 1.2.3.4;
}
}

```

2. Configure the interfaces using the following configuration statements:

```

interfaces {
  ge-0/0/0 {
    unit 0 {
      family inet {
        address 10.64.75.135/16;
      }
    }
  }
}

```

```

    }
  }
  ge-0/0/1 {
    unit 0 {
      family inet {
        address 10.100.54.1/16;
      }
    }
  }
  ge-0/0/2 {
    unit 0 {
      family inet {
        address 10.101.54.1/16;
      }
    }
  }
}

```

3. Configure routing options using the following configuration statements:

```

routing-options {
  static {
    route 0.0.0.0/0 next-hop 10.64.0.1;
    route 10.11.0.0/16 next-hop 10.64.0.1;
    route 172.0.0.0/8 next-hop 10.64.0.1;
    route 10.64.0.0/16 next-hop 10.64.0.1;
  }
}

```

4. Configure security options using the following configuration statements:

```

security {
  ike {
    traceoptions {
      file ike;
      flag all;
    }
    proposal prop1 {
      authentication-method pre-shared-keys;
      dh-group group2;
      authentication-algorithm sha1;
      encryption-algorithm 3des-cbc;
    }
    policy pol1 {
      mode aggressive;
      proposals prop1;
      pre-shared-key ascii-text "$9$YS4Z]mPQ6CuTz6Au0cSvWLxNbiHm";
    }
    gateway gateway1 {
      ike-policy pol1;
      dynamic {
        hostname gateway1.juniper.net;
        connections-limit 1000;
        ike-user-type group-ike-id;
      }
      external-interface ge-0/0/0;
      xauth access-profile infranet;
    }
  }
}

```

```

gateway gateway2 {
  ike-policy pol1;
  dynamic {
    hostname gateway2.juniper.net;
    connections-limit 1000;
    ike-user-type group-ike-id;
  }
  external-interface ge-0/0/0;
  xauth access-profile infranet;
}
}

```

5. Configure IPsec parameters using the following configuration statements:

```

ipsec {
  proposal prop1 {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
    lifetime-seconds 86400;
  }
  policy pol1 {
    proposals prop1;
  }
  vpn vpn1 {
    ike {
      gateway gateway1;
      ipsec-policy pol1;
    }
    establish-tunnels immediately;
  }
  vpn vpn2 {
    ike {
      gateway gateway2;
      ipsec-policy pol1;
    }
    establish-tunnels immediately;
  }
}
}

```

6. Configure screen options using the following configuration statements:

```

screen {
  ids-option untrust-screen {
    icmp {
      ping-death;
    }
    ip {
      source-route-option;
      tear-drop;
    }
    tcp {
      syn-flood {
        alarm-threshold 1024;
        attack-threshold 200;
        source-threshold 1024;
        destination-threshold 2048;
      }
    }
  }
}

```

```

        queue-size 2000;
        timeout 20;
    }
    land;
}
}
}

```

7. Configure zones using the following configuration statements:

```

zones {
  security-zone trust {
    tcp-rst;
    host-inbound-traffic {
      system-services {
        all;
      }
      protocols {
        all;
      }
    }
    interfaces {
      ge-0/0/0.0;
    }
  }
  security-zone untrust {
    host-inbound-traffic {
      system-services {
        all;
      }
      protocols {
        all;
      }
    }
    interfaces {
      ge-0/0/1.0;
    }
  }
  security-zone zone101 {
    host-inbound-traffic {
      system-services {
        all;
      }
      protocols {
        all;
      }
    }
    interfaces {
      ge-0/0/2.0;
    }
  }
}
}
}

```

8. Configure policies for UAC using the following configuration statements:

```

policies {
  inactive: from-zone trust to-zone trust {

```



```
policy default-permit {
  match {
    source-address any;
    destination-address any;
    application any;
  }
  then {
    permit;
  }
}
}
}
from-zone trust to-zone untrust {
inactive: policy default-permit {
  match {
    source-address any;
    destination-address any;
    application any;
  }
  then {
    permit;
  }
}
}
inactive: policy default-deny {
  match {
    source-address any;
    destination-address any;
    application any;
  }
  then {
    permit;
  }
}
}
policy pol1 {
  match {
    source-address any;
    destination-address any;
    application any;
  }
  then {
    permit {
      tunnel {
        ipsec-vpn vpn1;
      }
      application-services {
        uac-policy;
      }
    }
  }
  log {
    session-init;
    session-close;
  }
}
}
}
from-zone untrust to-zone trust {
  policy pol1 {
```

```
match {
  source-address any;
  destination-address any;
  application any;
}
then {
  permit;
  log {
    session-init;
    session-close;
  }
}
}
}
from-zone trust to-zone zone101 {
policy pol1 {
  match {
    source-address any;
    destination-address any;
    application any;
  }
  then {
    permit {
      tunnel {
        ipsec-vpn vpn2;
      }
      application-services {
        uac-policy;
      }
    }
    log {
      session-init;
      session-close;
    }
  }
}
}
policy test {
  match {
    source-address any;
    destination-address any;
    application any;
  }
  then {
    permit;
  }
}
}
}
default-policy {
  deny-all;
}
}
}
```

9. Configure RADIUS server authentication access using the following configuration statements:

```

access {
  profile infranet {
    authentication-order radius;
    radius-server {
      10.64.160.120 secret "$9$KBoWX-YgJHqfVwqfTzCAvWL";
    }
  }
}

```

10. Configure services for UAC using the following configuration statements:

```

services {
  unified-access-control {
    inactive: infranet-controller IC27 {
      address 3.23.1.2;
      interface ge-0/0/0.0;
      password "$9$Wjl8X-Vb2GDkev4aGUHkuOB";
    }
    infranet-controller prabaIC {
      address 10.64.160.120;
      interface ge-0/0/0.0;
      password "$9$jdkmT69pRhrz3hrev7Nik.";
    }
    traceoptions {
      flag all;
    }
  }
}

```

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Junos OS Enforcer Implementations Using IPsec on page 429

## Junos OS Enforcer and Infranet Agent Endpoint Security

- Understanding Endpoint Security Using the Infranet Agent with the Junos OS Enforcer on page 437
- Configuring Endpoint Security Using the Infranet Agent with the Junos OS Enforcer on page 438

### Understanding Endpoint Security Using the Infranet Agent with the Junos OS Enforcer

An Infranet agent helps you secure traffic on your network starting with the endpoints that initiate communications as follows:

1. The Infranet agent, which runs directly on the endpoint, checks that the endpoint is compliant with your Unified Access Control (UAC) Host Checker policies. You can use a wide variety of criteria within a UAC Host Checker policy to determine compliance. For example, you can configure the Host Checker policy to confirm that the endpoint is running antivirus software or a firewall or that the endpoint is not running specific types of malware or processes.
2. The Infranet agent transmits the compliance information to the Junos OS Enforcer.

3. The Junos OS Enforcer allows or denies the endpoint access to the resources on your network based on the Host Checker compliance results.

Because the Infranet agent runs directly on the endpoint, you can use the Infranet agent to check the endpoint for security compliance at any time. For instance, when a user tries to sign into the Infranet Controller, you can require the Infranet agent to send compliance results immediately—the user will not even see the sign-in page until the Infranet agent returns positive compliance results to the Infranet Controller. You can also configure the Infranet agent to check for compliance after the user signs in or periodically during the user session.

If the endpoints running the Infranet agent have appropriate access, they will automatically send their compliance results to the Infranet Controller and the Infranet Controller will update the authentication table entries accordingly and push them to the Junos OS Enforcer. The Junos OS Enforcer supports connections with the Odyssey Access Client and “agentless” Infranet agents.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Unified Access Control Administration Guide](#)
- Understanding UAC in a Junos OS Environment on page 421
- Configuring Endpoint Security Using the Infranet Agent with the Junos OS Enforcer on page 438

## Configuring Endpoint Security Using the Infranet Agent with the Junos OS Enforcer

To integrate the Infranet agent into a Junos OS-UAC deployment, no special configuration is required on the Junos OS Enforcer. You simply need to create security policies enabling access to the appropriate endpoints as you would for any other Junos OS-UAC deployment.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policies Overview on page 145
- Understanding Endpoint Security Using the Infranet Agent with the Junos OS Enforcer on page 437

## Junos OS Enforcer and Captive Portal

---

- Understanding the Captive Portal on the Junos OS Enforcer on page 439
- Understanding Captive Portal Configuration on the Junos OS Enforcer on page 440
- Example: Creating a Captive Portal Policy on the Junos OS Enforcer on page 441
- Understanding the Captive Portal Redirect URL Options on page 444
- Example: Configuring a Redirect URL for Captive Portal on page 445

## Understanding the Captive Portal on the Junos OS Enforcer

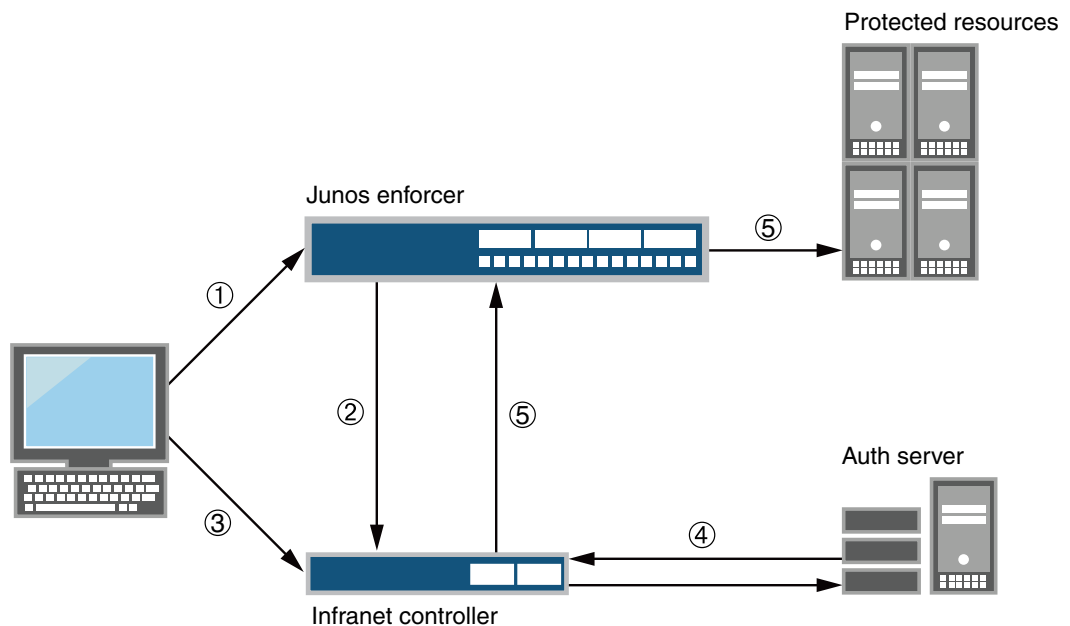
In a Unified Access Control (UAC) deployment, users might not be aware that they must first sign in to the Infranet Controller for authentication and endpoint security checking before they are allowed to access a protected resource behind the Junos OS Enforcers. To help users sign in to the Infranet Controller, you can configure the captive portal feature. The captive portal feature allows you to configure a policy in the Junos OS Enforcer that automatically redirects HTTP traffic destined for protected resources to the Infranet Controller or to a URL configured in the Junos OS Enforcer.

You can configure a captive portal for deployments that use either source IP enforcement or IPsec enforcement, or a combination of both enforcement methods.

Figure 38 on page 439 shows the captive portal feature enabled on a Junos OS Enforcer. Users accessing protected resources are automatically redirected to the Infranet Controller:

1. Users point to a protected resource using the browser.
2. The Junos OS Enforcer determines that the user is not authenticated and redirects the request to the Infranet Controller or another server.
3. Users enter their Infranet username and password to log in.
4. The Infranet Controller passes the user credentials to an authentication server.
5. After authentication, the Infranet Controller redirects the users to the protected resource they wanted to access.

Figure 38: Enabling the Captive Portal Feature on a Junos OS Enforcer



By default, the Junos OS Enforcer encodes and forwards to the Infranet Controller the protected resource URL that the user entered. The Infranet Controller uses the protected

resource URL to help users navigate to the protected resource. The manner in which the Infranet Controller uses the protected resource URL depends on whether or not the user's endpoint is running the Odyssey Access Client or Junos Pulse. If the user's endpoint is not running the Odyssey Access Client or Junos Pulse (that is, it is in an agentless or Java agent configuration), the Infranet Controller automatically opens a new browser window and uses HTTP to access the protected resource after the user signs in. If the endpoint is using the Odyssey Access Client, the Infranet Controller inserts a hypertext link in the webpage that automatically opens after the user signs in. The user must then click that hypertext link to access the protected resource by means of HTTP in the same browser window.

The Junos OS Enforcer supports the captive portal feature only for HTTP traffic. If you attempt to access a protected resource by using HTTPS or a non-browser application (such as an e-mail application), the Junos OS Enforcer does not redirect the traffic. When using HTTPS or a non-browser application, you must manually sign in to the Infranet Controller first before attempting to access protected resources.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding UAC in a Junos OS Environment on page 421](#)
- [Understanding Captive Portal Configuration on the Junos OS Enforcer on page 440](#)
- [Example: Creating a Captive Portal Policy on the Junos OS Enforcer on page 441](#)
- [Understanding the Captive Portal Redirect URL Options on page 444](#)
- [Example: Configuring a Redirect URL for Captive Portal on page 445](#)

## Understanding Captive Portal Configuration on the Junos OS Enforcer

To configure the captive portal feature, you create a security policy on the Junos OS Enforcer and then specify a redirection option for the captive portal security policy. You can choose to redirect traffic to an external server or to the Infranet Controller. You can also choose to redirect all traffic or unauthenticated traffic only.

- **Redirecting traffic to an external webserver**—You can configure the Junos OS Enforcer to redirect HTTP traffic to an external webserver instead of the Infranet Controller. For example, you can redirect HTTP traffic to a webpage that explains to users the requirement to sign in to the Infranet Controller before they can access the protected resource. You could also include a link to the Infranet Controller on that webpage to help users sign in.
- **Redirecting unauthenticated traffic**—Select this option if your deployment uses source IP only or a combination of source IP and IPsec. The Junos OS Enforcer redirects clear-text traffic from unauthenticated users to the currently connected Infranet Controller or to an IP address or domain name that you specify in a redirect URL. After a user signs in to the Infranet Controller and the user's endpoint system meets the requirements of the Infranet Controller's security policies, the Junos OS Enforcer allows the user's clear-text traffic to pass through in source IP deployments. For IPsec deployments, the Odyssey Access Client creates a VPN tunnel between the user and the Junos OS Enforcer. The Junos OS Enforcer then applies the VPN policy, allowing the encrypted traffic to pass through.

- Redirecting all traffic—Specify this option if you want to redirect all traffic to the URL that you specify in a redirect URL.
- Redirecting traffic with multiple Infranet Controllers—You can configure multiple Infranet Controllers on your Junos OS Enforcer, but it is connected to only one Infranet Controller at any given time. If the connection to the Infranet Controller fails, the Junos OS Enforcer tries to connect to next configured Infranet Controller. As a result, you cannot be sure which Infranet Controller is connected to the Junos OS Enforcer at any given time. To ensure that the Junos OS Enforcer redirects traffic to the connected Infranet Controller configure the default redirect URL or the %ic-ip% option in the URL.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding UAC in a Junos OS Environment on page 421](#)
- [Understanding the Captive Portal on the Junos OS Enforcer on page 439](#)
- [Example: Creating a Captive Portal Policy on the Junos OS Enforcer on page 441](#)
- [Understanding the Captive Portal Redirect URL Options on page 444](#)
- [Example: Configuring a Redirect URL for Captive Portal on page 445](#)

### Example: Creating a Captive Portal Policy on the Junos OS Enforcer

This example shows how to create a captive portal policy on the Junos OS Enforcer. In this example, you deploy a Junos OS Enforcer in front of the data center resources you want to protect and configure the captive portal feature on the Junos OS Enforcer. The Junos OS Enforcer then automatically redirects HTTP traffic destined for the protected resource to the Infranet Controller for authentication.

- [Requirements on page 441](#)
- [Overview on page 442](#)
- [Configuration on page 442](#)
- [Verification on page 443](#)

#### Requirements

Before you begin:

- Deploy the Infranet Controller in the network so that users can access the device. Use the internal port on the Infranet Controller to connect users, the Junos OS Enforcer (an SRX210 device in this example), and authentication servers. See “Configuring Communications Between the Junos OS Enforcer and the IC Series UAC Appliance (CLI Procedure)” on page 424.
- Set up security zones and interfaces on the Junos OS Enforcer. Make sure that end users are in a different security zone than protected resources. For example, protected resources in the data center are configured in the trusted zone and users in an untrusted zone. See “Example: Creating Security Zones” on page 114.
- Add individual users to either an external authentication server or the local authentication server. Set up roles and realms for individual users. You can provision

access to protected resources based on your network security needs. See the *Unified Access Control Administration Guide*.

## Overview

In this example, you want to protect the trusted zone from users on the LAN by making sure that only compliant and authenticated users are granted access. New users join your network every month. You want to configure the captive portal feature on your system so that unauthenticated users are redirected to the Infranet Controller automatically without requiring new users to remember to log in to the Infranet Controller.

The configuration instructions in this topic describe how to create a security policy called **my-policy**, specify a match condition for this policy, specify the captive portal policy as a part of the UAC policy, and set criteria for redirecting traffic to the Infranet Controller. In this example, the policy **my-policy**:

- Specifies the match condition to include any traffic from a previously configured zone called **trust** to another previously configured zone called **untrust**.
- Specifies the captive portal policy called **my-captive-portal-policy** as part of the UAC policy.
- Specifies the redirect-traffic criteria as **unauthenticated**.

## Configuration

### CLI Quick Configuration

To quickly create a captive portal policy, copy the following commands and paste them into the CLI:

```
[edit]
set security policies from-zone untrust to-zone trust policy my-policy match
  destination-address any source-address any application any
set security policies from-zone untrust to-zone trust policy my-policy then permit
  application-services uac-policy captive-portal my-captive-portal-policy
set services unified-access-control captive-portal my-captive-portal-policy redirect-traffic
  unauthenticated
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To create a captive portal policy on the Junos OS Enforcer:

1. Specify the match condition for the policy.

```
[edit security policies from-zone untrust to-zone trust policy my-policy]
user@host# set match destination-address any source-address any application
  any
```

2. Specify the captive portal policy as part of the UAC policy to be applied on the traffic that matches the conditions specified in the security policy.

```
[edit security policies from-zone untrust to-zone trust policy my-policy]
user@host# set then permit application-services uac-policy captive-portal
  my-captive-portal-policy
```



3. Redirect all unauthenticated traffic to the Infranet Controller.

```
[edit services unified-access-control]
user@host# set captive-portal my-captive-portal-policy redirect-traffic
unauthenticated
```

**Results** Confirm your configuration by entering the **show services** and **show security policies** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
user@host# show services
unified-access-control {
  captive-portal my-captive-portal-policy {
    redirect-traffic unauthenticated;
  }
}

[edit]
user@host# show security policies
...
from-zone untrust to-zone trust {
  policy my-policy {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit {
        application-services {
          uac-policy {
            captive-portal my-captive-portal-policy;
          }
        }
      }
    }
  }
}
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform this task:

- Verifying the Captive Portal Policy on page 443

### *Verifying the Captive Portal Policy*

**Purpose** Verify that the captive portal policy was created.

**Action** From operational mode, enter the **show security policies detail** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding UAC in a Junos OS Environment on page 421](#)
- [Understanding the Captive Portal on the Junos OS Enforcer on page 439](#)
- [Understanding Captive Portal Configuration on the Junos OS Enforcer on page 440](#)
- [Understanding the Captive Portal Redirect URL Options on page 444](#)
- [Example: Configuring a Redirect URL for Captive Portal on page 445](#)

## Understanding the Captive Portal Redirect URL Options

By default, after you configure a captive portal policy, the Junos OS Enforcer redirects HTTP traffic to the currently connected Infranet Controller by using HTTPS. To perform the redirection, the Junos OS Enforcer uses the IP address or domain name that you specified when you configured the Infranet Controller instance on the Junos OS Enforcer. The format of the URL that the Junos OS Enforcer uses for default redirection is:

```
https://%ic-ip%/?target = %dest-url% &enforcer = %enforcer-id% &policy = %policy-id% &dest-ip = %dest-ip%
```

If you configured your Junos OS Enforcer to work with multiple Infranet Controllers in a cluster, and the current Infranet Controller becomes disconnected, the Junos OS Enforcer automatically redirects HTTP traffic to the next active Infranet Controller in its configuration list. The Junos OS Enforcer redirects traffic to only one Infranet Controller at a time.

Otherwise, the browser displays a certificate warning to users when they sign in. You do not need to override the default redirection destination except in these situations:

- You are using a VIP for a cluster of Infranet Controller appliances and the Junos OS Enforcer is configured to connect to the Infranet Controller's physical IP addresses.
- You want to redirect traffic to a webserver instead of the Infranet Controller.
- If, because of split DNS or IP routing restrictions at your site, the Junos OS Enforcer uses a different address for the Infranet Controller than endpoints, you must specify the domain name or IP address that endpoints must use to access the Infranet Controller.

Table 39 on page 444 lists different options that you can configure in the redirect URL string.

**Table 39: Redirect URL String Options**

|                            |                                                                                |
|----------------------------|--------------------------------------------------------------------------------|
| <code>%dest-url%</code>    | Specifies the protected resource which the user is trying to access.           |
| <code>%enforcer-id%</code> | Specifies the ID assigned to the Junos OS Enforcer by the Infranet Controller. |

Table 39: Redirect URL String Options (*continued*)

|                          |                                                                                                                        |
|--------------------------|------------------------------------------------------------------------------------------------------------------------|
| <code>%policy-id%</code> | Specifies the encrypted policy ID for the captive portal security policy that redirected the traffic.                  |
| <code>%dest-ip%</code>   | Specifies the IP address or hostname of the protected resource which the user is trying to access.                     |
| <code>%ic-ip%</code>     | Specifies the IP address or hostname of the Infranet Controller to which the Junos OS Enforcer is currently connected. |

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding UAC in a Junos OS Environment on page 421](#)
- [Understanding the Captive Portal on the Junos OS Enforcer on page 439](#)
- [Understanding Captive Portal Configuration on the Junos OS Enforcer on page 440](#)
- [Example: Creating a Captive Portal Policy on the Junos OS Enforcer on page 441](#)
- [Example: Configuring a Redirect URL for Captive Portal on page 445](#)

### Example: Configuring a Redirect URL for Captive Portal

This example shows how to redirect traffic to the currently connected Infranet Controller or to an IP address or domain name that you specify in a redirect URL. We recommend the default configuration that redirects traffic to the Infranet Controller for authentication.

- [Requirements on page 445](#)
- [Overview on page 445](#)
- [Configuration on page 446](#)
- [Verification on page 446](#)

#### Requirements

Before you specify the redirect URL, make sure you configure the captive portal policy. For information about creating the captive portal policy, see “Example: Creating a Captive Portal Policy on the Junos OS Enforcer” on page 441.

#### Overview

In this example, you configure the URL to redirect traffic to the Infranet Controller and after authentication to forward the traffic automatically to the protected resource. The configuration instructions in this topic describe how to set the URL to `https://my-website.com`.

You can redirect traffic to the currently connected Infranet Controller or to an IP address or domain name that you specify in a redirect URL. We recommend the default configuration that redirects traffic to the Infranet Controller for authentication.

If you need to override the default redirection destination, you can specify any combination of redirect options:

- **https://IP or domain name/URL path/target=%dest-url%**—Forwards users to the protected resource automatically after authentication with the Infranet Controller or webserver. The Junos OS Enforcer replaces the `%dest-url%` parameter with the protected resource URL and then forwards the protected resource URL in encrypted form to the Infranet Controller.
- **https://IP or domain name/target=URL path**—Forwards users to the specified URL automatically after authentication with the Infranet Controller or webserver.
- **https://IP or domain name/URL path**—Redirects users to the Infranet Controller authentication page but not be forwarded to the protected resource after authentication. Users must manually open a new browser window and enter the protected resource URL again after signing in.
- **redirect-all**—Redirects all traffic to the URL that you specify in a redirect URL.

### Configuration

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To configure the redirect URL for the captive portal feature on the Junos OS Enforcer:

1. Specify the redirect URL for the preconfigured captive portal policy.
 

```
[edit services unified-access-control]
user@host# set captive-portal my-captive-portal-policy redirect-url
https://192.168.0.100/target=my-website.com
```
2. If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show services unified-access-control captive-portal my-captive-portal-policy** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding UAC in a Junos OS Environment on page 421](#)
- [Understanding the Captive Portal on the Junos OS Enforcer on page 439](#)
- [Understanding Captive Portal Configuration on the Junos OS Enforcer on page 440](#)
- [Understanding the Captive Portal Redirect URL Options on page 444](#)

## Junos OS Enforcer and Infranet Controller Cluster Failover

- Understanding Communications Between Junos OS Enforcer and a Cluster of Infranet Controllers on page 447
- Configuring Junos OS Enforcer Failover Options (CLI Procedure) on page 447

### Understanding Communications Between Junos OS Enforcer and a Cluster of Infranet Controllers

You can configure a Junos OS Enforcer to work with more than one Infranet Controller in a high availability configuration known as an Infranet Controller cluster. The Junos OS Enforcer communicates with only one Infranet Controller at a time; the other Infranet Controllers are used for failover. If the Junos OS Enforcer cannot connect to the first Infranet Controller you added to a cluster, it tries to connect to the failed Infranet Controller again. Then it fails over to the other Infranet Controllers in the cluster. It continues trying to connect to Infranet Controllers in the cluster until a connection occurs.

When the Junos OS Enforcer cannot establish a connection to an Infranet Enforcer, it preserves all its existing authentication table entries and Unified Access Control (UAC) policies and takes the timeout action that you specify. Timeout actions include:

- **close**—Close existing sessions and block any further traffic. This is the default option.
- **no-change**—Preserve existing sessions and require authentication for new sessions.
- **open**—Preserve existing sessions and allow new sessions access.

Once the Junos OS Enforcer can reestablish a connection to an Infranet Controller, the Infranet Controller compares the authentication table entries and UAC policies stored on the Junos OS Enforcer with the authentication table entries and policies stored on the Infranet Controller and reconciles the two as required.



**NOTE:** The Infranet Controllers configured on a Junos OS Enforcer should all be members of the same Infranet Controller cluster.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Unified Access Control Administration Guide](#)
- Understanding Junos OS Enforcer Policy Enforcement on page 427
- Understanding Junos OS Enforcer Policy Enforcement on page 427
- Configuring Junos OS Enforcer Failover Options (CLI Procedure) on page 447

### Configuring Junos OS Enforcer Failover Options (CLI Procedure)

To configure Infranet Controller failover processing, you must configure the Junos OS Enforcer to connect to a cluster of Infranet Controllers. The Junos OS Enforcer communicates with one of these Infranet Controllers at a time and uses the others for failover processing.

Before you begin:

1. Enable UAC through the relevant Junos OS security policies. See “Enabling UAC in a Junos OS Environment (CLI Procedure)” on page 423.
2. Configure the SRX Series or J Series device as a Junos OS Enforcer. During the configuration, define a cluster of Infranet Controllers to which the Junos OS Enforcer should connect. See “Enabling UAC in a Junos OS Environment (CLI Procedure)” on page 423.

To configure failover processing:

1. Specify how often (in seconds) the Junos OS Enforcer should expect a heartbeat signal from the Infranet Controller indicating an active connection:

```
user@host# set services unified-access-control interval seconds
```

2. Specify the interval (in seconds) at which the Junos OS Enforcer should consider the current connection timed out:

```
user@host# set services unified-access-control timeout seconds
```

3. Specify how the Junos OS Enforcer should handle all current and subsequent traffic sessions when its connection to an Infranet Controller cluster times out:

```
user@host# set services unified-access-control timeout-action (close | no-change | open)
```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Communications Between Junos OS Enforcer and a Cluster of IC Series UAC Appliances on page 447](#)

## PART 6

# Virtual Private Networks

- Internet Protocol Security on page 451
- Public Key Cryptography for Certificates on page 569
- Dynamic VPNs on page 597
- Group VPNs on page 655





# Internet Protocol Security

- VPN Overview on page 451
- Understanding IKE and IPsec Packet Processing on page 458
- Understanding Phase 1 of IKE Tunnel Negotiation on page 467
- Understanding Phase 2 of IKE Tunnel Negotiation on page 468
- Route-Based VPNs on page 470
- Policy-Based VPNs on page 488
- Hub-and-Spoke VPNs on page 506
- Configuring IPsec VPN Using the VPN Wizard on page 538
- Understanding IPv6 IKE and IPsec Packet Processing on page 538
- IPv6 IPsec Configuration Overview on page 543
- Example: Configuring an IPv6 IPsec Manual VPN on page 543
- Example: Configuring an IPv6 AutoKey IKE Policy-Based VPN on page 546
- Global SPI and VPN Monitoring Features on page 562
- Virtual Router Support for Route-Based VPNs on page 563
- Example: Configuring an st0 Interface in a Virtual Router on page 564
- Understanding Virtual Router Limitations on page 568

## VPN Overview

---

A virtual private network (VPN) provides a means for securely communicating among remote computers across a public WAN such as the Internet.

A VPN connection can link two LANs (site-to-site VPN) or a remote dial-up user and a LAN. The traffic that flows between these two points passes through shared resources such as routers, switches, and other network equipment that make up the public WAN. To secure VPN communication while passing through the WAN, the two participants create an IP Security (IPsec) tunnel.



**NOTE:** The term *tunnel* does not denote tunnel mode (see “Packet Processing in Tunnel Mode” on page 459). Instead, it refers to the IPsec connection.

IPsec is a suite of related protocols for cryptographically securing communications at the IP Packet Layer. IPsec also provides methods for the manual and automatic negotiation of security associations (SAs) and key distribution, all the attributes for which are gathered in a domain of interpretation (DOI). The IPsec DOI is a document containing definitions for all the security parameters required for the successful negotiation of a VPN tunnel—essentially, all the attributes required for SA and IKE negotiations. See RFC 2407 and RFC 2408 for more information.

This topic includes the following sections:

- IPsec VPN Topologies on page 452
- Comparison of Policy-Based VPNs and Route-Based VPNs on page 452
- Security Associations on page 453
- IPsec Key Management on page 454
- IPsec Security Protocols on page 456
- IPsec Tunnel Negotiation on page 457
- Distributed VPNs in SRX Series Services Gateways on page 458

## IPsec VPN Topologies

The following are some of the IPsec VPN topologies that Junos operating system (OS) supports:

- Site-to-site VPNs—Connects two sites in an organization together and allows secure communications between the sites.
- Hub-and-spoke VPNs—Connects branch offices to the corporate office in an enterprise network. You can also use this topology to connect spokes together by sending traffic through the hub.
- Remote access VPNs—Allows users working at home or traveling to connect to the corporate office and its resources. This topology is sometimes referred to as an *end-to-site tunnel*.

## Comparison of Policy-Based VPNs and Route-Based VPNs

Table 40 on page 452 summarizes the differences between policy-based VPNs and route-based VPNs.

**Table 40: Comparison Between Policy-Based VPNs and Route-Based VPNs**

| Policy-Based VPNs                                                                                                                                                              | Route-Based VPNs                                                                               |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| In policy-based VPNs, a tunnel is treated as an object that, together with source, destination, application, and action, constitutes a tunnel policy that permits VPN traffic. | In route-based VPNs, a policy does not specifically reference a VPN tunnel.                    |
| A tunnel policy specifically references a VPN tunnel by name.                                                                                                                  | A route determines which traffic is sent through the tunnel based on a destination IP address. |

Table 40: Comparison Between Policy-Based VPNs and Route-Based VPNs (*continued*)

| Policy-Based VPNs                                                                                                                                                                                                                   | Route-Based VPNs                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| The number of policy-based VPN tunnels that you can create is limited by the number of tunnels that the device supports.                                                                                                            | The number of route-based VPN tunnels that you create is limited by the number of st0 interfaces (for point-to-point VPNs) or the number of tunnels that the device supports, whichever is lower.                                                                                                                                                                                                     |
| With a policy-based VPN, although you can create numerous tunnel policies referencing the same VPN tunnel, each tunnel policy pair creates an individual IPsec SA with the remote peer. Each SA counts as an individual VPN tunnel. | Because the route, not the policy, determines which traffic goes through the tunnel, multiple policies can be supported with a single SA or VPN.                                                                                                                                                                                                                                                      |
| In a policy-based VPN, the action must be permit and must include a tunnel.                                                                                                                                                         | In a route-based VPN, the regulation of traffic is not coupled to the means of its delivery.                                                                                                                                                                                                                                                                                                          |
| The exchange of dynamic routing information is not supported in policy-based VPNs.                                                                                                                                                  | Route-based VPNs support the exchange of dynamic routing information through VPN tunnels. You can enable an instance of a dynamic routing protocol, such as OSPF, on an st0 interface that is bound to a VPN tunnel.                                                                                                                                                                                  |
| If you need more granularity than a route can provide to specify the traffic sent to a tunnel, using a policy-based VPN with security policies is the best choice.                                                                  | Route-based VPNs uses routes to specify the traffic sent to a tunnel; a policy does not specifically reference a VPN tunnel.                                                                                                                                                                                                                                                                          |
| With a policy-based VPN tunnel, you can consider a tunnel as an element in the construction of a policy.                                                                                                                            | <p>When the security device does a route lookup to find the interface through which it must send traffic to reach an address, it finds a route through a secure tunnel (st0) interface.</p> <p>With a route-based VPN tunnel, you can consider a tunnel as a means for delivering traffic, and can consider the policy as a method for either permitting or denying the delivery of that traffic.</p> |

## Security Associations

A security association (SA) is a unidirectional agreement between the VPN participants regarding the methods and parameters to use in securing a communication channel. Full bidirectional communication requires at least two SAs, one for each direction. Through the SA, an IPsec tunnel can provide the following security functions:

- Privacy (through encryption)
- Content integrity (through data authentication)
- Sender authentication and—if using certificates—nonrepudiation (through data origin authentication)

The security functions you employ depend on your needs. If you need only to authenticate the IP packet source and content integrity, you can authenticate the packet without applying any encryption. On the other hand, if you are concerned only with preserving privacy, you can encrypt the packet without applying any authentication mechanisms. Optionally, you can both encrypt and authenticate the packet. Most network security designers choose to encrypt, authenticate, and replay-protect their VPN traffic.

An IPsec tunnel consists of a pair of unidirectional SAs—one SA for each direction of the tunnel—that specify the security parameter index (SPI), destination IP address, and security protocol (Authentication Header [AH] or Encapsulating Security Payload [ESP]) employed. An SA groups together the following components for securing communications:

- Security algorithms and keys.
- Protocol mode, either transport or tunnel. Junos OS devices always use tunnel mode. (See “Packet Processing in Tunnel Mode” on page 459.)
- Key-management method, either manual key or AutoKey IKE. (See “IPsec Key Management” on page 454.)
- SA lifetime.

For inbound traffic, Junos OS looks up the SA by using the following triplet:

- Destination IP address.
- Security protocol, either AH or ESP. (See “IPsec Security Protocols” on page 456.)
- Security parameter index (SPI) value.

For outbound VPN traffic, the policy invokes the SA associated with the VPN tunnel.

## IPsec Key Management

The distribution and management of keys are critical to using VPNs successfully. Junos OS supports IPsec technology for creating VPN tunnels with three kinds of key creation mechanisms:

- Manual key
- AutoKey IKE with a preshared key or a certificate

You can choose your key creation mechanism—also called authentication method—during Phase 1 and Phase 2 proposal configuration. See “IPsec Tunnel Negotiation” on page 457.



**NOTE:** Manual key creation and AutoKey IKE with certificates are not supported with the dynamic VPN feature at this time.

---

This topic includes the following sections:

- Manual Key on page 454
- AutoKey IKE on page 455
- Diffie-Hellman Exchange on page 455

### Manual Key

---

With manual keys, administrators at both ends of a tunnel configure all the security parameters. This is a viable technique for small, static networks where the distribution, maintenance, and tracking of keys are not difficult. However, safely distributing manual-key configurations across great distances poses security issues. Aside from

passing the keys face-to-face, you cannot be completely sure that the keys have not been compromised while in transit. Also, whenever you want to change the key, you are faced with the same security issues as when you initially distributed it.

### AutoKey IKE

When you need to create and manage numerous tunnels, you need a method that does not require you to configure every element manually. IPsec supports the automated generation and negotiation of keys and security associations using the Internet Key Exchange (IKE) protocol. Junos OS refers to such automated tunnel negotiation as AutoKey IKE and supports AutoKey IKE with preshared keys and AutoKey IKE with certificates.

- AutoKey IKE with preshared keys—Using AutoKey IKE with preshared keys to authenticate the participants in an IKE session, each side must configure and securely exchange the preshared key in advance. In this regard, the issue of secure key distribution is the same as that with manual keys. However, once distributed, an autokey, unlike a manual key, can automatically change its keys at predetermined intervals using the IKE protocol. Frequently changing keys greatly improves security, and automatically doing so greatly reduces key-management responsibilities. However, changing keys increases traffic overhead; therefore, changing keys too often can reduce data transmission efficiency.



**NOTE:** A preshared key is a key for both encryption and decryption, which both participants must have before initiating communication.

- AutoKey IKE with certificates—When using certificates to authenticate the participants during an AutoKey IKE negotiation, each side generates a public-private key pair and acquires a certificate. As long as the issuing certificate authority (CA) is trusted by both sides, the participants can retrieve the peer's public key and verify the peer's signature. There is no need to keep track of the keys and SAs; IKE does it automatically.

### Diffie-Hellman Exchange

A Diffie-Hellman (DH) exchange allows participants to produce a shared secret value. The strength of the technique is that it allows participants to create the secret value over an unsecured medium without passing the secret value through the wire. There are five DH groups; Junos OS supports groups 1, 2, 5, and 14. The size of the prime modulus used in each group's calculation differs as follows:

- DH Group 1—768-bit modulus
- DH Group 2—1024-bit modulus
- DH Group 5—1536-bit modulus
- DH Group 14—2048-bit modulus



**NOTE:** The strength of DH Group 1 security has depreciated; therefore, we do not recommend its use.

The larger the modulus, the more secure the generated key is considered to be; however, the larger the modulus, the longer the key-generation process takes. Because the modulus for each DH group is a different size, the participants must agree to use the same group.



**NOTE:** If you configure multiple (up to four) proposals for Phase 1 negotiations, use the same DH group in all proposals. The same guideline applies to multiple proposals for Phase 2 negotiations.

## IPsec Security Protocols

IPsec uses two protocols to secure communications at the IP layer:

- Authentication Header (AH)—A security protocol for authenticating the source of an IP packet and verifying the integrity of its content
- Encapsulating Security Payload (ESP)—A security protocol for encrypting the entire IP packet (and authenticating its content)

You can choose your security protocols—also called *authentication and encryption algorithms*—during Phase 2 proposal configuration. See “IPsec Tunnel Negotiation” on page 457.

This topic includes the following sections:

- AH Protocol on page 456
- ESP Protocol on page 457

### AH Protocol

The Authentication Header (AH) protocol provides a means to verify the authenticity and integrity of the content and origin of a packet. You can authenticate the packet by the checksum calculated through a Hash Message Authentication Code (HMAC) using a secret key and either MD5 or SHA-1 hash functions.

- Message Digest 5 (MD5)—An algorithm that produces a 128-bit hash (also called a *digital signature* or *message digest*) from a message of arbitrary length and a 16-byte key. The resulting hash is used, like a fingerprint of the input, to verify content and source authenticity and integrity.
- Secure Hash Algorithm (SHA-1)—An algorithm that produces a 160-bit hash from a message of arbitrary length and a 20-byte key. It is generally regarded as more secure than MD5 because of the larger hashes it produces. Because the computational processing is done in the ASIC, the performance cost is negligible.



**NOTE:** For more information on MD5 hashing algorithms, see RFC 1321 and RFC 2403. For more information on SHA hashing algorithms, see RFC 2404. For more information on HMAC, see RFC 2104.

## ESP Protocol

The Encapsulating Security Payload (ESP) protocol provides a means to ensure privacy (encryption) and source authentication and content integrity (authentication). ESP in tunnel mode encapsulates the entire IP packet (header and payload) and then appends a new IP header to the now-encrypted packet. This new IP header contains the destination address needed to route the protected data through the network. (See “Packet Processing in Tunnel Mode” on page 459.)

With ESP, you can both encrypt and authenticate, encrypt only, or authenticate only. For encryption, you can choose one of the following encryption algorithms:

- Data Encryption Standard (DES)—A cryptographic block algorithm with a 56-bit key.
- Triple DES (3DES)—A more powerful version of DES in which the original DES algorithm is applied in three rounds, using a 168-bit key. DES provides significant performance savings but is considered unacceptable for many classified or sensitive material transfers.
- Advanced Encryption Standard (AES)—An emerging encryption standard which, when adopted by Internet infrastructures worldwide, will offer greater interoperability with other devices. Junos OS supports AES with 128-bit, 192-bit, and 256-bit keys.

For authentication, you can use either the MD5 or the SHA-1 algorithm.



**NOTE:** Even though it is possible to select **NULL** for encryption, it has been demonstrated that IPsec might be vulnerable to attack under such circumstances. Therefore, we suggest that you choose an encryption algorithm for maximum security.

## IPsec Tunnel Negotiation

To establish an AutoKey IKE IPsec tunnel, two phases of negotiation are required:

- In Phase 1, the participants establish a secure channel in which to negotiate the IPsec security associations (SAs).
- In Phase 2, the participants negotiate the IPsec SAs for encrypting and authenticating the ensuing exchanges of user data.

For a manual key IPsec tunnel, because all the SA parameters have been previously defined, there is no need to negotiate which SAs to use. In essence, the tunnel has already been established. When traffic matches a policy using that manual key tunnel or when a route involves the tunnel, the Juniper Networks device simply encrypts and authenticates the data, as you determined, and forwards it to the destination gateway.

The remote IKE gateway address can be in any virtual routing (VR) instance. VR is determined during IKE Phase 1 and Phase 2 negotiation. VR does not have to be configured in the IKE proposals. If the IKE gateway interface is moved from one VR to another, the existing IKE Phase 1 and Phase 2 negotiations for the IKE gateway are cleared, and new Phase 1 and Phase 2 negotiations are performed.



**NOTE:** The combinations of local IP addresses and remote gateway IP addresses of IPsec VPN tunnels configured across VRs have to be unique.



**NOTE:** When the loopback interface is used as the IKE gateway external interface, the physical interface for IKE negotiation should be in the same VR.

## Distributed VPNs in SRX Series Services Gateways

In the SRX3000 and SRX5000 lines, the IKE provides tunnel management for IPsec and authenticates end entities. The IKE performs a Diffie-Hellman (DH) key exchange to generate an IPsec tunnel between network devices. The IPsec tunnels generated by IKE are used to encrypt, decrypt, and authenticate user traffic between the network devices at the IP layer.

The VPN is created by distributing the IKE and IPsec workload among the multiple Services Processing Units (SPUs) of the platform. The IKE workload is distributed based on a key generated from the IKE packet's 4 tuples (source IP address, destination IP addresses, and UDP ports). The workload is distributed by assigning anchoring SPUs logically and mapping the logical SPUs to physical SPUs, based on the composition at that given time. This distribution prevents any change in the number and composition of SPUs in the device, which may happen due to hot swap or SPC failure. The SPU in a device communicates with the Routing Engine to create a distributed VPN.

In IPsec, the workload is distributed by the same algorithm that distributes the IKE. The Phase 2 SA for a given VPN tunnel termination points pair is exclusively owned by a particular SPU, and all IPsec packets belonging to this Phase 2 SA are forwarded to the anchoring SPU of that SA for IPsec processing.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Configuring a Policy-Based VPN on page 489](#)
- [Example: Configuring a Route-Based VPN on page 470](#)
- [Understanding IKE and IPsec Packet Processing on page 458](#)
- [Understanding Phase 1 of IKE Tunnel Negotiation on page 467](#)
- [Understanding Phase 2 of IKE Tunnel Negotiation on page 468](#)
- [Understanding Hub-and-Spoke VPNs on page 506](#)

## Understanding IKE and IPsec Packet Processing

An IPsec VPN tunnel consists of tunnel setup and applied security. During tunnel setup, the peers establish security associations (SAs), which define the parameters for securing traffic between themselves. (See “VPN Overview” on page 451.) After the tunnel is established, IPsec protects the traffic sent between the two tunnel endpoints by applying



the security parameters defined by the SAs during tunnel setup. Within the Junos OS implementation, IPsec is applied in tunnel mode, which supports the Encapsulating Security Payload (ESP) and Authentication Header (AH) protocols.

This topic includes the following sections:

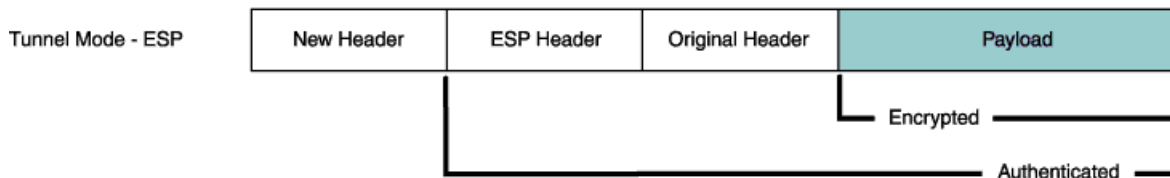
- Packet Processing in Tunnel Mode on page 459
- IKE Packet Processing on page 461
- IPsec Packet Processing on page 464

## Packet Processing in Tunnel Mode

IPsec operates in one of two modes—transport or tunnel. When both ends of the tunnel are hosts, you can use either mode. When at least one of the endpoints of a tunnel is a security gateway, such as a Junos OS router or firewall, you must use tunnel mode. Juniper Networks devices always operate in tunnel mode for IPsec tunnels.

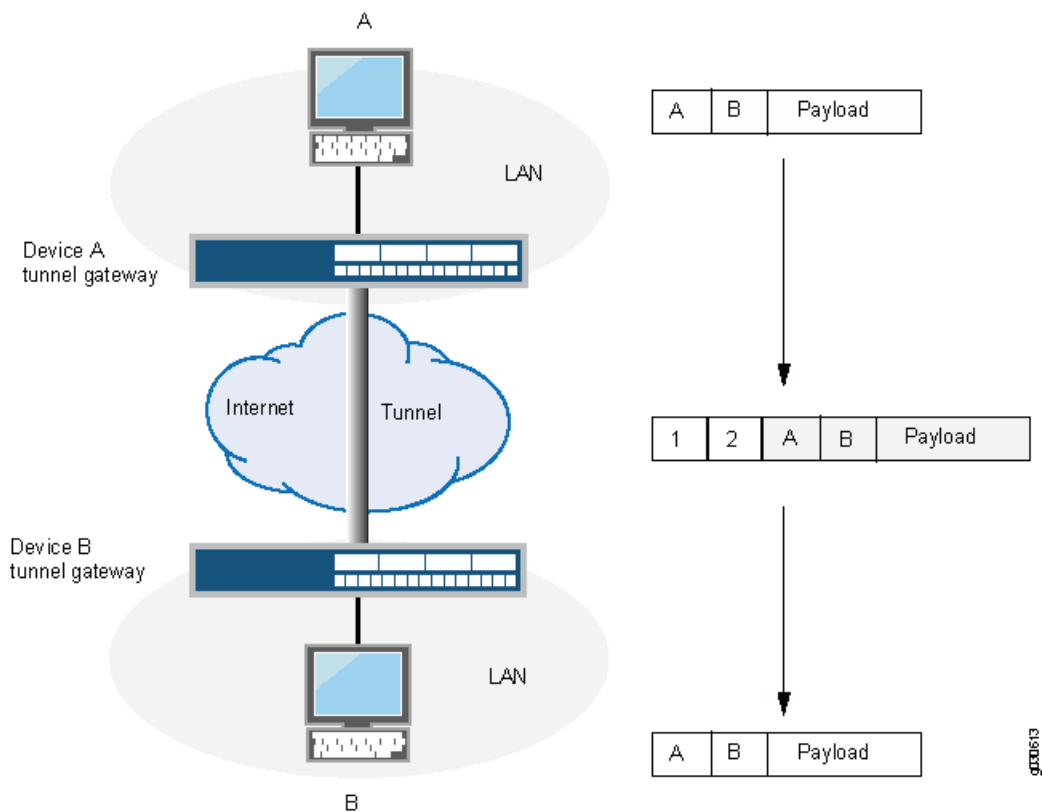
In tunnel mode, the entire original IP packet—payload and header—is encapsulated within another IP payload, and a new header is appended to it, as shown in Figure 39 on page 459. The entire original packet can be encrypted, authenticated, or both. With the Authentication Header (AH) protocol, the AH and new headers are also authenticated. With the Encapsulating Security Payload (ESP) protocol, the ESP header can also be authenticated.

Figure 39: Tunnel Mode



In a site-to-site VPN, the source and destination addresses used in the new header are the IP addresses of the outgoing interface. See Figure 40 on page 460.

Figure 40: Site-to-Site VPN in Tunnel Mode

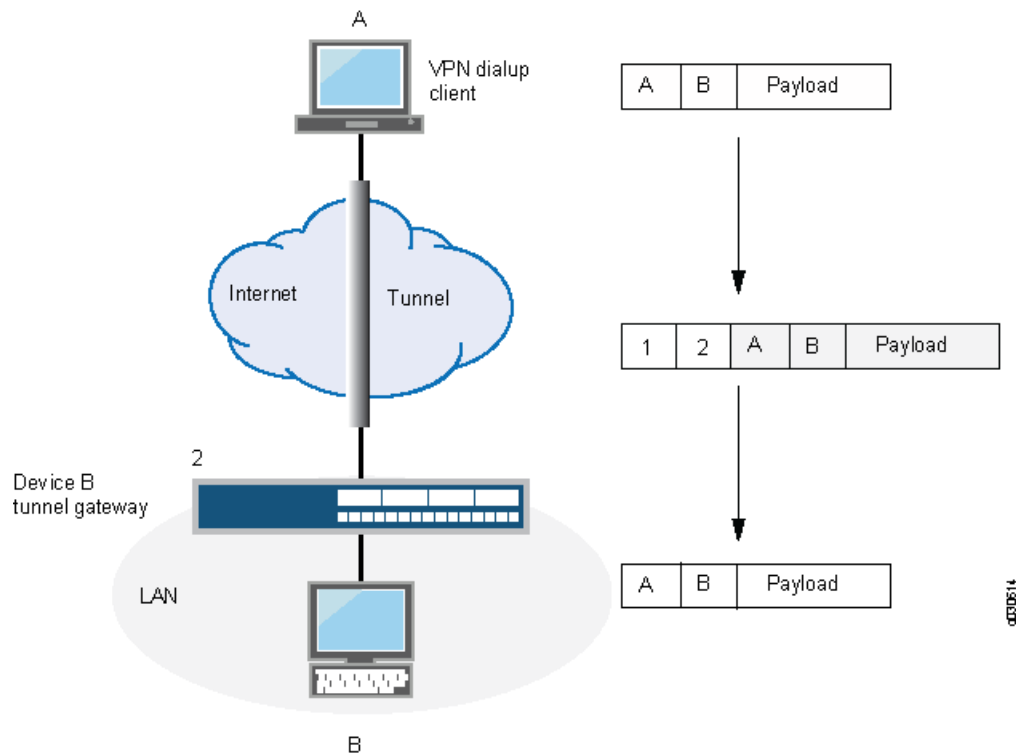


In a dial-up VPN, there is no tunnel gateway on the VPN dial-up client end of the tunnel; the tunnel extends directly to the client itself (see Figure 41 on page 461). In this case, on packets sent from the dial-up client, both the new header and the encapsulated original header have the same IP address: that of the client's computer.



**NOTE:** Some VPN clients, such as the dynamic VPN client and Netscreen-Remote, use a virtual inner IP address (also called a “sticky address”). Netscreen-Remote enables you to define the virtual IP address. The dynamic VPN client uses the virtual IP address assigned during the XAuth configuration exchange. In such cases, the virtual inner IP address is the source IP address in the original packet header of traffic originating from the client, and the IP address that the ISP dynamically assigns the dial-up client is the source IP address in the outer header.

Figure 41: Dial-Up VPN in Tunnel Mode

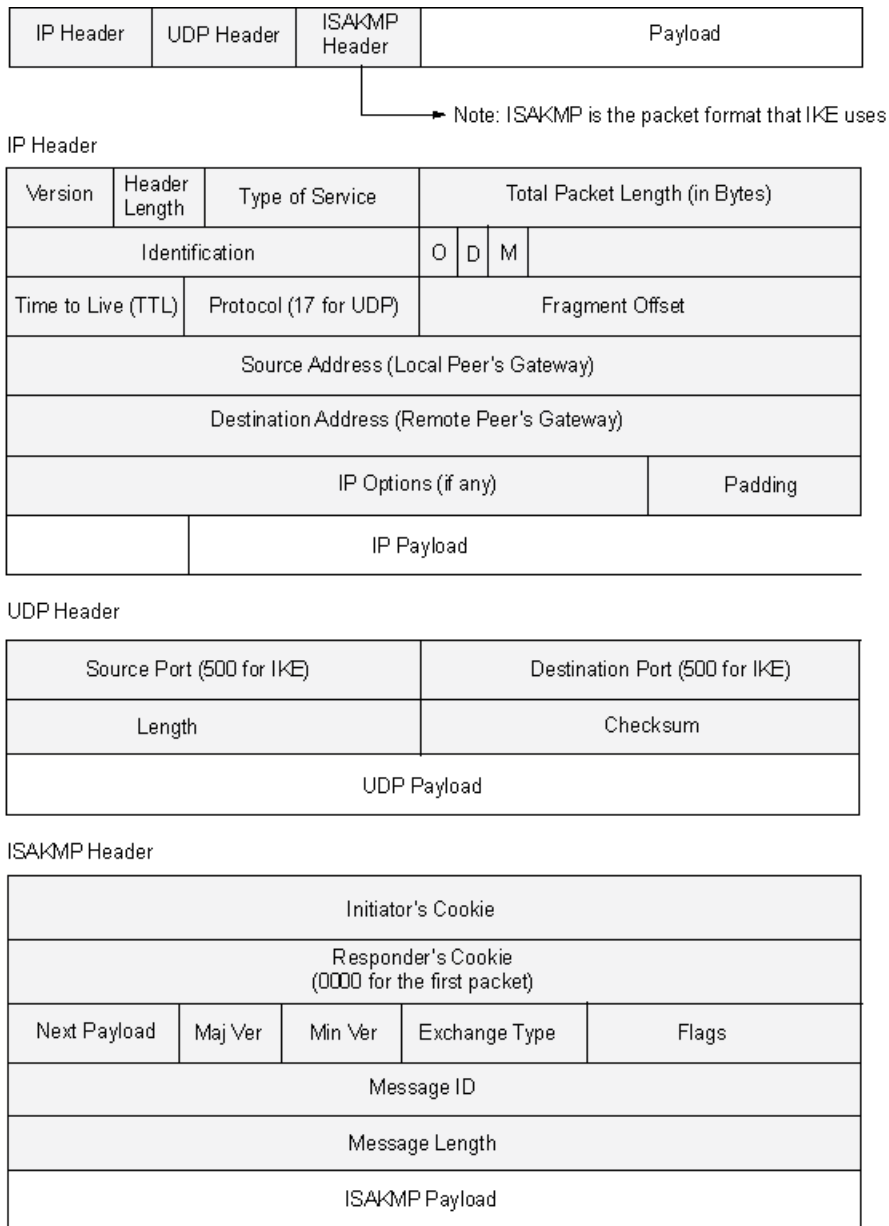


## IKE Packet Processing

When a cleartext packet arrives on a Juniper Networks device that requires tunneling, and no active Phase 2 SA exists for that tunnel, Junos OS begins IKE negotiations and drops the packet. The source and destination addresses in the IP packet header are those of the local and remote IKE gateways, respectively. In the IP packet payload, there is a UDP segment encapsulating an ISAKMP (IKE) packet. The format for IKE packets is the same for Phase 1 and Phase 2. See Figure 42 on page 462.

Meanwhile, the source host has sent the dropped packet again. Typically, by the time the second packet arrives, IKE negotiations are complete, and Junos OS protects the packet and all subsequent packets in the session—with IPsec before forwarding it.

Figure 42: IKE Packet for Phases 1 and 2



The Next Payload field contains a number indicating one of the following payload types:

- 0002—SA Negotiation Payload contains a definition for a Phase 1 or Phase 2 SA.
- 0004—Proposal Payload can be a Phase 1 or Phase 2 proposal.
- 0008—Transform Payload gets encapsulated in a proposal payload that gets encapsulated in an SA payload.
- 0010—Key Exchange (KE) Payload contains information necessary for performing a key exchange, such as a DH public value.

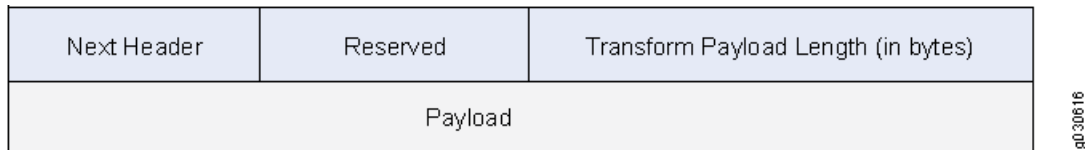
- 0020—Identification (IDx) Payload.
  - In Phase 1, IDi indicates the initiator ID, and IDr indicates the responder ID.
  - In Phase 2, IDui indicates the user initiator, and IDur indicates the user responder.

The IDs are IKE ID types such as FQDN, U-FQDN, IP address, and ASN.1\_DN.

- 0040—Certificate (CERT) Payload.
- 0080—Certificate Request (CERT\_REQ) Payload.
- 0100—Hash (HASH) Payload contains the digest output of a particular hash function.
- 0200—Signature (SIG) Payload contains a digital signature.
- 0400—Nonce (Nx) Payload contains some pseudorandom information necessary for the exchange).
- 0800—Notify Payload.
- 1000—ISAKMP Delete Payload.
- 2000—Vendor ID (VID) Payload can be included anywhere in Phase 1 negotiations. Junos OS uses it to mark support for NAT-T.

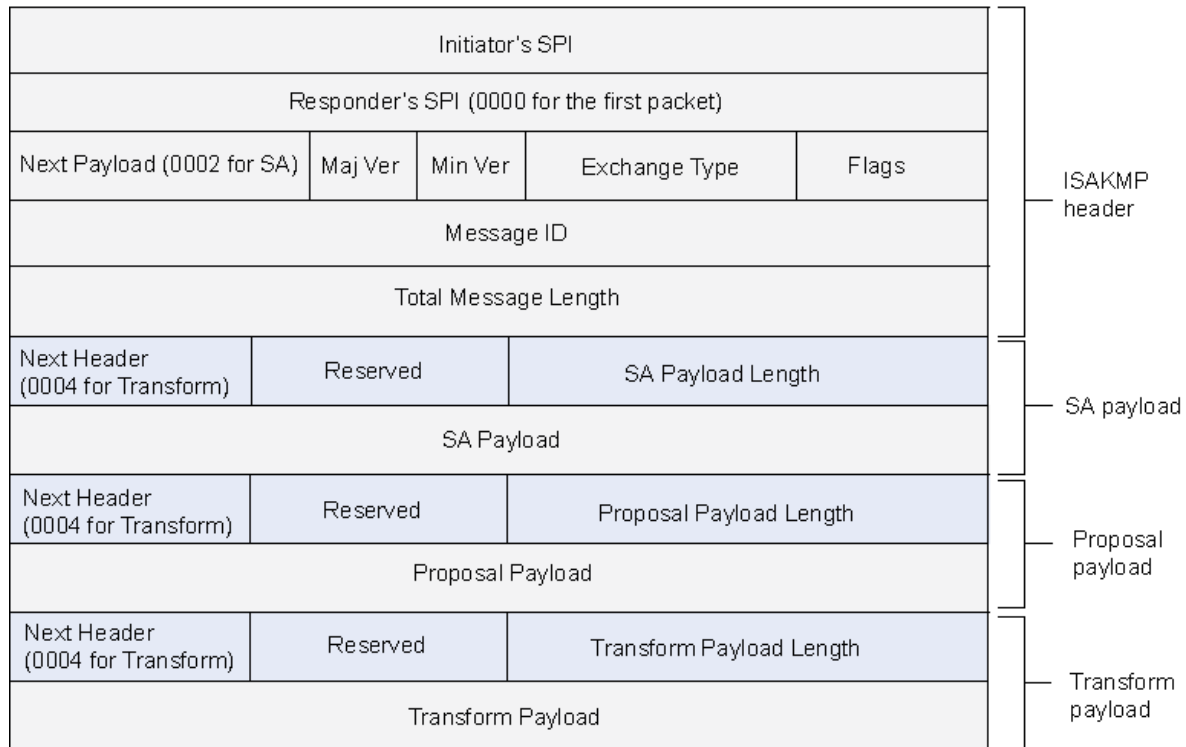
Each ISAKMP payload begins with the same generic header, as shown in Figure 43 on page 463.

**Figure 43: Generic ISAKMP Payload Header**



There can be multiple ISAKMP payloads chained together, with each subsequent payload type indicated by the value in the Next Header field. A value of **0000** indicates the last ISAKMP payload. See Figure 44 on page 464 for an example.

Figure 44: ISAKMP Header with Generic ISAKMP Payloads



g030617

## IPsec Packet Processing

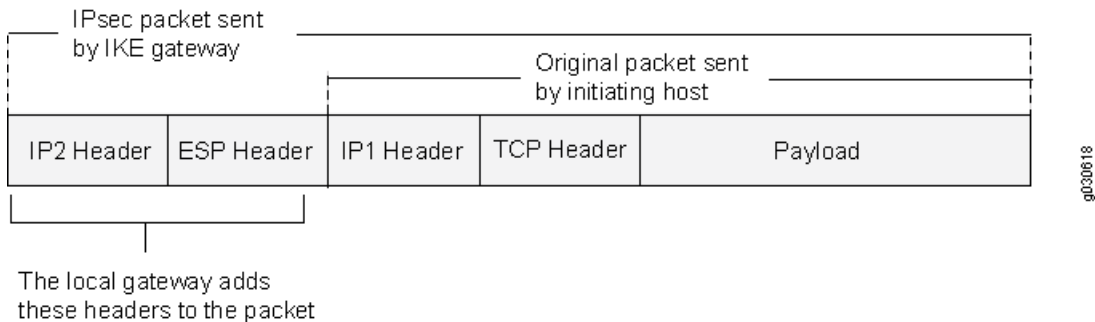
After IKE negotiations complete and the two IKE gateways have established Phase 1 and Phase 2 security associations (SAs), all subsequent packets are forwarded using the tunnel. If the Phase 2 SA specifies the Encapsulating Security Protocol (ESP) in tunnel mode, the packet looks like the one shown in Figure 45 on page 465. The device adds two additional headers to the original packet that the initiating host sends.



**NOTE:** For information about ESP, see “ESP Protocol” on page 457. For information about tunnel mode, see “Packet Processing in Tunnel Mode” on page 459.

As shown in Figure 45 on page 465, the packet that the initiating host constructs includes the payload, the TCP header, and the inner IP header (IP1).

Figure 45: IPsec Packet—ESP in Tunnel Mode



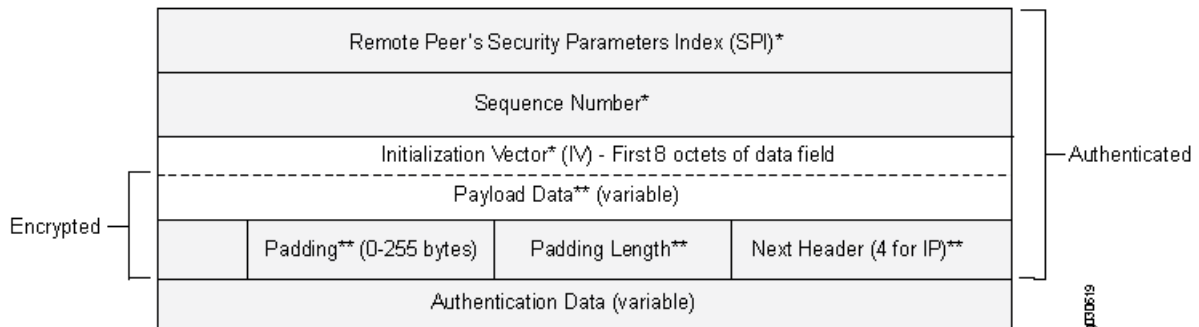
The router IP header (IP2), which Junos OS adds, contains the IP address of the remote gateway as the destination IP address and the IP address of the local router as the source IP address. Junos OS also adds an ESP header between the outer and inner IP headers. The ESP header contains information that allows the remote peer to properly process the packet when it receives it. This is shown in Figure 46 on page 465.

Figure 46: Outer IP Header (IP2) and ESP Header

Outer IP header (IP2)

|                                             |                    |                       |                                |         |   |
|---------------------------------------------|--------------------|-----------------------|--------------------------------|---------|---|
| Version                                     | Header             | Type of Service       | Total Packet Length (in Bytes) |         |   |
| Identification                              |                    |                       | O                              | D       | M |
| Fragment Offset                             | Time to Live (TTL) | Protocol (50 for ESP) | Header Checksum                |         |   |
| Source Address (Local Peer's Gateway)       |                    |                       |                                |         |   |
| Destination Address (Remote Peer's Gateway) |                    |                       |                                |         |   |
| IP Options (if any)                         |                    |                       |                                | Padding |   |
| Payload                                     |                    |                       |                                |         |   |

ESP Header



\* = Authenticated sections of the packet  
 \*\* = Encrypted sections of the packet

The Next Header field indicates the type of data in the payload field. In tunnel mode, this value is 4, indicating an IP packet is contained within the payload. See Figure 47 on page 466.

Figure 47: Inner IP Header (IP1) and TCP Header

Inner IP Header (IP1)

|                                      |                      |                 |                                |   |         |                 |
|--------------------------------------|----------------------|-----------------|--------------------------------|---|---------|-----------------|
| Version                              | Header               | Type of Service | Total Packet Length (in Bytes) |   |         |                 |
| Identification                       |                      |                 | O                              | D | M       | Fragment Offset |
| Time to Live (TTL)                   | Protocol (6 for TCP) |                 | Header Checksum                |   |         |                 |
| Source Address (Installing Host)     |                      |                 |                                |   |         |                 |
| Destination Address (Receiving Host) |                      |                 |                                |   |         |                 |
| IP Options (if any)                  |                      |                 |                                |   | Padding |                 |
| Payload                              |                      |                 |                                |   |         |                 |

TCP Header

|                        |          |             |                  |                                      |                                      |                                                |
|------------------------|----------|-------------|------------------|--------------------------------------|--------------------------------------|------------------------------------------------|
| Source Port            |          |             | Destination Port |                                      |                                      |                                                |
| Sequence Number        |          |             |                  |                                      |                                      |                                                |
| Acknowledgement Number |          |             |                  |                                      |                                      |                                                |
| Header Length          | Reserved | U<br>R<br>G | A<br>C<br>K      | P<br>R<br>S<br>S<br>S<br>S<br>S<br>S | R<br>S<br>S<br>S<br>S<br>S<br>S<br>S | W<br>I<br>N<br>D<br>O<br>W<br>S<br>I<br>Z<br>E |
| Checksum               |          |             | Urgent Pointer   |                                      |                                      |                                                |
| IP Options (if any)    |          |             |                  |                                      | Padding                              |                                                |
| Data                   |          |             |                  |                                      |                                      |                                                |

g030688

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [VPN Overview on page 451](#)
- [Understanding Phase 1 of IKE Tunnel Negotiation on page 467](#)
- [Understanding Phase 2 of IKE Tunnel Negotiation on page 468](#)
- [Understanding Hub-and-Spoke VPNs on page 506](#)
- [Example: Configuring a Policy-Based VPN on page 489](#)
- [Example: Configuring a Route-Based VPN on page 470](#)



## Understanding Phase 1 of IKE Tunnel Negotiation

Phase 1 of an AutoKey Internet Key Exchange (IKE) tunnel negotiation consists of the exchange of proposals for how to authenticate and secure the channel. The participants exchange proposals for acceptable security services such as:

- Encryption algorithms—Data Encryption Standard (DES), triple Data Encryption Standard (3DES), and Advanced Encryption Standard (AES). (See “IPsec Security Protocols” on page 456.)
- Authentication algorithms—Message Digest 5 (MD5) and Secure Hash Algorithm (SHA-1). (See “IPsec Security Protocols” on page 456.)
- Diffie-Hellman (DH) group. (See “Diffie-Hellman Exchange” on page 455.)
- Preshared key or RSA/DSA certificates. (See “IPsec Key Management” on page 454.)

A successful Phase 1 negotiation concludes when both ends of the tunnel agree to accept at least one set of the Phase 1 security parameters proposed and then process them. Juniper Networks devices support up to four proposals for Phase 1 negotiations, allowing you to define how restrictive a range of security parameters for key negotiation you will accept.

Junos OS provides the following predefined Phase 1 proposals:

- Standard—pre-g2-aes128-sha and pre-g2-3des-sha
- Compatible—pre-g2-3des-sha, pre-g2-3des-md5, pre-g2-des-sha, and pre-g2-des-md5
- Basic—pre-g1-des-sha and pre-g1-des-md5

You can also define custom Phase 1 proposals.

Phase 1 exchanges can take place in either main mode or aggressive mode. You can choose your mode during IKE policy configuration.

This topic includes the following sections:

- Main Mode on page 467
- Aggressive Mode on page 468

### Main Mode

In main mode, the initiator and recipient send three two-way exchanges (six messages total) to accomplish the following services:

- First exchange (messages 1 and 2)—Proposes and accepts the encryption and authentication algorithms.
- Second exchange (messages 3 and 4)—Executes a DH exchange, and the initiator and recipient each provide a pseudorandom number.
- Third exchange (messages 5 and 6)—Sends and verifies the identities of the initiator and recipient.

The information transmitted in the third exchange of messages is protected by the encryption algorithm established in the first two exchanges. Thus, the participants' identities are encrypted and therefore not transmitted "in the clear."

## Aggressive Mode

In aggressive mode, the initiator and recipient accomplish the same objectives as with main mode, but in only two exchanges, with a total of three messages:

- First message—The initiator proposes the security association (SA), initiates a DH exchange, and sends a pseudorandom number and its IKE identity.
- Second message—The recipient accepts the SA; authenticates the initiator; and sends a pseudorandom number, its IKE identity, and, if using certificates, the recipient's certificate.
- Third message—The initiator authenticates the recipient, confirms the exchange, and, if using certificates, sends the initiator's certificate.

Because the participants' identities are exchanged in the clear (in the first two messages), aggressive mode does not provide identity protection.



**NOTE:** When a dial-up VPN user negotiates an AutoKey IKE tunnel with a preshared key, aggressive mode must be used. Therefore, you must always use aggressive mode with the dynamic VPN feature. Note also that a dial-up VPN user can use an e-mail address, a fully qualified domain name (FQDN), or an IP address as its IKE ID. A dynamic peer can use either an e-mail address or an FQDN, but not an IP address.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [VPN Overview on page 451](#)
- [Understanding Phase 2 of IKE Tunnel Negotiation on page 468](#)
- [Example: Configuring a Policy-Based VPN on page 489](#)
- [Example: Configuring a Route-Based VPN on page 470](#)

## Understanding Phase 2 of IKE Tunnel Negotiation

After the participants have established a secure and authenticated channel, they proceed through Phase 2, in which they negotiate security associations (SAs) to secure the data to be transmitted through the IPsec tunnel.

Similar to the process for Phase 1, the participants exchange proposals to determine which security parameters to employ in the SA. A Phase 2 proposal also includes a security protocol—either Encapsulating Security Payload (ESP) or Authentication Header (AH)—and selected encryption and authentication algorithms. The proposal can also specify a Diffie-Hellman (DH) group, if Perfect Forward Secrecy (PFS) is desired.

Regardless of the mode used in Phase 1, Phase 2 always operates in quick mode and involves the exchange of three messages.

Juniper Networks devices support up to four proposals for Phase 2 negotiations, allowing you to define how restrictive a range of tunnel parameters you will accept. Junos OS provides the following predefined Phase 2 proposals:

- Standard—g2-esp-3des-sha and g2-esp-aes128-sha
- Compatible—nopfs-esp-3des-sha, nopfs-esp-3des-md5, nopfs-esp-des-sha, and nopfs-esp-des-md5
- Basic—nopfs-esp-des-sha and nopfs-esp-des-md5

You can also define custom Phase 2 proposals.

This topic includes the following sections:

- Proxy IDs on page 469
- Perfect Forward Secrecy on page 469
- Replay Protection on page 469

## Proxy IDs

In Phase 2, the peers exchange proxy IDs. A proxy ID consists of a local and remote IP address prefix. The proxy ID for both peers must match, which means that the local IP address specified for one peer must be the same as the remote IP address specified for the other peer.

## Perfect Forward Secrecy

PFS is a method for deriving Phase 2 keys independent from and unrelated to the preceding keys. Alternatively, the Phase 1 proposal creates the key (the SKEYID\_d key) from which all Phase 2 keys are derived. The SKEYID\_d key can generate Phase 2 keys with a minimum of CPU processing. Unfortunately, if an unauthorized party gains access to the SKEYID\_d key, all your encryption keys are compromised.

PFS addresses this security risk by forcing a new DH key exchange to occur for each Phase 2 tunnel. Using PFS is thus more secure, although the rekeying procedure in Phase 2 might take slightly longer with PFS enabled.

## Replay Protection

A replay attack occurs when an unauthorized person intercepts a series of packets and uses them later either to flood the system, causing a denial of service (DoS), or to gain entry to the trusted network. Junos OS provides a replay protection feature that enables devices to check every IPsec packet to see if it has been received previously. If packets arrive outside a specified sequence range, Junos OS rejects them. Use of this feature does not require negotiation, because packets are always sent with sequence numbers. You simply have the option of checking or not checking the sequence numbers.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- VPN Overview on page 451

- Example: Configuring a Policy-Based VPN on page 489
- Example: Configuring a Route-Based VPN on page 470

## Route-Based VPNs

---

- Understanding Route-Based IPsec VPNs on page 470
- Example: Configuring a Route-Based VPN on page 470

### Understanding Route-Based IPsec VPNs

With route-based VPNs, you can configure dozens of security policies to regulate traffic flowing through a single VPN tunnel between two sites, and there is just one set of IKE and IPsec SAs at work. Unlike policy-based VPNs, for route-based VPNs, a policy refers to a destination address, not a VPN tunnel. When Junos OS looks up a route to find the interface to use to send traffic to the packet's destination address, it finds a route through a secure tunnel interface (st0.x). The tunnel interface is bound to a specific VPN tunnel, and the traffic is routed to the tunnel if the policy action is permit.

Examples of where route-based VPNs can be used:

- There are overlapping subnets or IP addresses between the two LANs.
- A hub-and-spoke VPN topology is used in the network, and spoke-to-spoke traffic is required.
- Primary and backup VPNs are required.
- A dynamic routing protocol (for example, OSPF, RIP, or BGP) is running across the VPN.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- VPN Overview on page 451
- Example: Configuring a Hub-and-Spoke VPN on page 507
- Example: Configuring a Policy-Based VPN on page 489

### Example: Configuring a Route-Based VPN

This example shows how to configure a route-based IPsec VPN to allow data to be securely transferred between a branch office and the corporate office.

- Requirements on page 470
- Overview on page 471
- Configuration on page 475
- Verification on page 484

#### Requirements

---

This example uses the following hardware:

- SRX240 device
- SSG140 device

Before you begin, read “VPN Overview” on page 451.

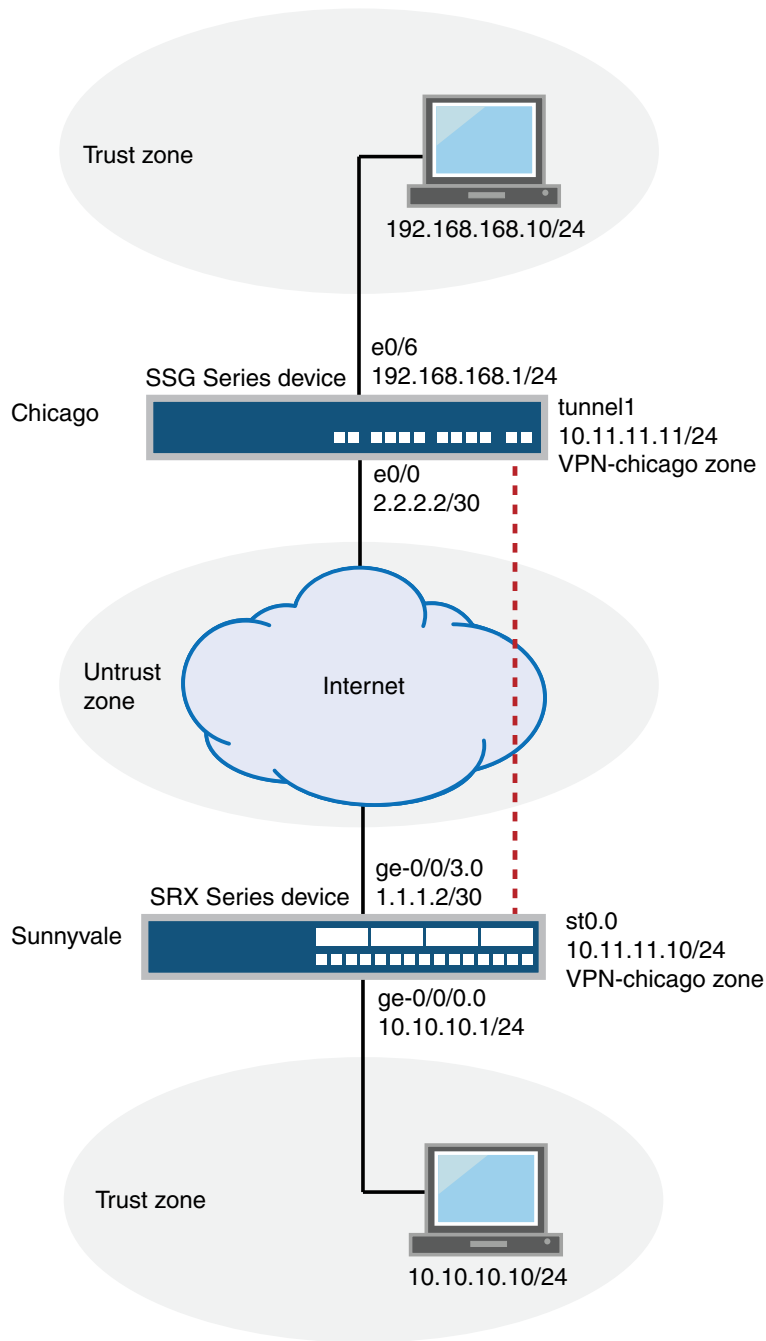
### Overview

---

In this example, you configure a route-based VPN for a branch office in Chicago, Illinois, because you want to conserve tunnel resources but still get granular restrictions on VPN traffic. Users in the Chicago office will use the VPN to connect to their corporate headquarters in Sunnyvale, California.

Figure 48 on page 472 shows an example of a route-based VPN topology. In this topology, the SRX Series device is located in Sunnyvale, and an SSG Series device (or a third-party device) is located in Chicago.

Figure 48: Route-Based VPN Topology



In this example, you configure interfaces, an IPv4 default route, security zones, and address books. Then you configure IKE Phase 1, IPsec Phase 2, security policy, and TCP-MSS parameters. See Table 41 on page 473 through Table 45 on page 474 for specific configuration parameters used in this example.

Table 41: Interface, Static Route, Security Zone, and Address Book Information

| Feature              | Name                      | Configuration Parameters                                                                                                                                                     |
|----------------------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Interfaces           | ge-0/0/0.0                | 10.10.10.1/24                                                                                                                                                                |
|                      | ge-0/0/3.0                | 1.1.1.2/30                                                                                                                                                                   |
|                      | st0.0 (tunnel interface)  | 10.11.11.10/24                                                                                                                                                               |
| Static routes        | 0.0.0.0/0 (default route) | The next hop is 1.1.1.1.                                                                                                                                                     |
|                      | 192.168.168.0/24          | The next hop is st0.0.                                                                                                                                                       |
| Security zones       | trust                     | <ul style="list-style-type: none"> <li>All system services are allowed.</li> <li>The ge-0/0/0.0 interface is bound to this zone.</li> </ul>                                  |
|                      | untrust                   | <ul style="list-style-type: none"> <li>IKE is the only allowed system service.</li> <li>The ge-0/0/3.0 interface is bound to this zone.</li> </ul>                           |
|                      | vpn-chicago               | The st0.0 interface is bound to this zone.                                                                                                                                   |
| Address book entries | sunnyvale                 | <ul style="list-style-type: none"> <li>This address is for the trust zone's address book.</li> <li>The address for this address book entry is 10.10.10.0/24.</li> </ul>      |
|                      | chicago                   | <ul style="list-style-type: none"> <li>This address is for the untrust zone's address book.</li> <li>The address for this address book entry is 192.168.168.0/24.</li> </ul> |

Table 42: IKE Phase 1 Configuration Parameters

| Feature  | Name                | Configuration Parameters                                                                                                                                                                                          |
|----------|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Proposal | ike-phase1-proposal | <ul style="list-style-type: none"> <li>Authentication method: pre-shared-keys</li> <li>Diffie-Hellman group: group2</li> <li>Authentication algorithm: sha1</li> <li>Encryption algorithm: aes-128-cbc</li> </ul> |
| Policy   | ike-phase1-policy   | <ul style="list-style-type: none"> <li>Mode: main</li> <li>Proposal reference: ike-phase1-proposal</li> <li>IKE Phase 1 policy authentication method: pre-shared-key ascii-text</li> </ul>                        |
| Gateway  | gw-chicago          | <ul style="list-style-type: none"> <li>IKE policy reference: ike-phase1-policy</li> <li>External interface: ge-0/0/3.0</li> <li>Gateway address: 2.2.2.2</li> </ul>                                               |

Table 43: IPsec Phase 2 Configuration Parameters

| Feature  | Name                  | Configuration Parameters                                                                                                                                                   |
|----------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Proposal | ipsec-phase2-proposal | <ul style="list-style-type: none"> <li>Protocol: esp</li> <li>Authentication algorithm: hmac-sha1-96</li> <li>Encryption algorithm: aes-128-cbc</li> </ul>                 |
| Policy   | ipsec-phase2-policy   | <ul style="list-style-type: none"> <li>Proposal reference: ipsec-phase2-proposal</li> <li>PFS: Diffie-Hellman group2</li> </ul>                                            |
| VPN      | ike-vpn-chicago       | <ul style="list-style-type: none"> <li>IKE gateway reference: gw-chicago</li> <li>IPsec policy reference: ipsec-phase2-policy</li> <li>Bind to interface: st0.0</li> </ul> |

Table 44: Security Policy Configuration Parameters

| Purpose                                                                          | Name       | Configuration Parameters                                                                                                                                                                                                            |
|----------------------------------------------------------------------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| The security policy permits traffic from the trust zone to the vpn-chicago zone. | vpn-tr-chi | <ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address sunnyvale</li> <li>destination-address chicago</li> <li>application any</li> </ul> </li> <li>Action: permit</li> </ul> |
| The security policy permits traffic from the vpn-chicago zone to the trust zone. | vpn-chi-tr | <ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address chicago</li> <li>destination-address sunnyvale</li> <li>application any</li> </ul> </li> <li>Action: permit</li> </ul> |

Table 45: TCP-MSS Configuration Parameters

| Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Configuration Parameters |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| <p>TCP-MSS is negotiated as part of the TCP three-way handshake and limits the maximum size of a TCP segment to better fit the MTU limits on a network. For VPN traffic, the IPsec encapsulation overhead, along with the IP and frame overhead, can cause the resulting ESP packet to exceed the MTU of the physical interface, which causes fragmentation. Fragmentation increases bandwidth and device resources.</p> <p><b>NOTE:</b> We recommend a value of 1350 as the starting point for most Ethernet-based networks with an MTU of 1500 or greater. You might need to experiment with different TCP-MSS values to obtain optimal performance. For example, you might need to change the value if any device in the path has a lower MTU, or if there is any additional overhead such as PPP or Frame Relay.</p> | MSS value: 1350          |



## Configuration

- Configuring Interface, Static Route, Security Zone, and Address Book Information on page 475
- Configuring IKE on page 478
- Configuring IPsec on page 479
- Configuring Security Policies on page 481
- Configuring TCP-MSS on page 482
- Configuring the SSG Series Device on page 483

### *Configuring Interface, Static Route, Security Zone, and Address Book Information*

**CLI Quick Configuration** To quickly configure interface, static route, security zone, and address book information, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet address 10.10.10.1/24
set interfaces ge-0/0/3 unit 0 family inet address 1.1.1.2/30
set interfaces st0 unit 0 family inet address 10.11.10/24
set routing-options static route 0.0.0.0/0 next-hop 1.1.1.1
set routing-options static route 192.168.168.0/24 next-hop st0.0
set security zones security-zone untrust interfaces ge-0/0/3.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust address-book address sunnyvale 10.10.10.0/24
set security zones security-zone vpn-chicago interfaces st0.0
set security zones security-zone vpn-chicago address-book address chicago
192.168.168.0/24
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure interface, static route, security zone, and address book information:

1. Configure Ethernet interface information.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.10.10.1/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 1.1.1.2/30
user@host# set interfaces st0 unit 0 family inet address 10.11.10/24
```

2. Configure static route information.

```
[edit]
user@host# set routing-options static route 0.0.0.0/0 next-hop 1.1.1.1
user@host# set routing-options static route 192.168.168.0/24 next-hop st0.0
```

3. Configure the untrust security zone.

```
[edit ]
user@host# edit security zones security-zone untrust
```

4. Assign an interface to the security zone.

- ```
[edit security zones security-zone untrust]
user@host# set interfaces ge-0/0/3.0
```
5. Specify allowed system services for the security zone.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services ike
```
  6. Configure the trust security zone.

```
[edit]
user@host# edit security zones security-zone trust
```
  7. Assign an interface to the trust security zone.

```
[edit security zones security-zone trust]
user@host# set interfaces ge-0/0/0.0
```
  8. Specify allowed system services for the trust security zone.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
```
  9. Configure the address book entry for the trust security zone.

```
[edit security zones security-zone trust]
user@host# set address-book address sunnyvale 10.10.10.0/24
```
  10. Configure the vpn-chicago security zone.

```
[edit]
user@host# edit security zones security-zone vpn-chicago
```
  11. Assign an interface to the security zone.

```
[edit security zones security-zone vpn-chicago]
user@host# set interfaces st0.0
```
  12. Configure the address book entry for the vpn-chicago zone.

```
[edit security zones security-zone vpn-chicago]
user@host# set address-book address chicago 192.168.168.0/24
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show routing-options**, and **show security zones** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 10.10.10.1/24;
    }
  }
}
ge-0/0/3 {
  unit 0 {
    family inet {
      address 1.1.1.2/30
```

```

    }
  }
}
st0{
  unit 0 {
    family inet {
      address 10.11.11.10/24
    }
  }
}

[edit]
user@host# show routing-options
static {
  route 0.0.0.0/0 next-hop 1.1.1.1;
  route 192.168.168.0/24 next-hop st0.0;
}

[edit]
user@host# show security zones
security-zone untrust {
  host-inbound-traffic {
    system-services {
      ike;
    }
  }
  interfaces {
    ge-0/0/3.0;
  }
}
security-zone trust {
  address-book {
    address sunnyvale 10.10.10.0/24;
  }
  host-inbound-traffic {
    system-services {
      all;
    }
  }
  interfaces {
    ge-0/0/0.0;
  }
}
security-zone vpn-chicago {
  host-inbound-traffic {
    address-book {
      address chicago 192.168.168.0/24;
    }
  }
  interfaces {
    st0.0;
  }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring IKE

#### CLI Quick Configuration

To quickly configure IKE, copy the following commands and paste them into the CLI:

```
[edit]
set security ike proposal ike-phase1-proposal authentication-method pre-shared-keys
set security ike proposal ike-phase1-proposal dh-group group2
set security ike proposal ike-phase1-proposal authentication-algorithm sha1
set security ike proposal ike-phase1-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-phase1-policy mode main
set security ike policy ike-phase1-policy proposals ike-phase1-proposal
set security ike policy ike-phase1-policy pre-shared-key ascii-text 395psksecr3t
set security ike gateway gw-chicago external-interface ge-0/0/3.0
set security ike gateway gw-chicago ike-policy ike-phase1-policy
set security ike gateway gw-chicago address 2.2.2.2
```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure IKE:

1. Create the IKE Phase 1 proposal.

```
[edit security ike]
user@host# set proposal ike-phase1-proposal
```

2. Define the IKE proposal authentication method.

```
[edit security ike proposal ike-phase1-proposal]
user@host# set authentication-method pre-shared-keys
```

3. Define the IKE proposal Diffie-Hellman group.

```
[edit security ike proposal ike-phase1-proposal]
user@host# set dh-group group2
```

4. Define the IKE proposal authentication algorithm.

```
[edit security ike proposal ike-phase1-proposal]
user@host# set authentication-algorithm sha1
```

5. Define the IKE proposal encryption algorithm.

```
[edit security ike proposal ike-phase1-proposal]
user@host# set encryption-algorithm aes-128-cbc
```

6. Create an IKE Phase 1 policy.

```
[edit security ike]
user@host# set policy ike-phase1-policy
```

7. Set the IKE Phase 1 policy mode.

```
[edit security ike policy ike-phase1-policy]
user@host# set mode main
```

8. Specify a reference to the IKE proposal.

```
[edit security ike policy ike-phase1-policy]
user@host# set proposals ike-phase1-proposal
```

9. Define the IKE Phase 1 policy authentication method.
 

```
[edit security ike policy ike-phase1-policy]
user@host# set pre-shared-key ascii-text 395psksecr3t
```
10. Create an IKE Phase 1 gateway and define its external interface.
 

```
[edit security ike]
user@host# set gateway gw-chicago external-interface ge-0/0/3.0
```
11. Define the IKE Phase 1 policy reference.
 

```
[edit security ike gateway gw-chicago]
user@host# set ike-policy ike-phase1-policy
```
12. Define the IKE Phase 1 gateway address.
 

```
[edit security ike gateway gw-chicago]
user@host# set address 2.2.2.2
```

**Results** From configuration mode, confirm your configuration by entering the **show security ike** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ike
proposal ike-phase1-proposal {
  authentication-method pre-shared-keys;
  dh-group group2;
  authentication-algorithm sha1;
  encryption-algorithm aes-128-cbc;
}
policy ike-phase1-policy {
  mode main;
  proposals ike-phase1-proposal;
  pre-shared-key ascii-text "$9$9VMTp1RvWLdwYKMJDkmF3ylKM87Vb2oZjws5F"; ##
  SECRET-DATA
}
gateway gw-chicago {
  ike-policy ike-phase1-policy;
  address 2.2.2.2;
  external-interface ge-0/0/3.0;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring IPsec

**CLI Quick Configuration** To quickly configure IPsec, copy the following commands and paste them into the CLI:

```
[edit]
set security ipsec proposal ipsec-phase2-proposal protocol esp
set security ipsec proposal ipsec-phase2-proposal authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec-phase2-proposal encryption-algorithm aes-128-cbc
set security ipsec policy ipsec-phase2-policy proposals ipsec-phase2-proposal
set security ipsec policy ipsec-phase2-policy perfect-forward-secrecy keys group2
set security ipsec vpn ike-vpn-chicago ike gateway gw-chicago
set security ipsec vpn ike-vpn-chicago ike ipsec-policy ipsec-phase2-policy
```

```
set security ipsec vpn ike-vpn-chicago bind-interface st0.0
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure IPsec:

1. Create an IPsec Phase 2 proposal.  
[edit]  
user@host# set security ipsec proposal ipsec-phase2-proposal
2. Specify the IPsec Phase 2 proposal protocol.  
[edit security ipsec proposal ipsec-phase2-proposal]  
user@host# set protocol esp
3. Specify the IPsec Phase 2 proposal authentication algorithm.  
[edit security ipsec proposal ipsec-phase2-proposal]  
user@host# set authentication-algorithm hmac-sha1-96
4. Specify the IPsec Phase 2 proposal encryption algorithm.  
[edit security ipsec proposal ipsec-phase2-proposal]  
user@host# set encryption-algorithm aes-128-cbc
5. Create the IPsec Phase 2 policy.  
[edit security ipsec]  
user@host# set policy ipsec-phase2-policy
6. Specify the IPsec Phase 2 proposal reference.  
[edit security ipsec policy ipsec-phase2-policy]  
user@host# set proposals ipsec-phase2-proposal
7. Specify IPsec Phase 2 PFS to use Diffie-Hellman group 2.  
[edit security ipsec policy ipsec-phase2-policy]  
user@host# set perfect-forward-secrecy keys group2
8. Specify the IKE gateway.  
[edit security ipsec]  
user@host# set vpn ike-vpn-chicago ike gateway gw-chicago
9. Specify the IPsec Phase 2 policy.  
[edit security ipsec]  
user@host# set vpn ike-vpn-chicago ike ipsec-policy ipsec-phase2-policy
10. Specify the interface to bind.  
[edit security ipsec]  
user@host# set vpn ike-vpn-chicago bind-interface st0.0

**Results** From configuration mode, confirm your configuration by entering the `show security ipsec` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ipsec
proposal ipsec-phase2-proposal {
  protocol esp;
  authentication-algorithm hmac-sha1-96;
  encryption-algorithm aes-128-cbc;
}
policy ipsec-phase2-policy {
  perfect-forward-secrecy {
    keys group2;
  }
  proposals ipsec-phase2-proposal;
}
vpn ike-vpn-chicago {
  bind-interface st0.0;
  ike {
    gateway gw-chicago;
    ipsec-policy ipsec-phase2-policy;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring Security Policies

#### CLI Quick Configuration

To quickly configure security policies, copy the following commands and paste them into the CLI:

```
[edit]
set security policies from-zone trust to-zone vpn-chicago policy vpn-tr-chi match
  source-address sunnyvale
set security policies from-zone trust to-zone vpn-chicago policy vpn-tr-chi match
  destination-address chicago
set security policies from-zone trust to-zone vpn-chicago policy vpn-tr-chi match
  application any
set security policies from-zone trust to-zone vpn-chicago policy vpn-tr-chi then permit
set security policies from-zone vpn-chicago to-zone trust policy vpn-chi-tr match
  source-address chicago
set security policies from-zone vpn-chicago to-zone trust policy vpn-chi-tr match
  destination-address sunnyvale
set security policies from-zone vpn-chicago to-zone trust policy vpn-chi-tr match
  application any
set security policies from-zone vpn-chicago to-zone trust policy vpn-chi-tr then permit
```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure security policies:

1. Create the security policy to permit traffic from the trust zone to the vpn-chicago zone.

```
[edit security policies from-zone trust to-zone vpn-chicago]
user@host# set policy vpn-tr-chi match source-address sunnyvale
user@host# set policy vpn-tr-chi match destination-address chicago
```

```

user@host# set policy vpn-tr-chi match application any
user@host# set policy vpn-tr-chi then permit

```

2. Create the security policy to permit traffic from the vpn-chicago zone to the trust zone.

```

[edit security policies from-zone vpn-chicago to-zone trust]
user@host# set policy vpn-chi-tr match source-address sunnyvale
user@host# set policy vpn-chi-tr match destination-address chicago
user@host# set policy vpn-chi-tr match application any
user@host# set policy vpn-chi-tr then permit

```

**Results** From configuration mode, confirm your configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show security policies
from-zone trust to-zone vpn-chicago {
  policy vpn-tr-vpn {
    match {
      source-address sunnyvale;
      destination-address chicago;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone vpn-chicago to-zone trust {
  policy vpn-tr-vpn {
    match {
      source-address chicago;
      destination-address sunnyvale;
      application any;
    }
    then {
      permit;
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring TCP-MSS

**CLI Quick Configuration** To quickly configure TCP-MSS information, copy the following commands and paste them into the CLI:

```

[edit]
set security flow tcp-mss ipsec-vpn mss 1350

```



**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure TCP-MSS information:

1. Configure TCP-MSS information.

```
[edit]
user@host# set security flow tcp-mss ipsec-vpn mss 1350
```

**Results** From configuration mode, confirm your configuration by entering the **show security flow** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security flow
tcp-mss {
  ipsec-vpn {
    mss 1350;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

#### *Configuring the SSG Series Device*

**CLI Quick Configuration** For reference, the configuration for the SSG Series device is provided. For information about configuring SSG Series devices, see the *Concepts and Examples ScreenOS Reference Guide*, which is located at <http://www.juniper.net/techpubs>.

To quickly configure the SSG Series device, copy the following commands and paste them into the CLI:

```
set zone name vpn-chicago
set interface ethernet0/6 zone Trust
set interface ethernet0/0 zone Untrust
set interface tunnel.1 zone vpn-chicago
set interface ethernet0/6 ip 192.168.168.1/24
set interface ethernet0/6 route
set interface ethernet0/0 ip 2.2.2.2/30
set interface ethernet0/0 route
set interface tunnel.1 ip 10.11.11.1/24
set flow tcp-mss 1350
set address Trust "192.168.168-net" 192.168.168.0 255.255.255.0
set address vpn-chicago "10.10.10-net" 10.10.10.0 255.255.255.0
set ike gateway corp-ike address 1.1.1.2 Main outgoing-interface ethernet0/0 preshare
  395psksecr3t sec-level standard
set vpn corp-vpn gateway corp-ike replay tunnel idletime 0 sec-level standard
set vpn corp-vpn monitor optimized rekey
set vpn corp-vpn bind interface tunnel.1
set policy from Trust to Untrust "ANY" "ANY" "ANY" nat src permit
set policy from Trust to vpn-chicago "192.168.168-net" "10.10.10-net" "ANY" permit
set policy from vpn-chicago to Trust "10.10.10-net" "192.168.168-net" "ANY" permit
set route 10.10.10.0/24 interface tunnel.1
```

```
set route 0.0.0.0/0 interface ethernet0/0 gateway 2.2.2.1
```

## Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying the IKE Phase 1 Status on page 484
- Verifying the IPsec Phase 2 Status on page 485
- Reviewing Statistics and Errors for an IPsec Security Association on page 487
- Testing Traffic Flow Across the VPN on page 487

### Verifying the IKE Phase 1 Status

**Purpose** Verify the IKE Phase 1 status.

**Action**



**NOTE:** Before starting the verification process, you need to send traffic from a host in the 10.10.10/24 network to a host in the 192.168.168/24 network. For route-based VPNs, traffic can be initiated by the SRX Series device through the tunnel. We recommend that when testing IPsec tunnels, test traffic be sent from a separate device on one side of the VPN to a second device on the other side of the VPN. For example, initiate a ping from 10.10.10.10 to 192.168.168.10.

From operational mode, enter the **show security ike security-associations** command. After obtaining an index number from the command, use the **show security ike security-associations index *index\_number* detail** command.

```
user@host> show security ike security-associations
Index  Remote Address  State  Initiator cookie  Responder cookie  Mode
1      2.2.2.2          UP     744a594d957dd513  1e1307db82f58387  Main
```

```
user@host> show security ike security-associations index 1 detail
IKE peer 2.2.2.2, Index 1,
  Role: Responder, State: UP
  Initiator cookie: 744a594d957dd513, Responder cookie: 1e1307db82f58387
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 1.1.1.2:500, Remote: 2.2.2.2:500
  Lifetime: Expires in 28570 seconds
  Algorithms:
    Authentication      : sha1
    Encryption          : aes-cbc (128 bits)
    Pseudo random function: hmac-sha1
  Traffic statistics:
    Input bytes       :           852
    Output bytes      :           940
    Input packets     :             5
    Output packets    :             5
  Flags: Caller notification sent
  IPSec security associations: 1 created, 0 deleted
  Phase 2 negotiations in progress: 0
```

**Meaning** The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the `show security ike security-associations index detail` command to get more information about the SA.
- Remote Address—Verify that the remote IP address is correct.
- State
  - UP—The Phase 1 SA has been established.
  - DOWN—There was a problem establishing the Phase 1 SA.
- Mode—Verify that the correct mode is being used.

Verify that the following are correct in your configuration:

- External interfaces (the interface must be the one that receives IKE packets)
- IKE policy parameters
- Preshared key information
- Phase 1 proposal parameters (must match on both peers)

The `show security ike security-associations index 1 detail` command lists additional information about the security association with an index number of 1:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Role information



**NOTE:** Troubleshooting is best performed on the peer using the responder role.

- Initiator and responder information
- Number of IPsec SAs created
- Number of Phase 2 negotiations in progress

#### *Verifying the IPsec Phase 2 Status*

**Purpose** Verify the IPsec Phase 2 status.

**Action** From operational mode, enter the **show security ipsec security-associations** command. After obtaining an index number from the command, use the **show security ipsec security-associations index *index\_number* detail** command.

```
user@host> show security ipsec security-associations
total configured sa: 2
ID      Gateway          Port  Algorithm          SPI      Life:sec/kb  Mon vsys
-----
<16384  2.2.2.2          500   ESP:aes-128/sha1   76d64d1d 3363/ unlim  -   0
>16384  2.2.2.2          500   ESP:aes-128/sha1   a1024ee2 3363/ unlim  -   0
```

```
user@host> show security ipsec security-associations index 16384 detail
```

```
Virtual-system: Root
Local Gateway: 1.1.1.2, Remote Gateway: 2.2.2.2
Local Identity: ipv4_subnet(any:0,[0..7]=10.10.10.0/24)
Remote Identity: ipv4_subnet(any:0,[0..7]=192.168.168.0/24)
DF-bit: clear

Direction: inbound, SPI: 1993755933, AUX-SPI: 0
Hard lifetime: Expires in 3352 seconds
Lifesize Remaining: Unlimited
Soft lifetime: Expires in 2775 seconds
Mode: tunnel, Type: dynamic, State: installed, VPN Monitoring: -
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
```

```
Anti-replay service: enabled, Replay window size: 32
```

```
Direction: outbound, SPI: 2701283042, AUX-SPI: 0
Hard lifetime: Expires in 3352 seconds
Lifesize Remaining: Unlimited
Soft lifetime: Expires in 2775 seconds
Mode: tunnel, Type: dynamic, State: installed, VPN Monitoring: -
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc
(128 bits)
Anti-replay service: enabled, Replay window size: 32
```

**Meaning** The output from the **show security ipsec security-associations** command lists the following information:

- The ID number is 16384. Use this value with the **show security ipsec security-associations index** command to get more information about this particular SA.
- There is one IPsec SA pair using port 500, which indicates that no NAT-traversal is implemented. (NAT-traversal uses port 4500 or another random high-number port.)
- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 3363/ unlim value indicates that the Phase 2 lifetime expires in 3363 seconds, and that no lifesize has been specified, which indicates that it is unlimited. Phase 2 lifetime can differ from Phase 1 lifetime, as Phase 2 is not dependent on Phase 1 after the VPN is up.
- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U indicates that monitoring is up, and D indicates that monitoring is down.
- The virtual system (vsys) is the root system, and it always lists 0.

The output from the `show security ipsec security-associations index 16384 detail` command lists the following information:

- The local identity and remote identity make up the proxy ID for the SA.  
A proxy ID mismatch is one of the most common causes for a Phase 2 failure. If no IPsec SA is listed, confirm that Phase 2 proposals, including the proxy ID settings, are correct for both peers. For route-based VPNs, the default proxy ID is local=0.0.0.0/0, remote=0.0.0.0/0, and service=any. Issues can occur with multiple route-based VPNs from the same peer IP. In this case, a unique proxy ID for each IPsec SA must be specified. For some third-party vendors, the proxy ID must be manually entered to match.
- Another common reason for Phase 2 failure is not specifying the ST interface binding. If IPsec cannot complete, check the `kmd log` or `set traceoptions`.

### *Reviewing Statistics and Errors for an IPsec Security Association*

**Purpose** Review ESP and authentication header counters and errors for an IPsec security association.

**Action** From operational mode, enter the `show security ipsec statistics index index_number` command, using the index number of the VPN for which you want to see statistics.

```
user@host> show security ipsec statistics index 16384
ESP Statistics:
  Encrypted bytes:          920
  Decrypted bytes:         6208
  Encrypted packets:        5
  Decrypted packets:       87
AH Statistics:
  Input bytes:              0
  Output bytes:             0
  Input packets:           0
  Output packets:          0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

You can also use the `show security ipsec statistics` command to review statistics and errors for all SAs.

To clear all IPsec statistics, use the `clear security ipsec statistics` command.

**Meaning** If you see packet loss issues across a VPN, you can run the `show security ipsec statistics` or `show security ipsec statistics detail` command several times to confirm that the encrypted and decrypted packet counters are incrementing. You should also check whether the other error counters are incrementing.

### *Testing Traffic Flow Across the VPN*

**Purpose** Verify the traffic flow across the VPN.

**Action** You can use the **ping** command from the SRX Series device to test traffic flow to a remote host PC. Make sure that you specify the source interface so that the route lookup is correct and the appropriate security zones are referenced during policy lookup.

From operational mode, enter the **ping** command.

```

ssg-> ping 192.168.168.10 interface ge-0/0/0 count 5
PING 192.168.168.10 (192.168.168.10): 56 data bytes
64 bytes from 192.168.168.10: icmp_seq=0 ttl=127 time=8.287 ms
64 bytes from 192.168.168.10: icmp_seq=1 ttl=127 time=4.119 ms
64 bytes from 192.168.168.10: icmp_seq=2 ttl=127 time=5.399 ms
64 bytes from 192.168.168.10: icmp_seq=3 ttl=127 time=4.361 ms
64 bytes from 192.168.168.10: icmp_seq=4 ttl=127 time=5.137 ms

--- 192.168.168.10 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.119/5.461/8.287/1.490 ms

```

You can also use the **ping** command from the SSG Series device.

```

user@host> ping 10.10.10.10 from ethernet0/6
Type escape sequence to abort
Sending 5, 100-byte ICMP Echos to 10.10.10.10, timeout is 1 seconds from
ethernet0/6
!!!!
Success Rate is 100 percent (5/5), round-trip time min/avg/max=4/4/5 ms

```

**Meaning** If the **ping** command fails from the SRX Series or SSG Series device, there might be a problem with the routing, security policies, end host, or encryption and decryption of ESP packets.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [VPN Overview on page 451](#)
  - [Example: Configuring a Hub-and-Spoke VPN on page 507](#)
  - [Example: Configuring a Policy-Based VPN on page 489](#)

## Policy-Based VPNs

- [Understanding Policy-Based IPsec VPNs on page 488](#)
- [Example: Configuring a Policy-Based VPN on page 489](#)

### Understanding Policy-Based IPsec VPNs

For policy-based IPsec VPNs, a security policy specifies as its action the VPN tunnel to be used for transit traffic that meets the policy's match criteria. A VPN is configured independent of a policy statement. The policy statement refers to the VPN by name to specify the traffic that is allowed access to the tunnel. For policy-based VPNs, each policy creates an individual IPsec security association (SA) with the remote peer, each of which counts as an individual VPN tunnel. For example, if a policy contains a group source address and a group destination address, whenever one of the users belonging to the address set attempts to communicate with any one of the hosts specified as the destination address, a new tunnel is negotiated and established. Because each tunnel

requires its own negotiation process and separate pair of SAs, the use of policy-based IPsec VPNs can be more resource-intensive than route-based VPNs.

Examples of where policy-based VPNs can be used:

- You are implementing a dial-up VPN.
- You require more granularity than a route can provide when determining which traffic is sent to a tunnel (for example, you need to specify that traffic to a certain destination goes through the tunnel only if the traffic originated from a particular source).
- The remote VPN device is a non-Juniper device that requires separate SAs for each remote subnet.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [VPN Overview on page 451](#)
- [Example: Configuring a Route-Based VPN on page 470](#)
- [Example: Configuring a Hub-and-Spoke VPN on page 507](#)
- [Example: Configuring a Policy-Based VPN on page 489](#)

### Example: Configuring a Policy-Based VPN

This example shows how to configure a policy-based IPsec VPN to allow data to be securely transferred between a branch office and the corporate office.

- [Requirements on page 489](#)
- [Overview on page 489](#)
- [Configuration on page 493](#)
- [Verification on page 502](#)

#### Requirements

This example uses the following hardware:

- SRX240 device
- SSG140 device

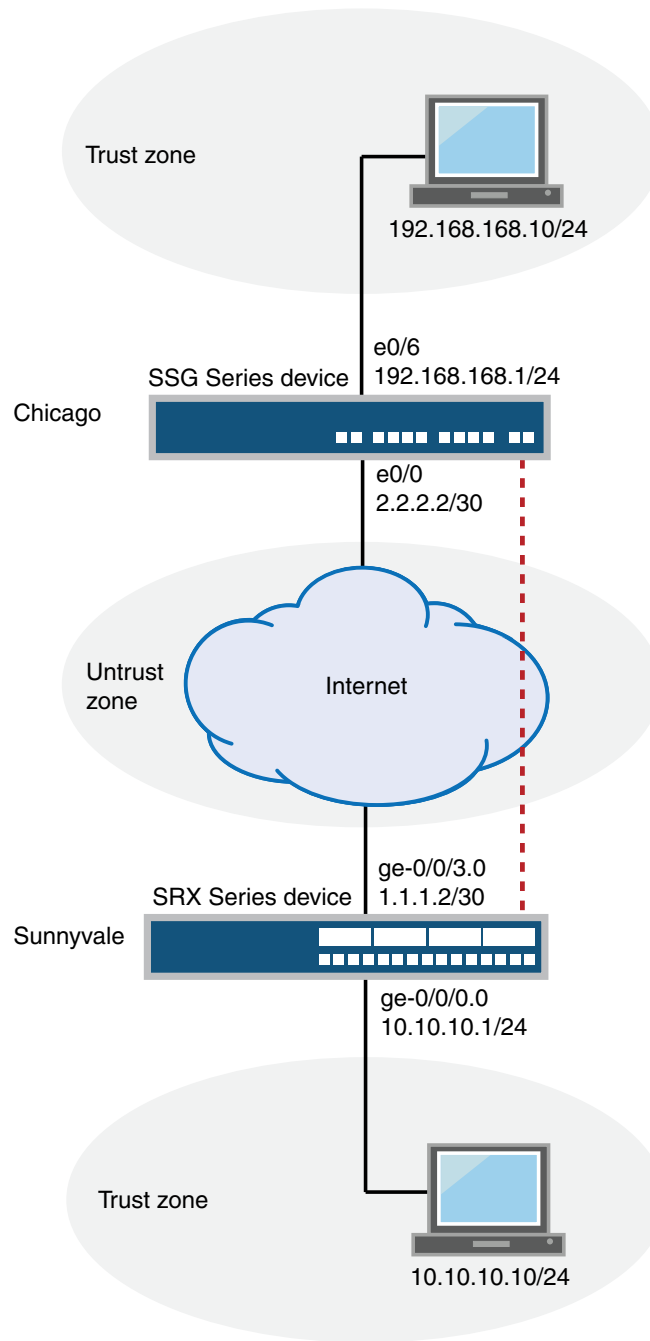
Before you begin, read “VPN Overview” on page 451.

#### Overview

In this example, you configure a policy-based VPN for a branch office in Chicago, Illinois, because you do not need to conserve tunnel resources or configure many security policies to filter traffic through the tunnel. Users in the Chicago office will use the VPN to connect to their corporate headquarters in Sunnyvale, California.

Figure 49 on page 490 shows an example of a policy-based VPN topology. In this topology, the SRX Series device is located in Sunnyvale, and an SSG Series device (or it can be another third-party device) is located in Chicago.

Figure 49: Policy-Based VPN Topology



IKE IPsec tunnel negotiation occurs in two phases. In Phase 1, participants establish a secure channel in which to negotiate the IPsec security association (SA). In Phase 2, participants negotiate the IPsec SA for authenticating traffic that will flow through the tunnel. Just as there are two phases to tunnel negotiation, there are two phases to tunnel configuration.



In this example, you configure interfaces, an IPv4 default route, security zones, and address books. Then you configure IKE Phase 1, IPsec Phase 2, security policy, and TCP-MSS parameters. See Table 46 on page 491 through Table 50 on page 493.

**Table 46: Interface, Security Zone, and Address Book Information**

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/0.0	10.10.10.1/24
	ge-0/0/3.0	1.1.1.2/30
Security zones	trust	<ul style="list-style-type: none"> <li>All system services are allowed.</li> <li>The ge-0/0/0.0 interface is bound to this zone.</li> </ul>
	untrust	<ul style="list-style-type: none"> <li>IKE is the only allowed system service.</li> <li>The ge-0/0/3.0 interface is bound to this zone.</li> </ul>
Address book entries	sunnyvale	<ul style="list-style-type: none"> <li>This address is for the trust zone's address book.</li> <li>The address for this address book entry is 10.10.10.0/24.</li> </ul>
	chicago	<ul style="list-style-type: none"> <li>This address is for the untrust zone's address book.</li> <li>The address for this address book entry is 192.168.168.0/24.</li> </ul>

**Table 47: IKE Phase 1 Configuration Parameters**

Feature	Name	Configuration Parameters
Proposal	ike-phase1-proposal	<ul style="list-style-type: none"> <li>Authentication method: pre-shared-keys</li> <li>Diffie-Hellman group: group2</li> <li>Authentication algorithm: sha1</li> <li>Encryption algorithm: aes-128-cbc</li> </ul>
Policy	ike-phase1-policy	<ul style="list-style-type: none"> <li>Mode: main</li> <li>Proposal reference: ike-phase1-proposal</li> <li>IKE Phase 1 policy authentication method: pre-shared-key ascii-text</li> </ul>
Gateway	gw-chicago	<ul style="list-style-type: none"> <li>IKE policy reference: ike-phase1-policy</li> <li>External interface: ge-0/0/3.0</li> <li>Gateway address: 2.2.2.2</li> </ul>

Table 48: IPsec Phase 2 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ipsec-phase2-proposal	<ul style="list-style-type: none"> <li>Protocol: esp</li> <li>Authentication algorithm: hmac-sha1-96</li> <li>Encryption algorithm: aes-128-cbc</li> </ul>
Policy	ipsec-phase2-policy	<ul style="list-style-type: none"> <li>Proposal reference: ipsec-phase2-proposal</li> <li>PFS: Diffie-Hellman group2</li> </ul>
VPN	ike-vpn-chicago	<ul style="list-style-type: none"> <li>IKE gateway reference: gw-chicago</li> <li>IPsec policy reference: ipsec-phase2-policy</li> </ul>

Table 49: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters
This security policy permits traffic from the trust zone to the untrust zone.	vpn-tr-untr	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address sunnyvale</li> <li>destination-address chicago</li> <li>application any</li> </ul> </li> <li>Permit action: tunnel ipsec-vpn ike-vpn-chicago</li> <li>Permit action: tunnel pair-policy vpn-untr-tr</li> </ul>
This security policy permits traffic from the untrust zone to the trust zone.	vpn-untr-tr	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address chicago</li> <li>destination-address sunnyvale</li> <li>application any</li> </ul> </li> <li>Permit action: tunnel ipsec-vpn ike-vpn-chicago</li> <li>Permit action: tunnel pair-policy vpn-tr-untr</li> </ul>
<p>This security policy permits all traffic from the trust zone to the untrust zone.</p> <p><b>NOTE:</b> You must put the vpn-tr-untr policy before the permit-any security policy. Junos OS performs a security policy lookup starting at the top of the list. If the permit-any policy comes before the vpn-tr-untr policy, all traffic from the trust zone will match the permit-any policy and be permitted. Thus, no traffic will ever match the vpn-tr-untr policy.</p>	permit-any	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address any</li> <li>source-destination any</li> <li>application any</li> </ul> </li> <li>Action: permit</li> </ul>

Table 50: TCP-MSS Configuration Parameters

Purpose	Configuration Parameters
<p>TCP-MSS is negotiated as part of the TCP three-way handshake and limits the maximum size of a TCP segment to better fit the maximum transmission unit (MTU) limits on a network. This is especially important for VPN traffic, as the IPsec encapsulation overhead, along with the IP and frame overhead, can cause the resulting Encapsulating Security Payload (ESP) packet to exceed the MTU of the physical interface, thus causing fragmentation. Fragmentation results in increased use of bandwidth and device resources.</p>	MSS value: 1350
<p><b>NOTE:</b> We recommend a value of 1350 as the starting point for most Ethernet-based networks with an MTU of 1500 or greater. You might need to experiment with different TCP-MSS values to obtain optimal performance. For example, you might need to change the value if any device in the path has a lower MTU, or if there is any additional overhead such as PPP or Frame Relay.</p>	

### Configuration

#### *Configuring Basic Network, Security Zone, and Address Book Information*

**CLI Quick Configuration** To quickly configure basic network, security zone, and address book information, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet address 10.10.10.1/24
set interfaces ge-0/0/3 unit 0 family inet address 1.1.1.2/30
set routing-options static route 0.0.0.0/0 next-hop 1.1.1.1
set security zones security-zone untrust interfaces ge-0/0/3.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone trust address-book address chicago 192.168.168.0/24
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust address-book address sunnyvale 10.10.10.0/24
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure basic network, security zone, and address book information:

1. Configure Ethernet interface information.
 

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.10.10.1/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 1.1.1.2/30
```
2. Configure static route information.
 

```
[edit]
user@host# set routing-options static route 0.0.0.0/0 next-hop 1.1.1.1
```
3. Configure the untrust security zone.
 

```
[edit ]
user@host# edit security zones security-zone untrust
```
4. Assign an interface to the security zone.

- ```
[edit security zones security-zone untrust]
user@host# set interfaces ge-0/0/3.0
```
5. Specify allowed system services for the security zone.

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services ike
```
  6. Configure the address book entry for the untrust zone.

```
[edit security zones security-zone untrust]
user@host# set address-book address chicago 192.168.168.0/24
```
  7. Configure the trust security zone.

```
[edit]
user@host# edit security zones security-zone trust
```
  8. Assign an interface to the security zone.

```
[edit security zones security-zone trust]
user@host# set interfaces ge-0/0/0.0
```
  9. Specify allowed system services for the security zone.

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
```
  10. Configure the address book entry for the trust zone.

```
[edit security zones security-zone trust]
user@host# set address-book address sunnyvale 10.10.10.0/24
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show routing-options**, and **show security zones** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 10.10.10.1/24;
    }
  }
}
ge-0/0/3 {
  unit 0 {
    family inet {
      address 1.1.1.2/30
    }
  }
}
[edit]
user@host# show routing-options
static {
  route 0.0.0.0/0 next-hop 1.1.1.1;
}
```

```
[edit]
user@host# show security zones
security-zone untrust {
  address-book {
    address chicago 192.168.168.0/24{
    }
  }
  host-inbound-traffic {
    system-services {
      ike;
    }
  }
  interfaces {
    ge-0/0/3.0;
  }
}
security-zone trust {
  address-book {
    address sunnyvale 10.10.10.0/24{
    }
  }
  host-inbound-traffic {
    system-services {
      all;
    }
  }
  interfaces {
    ge-0/0/0.0;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring IKE

#### CLI Quick Configuration

To quickly configure IKE, copy the following commands and paste them into the CLI:

```
[edit]
set security ike proposal ike-phase1-proposal authentication-method pre-shared-keys
set security ike proposal ike-phase1-proposal dh-group group2
set security ike proposal ike-phase1-proposal authentication-algorithm sha1
set security ike proposal ike-phase1-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-phase1-policy mode main
set security ike policy ike-phase1-policy proposals ike-phase1-proposal
set security ike policy ike-phase1-policy pre-shared-key ascii-text 395psksecr3t
set security ike gateway gw-chicago external-interface ge-0/0/3.0
set security ike gateway gw-chicago ike-policy ike-phase1-policy
set security ike gateway gw-chicago address 2.2.2.2
```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure IKE:

1. Create the IKE Phase 1 proposal.

```
[edit security ike]
```

- ```
user@host# set proposal ike-phase1-proposal
```
2. Define the IKE proposal authentication method.  

```
[edit security ike proposal ike-phase1-proposal]  
user@host# set authentication-method pre-shared-keys
```
3. Define the IKE proposal Diffie-Hellman group.  

```
[edit security ike proposal ike-phase1-proposal]  
user@host# set dh-group group2
```
4. Define the IKE proposal authentication algorithm.  

```
[edit security ike proposal ike-phase1-proposal]  
user@host# set authentication-algorithm sha1
```
5. Define the IKE proposal encryption algorithm.  

```
[edit security ike proposal ike-phase1-proposal]  
user@host# set encryption-algorithm aes-128-cbc
```
6. Create an IKE Phase 1 policy.  

```
[edit security ike]  
user@host# set policy ike-phase1-policy
```
7. Set the IKE Phase 1 policy mode.  

```
[edit security ike policy ike-phase1-policy]  
user@host# set mode main
```
8. Specify a reference to the IKE proposal.  

```
[edit security ike policy ike-phase1-policy]  
user@host# set proposals ike-phase1-proposal
```
9. Define the IKE Phase 1 policy authentication method.  

```
[edit security ike policy ike-phase1-policy]  
user@host# set pre-shared-key ascii-text 395psksecr3t
```
10. Create an IKE Phase 1 gateway and define its external interface.  

```
[edit security ike]  
user@host# set gateway gw-chicago external-interface ge-0/0/3.0
```
11. Define the IKE Phase 1 policy reference.  

```
[edit security ike gateway gw-chicago]  
user@host# set ike-policy ike-phase1-policy
```
12. Create an IKE Phase 1 gateway and define its external interface.  

```
[edit security ike gateway gw-chicago]  
user@host# set gateway gw-chicago external-interface ge-0/0/3.0
```
13. Define the IKE Phase 1 policy reference.  

```
[edit security ike gateway gw-chicago]  
user@host# set ike-policy ike-phase1-policy
```

**Results** From configuration mode, confirm your configuration by entering the **show security ike** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ike
proposal ike-phase1-proposal {
  authentication-method pre-shared-keys;
  dh-group group2;
  authentication-algorithm sha1;
  encryption-algorithm aes-128-cbc;
}
policy ike-phase1-policy {
  mode main;
  proposals ike-phase1-proposal;
  pre-shared-key ascii-text "$9$9VMTp1RvWLdwYKMJDkmF3ylKM87Vb2oZjws5F"; ##
  SECRET-DATA
}
gateway gw-chicago {
  ike-policy ike-phase1-policy;
  address 2.2.2.2;
  external-interface ge-0/0/3.0;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring IPsec

**CLI Quick Configuration** To quickly configure IPsec, copy the following commands and paste them into the CLI:

```
[edit]
set security ipsec proposal ipsec-phase2-proposal protocol esp
set security ipsec proposal ipsec-phase2-proposal authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec-phase2-proposal encryption-algorithm aes-128-cbc
set security ipsec policy ipsec-phase2-policy proposals ipsec-phase2-proposal
set security ipsec policy ipsec-phase2-policy perfect-forward-secrecy keys group2
set security ipsec vpn ike-vpn-chicago ike gateway gw-chicago
set security ipsec vpn ike-vpn-chicago ike ipsec-policy ipsec-phase2-policy
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure IPsec:

1. Create an IPsec Phase 2 proposal.

```
[edit]
user@host# set security ipsec proposal ipsec-phase2-proposal
```

2. Specify the IPsec Phase 2 proposal protocol.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@host# set protocol esp
```

3. Specify the IPsec Phase 2 proposal authentication algorithm.

- ```
[edit security ipsec proposal ipsec-phase2-proposal]
user@host# set authentication-algorithm hmac-sha1-96
```
4. Specify the IPsec Phase 2 proposal encryption algorithm.
 

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@host# set encryption-algorithm aes-128-cbc
```
  5. Create the IPsec Phase 2 policy.
 

```
[edit security ipsec]
user@host# set policy ipsec-phase2-policy
```
  6. Specify the IPsec Phase 2 proposal reference.
 

```
[edit security ipsec policy ipsec-phase2-policy]
user@host# set proposals ipsec-phase2-proposal
```
  7. Specify IPsec Phase 2 PFS to use Diffie-Hellman group 2.
 

```
[edit security ipsec policy ipsec-phase2-policy]
user@host# set perfect-forward-secrecy keys group2
```
  8. Specify the IKE gateway.
 

```
[edit security ipsec]
user@host# set vpn ike-vpn-chicago ike gateway gw-chicago
```
  9. Specify the IPsec Phase 2 policy.
 

```
[edit security ipsec]
user@host# set vpn ike-vpn-chicago ike ipsec-policy ipsec-phase2-policy
```

**Results** From configuration mode, confirm your configuration by entering the **show security ipsec** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ipsec
proposal ipsec-phase2-proposal {
  protocol esp;
  authentication-algorithm hmac-sha1-96;
  encryption-algorithm aes-128-cbc;
}
policy ipsec-phase2-policy {
  perfect-forward-secrecy {
    keys group2;
  }
  proposals ipsec-phase2-proposal;
}
vpn ike-vpn-chicago {
  ike {
    gateway gw-chicago;
    ipsec-policy ipsec-phase2-policy;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.



**Configuring Security Policies**

**CLI Quick Configuration** To quickly configure security policies, copy the following commands and paste them into the CLI:

```
[edit]
set security policies from-zone trust to-zone untrust policy vpn-tr-untr match
  source-address sunnyvale
set security policies from-zone trust to-zone untrust policy vpn-tr-untr match
  destination-address chicago
set security policies from-zone trust to-zone untrust policy vpn-tr-untr match application
  any
set security policies from-zone trust to-zone untrust policy vpn-tr-untr then permit tunnel
  ipsec-vpn ike-vpn-chicago
set security policies from-zone trust to-zone untrust policy vpn-tr-untr then permit tunnel
  pair-policy vpn-untr-tr
set security policies from-zone untrust to-zone trust policy vpn-untr-tr match
  source-address chicago
set security policies from-zone untrust to-zone trust policy vpn-untr-tr match
  destination-address sunnyvale
set security policies from-zone untrust to-zone trust policy vpn-untr-tr match application
  any
set security policies from-zone untrust to-zone trust policy vpn-untr-tr then permit tunnel
  ipsec-vpn ike-vpn-chicago
set security policies from-zone untrust to-zone trust policy vpn-untr-tr then permit tunnel
  pair-policy vpn-tr-untr
set security policies from-zone trust to-zone untrust policy permit-any match
  source-address any
set security policies from-zone trust to-zone untrust policy permit-any match
  destination-address any
set security policies from-zone trust to-zone untrust policy permit-any match application
  any
set security policies from-zone trust to-zone untrust policy permit-any then permit
insert security policies from-zone trust to-zone untrust policy vpn-tr-untr before policy
  permit-any
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure security policies:

1. Create the security policy to permit traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy vpn-tr-untr match source-address sunnyvale
user@host# set policy vpn-tr-untr match destination-address chicago
user@host# set policy vpn-tr-untr match application any
user@host# set policy vpn-tr-untr then permit tunnel ipsec-vpn ike-vpn-chicago
user@host# set policy vpn-tr-untr then permit tunnel pair-policy vpn-untr-tr
```

2. Create the security policy to permit traffic from the untrust zone to the trust zone.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy vpn-untr-tr match source-address sunnyvale
user@host# set policy vpn-untr-tr match destination-address chicago
```

```

user@host# set policy vpn-untr-tr match application any
user@host# set policy vpn-untr-tr then permit tunnel ipsec-vpn ike-vpn-chicago
user@host# set policy vpn-untr-tr then permit tunnel pair-policy vpn-tr-untr

```

3. Create the security policy to permit traffic from the trust zone to the untrust zone.

```

[edit security policies from-zone trust to-zone untrust]
user@host# set policy permit-any match source-address any
user@host# set policy vpn-untr-tr match destination-address any
user@host# set policy vpn-untr-tr match application any
user@host# set policy vpn-untr-tr then permit

```

4. Reorder the security policies so that the vpn-tr-untr security policy is placed above the permit-any security policy.

```

[edit security policies from-zone trust to-zone untrust]
user@host# insert policy vpn-tr-untr before policy permit-any

```

**Results** From configuration mode, confirm your configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show security policies
from-zone trust to-zone untrust {
  policy vpn-tr-untr {
    match {
      source-address sunnyvale;
      destination-address chicago;
      application any;
    }
    then {
      permit {
        tunnel {
          ipsec-vpn ike-vpn-chicago;
          pair-policy vpn-untr-tr;
        }
      }
    }
  }
  policy permit-any {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit
    }
  }
}
from-zone untrust to-zone trust {
  policy vpn-untr-tr {
    match {
      source-address chicago;
      destination-address sunnyvale;
    }
  }
}

```

```

        application any;
    }
    then {
        permit {
            tunnel {
                ipsec-vpn ike-vpn-chicago;
                pair-policy vpn-tr-untr;
            }
        }
    }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring TCP-MSS

**CLI Quick Configuration** To quickly configure TCP-MSS information, copy the following commands and paste them into the CLI:

```

[edit]
set security flow tcp-mss ipsec-vpn mss 1350

```

**Step-by-Step Procedure** To configure TCP-MSS information:

1. Configure TCP-MSS information.

```

[edit]
user@host# set security flow tcp-mss ipsec-vpn mss 1350

```

**Results** From configuration mode, confirm your configuration by entering the **show security flow** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show security flow
tcp-mss {
    ipsec-vpn {
        mss 1350;
    }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring the SSG Series Device

**CLI Quick Configuration** For reference, the configuration for the SSG Series device is provided. For information about configuring SSG Series devices, see the *Concepts and Examples ScreenOS Reference Guide*, which is located at <http://www.juniper.net/techpubs>.

To quickly configure the SSG Series device, copy the following commands and paste them into the CLI:

```

set interface ethernet0/6 zone Trust
set interface ethernet0/0 zone Untrust
set interface ethernet0/6 ip 192.168.168.1/24

```

```

set interface ethernet0/6 route
set interface ethernet0/0 ip 2.2.2.2/30
set interface ethernet0/0 route
set flow tcp-mss 1350
set address Trust "local-net" 192.168.168.0 255.255.255.0
set address Untrust "corp-net" 10.10.10.0 255.255.255.0
set ike gateway corp-ike address 1.1.1.2 Main outgoing-interface ethernet0/0 preshare
  395psksecr3t sec-level standard
set vpn corp-vpn gateway corp-ike replay tunnel idletime 0 sec-level standard
set policy id 11 from Trust to Untrust "local-net" "corp-net" "ANY" tunnel vpn "corp-vpn"
  pair-policy 10
set policy id 10 from Untrust to Trust "corp-net" "local-net" "ANY" tunnel vpn "corp-vpn"
  pair-policy 11
set policy id 1 from Trust to Untrust "ANY" "ANY" "ANY" nat src permit
set route 0.0.0.0/0 interface ethernet0/0 gateway 2.2.2.1

```

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying the IKE Phase 1 Status on page 502
- Verifying the IPsec Phase 2 Status on page 504
- Reviewing Statistics and Errors for an IPsec Security Association on page 505

#### Verifying the IKE Phase 1 Status

**Purpose** Verify the IKE Phase 1 status.

#### Action



**NOTE:** Before starting the verification process, you need to send traffic from a host in the 10.10.10/24 network to a host in the 192.168.168/24 network. For policy-based VPNs, a separate host must generate the traffic; traffic initiated from the SRX Series device will not match the VPN policy. We recommend that the test traffic be from a separate device on one side of the VPN to a second device on the other side of the VPN. For example, initiate ping from 10.10.10.10 to 192.168.168.10.

From operational mode, enter the **show security ike security-associations** command. After obtaining an index number from the command, use the **show security ike security-associations index *index\_number* detail** command.

```

user@host> show security ike security-associations
Index  Remote Address  State  Initiator cookie  Responder cookie  Mode
4      2.2.2.2          UP     5e1db3f9d50b0de6  e50865d9ebf134f8  Main

```

```

user@host> show security ike security-associations index 4 detail
IKE peer 2.2.2.2, Index 4,
  Role: Responder, State: UP
  Initiator cookie: 5e1db3f9d50b0de6, Responder cookie: e50865d9ebf134f8
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 1.1.1.2:500, Remote: 2.2.2.2:500
  Lifetime: Expires in 28770 seconds
  Algorithms:

```

```

Authentication      : sha1
Encryption          : aes-128-cbc
Pseudo random function: hmac-sha1
Traffic statistics:
Input bytes       :           852
Output bytes      :           856
Input packets     :             5
Output packets    :             4
Flags: Caller notification sent
IPSec security associations: 1 created, 0 deleted
Phase 2 negotiations in progress: 0

```

**Meaning** The **show security ike security-associations** command lists all active IKE Phase 1 security associations (SAs). If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the **show security ike security-associations index detail** command to get more information about the SA.
- Remote Address—Verify that the remote IP address is correct.
- State
  - UP—The Phase 1 SA has been established.
  - DOWN—There was a problem establishing the Phase 1 SA.
- Mode—Verify that the correct mode is being used.

Verify that the following are correct in your configuration:

- External interfaces (the interface must be the one that receives IKE packets)
- IKE policy parameters
- Preshared key information
- Phase 1 proposal parameters (must match on both peers)

The **show security ike security-associations index 1 detail** command lists additional information about the security association with an index number of 1:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Initiator and responder role information



**NOTE:** Troubleshooting is best performed on the peer using the responder role.

- Number of IPsec SAs created
- Number of Phase 2 negotiations in progress

### Verifying the IPsec Phase 2 Status

**Purpose** Verify the IPsec Phase 2 status.

**Action** From operational mode, enter the **show security ipsec security-associations** command. After obtaining an index number from the command, use the **show security ipsec security-associations index *index\_number* detail** command.

```
user@host> show security ipsec security-associations
total configured sa: 2
ID      Gateway      Port  Algorithm      SPI      Life:sec/kb  Mon vsys
<2     2.2.2.2      500   ESP:aes-128/sha1  a63eb26f 3565/ unlim  -   0
>2     2.2.2.2      500   ESP:aes-128/sha1  a1024ed9 3565/ unlim  -   0
```

```
user@host> show security ipsec security-associations index 2 detail
Virtual-system: Root
Local Gateway: 1.1.1.2, Remote Gateway: 2.2.2.2
Local Identity: ipv4_subnet(any:0,[0..7]=10.10.10.0/24)
Remote Identity: ipv4_subnet(any:0,[0..7]=192.168.168.0/24)
DF-bit: clear
Policy-name: vpnpolicy-unt-tr

Direction: inbound, SPI: 2789126767, AUX-SPI: 0
Hard lifetime: Expires in 3558 seconds
Lifesize Remaining: Unlimited
Soft lifetime: Expires in 2986 seconds
Mode: tunnel, Type: dynamic, State: installed, VPN Monitoring: -
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)

Anti-replay service: enabled, Replay window size: 32

Direction: outbound, SPI: 2701283033,, AUX-SPI: 0
Hard lifetime: Expires in 3558 seconds
Lifesize Remaining: Unlimited
Soft lifetime: Expires in 2986 seconds
Mode: tunnel, Type: dynamic, State: installed, VPN Monitoring: -
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc
Anti-replay service: enabled, Replay window size: 32
```

**Meaning** The output from the **show security ipsec security-associations** command lists the following information:

- The ID number is 2. Use this value with the **show security ipsec security-associations index** command to get more information about this particular SA.
- There is one IPsec SA pair using port 500, which indicates that no NAT-traversal is implemented. (NAT-traversal uses port 4500 or another random high-number port.)
- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 3565/ unlim value indicates that the Phase 2 lifetime expires in 3565 seconds, and that no lifesize has been specified, which indicates that it is unlimited. Phase 2 lifetime can differ from Phase 1 lifetime, as Phase 2 is not dependent on Phase 1 after the VPN is up.

- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U (up) or D (down) is listed.
- The virtual system (vsys) is the root system, and it always lists 0.

The output from the `show security ipsec security-associations index 16384 detail` command lists the following information:

- The local identity and remote identity make up the proxy ID for the SA.

A proxy ID mismatch is one of the most common reasons for a Phase 2 failure. For policy-based VPNs, the proxy ID is derived from the security policy. The local address and remote address are derived from the address book entries, and the service is derived from the application configured for the policy. If Phase 2 fails because of a proxy ID mismatch, you can use the policy to confirm which address book entries are configured. Verify that the addresses match the information being sent. Check the service to ensure that the ports match the information being sent.



**NOTE:** For some third-party vendors, the proxy ID must be manually entered to match.

### *Reviewing Statistics and Errors for an IPsec Security Association*

**Purpose** Review ESP and authentication header counters and errors for an IPsec security association.

**Action** From operational mode, enter the `show security ipsec statistics index index_number` command, using the index number of the VPN for which you want to see statistics.

```
user@host> show security ipsec statistics index 2
ESP Statistics:
  Encrypted bytes:          920
  Decrypted bytes:         6208
  Encrypted packets:        5
  Decrypted packets:       87
AH Statistics:
  Input bytes:              0
  Output bytes:             0
  Input packets:            0
  Output packets:          0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

You can also use the `show security ipsec statistics` command to review statistics and errors for all SAs.

To clear all IPsec statistics, use the `clear security ipsec statistics` command.

**Meaning** If you see packet loss issues across a VPN, you can run the `show security ipsec statistics` or `show security ipsec statistics detail` command several times to confirm that the

encrypted and decrypted packet counters are incrementing. You should also check if the other error counters are incrementing.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- VPN Overview on page 451
- Example: Configuring a Route-Based VPN on page 470
- Example: Configuring a Hub-and-Spoke VPN on page 507

## Hub-and-Spoke VPNs

- Understanding Hub-and-Spoke VPNs on page 506
- Example: Configuring a Hub-and-Spoke VPN on page 507

### Understanding Hub-and-Spoke VPNs

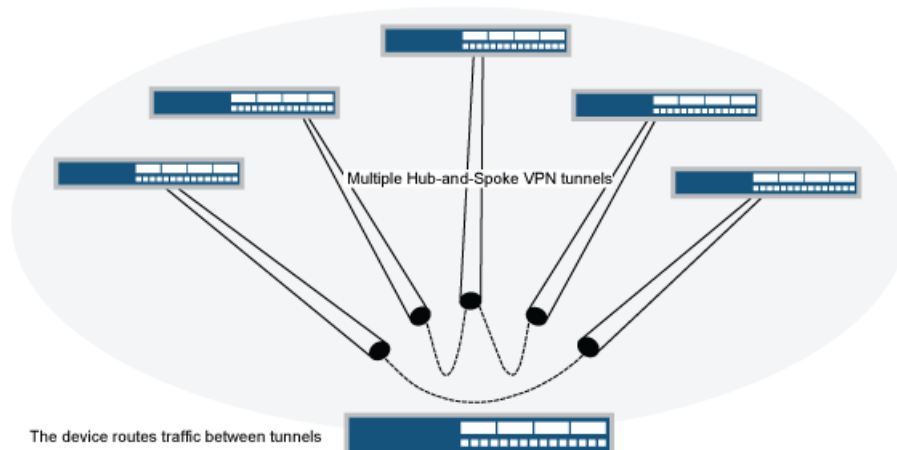
If you create two VPN tunnels that terminate at a device, you can set up a pair of routes so that the device directs traffic exiting one tunnel to the other tunnel. You also need to create a policy to permit the traffic to pass from one tunnel to the other. Such an arrangement is known as *hub-and-spoke VPN*. (See Figure 50 on page 506.)

You can also configure multiple VPNs and route traffic between any two tunnels.



**NOTE:** SRX Series devices support only the route-based hub-and-spoke feature.

Figure 50: Multiple Tunnels in a Hub-and-Spoke VPN Configuration



#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Hub-and-Spoke VPN Configuration Overview
- Example: Configuring a Hub-and-Spoke VPN on page 507



## Example: Configuring a Hub-and-Spoke VPN

This example shows how to configure a hub-and-spoke IPsec VPN for an enterprise-class deployment.

- Requirements on page 507
- Overview on page 507
- Configuration on page 513
- Verification on page 532

### Requirements

---

This example uses the following hardware:

- SRX240 device
- SRX5800 device
- SSG140 device

Before you begin, read “VPN Overview” on page 451.

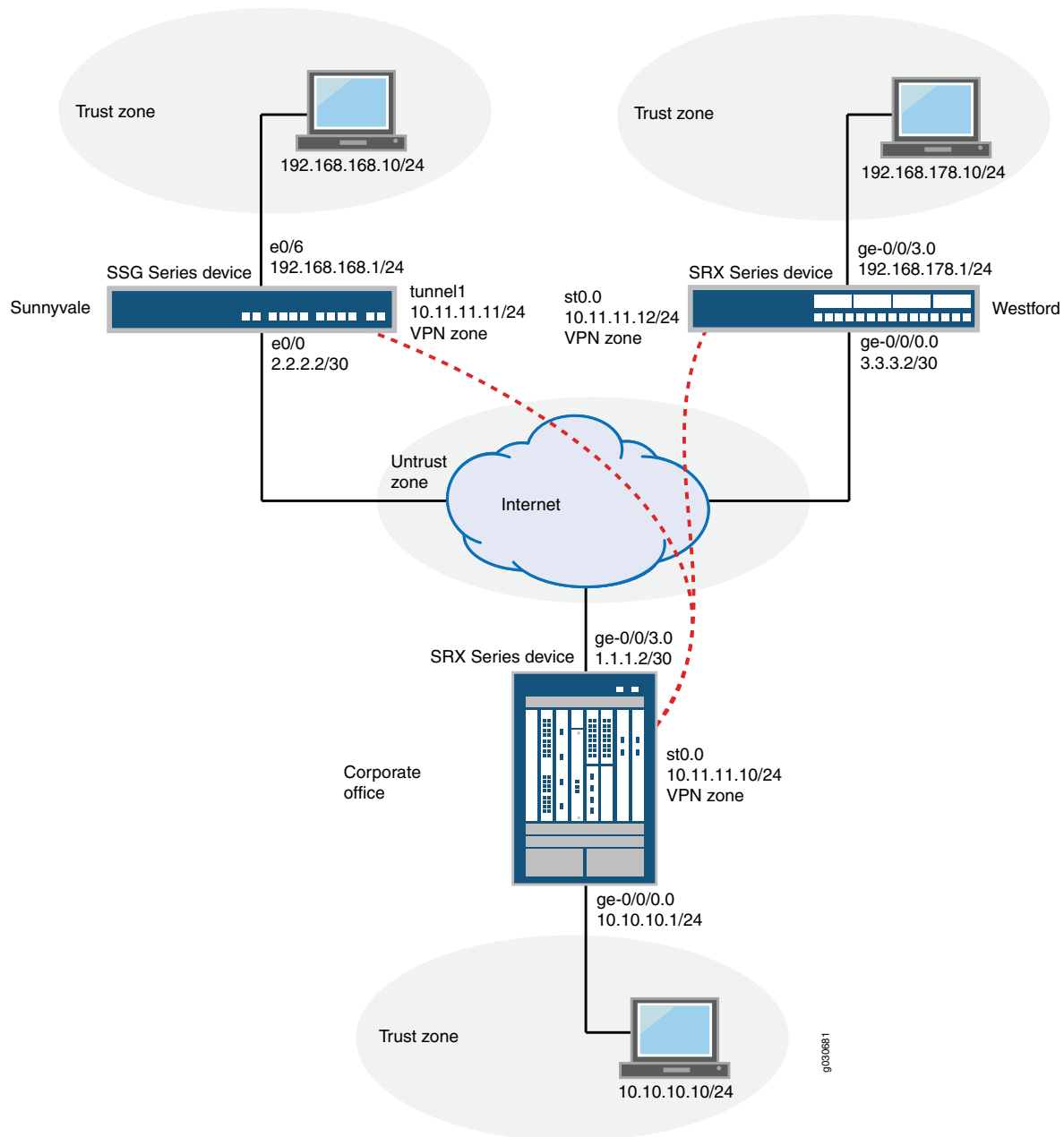
### Overview

---

This example describes how to configure a hub-and-spoke VPN typically found in branch deployments. The hub is the corporate office, and there are two spokes—a branch office in Sunnyvale, California, and a branch office in Westford, Massachusetts. Users in the branch offices will use the VPN to securely transfer data with the corporate office.

Figure 51 on page 508 shows an example of a hub-and-spoke VPN topology. In this topology, an SRX5800 device is located at the corporate office. An SRX240 device is located at the Westford branch, and an SSG140 device is located at the Sunnyvale branch.

Figure 51: Hub-and-Spoke VPN Topology



In this example, you configure the corporate office hub, the Westford spoke, and the Sunnyvale spoke. First you configure interfaces, IPv4 static and default routes, security zones, and address books. Then you configure IKE Phase 1 and IPsec Phase 2 parameters, and bind the `st0.0` interface to the IPsec VPN. On the hub, you configure `st0.0` for multipoint and add a static NHTB table entry for the Sunnyvale spoke. Finally, you configure security policy and TCP-MSS parameters. See Table 51 on page 509 through Table 55 on page 513 for specific configuration parameters used in this example.

Table 51: Interface, Security Zone, and Address Book Information

| Hub or Spoke | Feature              | Name          | Configuration Parameters                                                                                                                                                      |
|--------------|----------------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hub          | Interfaces           | ge-0/0/0.0    | 10.10.10.1/24                                                                                                                                                                 |
|              |                      | ge-0/0/3.0    | 1.1.1.2/30                                                                                                                                                                    |
|              |                      | st0           | 10.11.11.10/24                                                                                                                                                                |
| Spoke        | Interfaces           | ge-0/0/0.0    | 3.3.3.2/30                                                                                                                                                                    |
|              |                      | ge-0/0/3.0    | 192.168.178.1/24                                                                                                                                                              |
|              |                      | st0           | 10.11.11.12/24                                                                                                                                                                |
| Hub          | Security zones       | trust         | <ul style="list-style-type: none"> <li>All system services are allowed.</li> <li>The ge-0/0/0.0 interface is bound to this zone.</li> </ul>                                   |
|              |                      | untrust       | <ul style="list-style-type: none"> <li>IKE is the only allowed system service.</li> <li>The ge-0/0/3.0 interface is bound to this zone.</li> </ul>                            |
|              |                      | vpn           | The st0.0 interface is bound to this zone.                                                                                                                                    |
| Spoke        | Security zones       | trust         | <ul style="list-style-type: none"> <li>All system services are allowed.</li> <li>The ge-0/0/3.0 interface is bound to this zone.</li> </ul>                                   |
|              |                      | untrust       | <ul style="list-style-type: none"> <li>IKE is the only allowed system service.</li> <li>The ge-0/0/0.0 interface is bound to this zone.</li> </ul>                            |
|              |                      | vpn           | The st0.0 interface is bound to this zone.                                                                                                                                    |
| Hub          | Address book entries | local-net     | <ul style="list-style-type: none"> <li>This address is for the trust zone's address book.</li> <li>The address for this address book entry is 10.10.10.0/24.</li> </ul>       |
|              |                      | sunnyvale-net | <ul style="list-style-type: none"> <li>This address book is for the vpn zone's address book.</li> <li>The address for this address book entry is 192.168.168.0/24.</li> </ul> |

Table 51: Interface, Security Zone, and Address Book Information (*continued*)

| Hub or Spoke | Feature              | Name          | Configuration Parameters                                                                                                                                                       |
|--------------|----------------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              |                      | westford-net  | <ul style="list-style-type: none"> <li>This address is for the vpn zone's address book.</li> <li>The address for this address book entry is 192.168.178.0/24.</li> </ul>       |
| Spoke        | Address book entries | local-net     | <ul style="list-style-type: none"> <li>This address is for the trust zone's address book.</li> <li>The address for this address book entry is 192.168.168.178.0/24.</li> </ul> |
|              |                      | corp-net      | <ul style="list-style-type: none"> <li>This address is for the vpn zone's address book.</li> <li>The address for this address book entry is 10.10.10.0/24.</li> </ul>          |
|              |                      | sunnyvale-net | <ul style="list-style-type: none"> <li>This address is for the vpn zone's address book.</li> <li>The address for this address book entry is 192.168.168.0/24.</li> </ul>       |

Table 52: IKE Phase 1 Configuration Parameters

| Hub or Spoke | Feature  | Name                | Configuration Parameters                                                                                                                                                                                          |
|--------------|----------|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hub          | Proposal | ike-phase1-proposal | <ul style="list-style-type: none"> <li>Authentication method: pre-shared-keys</li> <li>Diffie-Hellman group: group2</li> <li>Authentication algorithm: sha1</li> <li>Encryption algorithm: aes-128-cbc</li> </ul> |
|              | Policy   | ike-phase1-policy   | <ul style="list-style-type: none"> <li>Mode: main</li> <li>Proposal reference: ike-phase1-proposal</li> <li>IKE Phase 1 policy authentication method: pre-shared-key<br/>ascii-text</li> </ul>                    |
|              | Gateway  | gw-westford         | <ul style="list-style-type: none"> <li>IKE policy reference: ike-phase1-policy</li> <li>External interface: ge-0/0/3.0</li> <li>Gateway address: 3.3.3.2</li> </ul>                                               |
|              |          | gw-sunnyvale        | <ul style="list-style-type: none"> <li>IKE policy reference: ike-phase1-policy</li> <li>External interface: ge-0/0/3.0</li> <li>Gateway address: 2.2.2.2</li> </ul>                                               |

Table 52: IKE Phase 1 Configuration Parameters (*continued*)

| Hub or Spoke | Feature  | Name                | Configuration Parameters                                                                                                                                                                                                  |
|--------------|----------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Spoke        | Proposal | ike-phase1-proposal | <ul style="list-style-type: none"> <li>• Authentication method: pre-shared-keys</li> <li>• Diffie-Hellman group: group2</li> <li>• Authentication algorithm: sha1</li> <li>• Encryption algorithm: aes-128-cbc</li> </ul> |
|              | Policy   | ike-phase1-policy   | <ul style="list-style-type: none"> <li>• Mode: main</li> <li>• Proposal reference: ike-phase1-proposal</li> <li>• IKE Phase 1 policy authentication method: pre-shared-key<br/>ascii-text</li> </ul>                      |
|              | Gateway  | gw-corporate        | <ul style="list-style-type: none"> <li>• IKE policy reference: ike-phase1-policy</li> <li>• External interface: ge-0/0/0.0</li> <li>• Gateway address: 1.1.1.2</li> </ul>                                                 |

Table 53: IPsec Phase 2 Configuration Parameters

| Hub or Spoke | Feature  | Name                  | Configuration Parameters                                                                                                                                                           |
|--------------|----------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hub          | Proposal | ipsec-phase2-proposal | <ul style="list-style-type: none"> <li>• Protocol: esp</li> <li>• Authentication algorithm: hmac-sha1-96</li> <li>• Encryption algorithm: aes-128-cbc</li> </ul>                   |
|              | Policy   | ipsec-phase2-policy   | <ul style="list-style-type: none"> <li>• Proposal reference: ipsec-phase2-proposal</li> <li>• PFS: Diffie-Hellman group2</li> </ul>                                                |
|              | VPN      | vpn-sunnyvale         | <ul style="list-style-type: none"> <li>• IKE gateway reference: gw-sunnyvale</li> <li>• IPsec policy reference: ipsec-phase2-policy</li> <li>• Bind to interface: st0.0</li> </ul> |
|              |          | vpn-westford          | <ul style="list-style-type: none"> <li>• IKE gateway reference: gw-westford</li> <li>• IPsec policy reference: ipsec-phase2-policy</li> <li>• Bind to interface: st0.0</li> </ul>  |
| Spoke        | Proposal | ipsec-phase2-proposal | <ul style="list-style-type: none"> <li>• Protocol: esp</li> <li>• Authentication algorithm: hmac-sha1-96</li> <li>• Encryption algorithm: aes-128-cbc</li> </ul>                   |
|              | Policy   | ipsec-phase2-policy   | <ul style="list-style-type: none"> <li>• Proposal reference: ipsec-phase2-proposal</li> <li>• PFS: Diffie-Hellman group2</li> </ul>                                                |
|              | VPN      | vpn-corporate         | <ul style="list-style-type: none"> <li>• IKE gateway reference: gw-corporate</li> <li>• IPsec policy reference: ipsec-phase2-policy</li> <li>• Bind to interface: st0.0</li> </ul> |

Table 54: Security Policy Configuration Parameters

| Hub or Spoke | Purpose                                                                      | Name            | Configuration Parameters                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------|------------------------------------------------------------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hub          | The security policy permits traffic from the trust zone to the vpn zone.     | local-to-spokes | <ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address local-net</li> <li>destination-address sunnyvale-net</li> <li>destination-address westford-net</li> <li>application any</li> </ul> </li> </ul>                                                                                                                                                                                                                                                    |
|              | The security policy permits traffic from the vpn zone to the trust zone.     | spokes-to-local | <ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address sunnyvale-net</li> <li>source-address westford-net</li> <li>destination-address local-net</li> <li>application any</li> </ul> </li> </ul>                                                                                                                                                                                                                                                         |
|              | The security policy permits intrazone traffic.                               | spoke-to-spoke  | <ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address any</li> <li>destination-address any</li> <li>application any</li> </ul> </li> </ul>                                                                                                                                                                                                                                                                                                              |
| Spoke        | The security policy permits traffic from the trust zone to the vpn zone.     | to-corp         | <ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address local-net</li> <li>destination-address corp-net</li> <li>destination-address sunnyvale-net</li> <li>application any</li> </ul> </li> </ul>                                                                                                                                                                                                                                                        |
|              | The security policy permits traffic from the vpn zone to the trust zone.     | from-corp       | <ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address corp-net</li> <li>source-address sunnyvale-net</li> <li>destination-address local-net</li> <li>application any</li> </ul> </li> </ul>                                                                                                                                                                                                                                                             |
|              | The security policy permits traffic from the untrust zone to the trust zone. | permit-any      | <ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address any</li> <li>source-destination any</li> <li>application any</li> </ul> </li> <li>Permit action: source-nat interface</li> </ul> <p>By specifying <b>source-nat interface</b>, the SRX Series device translates the source IP address and port for outgoing traffic, using the IP address of the egress interface as the source IP address and a random high-number port for the source port.</p> |

Table 55: TCP-MSS Configuration Parameters

| Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Configuration Parameters |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| <p>TCC-MSS is negotiated as part of the TCP three-way handshake and limits the maximum size of a TCP segment to better fit the MTU limits on a network. For VPN traffic, the IPsec encapsulation overhead, along with the IP and frame overhead, can cause the resulting ESP packet to exceed the MTU of the physical interface, which causes fragmentation. Fragmentation results in increased use of bandwidth and device resources.</p> <p><b>NOTE:</b> The value of 1350 is a recommended starting point for most Ethernet-based networks with an MTU of 1500 or greater. You might need to experiment with different TCP-MSS values to obtain optimal performance. For example, you might need to change the value if any device in the path has a lower MTU, or if there is any additional overhead such as PPP or Frame Relay.</p> | MSS value: 1350          |

### Configuration

- Configuring Basic Network, Security Zone, and Address Book Information for the Hub on page 513
- Configuring IKE for the Hub on page 516
- Configuring IPsec for the Hub on page 518
- Configuring Security Policies for the Hub on page 520
- Configuring TCP-MSS for the Hub on page 522
- Configuring Basic Network, Security Zone, and Address Book Information for the Westford Spoke on page 523
- Configuring IKE for the Westford Spoke on page 526
- Configuring IPsec for the Westford Spoke on page 528
- Configuring Security Policies for the Westford Spoke on page 529
- Configuring TCP-MSS for the Westford Spoke on page 531
- Configuring the Sunnyvale Spoke on page 531

#### *Configuring Basic Network, Security Zone, and Address Book Information for the Hub*

**CLI Quick Configuration** To quickly configure basic network, security zone, and address book information for the hub, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet address 10.10.10.1/24
set interfaces ge-0/0/3 unit 0 family inet address 1.1.1.2/30
set interfaces st0 unit 0 family inet address 10.11.11.10/24
set routing-options static route 0.0.0.0/0 next-hop 1.1.1.1
set routing-options static route 192.168.168.0/24 next-hop 10.11.11.11
set routing-options static route 192.168.178.0/24 next-hop 10.11.11.12
set security zones security-zone untrust interfaces ge-0/0/3.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust address-book address local-net 10.10.10.0/24
set security zones security-zone vpn interfaces st0.0
set security zones security-zone vpn address-book address sunnyvale-net 192.168.168.0/24
set security zones security-zone vpn address-book address westford-net 192.168.178.0/24
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure basic network, security zone, and address book information for the hub:

1. Configure Ethernet interface information.  

```
[edit]
user@hub# set interfaces ge-0/0/0 unit 0 family inet address 10.10.10.1/24
user@hub# set interfaces ge-0/0/3 unit 0 family inet address 1.1.1.2/30
user@hub# set interfaces st0 unit 0 family inet address 10.11.11.10/24
```
2. Configure static route information.  

```
[edit]
user@hub# set routing-options static route 0.0.0.0/0 next-hop 1.1.1.1
user@hub# set routing-options static route 192.168.168.0/24 next-hop 10.11.11.11
user@hub# set routing-options static route 192.168.178.0/24 next-hop 10.11.11.12
```
3. Configure the untrust security zone.  

```
[edit ]
user@hub# set security zones security-zone untrust
```
4. Assign an interface to the untrust security zone.  

```
[edit security zones security-zone untrust]
user@hub# set interfaces ge-0/0/3.0
```
5. Specify allowed system services for the untrust security zone.  

```
[edit security zones security-zone untrust]
user@hub# set host-inbound-traffic system-services ike
```
6. Configure the trust security zone.  

```
[edit]
user@hub# edit security zones security-zone trust
```
7. Assign an interface to the trust security zone.  

```
[edit security zones security-zone trust]
user@hub# set interfaces ge-0/0/0.0
```
8. Specify allowed system services for the trust security zone.  

```
[edit security zones security-zone trust]
user@hub# set host-inbound-traffic system-services all
```
9. Configure an address book entry for the trust security zone.  

```
[edit security zones security-zone trust]
user@hub# set address-book address local-net 10.10.10.0/24
```
10. Configure the vpn security zone.  

```
[edit]
user@hub# edit security zones security-zone vpn
```
11. Assign an interface to the vpn security zone.  

```
[edit security zones security-zone vpn]
```



```
user@hub# set interfaces st0.0
```

12. Configure an address book entry for the trust security zone.

```
[edit security zones security-zone trust]
user@hub# set address-book address local-net 10.10.10.0/24
```

13. Configure address book entries for the vpn security zone.

```
[edit security zones security-zone vpn]
user@hub# set address-book address sunnyvale-net 192.168.168.0/24
[edit security zones security-zone vpn]
user@hub# set address-book address westford-net 192.168.178.0/24
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show routing-options**, and **show security zones** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@hub# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 10.10.10.1/24;
    }
  }
}
ge-0/0/3 {
  unit 0 {
    family inet {
      address 1.1.1.2/30
    }
  }
}
st0 {
  unit 0 {
    family inet {
      address 10.11.11.10/24
    }
  }
}

[edit]
user@hub# show routing-options
static {
  route 0.0.0.0/0 next-hop 1.1.1.1;
  route 192.168.168.0/24 next-hop 10.11.11.11;
  route 192.168.178.0/24 next-hop 10.11.11.12;
}

[edit]
user@hub# show security zones
security-zone untrust {
  host-inbound-traffic {
    system-services {
      ike;
    }
  }
}
```

```

    }
  }
  interfaces {
    ge-0/0/3.0;
  }
}
security-zone trust {
  address-book {
    address local-net 10.10.10.0/24 {
    }
  }
  host-inbound-traffic {
    system-services {
      all;
    }
  }
  interfaces {
    ge-0/0/0.0;
  }
}
security-zone vpn {
  host-inbound-traffic {
    address-book {
      address sunnyvale-net 192.168.168.0/24;
      address westford--net 192.168.178.0/24;
    }
  }
  interfaces {
    st0.0;
  }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### *Configuring IKE for the Hub*

#### **CLI Quick Configuration**

To quickly configure IKE for the hub, copy the following commands and paste them into the CLI:

```

[edit]
set security ike proposal ike-phase1-proposal authentication-method pre-shared-keys
set security ike proposal ike-phase1-proposal dh-group group2
set security ike proposal ike-phase1-proposal authentication-algorithm sha1
set security ike proposal ike-phase1-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-phase1-policy mode main
set security ike policy ike-phase1-policy proposals ike-phase1-proposal
set security ike policy ike-phase1-policy pre-shared-key ascii-text 395psksecr3t
set security ike gateway gw-westford external-interface ge-0/0/3.0
set security ike gateway gw-westford ike-policy ike-phase1-policy
set security ike gateway gw-westford address 3.3.3.2
set security ike gateway gw-sunnyvale external-interface ge-0/0/3.0
set security ike gateway gw-sunnyvale ike-policy ike-phase1-policy
set security ike gateway gw-sunnyvale address 2.2.2.2

```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure IKE for the hub:

1. Create the IKE Phase 1 proposal.
 

```
[edit security ike]
user@hub# set proposal ike-phase1-proposal
```
2. Define the IKE proposal authentication method.
 

```
[edit security ike proposal ike-phase1-proposal]
user@hub# set authentication-method pre-shared-keys
```
3. Define the IKE proposal Diffie-Hellman group.
 

```
[edit security ike proposal ike-phase1-proposal]
user@hub# set dh-group group2
```
4. Define the IKE proposal authentication algorithm.
 

```
[edit security ike proposal ike-phase1-proposal]
user@hub# set authentication-algorithm sha1
```
5. Define the IKE proposal encryption algorithm.
 

```
[edit security ike proposal ike-phase1-proposal]
user@hub# set encryption-algorithm aes-128-cbc
```
6. Create an IKE Phase 1 policy.
 

```
[edit security ike]
user@hub# set policy ike-phase1-policy
```
7. Set the IKE Phase 1 policy mode.
 

```
[edit security ike policy ike-phase1-policy]
user@hub# set mode main
```
8. Specify a reference to the IKE proposal.
 

```
[edit security ike policy ike-phase1-policy]
user@hub# set proposals ike-phase1-proposal
```
9. Define the IKE Phase 1 policy authentication method.
 

```
[edit security ike policy ike-phase1-policy]
user@hub# set pre-shared-key ascii-text 395psksecr3t
```
10. Create an IKE Phase 1 gateway and define its external interface.
 

```
[edit security ike]
user@hub# set gateway gw-westford external-interface ge-0/0/3.0
```
11. Define the IKE Phase 1 policy reference.
 

```
[edit security ike]
user@hub# set gateway gw-westford ike-policy ike-phase1-policy
```
12. Define the IKE Phase 1 gateway address.
 

```
[edit security ike]
```

```
user@hub# set gateway gw-westford address 3.3.3.2
```

13. Create an IKE Phase 1 gateway and define its external interface.

```
[edit security ike]
user@hub# set gateway gw-sunnyvale external-interface ge-0/0/3.0
```

14. Define the IKE Phase 1 policy reference.

```
[edit security ike gateway]
user@hub# set gateway gw-sunnyvale ike-policy ike-phase1-policy
```

15. Define the IKE Phase 1 gateway address.

```
[edit security ike gateway]
user@hub# set gateway gw-sunnyvale address 2.2.2.2
```

**Results** From configuration mode, confirm your configuration by entering the **show security ike** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@hub# show security ike
proposal ike-phase1-proposal {
  authentication-method pre-shared-keys;
  dh-group group2;
  authentication-algorithm sha1;
  encryption-algorithm aes-128-cbc;
}
policy ike-phase1-policy {
  mode main;
  proposals ike-phase1-proposal;
  pre-shared-key ascii-text "$9$9VMTp1RvWLdwYKMJDkmF3ylKM87Vb2oZjws5F"; ##
  SECRET-DATA
}
gateway gw-sunnyvale {
  ike-policy ike-phase1-policy;
  address 2.2.2.2;
  external-interface ge-0/0/3.0;
}
gateway gw-westford {
  ike-policy ike-phase1-policy;
  address 3.3.3.2;
  external-interface ge-0/0/3.0;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### *Configuring IPsec for the Hub*

**CLI Quick Configuration** To quickly configure IPsec for the hub, copy the following commands and paste them into the CLI:

```
[edit]
set security ipsec proposal ipsec-phase2-proposal protocol esp
set security ipsec proposal ipsec-phase2-proposal authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec-phase2-proposal encryption-algorithm aes-128-cbc
set security ipsec policy ipsec-phase2-policy proposals ipsec-phase2-proposal
```

```

set security ipsec policy ipsec-phase2-policy perfect-forward-secrecy keys group2
set security ipsec vpn vpn-westford ike gateway gw-westford
set security ipsec vpn vpn-westford ike ipsec-policy ipsec-phase2-policy
set security ipsec vpn vpn-westford bind-interface st0.0
set security ipsec vpn vpn-sunnyvale ike gateway gw-sunnyvale
set security ipsec vpn vpn-sunnyvale ike ipsec-policy ipsec-phase2-policy
set security ipsec vpn vpn-sunnyvale bind-interface st0.0
set interfaces st0 unit 0 multipoint
set interfaces st0 unit 0 family inet next-hop-tunnel 10.11.11.11 ipsec-vpn vpn-sunnyvale

```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure IPsec for the hub:

1. Create an IPsec Phase 2 proposal.
 

```
[edit]
user@hub# set security ipsec proposal ipsec-phase2-proposal
```
2. Specify the IPsec Phase 2 proposal protocol.
 

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@hub# set protocol esp
```
3. Specify the IPsec Phase 2 proposal authentication algorithm.
 

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@hub# set authentication-algorithm hmac-sha1-96
```
4. Specify the IPsec Phase 2 proposal encryption algorithm.
 

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@hub# set encryption-algorithm aes-128-cbc
```
5. Create the IPsec Phase 2 policy.
 

```
[edit security ipsec]
user@hub# set policy ipsec-phase2-policy
```
6. Specify the IPsec Phase 2 proposal reference.
 

```
[edit security ipsec policy ipsec-phase2-policy]
user@hub# set proposals ipsec-phase2-proposal
```
7. Specify IPsec Phase 2 PFS to use Diffie-Hellman group 2.
 

```
[edit security ipsec policy ipsec-phase2-policy]
user@host# set perfect-forward-secrecy keys group2
```
8. Specify the IKE gateways.
 

```
[edit security ipsec]
user@hub# set vpn vpn-westford ike gateway gw-westford
user@hub# set vpn vpn-sunnyvale ike gateway gw-sunnyvale
```
9. Specify the IPsec Phase 2 policies.
 

```
[edit security ipsec]
user@hub# set vpn vpn-westford ike ipsec-policy ipsec-phase2-policy
user@hub# set vpn vpn-sunnyvale ike ipsec-policy ipsec-phase2-policy
```

10. Specify the interface to bind.

```
[edit security ipsec]
user@hub# set vpn vpn-westford bind-interface st0.0
user@hub# set vpn vpn-sunnyvale bind-interface st0.0
```

11. Configure the st0 interface as multipoint.

```
[edit]
user@hub# set interfaces st0 unit 0 multipoint
```

12. Add static NHTB table entries for the Sunnyvale and Westford offices.

```
[edit]
user@hub# set interfaces st0 unit 0 family inet next-hop-tunnel 10.11.11.11 ipsec-vpn
vpn-sunnyvale
user@hub# set interfaces st0 unit 0 family inet next-hop-tunnel 10.11.11.12 ipsec-vpn
vpn-westford
```

**Results** From configuration mode, confirm your configuration by entering the **show security ipsec** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@hub# show security ipsec
proposal ipsec-phase2-proposal {
  protocol esp;
  authentication-algorithm hmac-sha1-96;
  encryption-algorithm aes-128-cbc;
}
policy ipsec-phase2-policy {
  perfect-forward-secrecy {
    keys group2;
  }
  proposals ipsec-phase2-proposal;
}
vpn vpn-sunnyvale {
  bind-interface st0.0;
  ike {
    gateway gw-sunnyvale;
    ipsec-policy ipsec-phase2-policy;
  }
}
vpn vpn-westford {
  bind-interface st0.0;
  ike {
    gateway gw-westford;
    ipsec-policy ipsec-phase2-policy;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### *Configuring Security Policies for the Hub*

**CLI Quick Configuration** To quickly configure security policies for the hub, copy the following commands and paste them into the CLI:

```

[edit]
set security policies from-zone trust to-zone vpn policy local-to-spokes match
  source-address local-net
set security policies from-zone trust to-zone vpn policy local-to-spokes match
  destination-address sunnyvale-net
set security policies from-zone trust to-zone vpn policy local-to-spokes match
  destination-address westford-net
set security policies from-zone trust to-zone vpn policy local-to-spokes match application
  any
set security policies from-zone trust to-zone vpn policy local-to-spokes then permit
set security policies from-zone vpn to-zone trust policy spokes-to-local match
  source-address sunnyvale-net
set security policies from-zone vpn to-zone trust policy spokes-to-local match
  source-address westford-net
set security policies from-zone vpn to-zone trust policy spokes-to-local match
  destination-address local-net
set security policies from-zone vpn to-zone trust policy spokes-to-local match application
  any
set security policies from-zone vpn to-zone trust policy spokes-to-local then permit
set security policies from-zone vpn to-zone vpn policy spoke-to-spoke match
  source-address any
set security policies from-zone vpn to-zone vpn policy spoke-to-spoke match
  destination-address any
set security policies from-zone vpn to-zone vpn policy spoke-to-spoke match application
  any
set security policies from-zone vpn to-zone vpn policy spoke-to-spoke then permit

```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure security policies for the hub:

1. Create the security policy to permit traffic from the trust zone to the vpn zone.

```

[edit security policies from-zone trust to-zone vpn]
user@hub# set policy local-to-spokes match source-address local-net
user@hub# set policy local-to-spokes match destination-address sunnyvale-net
user@hub# set policy local-to-spokes match destination-address westford-net
user@hub# set policy local-to-spokes match application any
user@hub# set policy local-to-spokes then permit

```

2. Create the security policy to permit traffic from the vpn zone to the trust zone.

```

[edit security policies from-zone vpn to-zone trust]
user@hub# set policy spokes-to-local match source-address sunnyvale-net
user@hub# set policy spokes-to-local match source-address westford-net
user@hub# set policy spokes-to-local match destination-address local-net
user@hub# set policy spokes-to-local match application any
user@hub# set policy spokes-to-local then permit

```

3. Create the security policy to permit intrazone traffic.

```

[edit security policies from-zone vpn to-zone vpn]
user@hub# set policy spoke-to-spoke match source-address any
user@hub# set policy spoke-to-spoke match destination-address any
user@hub# set policy spoke-to-spoke match application any

```

```
user@hub# set policy spoke-to-spoke then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@hub# show security policies
from-zone trust to-zone vpn {
  policy local-to-spokes {
    match {
      source-address local-net;
      destination-address [ sunnyvale-net westford-net ];
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone vpn to-zone trust {
  policy spokes-to-local {
    match {
      source-address [ sunnyvale-net westford-net ];
      destination-address local-net;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone vpn to-zone vpn {
  policy spoke-to-spoke {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### *Configuring TCP-MSS for the Hub*

**CLI Quick Configuration** To quickly configure TCP-MSS for the hub, copy the following commands and paste them into the CLI:

```
[edit]
set security flow tcp-mss ipsec-vpn mss 1350
```



**Step-by-Step Procedure** To configure TCP-MSS information for the hub:

1. Configure TCP-MSS information.

```
[edit]
user@hub# set security flow tcp-mss ipsec-vpn mss 1350
```

**Results** From configuration mode, confirm your configuration by entering the **show security flow** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@hub# show security flow
tcp-mss {
  ipsec-vpn {
    mss 1350;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### *Configuring Basic Network, Security Zone, and Address Book Information for the Westford Spoke*

**CLI Quick Configuration** To quickly configure basic network, security zone, and address book information for the Westford spoke, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet address 3.3.3.2/30
set interfaces ge-0/0/3 unit 0 family inet address 192.168.178.1/24
set interfaces st0 unit 0 family inet address 10.11.11.12/24
set routing-options static route 0.0.0.0/0 next-hop 3.1.1.1
set routing-options static route 10.10.10.0/24 next-hop 10.11.11.10
set routing-options static route 192.168.168.0/24 next-hop 10.11.11.10
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone untrust interfaces ge-0/0/3.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone vpn address-book address local-net 192.168.178.0/24
set security zones security-zone vpn interfaces st0.0
set security zones security-zone trust address-book address corp-net 10.10.10.0/24
set security zones security-zone vpn address-book address sunnyvale-net 192.168.168.0/24
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure basic network, security zone, and address book information for the Westford spoke:

1. Configure Ethernet interface information.

```
[edit]
user@spoke# set interfaces ge-0/0/0 unit 0 family inet address 3.3.3.2/30
user@spoke# set interfaces ge-0/0/3 unit 0 family inet address 192.168.178.1/24
user@spoke# set interfaces st0 unit 0 family inet address 10.11.11.12/24
```

2. Configure static route information.  

```
[edit]  
user@spoke# set routing-options static route 0.0.0.0/0 next-hop 3.1.1.1  
user@spoke# set routing-options static route 10.10.10.0/24 next-hop 10.11.11.10  
user@spoke# set routing-options static route 192.168.168.0/24 next-hop 10.11.11.10
```
3. Configure the untrust security zone.  

```
[edit]  
user@spoke# set security zones security-zone untrust
```
4. Assign an interface to the security zone.  

```
[edit security zones security-zone untrust]  
user@spoke# set interfaces ge-0/0/0.0
```
5. Specify allowed system services for the untrust security zone.  

```
[edit security zones security-zone untrust]  
user@spoke# set host-inbound-traffic system-services ike
```
6. Configure the trust security zone.  

```
[edit]  
user@spoke# edit security zones security-zone trust
```
7. Assign an interface to the trust security zone.  

```
[edit security zones security-zone trust]  
user@spoke# set interfaces ge-0/0/3.0
```
8. Specify allowed system services for the trust security zone.  

```
[edit security zones security-zone trust]  
user@spoke# set host-inbound-traffic system-services all
```
9. Configure an address book entry for the trust security zone.  

```
[edit security zones security-zone trust]  
user@spoke# set address-book address local-net 192.168.178.0/24
```
10. Configure the vpn security zone.  

```
[edit]  
user@spoke# edit security zones security-zone vpn
```
11. Assign an interface to the vpn security zone.  

```
[edit security zones security-zone vpn]  
user@spoke# set interfaces st0.0
```
12. Configure address book entries for the vpn security zone.  

```
[edit security zones security-zone vpn]  
user@spoke# set address-book address corp-net 10.10.10.0/24  
[edit security zones security-zone vpn]  
user@spoke# set address-book address sunnyvale-net 192.168.168.0/24
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show routing-options**, and **show security zones** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@spoke# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 3.3.3.2/30;
    }
  }
}
ge-0/0/3 {
  unit 0 {
    family inet {
      address 192.168.178.1/24;
    }
  }
}
st0 {
  unit 0 {
    family inet {
      address 10.11.11.10/24;
    }
  }
}

[edit]
user@spoke# show routing-options
static {
  route 0.0.0.0/0 next-hop 1.1.1.1;
  route 192.168.168.0/24 next-hop 10.11.11.11;
  route 10.10.10.0/24 next-hop 10.11.11.10;
}

[edit]
user@spoke# show security zones
security-zone untrust {
  host-inbound-traffic {
    system-services {
      ike;
    }
  }
  interfaces {
    ge-0/0/0.0;
  }
}
security-zone trust {
  address-book {
    address local-net 192.168.178.0/24;
  }
  host-inbound-traffic {
    system-services {
      all;
    }
  }
  interfaces {
    ge-0/0/3.0;
  }
}
```

```

security-zone vpn {
  address-book {
    address corp-net 10.10.10.0/24;
    address sunnyvale-net 192.168.168.0/24;
  }
  interfaces {
    st0.0;
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring IKE for the Westford Spoke

**CLI Quick Configuration** To quickly configure IKE for the Westford spoke, copy the following commands and paste them into the CLI:

```

[edit]
set security ike proposal ike-phase1-proposal authentication-method pre-shared-keys
set security ike proposal ike-phase1-proposal dh-group group2
set security ike proposal ike-phase1-proposal authentication-algorithm sha1
set security ike proposal ike-phase1-proposal encryption-algorithm aes-128-cbc
set security ike policy ike-phase1-policy mode main
set security ike policy ike-phase1-policy proposals ike-phase1-proposal
set security ike policy ike-phase1-policy pre-shared-key ascii-text 395psksecr3t
set security ike gateway gw-corporate external-interface ge-0/0/0.0
set security ike gateway gw-corporate ike-policy ike-phase1-policy
set security ike gateway gw-corporate address 1.1.1.2

```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure IKE for the Westford spoke:

1. Create the IKE Phase 1 proposal.

```

[edit security ike]
user@spoke# set proposal ike-phase1-proposal

```

2. Define the IKE proposal authentication method.

```

[edit security ike proposal ike-phase1-proposal]
user@spoke# set authentication-method pre-shared-keys

```

3. Define the IKE proposal Diffie-Hellman group.

```

[edit security ike proposal ike-phase1-proposal]
user@spoke# set dh-group group2

```

4. Define the IKE proposal authentication algorithm.

```

[edit security ike proposal ike-phase1-proposal]
user@spoke# set authentication-algorithm sha1

```

5. Define the IKE proposal encryption algorithm.

```

[edit security ike proposal ike-phase1-proposal]
user@spoke# set encryption-algorithm aes-128-cbc

```

6. Create an IKE Phase 1 policy.
 

```
[edit security ike]
user@spoke# set policy ike-phase1-policy
```
7. Set the IKE Phase 1 policy mode.
 

```
[edit security ike policy ike-phase1-policy]
user@spoke# set mode main
```
8. Specify a reference to the IKE proposal.
 

```
[edit security ike policy ike-phase1-policy]
user@spoke# set proposals ike-phase1-proposal
```
9. Define the IKE Phase 1 policy authentication method.
 

```
[edit security ike policy ike-phase1-policy]
user@spoke# set pre-shared-key ascii-text 395psksecr3t
```
10. Create an IKE Phase 1 gateway and define its external interface.
 

```
[edit security ike]
user@spoke# set gateway gw-corporate external-interface ge-0/0/0.0
```
11. Define the IKE Phase 1 policy reference.
 

```
[edit security ike]
user@spoke# set gateway gw-corporate ike-policy ike-phase1-policy
```
12. Define the IKE Phase 1 gateway address.
 

```
[edit security ike]
user@spoke# set gateway gw-corporate address 1.1.1.2
```

**Results** From configuration mode, confirm your configuration by entering the **show security ike** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@spoke# show security ike
proposal ike-phase1-proposal {
  authentication-method pre-shared-keys;
  dh-group group2;
  authentication-algorithm sha1;
  encryption-algorithm aes-128-cbc;
}
policy ike-phase1-policy {
  mode main;
  proposals ike-phase1-proposal;
  pre-shared-key ascii-text "$9$9VMTp1RvWLdwYKMJDkmF3ylKM87Vb2oZjws5F"; ##
  SECRET-DATA
}
gateway gw-corporate {
  ike-policy ike-phase1-policy;
  address 1.1.1.2;
  external-interface ge-0/0/0.0;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

*Configuring IPsec for the Westford Spoke*

**CLI Quick Configuration** To quickly configure IPsec for the Westford spoke, copy the following commands and paste them into the CLI:

```
[edit]
set security ipsec proposal ipsec-phase2-proposal protocol esp
set security ipsec proposal ipsec-phase2-proposal authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec-phase2-proposal encryption-algorithm aes-128-cbc
set security ipsec policy ipsec-phase2-policy proposals ipsec-phase2-proposal
set security ipsec policy ipsec-phase2-policy perfect-forward-secrecy keys group2
set security ipsec vpn vpn-corporate ike gateway gw-corporate
set security ipsec vpn vpn-corporate ike ipsec-policy ipsec-phase2-policy
set security ipsec vpn vpn-corporate bind-interface st0.0
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure IPsec for the Westford spoke:

1. Create an IPsec Phase 2 proposal.

```
[edit]
user@spoke# set security ipsec proposal ipsec-phase2-proposal
```

2. Specify the IPsec Phase 2 proposal protocol.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@spoke# set protocol esp
```

3. Specify the IPsec Phase 2 proposal authentication algorithm.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@spoke# set authentication-algorithm hmac-sha1-96
```

4. Specify the IPsec Phase 2 proposal encryption algorithm.

```
[edit security ipsec proposal ipsec-phase2-proposal]
user@spoke# set encryption-algorithm aes-128-cbc
```

5. Create the IPsec Phase 2 policy.

```
[edit security ipsec]
user@spoke# set policy ipsec-phase2-policy
```

6. Specify the IPsec Phase 2 proposal reference.

```
[edit security ipsec policy ipsec-phase2-policy]
user@spoke# set proposals ipsec-phase2-proposal
```

7. Specify IPsec Phase 2 PFS to use Diffie-Hellman group 2.

```
[edit security ipsec policy ipsec-phase2-policy]
user@host# set perfect-forward-secrecy keys group2
```

8. Specify the IKE gateway.

```
[edit security ipsec]
user@spoke# set vpn vpn-corporate ike gateway gw-corporate
```

9. Specify the IPsec Phase 2 policy.

```
[edit security ipsec]
user@spoke# set vpn vpn-corporate ike ipsec-policy ipsec-phase2-policy
```

10. Specify the interface to bind.

```
[edit security ipsec]
user@spoke# set vpn vpn-corporate bind-interface st0.0
```

**Results** From configuration mode, confirm your configuration by entering the **show security ipsec** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@spoke# show security ipsec
proposal ipsec-phase2-proposal {
  protocol esp;
  authentication-algorithm hmac-sha1-96;
  encryption-algorithm aes-128-cbc;
}
policy ipsec-phase2-policy {
  perfect-forward-secrecy {
    keys group2;
  }
  proposals ipsec-phase2-proposal;
}
vpn vpn-corporate {
  bind-interface st0.0;
  ike {
    gateway gw-corporate;
    ipsec-policy ipsec-phase2-policy;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### *Configuring Security Policies for the Westford Spoke*

**CLI Quick Configuration** To quickly configure security policies for the Westford spoke, copy the following commands and paste them into the CLI:

```
[edit]
set security policies from-zone trust to-zone vpn policy to-corporate match source-address local-net
set security policies from-zone trust to-zone vpn policy to-corporate match destination-address corp-net
set security policies from-zone trust to-zone vpn policy to-corporate match destination-address sunnyvale-net
set security policies from-zone trust to-zone vpn policy to-corporate application any
set security policies from-zone trust to-zone vpn policy to-corporate then permit
set security policies from-zone vpn to-zone trust policy from-corporate match source-address corp-net
set security policies from-zone vpn to-zone trust policy from-corporate match source-address sunnyvale-net
```

```

set security policies from-zone vpn to-zone trust policy from-corporate match
  destination-address local-net
set security policies from-zone vpn to-zone trust policy from-corporate application any
set security policies from-zone vpn to-zone trust policy from-corporate then permit

```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure security policies for the Westford spoke:

1. Create the security policy to permit traffic from the trust zone to the vpn zone.

```

[edit security policies from-zone trust to-zone vpn]
user@spoke# set policy to-corp match source-address local-net
user@spoke# set policy to-corp match destination-address corp-net
user@spoke# set policy to-corp match destination-address sunnyvale-net
user@spoke# set policy to-corp match application any
user@spoke# set policy to-corp then permit

```

2. Create the security policy to permit traffic from the vpn zone to the trust zone.

```

[edit security policies from-zone vpn to-zone trust]
user@spoke# set policy spokes-to-local match source-address corp-net
user@spoke# set policy spokes-to-local match source-address sunnyvale-net
user@spoke# set policy spokes-to-local match destination-address local-net
user@spoke# set policy spokes-to-local match application any
user@spoke# set policy spokes-to-local then permit

```

**Results** From configuration mode, confirm your configuration by entering the `show security policies` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@spoke# show security policies
from-zone trust to-zone vpn {
  policy to-corp {
    match {
      source-address local-net;
      destination-address [ sunnyvale-net westford-net ];
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone vpn to-zone trust {
  policy spokes-to-local {
    match {
      source-address [ sunnyvale-net westford-net ];
      destination-address local-net;
      application any;
    }
    then {
      permit;
    }
  }
}

```



```

    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### *Configuring TCP-MSS for the Westford Spoke*

**CLI Quick Configuration** To quickly configure TCP-MSS for the Westford spoke, copy the following commands and paste them into the CLI:

```

[edit]
set security flow tcp-mss ipsec-vpn mss 1350

```

**Step-by-Step Procedure** To configure TCP-MSS for the Westford spoke:

1. Configure TCP-MSS information.

```

[edit]
user@spoke# set security flow tcp-mss ipsec-vpn mss 1350

```

**Results** From configuration mode, confirm your configuration by entering the **show security flow** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@spoke# show security flow
tcp-mss {
  ipsec-vpn {
    mss 1350;
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### *Configuring the Sunnyvale Spoke*

**CLI Quick Configuration** This example uses an SSG Series device for the Sunnyvale spoke. For reference, the configuration for the SSG Series device is provided. For information about configuring SSG Series devices, see the *Concepts and Examples ScreenOS Reference Guide*, which is located at <http://www.juniper.net/techpubs>.

To quickly configure the Sunnyvale spoke, copy the following commands and paste them into the CLI:

```

[edit]
set zone name "VPN"
set interface ethernet0/6 zone "Trust"
set interface "tunnel.1" zone "VPN"
set interface ethernet0/6 ip 192.168.168.1/24
set interface ethernet0/6 route
set interface ethernet0/0 ip 2.2.2.2/30
set interface ethernet0/0 route
set interface tunnel.1 ip 10.11.11.11/24
set flow tcp-mss 1350

```

```

set address "Trust" "sunnyvale-net" 192.168.168.0 255.255.255.0
set address "VPN" "corp-net" 10.10.10.0 255.255.255.0
set address "VPN" "westford-net" 192.168.178.0 255.255.255.0
set ike gateway "corp-ike" address 1.1.1.2 Main outgoing-interface ethernet0/0 preshare
  "395psksecr3t" sec-level standard
set vpn corp-vpn monitor optimized rekey
set vpn "corp-vpn" bind interface tunnel.1
set vpn "corp-vpn" gateway "corp-ike" replay tunnel idletime 0 sec-level standard
set policy id 1 from "Trust" to "Untrust" "ANY" "ANY" "ANY" nat src permit
set policy id 2 from "Trust" to "VPN" "sunnyvale-net" "corp-net" "ANY" permit
set policy id 2
exit
set dst-address "westford-net"
exit
set policy id 3 from "VPN" to "Trust" "corp-net" "sunnyvale-net" "ANY" permit
set policy id 3
set src-address "westford-net"
exit
set route 10.10.10.0/24 interface tunnel.1
set route 192.168.178.0/24 interface tunnel.1
set route 0.0.0.0/0 interface ethernet0/0 gateway 2.2.2.1

```

## Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying the IKE Phase 1 Status on page 532
- Verifying the IPsec Phase 2 Status on page 534
- Verifying Next-Hop Tunnel Bindings on page 535
- Verifying Static Routes for Remote Peer Local LANs on page 536
- Reviewing Statistics and Errors for an IPsec Security Association on page 536
- Testing Traffic Flow Across the VPN on page 537

### Verifying the IKE Phase 1 Status

**Purpose** Verify the IKE Phase 1 status.

**Action**



**NOTE:** Before starting the verification process, you need to send traffic from a host in the 10.10.10/24 network to a host in the 192.168.168/24 and 192.168.178/24 networks to bring the tunnels up. For route-based VPNs, you can send traffic initiated from the SRX Series device through the tunnel. We recommend that when testing IPsec tunnels, you send test traffic from a separate device on one side of the VPN to a second device on the other side of the VPN. For example, initiate a ping from 10.10.10.10 to 192.168.168.10.

From operational mode, enter the **show security ike security-associations** command. After obtaining an index number from the command, use the **show security ike security-associations index *index\_number* detail** command.

```
user@hub> show security ike security-associations
```

| Index | Remote Address | State | Initiator cookie | Responder cookie | Mode |
|-------|----------------|-------|------------------|------------------|------|
| 6     | 3.3.3.2        | UP    | 94906ae2263bbd8e | 1c35e4c3fc54d6d3 | Main |
| 7     | 2.2.2.2        | UP    | 7e7a1c0367dfe73c | f284221c656a5fbc | Main |

```

user@hub> show security ike security-associations index 6 detail
IKE peer 3.3.3.2, Index 6,
  Role: Responder, State: UP
  Initiator cookie: 94906ae2263bbd8e,, Responder cookie: 1c35e4c3fc54d6d3
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 1.1.1.2:500, Remote: 3.3.3.2:500
  Lifetime: Expires in 3571 seconds
  Algorithms:
    Authentication      : sha1
    Encryption          : aes-cbc (128 bits)
    Pseudo random function: hmac-sha1
  Traffic statistics:
    Input bytes      :          1128
    Output bytes     :           988
    Input packets   :             6
    Output packets  :             5
  Flags: Caller notification sent
  IPSec security associations: 1 created, 0 deleted
  Phase 2 negotiations in progress: 1
    Negotiation type: Quick mode, Role: Responder, Message ID: 1350777248
    Local: 1.1.1.2:500, Remote: 3.3.3.2:500
    Local identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
    Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
    Flags: Caller notification sent, Waiting for done

```

**Meaning** The `show security ike security-associations` command lists all active IKE Phase 1 SAs. If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the `show security ike security-associations index detail` command to get more information about the SA.
- Remote Address—Verify that the remote IP address is correct.
- State
  - UP—The Phase 1 SA has been established.
  - DOWN—There was a problem establishing the Phase 1 SA.
- Mode—Verify that the correct mode is being used.

Verify that the following information is correct in your configuration:

- External interfaces (the interface must be the one that receives IKE packets)
- IKE policy parameters
- Preshared key information
- Phase 1 proposal parameters (must match on both peers)

The **show security ike security-associations index 1 detail** command lists additional information about the security association with an index number of 1:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Initiator and responder role information



**NOTE:** Troubleshooting is best performed on the peer using the responder role.

- Number of IPsec SAs created
- Number of Phase 2 negotiations in progress

### Verifying the IPsec Phase 2 Status

**Purpose** Verify the IPsec Phase 2 status.

**Action** From operational mode, enter the **show security ipsec security-associations** command. After obtaining an index number from the command, use the **show security ipsec security-associations index *index\_number* detail** command.

```
user@hub> show security ipsec security-associations
total configured sa: 4
ID      Gateway      Port  Algorithm      SPI      Life:sec/kb  Mon vsys
-----
<16384  2.2.2.2      500   ESP:aes-128/sha1  b2fc36f8 3364/ unlim - 0
>16384  2.2.2.2      500   ESP:aes-128/sha1  5d73929e 3364/ unlim - 0
ID      Gateway      Port  Algorithm      SPI      Life:sec/kb  Mon vsys
-----
<16385  3.3.3.2      500   ESP:3des/sha1    70f789c6 28756/unlim - 0
>16385  3.3.3.2      500   ESP:3des/sha1    80f4126d 28756/unlim - 0
```

```
user@hub> show security ipsec security-associations index 16385 detail
Virtual-system: Root
Local Gateway: 1.1.1.2, Remote Gateway: 3.3.3.2
Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/24)
Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
DF-bit: clear
Direction: inbound, SPI: 1895270854, AUX-SPI: 0
Hard lifetime: Expires in 28729 seconds
Lifeseize Remaining: Unlimited
Soft lifetime: Expires in 28136 seconds
Mode: tunnel, Type: dynamic, State: installed, VPN Monitoring: -
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)

Anti-replay service: enabled, Replay window size: 32

Direction: outbound, SPI: 2163479149, AUX-SPI: 0
Hard lifetime: Expires in 28729 seconds
Lifeseize Remaining: Unlimited
Soft lifetime: Expires in 28136 seconds
```

```
Mode: tunnel, Type: dynamic, State: installed, VPN Monitoring: -
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
```

```
Anti-replay service: enabled, Replay window size: 32
```

**Meaning** The output from the `show security ipsec security-associations` command lists the following information:

- The ID number is 16385. Use this value with the `show security ipsec security-associations index` command to get more information about this particular SA.
- There is one IPsec SA pair using port 500, which indicates that no NAT-traversal is implemented. (NAT-traversal uses port 4500 or another random high-number port.)
- The SPIs, lifetime (in seconds), and usage limits (or lifesize in KB) are shown for both directions. The 28756/ unlim value indicates that the Phase 2 lifetime expires in 28756 seconds, and that no lifesize has been specified, which indicates that it is unlimited. Phase 2 lifetime can differ from Phase 1 lifetime, as Phase 2 is not dependent on Phase 1 after the VPN is up.
- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U indicates that monitoring is up, and D indicates that monitoring is down.
- The virtual system (vsys) is the root system, and it always lists 0.

The output from the `show security ipsec security-associations index 16385 detail` command lists the following information:

- The local identity and remote identity make up the proxy ID for the SA.  
A proxy ID mismatch is one of the most common causes for a Phase 2 failure. If no IPsec SA is listed, confirm that Phase 2 proposals, including the proxy ID settings, are correct for both peers. For route-based VPNs, the default proxy ID is local=0.0.0.0/0, remote=0.0.0.0/0, and service=any. Issues can occur with multiple route-based VPNs from the same peer IP. In this case, a unique proxy ID for each IPsec SA must be specified. For some third-party vendors, the proxy ID must be manually entered to match.
- Another common reason for Phase 2 failure is not specifying the ST interface binding. If IPsec cannot complete, check the kmd log or set traceoptions.

### *Verifying Next-Hop Tunnel Bindings*

**Purpose** After Phase 2 is complete for all peers, verify the next-hop tunnel bindings.

**Action** From operational mode, enter the `show security ipsec next-hop-tunnels` command.

```
user@hub> show security ipsec next-hop-tunnels
Next-hop gateway  interface  IPsec VPN name  Flag
10.11.11.11       st0.0      sunnyvale-vpn   Static
10.11.11.12       st0.0      westford-vpn    Auto
```

**Meaning** The next-hop gateways are the IP addresses for the st0 interfaces of all remote spoke peers. The next hop should be associated with the correct IPsec VPN name. If no NHTB

entry exists, there is no way for the hub device to differentiate which IPsec VPN is associated with which next hop.

The Flag field has one of the following values:

- **Static**— NHTB was manually configured in the st0.0 interface configurations, which is required if the peer is not an SRX Series device.
- **Auto**— NHTB was not configured, but the entry was automatically populated into the NHTB table during Phase 2 negotiations between two SRX Series devices

There is no NHTB table for any of the spoke sites in this example. From the spoke perspective, the st0 interface is still a point-to-point link with only one IPsec VPN binding.

### *Verifying Static Routes for Remote Peer Local LANs*

**Purpose** Verify that the static route references the spoke peer's st0 IP address.

**Action** From operational mode, enter the **show route** command.

```
user@hub> show route 192.168.168.10
inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
192.168.168.0/24    *[Static/5] 00:08:33
                  > to 10.11.11.11 via st0.0
```

```
user@hub> show route 192.168.178.10
inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
192.168.178.0/24    *[Static/5] 00:04:04
                  > to 10.11.11.12 via st0.0
```

The next hop is the remote peer's st0 IP address, and both routes point to st0.0 as the outgoing interface.

### *Reviewing Statistics and Errors for an IPsec Security Association*

**Purpose** Review ESP and authentication header counters and errors for an IPsec security association.

**Action** From operational mode, enter the **show security ipsec statistics index** command.

```
user@hub> show security ipsec statistics index 16385
ESP Statistics:
  Encrypted bytes:          920
  Decrypted bytes:         6208
  Encrypted packets:        5
  Decrypted packets:       87
AH Statistics:
  Input bytes:              0
  Output bytes:             0
  Input packets:           0
  Output packets:          0
Errors:
  AH authentication failures: 0, Replay errors: 0
```

```
ESP authentication failures: 0, ESP decryption failures: 0
Bad headers: 0, Bad trailers: 0
```

You can also use the **show security ipsec statistics** command to review statistics and errors for all SAs.

To clear all IPsec statistics, use the **clear security ipsec statistics** command.

**Meaning** If you see packet loss issues across a VPN, you can run the **show security ipsec statistics** or **show security ipsec statistics detail** command several times to confirm that the encrypted and decrypted packet counters are incrementing. You should also check whether the other error counters are incrementing.

### *Testing Traffic Flow Across the VPN*

**Purpose** Verify the traffic flow across the VPN.

**Action** You can use the **ping** command from the SRX Series device to test traffic flow to a remote host PC. Make sure that you specify the source interface so that the route lookup is correct and the appropriate security zones are referenced during policy lookup.

From operational mode, enter the **ping** command.

```
user@hub> ping 192.168.168.10 interface ge-0/0/0 count 5
PING 192.168.168.10 (192.168.168.10): 56 data bytes
64 bytes from 192.168.168.10: icmp_seq=0 ttl=127 time=8.287 ms
64 bytes from 192.168.168.10: icmp_seq=1 ttl=127 time=4.119 ms
64 bytes from 192.168.168.10: icmp_seq=2 ttl=127 time=5.399 ms
64 bytes from 192.168.168.10: icmp_seq=3 ttl=127 time=4.361 ms
64 bytes from 192.168.168.10: icmp_seq=4 ttl=127 time=5.137 ms

--- 192.168.168.10 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.119/5.461/8.287/1.490 ms
```

You can also use the **ping** command from the SSG Series device.

```
user@hub> ping 10.10.10.10 from ethernet0/6
Type escape sequence to abort
Sending 5, 100-byte ICMP Echos to 10.10.10.10, timeout is 1 seconds from
ethernet0/6
!!!!
Success Rate is 100 percent (5/5), round-trip time min/avg/max=4/4/5 ms

ssg-> ping 192.168.178.10 from ethernet0/6
Type escape sequence to abort
Sending 5, 100-byte ICMP Echos to 192.168.178.10, timeout is 1 seconds from
ethernet0/6
!!!!
Success Rate is 100 percent (5/5), round-trip time min/avg/max=8/8/10 ms
```

**Meaning** If the **ping** command fails from the SRX Series or SSG Series device, there might be a problem with the routing, security policies, end host, or encryption and decryption of ESP packets.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding Hub-and-Spoke VPNs on page 506](#)
  - [Example: Configuring a Route-Based VPN on page 470](#)
  - [Example: Configuring a Policy-Based VPN on page 489](#)

## Configuring IPsec VPN Using the VPN Wizard

---

The VPN Wizard enables you to perform basic IPsec VPN configuration, including both Phase 1 and Phase 2. For more advanced configuration, use the J-Web interface or the CLI.

To configure IPsec VPN using the VPN Wizard:

1. Select **Configure>Wizards>VPN Wizard** in the J-Web interface.
2. Click the Launch VPN Wizard button.
3. Follow the wizard prompts.

The upper left area of the wizard page shows where you are in the configuration process. The lower left area of the page shows field-sensitive help. When you click a link under the Resources heading, the document opens in your browser. If the document opens in a new tab, be sure to close only the tab (not the browser window) when you close the document.

- Related Documentation**
- [VPN Overview on page 451](#)
  - [Understanding Phase 1 of IKE Tunnel Negotiation on page 467](#)
  - [Understanding Phase 2 of IKE Tunnel Negotiation on page 468](#)

## Understanding IPv6 IKE and IPsec Packet Processing

---

An IPv6 IPsec VPN implementation involves the exchange of IPv6 packets within an IPv6 tunnel set up between two IPv6 tunnel endpoints. (See “VPN Overview” on page 451.)

This topic includes the following sections:

- [Packet Processing in IPv6 6in6 Tunnel Mode on page 538](#)
- [IPv6 IKE Packet Processing on page 539](#)
- [IPv6 IPsec Packet Processing on page 540](#)

### Packet Processing in IPv6 6in6 Tunnel Mode

IPv6 VPN 6in6 tunneling is a technique for exchanging IPv6 packets within an IPv6 IPsec tunnel between two site-to-site endpoints. In this mode, the original IPv6 packet is encapsulated inside another IPv6 packet where both the outer and inner headers are IPv6. The IPv6 addresses of the outer IPv6 header represent the tunnel endpoints, while the IPv6 addresses of the inner IPv6 header represent the final source and destination



addresses. Unlike the transport mode, where the original IP header is retained, in the 6in6 tunneling mode, the entire original IPv6 packet (payload and header) is encapsulated by appending a new outer IPv6 header, IPsec headers (AH or ESP), followed by the inner IPv6 header, and the original IPv6 payload. The entire original IPv6 packet can be encrypted, authenticated, or both. The Authentication Header (AH) protocol provides authentication, while the Encapsulation Security Payload (ESP) protocol provides encryption as well as authentication for the IPv6 packets.

## IPv6 IKE Packet Processing

Internet Key Exchange (IKE) is part of the IPsec suite of protocols. It automatically enables two tunnel endpoints to set up security associations (SAs) and negotiate secret keys with each other. There is no need to manually configure the security parameters. IKE also provides authentication for communicating peers.

IKE packet processing in IPv6 networks involves the following elements:

- ISAKMP Identification Payload

Internet Security Association and Key Management Protocol (ISAKMP) identification payload is used to identify and authenticate the communicating IPv6 peers. Two new ID types—ID\_IPV6\_ADDR and ID\_IPV6\_ADDR\_SUBNET—are enabled for IPv6. The ID type indicates the type of identification to be used. The ID\_IPV6\_ADDR type specifies a single 16-octet IPv6 address. This ID type represents an IPv6 address. The ID\_IPV6\_ADDR\_SUBNET type specifies a range of IPv6 addresses represented by two 16-octet values. This ID type represents an IPv6 network mask. Table 56 on page 539 lists the ID types and their assigned values in the identification payload.

**Table 56: ISAKMP ID Types and Their Values**

| ID Type             | Value |
|---------------------|-------|
| RESERVED            | 0     |
| ID_IPV4_ADDR        | 1     |
| ID_FQDN             | 2     |
| ID_USER_FQDN        | 3     |
| ID_IPV4_ADDR_SUBNET | 4     |
| ID_IPV6_ADDR        | 5     |
| ID_IPV6_ADDR_SUBNET | 6     |
| ID_IPV4_ADDR_RANGE  | 7     |
| ID_IPV6_ADDR_RANGE  | 8     |
| ID_DER_ASN1_DN      | 9     |

Table 56: ISAKMP ID Types and Their Values (*continued*)

| ID Type        | Value |
|----------------|-------|
| ID_DER_ASN1_GN | 10    |
| ID_KEY_ID      | 11    |
| ID_LIST        | 12    |

The ID\_IPV6\_ADDR\_RANGE type specifies a range of IPv6 addresses represented by two 16-octet values. The first octet value represents the starting IPv6 address and the second octet value represents the ending IPv6 address in the range. All IPv6 addresses falling between the first and last IPv6 addresses are considered to be part of the list.



**NOTE:** Two ID types in ISAKMP identification payload—ID\_IPV6\_ADDR\_RANGE and ID\_IPV4\_ADDR\_RANGE—are not supported in this release.

- Proxy ID

A proxy ID is used during Phase 2 of IKE negotiation. It is generated before an IPsec tunnel is established. A proxy ID identifies the SA to be used for the VPN. Two proxy IDs are generated—local and remote. The local proxy ID refers to the local IPv6 address/network and subnet mask. The remote proxy ID refers to the remote IPv6 address/network and subnet mask.

- Security Association

An SA is an agreement between VPN participants to support secure communication. SAs are differentiated based on three parameters—security parameter index (SPI), destination IPv6 address, and security protocol (either AH or ESP). The SPI is a unique value assigned to an SA to help identify an SA among multiple SAs. In an IPv6 packet, the SA is identified from the destination address in the outer IPv6 header and the security protocol is identified from either the AH or the ESP header.

## IPv6 IPsec Packet Processing

After IKE negotiations are completed and the two IKE gateways have established Phase 1 and Phase 2 security associations (SAs), IPv6 IPsec employs authentication and encryption technologies to secure the IPv6 packets.

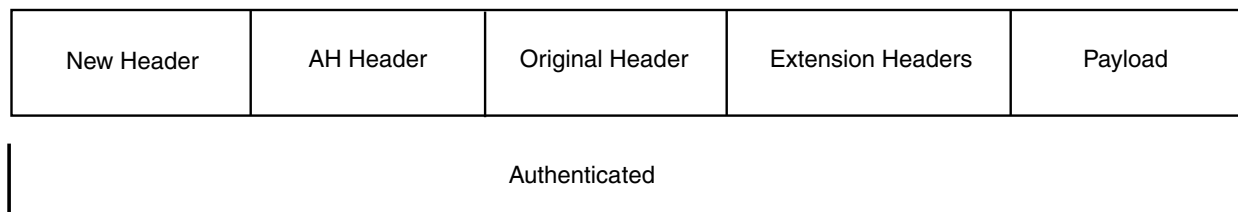
This topic includes the following sections:

- AH Protocol in IPv6 on page 541
- ESP Protocol in IPv6 on page 541
- Integrity Check Value (ICV) Calculation in IPv6 on page 541
- Header Construction in IPv6 Tunnel Mode on page 542

### AH Protocol in IPv6

The AH protocol provides data integrity and data authentication for IPv6 packets. IPv6 IPsec uses extension headers (for example, hop-by-hop and routing options) that must be arranged in a particular way in the IPv6 datagram. In IPv6 AH tunnel mode, the AH header immediately follows the new outer IPv6 header similar to that in IPv4 AH tunnel mode. The extension headers are placed after the original inner IPv6 header. Therefore, in IPv6 AH tunnel mode, the entire IPv6 packet is encapsulated by adding a new outer IPv6 header, followed by an authentication header, an inner IPv6 header, extension headers, and the rest of the original IPv6 datagram as shown in Figure 52 on page 541.

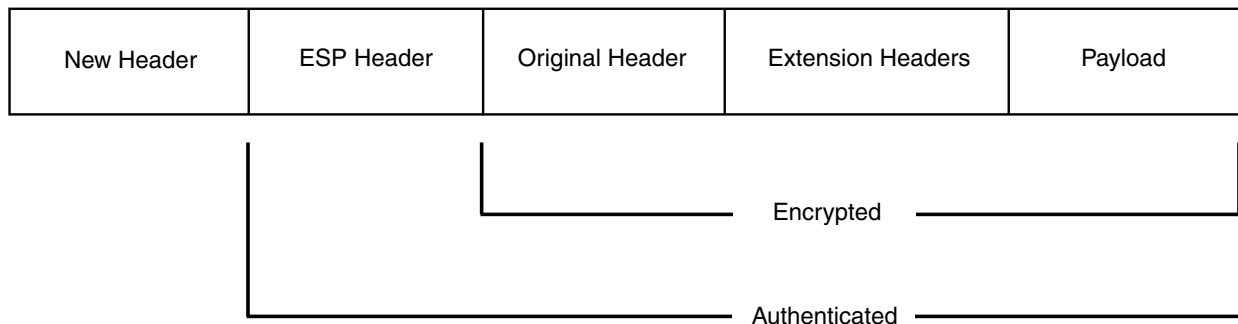
Figure 52: IPv6 AH Tunnel Mode



### ESP Protocol in IPv6

ESP protocol provides both encryption and authentication for IPv6 packets. Because IPv6 IPsec uses extension headers (for example, hop-by-hop and routing options) in the IPv6 datagram, the most important difference between IPv6 ESP tunnel mode and IPv4 ESP tunnel mode is the placement of extension headers in the packet layout. In IPv6 ESP tunnel mode, the ESP header immediately follows the new outer IPv6 header similar to that in IPv4 ESP tunnel mode. Therefore, in IPv6 ESP tunnel mode, the entire IPv6 packet is encapsulated by adding a new outer IPv6 header, followed by an ESP header, an inner IPv6 header, extension headers, and the rest of the original IPv6 datagram as shown in Figure 53 on page 541.

Figure 53: IPv6 ESP Tunnel Mode



### Integrity Check Value (ICV) Calculation in IPv6

AH protocol verifies the integrity of the IPv6 packet by computing an Integrity Check Value (ICV) on the packet contents. ICV is usually built over an authentication algorithm such as MD5 or SHA-1. The IPv6 ICV calculations differ from that in IPv4 in terms of two header fields—mutable header and optional extension header.

You can calculate the AH ICV over the IPv6 header fields that are either immutable in transit or predictable in value upon arrival at the tunnel endpoints. You can also calculate the AH ICV over the AH header and the upper level protocol data (considered to be immutable in transit). You can calculate the ESP ICV over the entire IPv6 packet, excluding the new outer IPv6 header and the optional extension headers.



**NOTE:** Unlike IPv4, IPv6 has a method for tagging options as mutable in transit. IPv6 optional extension headers contain a flag that indicates mutability. This flag determines the appropriate processing.

### Header Construction in IPv6 Tunnel Mode

In IPv6 tunnel mode, the source and destination addresses of the outer IPv6 header represent the tunnel endpoints, while the source and destination addresses of the inner IPv6 header represent the final source and destination addresses. Table 57 on page 542 summarizes the differences between the outer IPv6 header and the inner IPv6 header.

**Table 57: Comparison Between Outer Headers and Inner Headers**

| Header Fields     | Outer Header                  | Inner Header |
|-------------------|-------------------------------|--------------|
| version           | 6                             | No change.   |
| DS field          | Copied from the inner header. | No change.   |
| ECN field         | Copied from the inner header. | Constructed. |
| flow label        | Copied from the inner header. | No change.   |
| payload length    | Constructed.                  | No change.   |
| next header       | AH, ESP, and routing header.  | No change.   |
| hop limit         | 64.                           | Decrement.   |
| src address       | Constructed.                  | No change.   |
| dest address      | Constructed.                  | No change.   |
| extension headers | Never copied.                 | No change.   |



**NOTE:** This release supports IPv6 6in6 site-to-site VPN only. The IPv6 6in6 site-to-site VPN uses IPv6 address as the IKE identity in this release.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- VPN Overview on page 451

- IPv6 IPsec Configuration Overview on page 543
- Example: Configuring an IPv6 IPsec Manual VPN on page 543
- Example: Configuring an IPv6 AutoKey IKE Policy-Based VPN on page 546

## IPv6 IPsec Configuration Overview

---

Juniper Networks supports two types of IPv6 IPsec VPN configurations—Manual and AutoKey IKE with preshared keys.

- Manual VPN—In a Manual VPN configuration, the secret keys and security associations (SAs) are manually configured on the tunnel endpoints using the Manual key mechanism. To create an IPv6 IPsec Manual VPN, see “Example: Configuring an IPv6 IPsec Manual VPN” on page 543.
- AutoKey IKE VPN—In an AutoKey IKE VPN configuration, the secret keys and SAs are automatically created using the AutoKey IKE mechanism. To set up an IPv6 AutoKey IKE VPN, two phases of negotiations are required—Phase 1 and Phase 2.
  - Phase 1—In this phase, the participants establish a secure channel for negotiating the IPsec SAs. For more information on Phase 1 negotiations, see “Understanding Phase 1 of IKE Tunnel Negotiation” on page 467.
  - Phase 2—In this phase, the participants negotiate the IPsec SAs for authenticating and encrypting the IPv6 data packets. For more information on Phase 2 negotiations, see “Understanding Phase 2 of IKE Tunnel Negotiation” on page 468.

To create an IPv6 AutoKey IKE policy-based VPN, see “Example: Configuring an IPv6 AutoKey IKE Policy-Based VPN” on page 546.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding IPv6 IKE and IPsec Packet Processing on page 538
- Example: Configuring an IPv6 IPsec Manual VPN on page 543
- Example: Configuring an IPv6 AutoKey IKE Policy-Based VPN on page 546

## Example: Configuring an IPv6 IPsec Manual VPN

---

This example shows how to configure an IPv6 IPsec Manual VPN.

- Requirements on page 544
- Overview on page 544
- Configuration on page 544
- Verification on page 545

## Requirements

Before you begin:

- Understand how VPNs work. See “VPN Overview” on page 451.
- Understand IPv6 IPsec packet processing. See “Understanding IPv6 IKE and IPsec Packet Processing” on page 538.

## Overview

In a Manual VPN configuration, the secret keys are manually configured on the two IPsec endpoints.

In this example, you:

- Configure the authentication parameters for a VPN named vpn-sunnyvale.
- Configure the encryption parameters for vpn-sunnyvale.
- Specify the outgoing interface for the SA.
- Specify the IPv6 address of the peer.
- Define the IPsec protocol. Select the ESP protocol because the configuration includes both authentication and encryption.
- Configure a security parameter index (SPI).

## Configuration

**CLI Quick Configuration** To quickly configure security algorithms, copy the following commands and paste them into the CLI.

```
[edit]
set security ipsec vpn vpn-sunnyvale manual authentication algorithm hmac-md5-96
  key ascii-text 1111111111111111
set security ipsec vpn vpn-sunnyvale manual encryption algorithm 3des-cbc key ascii-text
  111111111111111111111111
set security ipsec vpn vpn-sunnyvale manual external-interface ge-0/0/14.0
set security ipsec vpn vpn-sunnyvale manual gateway 1212::1112
set security ipsec vpn vpn-sunnyvale manual protocol esp
set security ipsec vpn vpn-sunnyvale manual spi 12435
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode.

To configure security algorithms:

1. Configure the authentication parameters.

```
[edit security ipsec vpn vpn-sunnyvale manual]
user@host# set authentication algorithm hmac-md5-96 key ascii-text 1111111111111111
```

2. Configure the encryption parameters.

- ```
[edit security ipsec vpn vpn-sunnyvale manual]
user@host# set encryption algorithm 3des-cbc key ascii-text 11111111111111111111
```
3. Specify the outgoing interface for the SA.

```
[edit security ipsec vpn vpn-sunnyvale manual]
user@host# set external-interface ge-0/0/14.0
```
  4. Specify the IPv6 address of the peer.

```
[edit security ipsec vpn vpn-sunnyvale manual]
user@host# set gateway 1212::1112
```
  5. Define the IPsec protocol.

```
[edit security ipsec vpn vpn-sunnyvale manual]
user@host# set protocol esp
```
  6. Configure an SPI.

```
[edit security ipsec vpn vpn-sunnyvale manual]
user@host# set spi 12435
```

**Results** From configuration mode, confirm your configuration by entering the **show security ipsec vpn vpn-sunnyvale** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
[user@host]show security ipsec vpn vpn-sunnyvale
manual {
  gateway 1212::1112 ;
  external-interface ge-0/0/14.0 ;
  protocol esp ;
  spi 12435 ;
  authentication {
    algorithm hmac-md5-96 ;
    key ascii-text $9$P5369Ap01R3nSreK8LZUDimfTz36CtmP01REyrs2goUjHqm" ;##
    SECRET DATA
  }
  encryption {
    algorithm 3des-cbc ;
    key ascii-text $9$DRimfTz36tmP01REyrs2goUjHqmQFUD/CtpB1xN-V24aZU"; ##
    SECRET DATA
  }
}
```

## Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Security Algorithms on page 545

### Verifying Security Algorithms

**Purpose** Determine if security algorithms are applied or not.

**Action** From operational mode, enter the **show security ipsec security-associations** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding IPv6 IKE and IPsec Packet Processing](#) on page 538
  - [IPv6 IPsec Configuration Overview](#) on page 543
  - [Example: Configuring an IPv6 AutoKey IKE Policy-Based VPN](#) on page 546

## Example: Configuring an IPv6 AutoKey IKE Policy-Based VPN

---

This example shows how to configure a policy-based IPv6 AutoKey IKE VPN to allow IPv6 data to be securely transferred between the branch office and the corporate office.

- [Requirements](#) on page 546
- [Overview](#) on page 546
- [Configuration](#) on page 550
- [Verification](#) on page 558

### Requirements

This example uses the following hardware:

- SRX240 device

Before you begin:

- Understand how VPNs work. See “VPN Overview” on page 451.
- Understand IPv6 IKE and IPsec packet processing. See “Understanding IPv6 IKE and IPsec Packet Processing” on page 538.

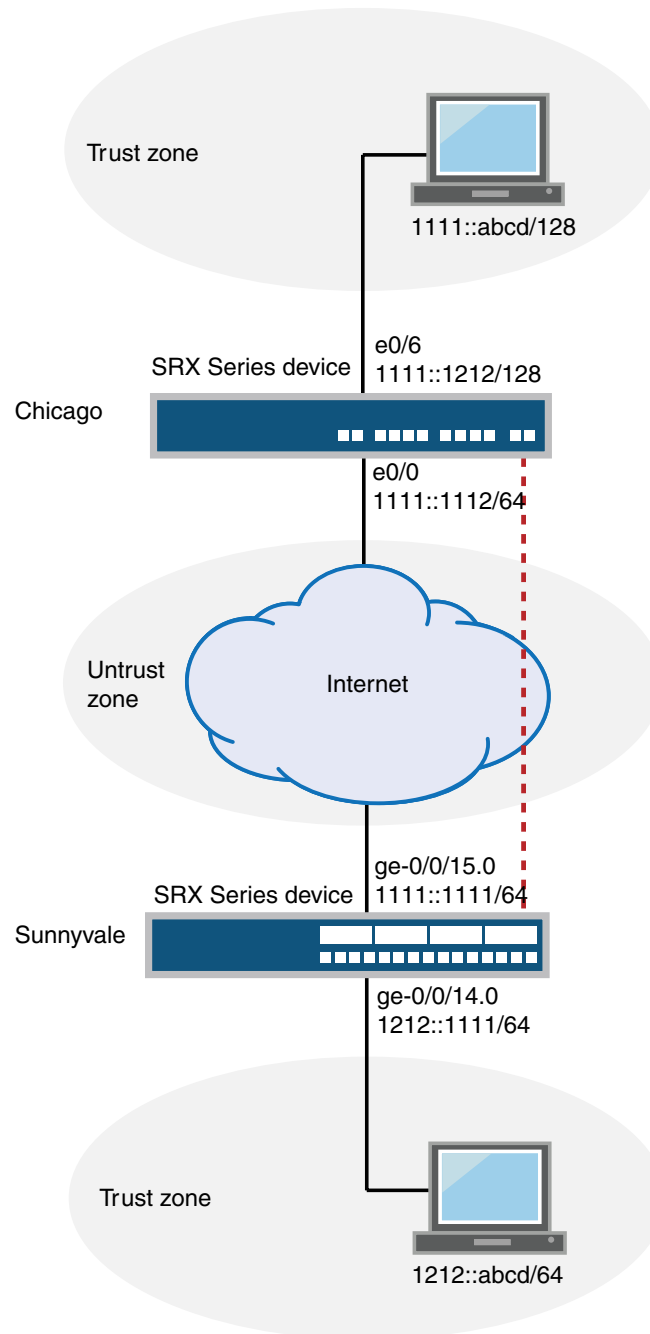
### Overview

In this example, you configure an IPv6 IKE policy-based VPN for a branch office in Chicago, Illinois, because you do not need to conserve tunnel resources or configure many security policies to filter traffic through the tunnel. Users in the Chicago office will use the VPN to connect to their corporate headquarters in Sunnyvale, California.

Figure 54 on page 547 shows an example of an IPv6 IKE policy-based VPN topology. In this topology, one SRX Series device is located in Sunnyvale, and another SRX Series device (this can be a second SRX Series device or a third-party device) is located in Chicago.



Figure 54: IPv6 IKE Policy-Based VPN Topology



In this example, you configure interfaces, an IPv6 default route, security zones, and address books. Then you configure IKE Phase 1, IPsec Phase 2, a security policy, and TCP-MSS parameters. See Table 58 on page 548 through Table 62 on page 550.

Table 58: Interface, Security Zone, and Address Book Information

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/14.0	1212::1111/64
	ge-0/0/15.0	1111::1111/64
Security zones	trust	<ul style="list-style-type: none"> <li>All system services are allowed.</li> <li>The ge-0/0/14.0 interface is bound to this zone.</li> </ul>
	untrust	<ul style="list-style-type: none"> <li>IKE is the only allowed system service.</li> <li>The ge-0/0/15.0 interface is bound to this zone.</li> </ul>
Address book entries	sunnyvale	<ul style="list-style-type: none"> <li>This address is for the trust zone's address book.</li> <li>The address for this address book entry is 1212::abcd/64.</li> </ul>
	chicago	<ul style="list-style-type: none"> <li>This address is for the untrust zone's address book.</li> <li>The address for this address book entry is 1111::abcd/128.</li> </ul>

Table 59: IPv6 IKE Phase 1 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ipv6-ike-phase1-proposal	<ul style="list-style-type: none"> <li>Authentication method: pre-shared-keys</li> <li>Diffie-Hellman group: group2</li> <li>Authentication algorithm: sha1</li> <li>Encryption algorithm: aes-128-cbc</li> </ul>
Policy	ipv6-ike-phase1-policy	<ul style="list-style-type: none"> <li>Mode: Aggressive</li> <li>Proposal reference: ipv6-ike-phase1-proposal</li> <li>IKE Phase 1 policy authentication method: pre-shared-key ascii-text</li> </ul>
Gateway	gw-chicago	<ul style="list-style-type: none"> <li>IKE policy reference: ipv6-ike-phase1-policy</li> <li>External interface: ge-0/0/15.0</li> <li>Gateway address: 1111::1112/64</li> </ul>

Table 60: IPv6 IPsec Phase 2 Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	ipv6-ipsec-phase2-proposal	<ul style="list-style-type: none"> <li>Protocol: esp</li> <li>Authentication algorithm: hmac-sha1-96</li> <li>Encryption algorithm: aes-128-cbc</li> </ul>

Table 60: IPv6 IPsec Phase 2 Configuration Parameters (*continued*)

Feature	Name	Configuration Parameters
Policy	ipv6-ipsec-phase2-policy	<ul style="list-style-type: none"> <li>Proposal reference: ipv6-ipsec-phase2-proposal</li> <li>PFS: Diffie-Hellman group2</li> </ul>
VPN	ipv6-ike-vpn-chicago	<ul style="list-style-type: none"> <li>IKE gateway reference: gw-chicago</li> <li>IPsec policy reference: ipv6-ipsec-phase2-policy</li> </ul>

Table 61: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters
This security policy permits traffic from the trust zone to the untrust zone.	ipv6-vpn-tr-untr	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address sunnyvale</li> <li>destination-address chicago</li> <li>application any</li> </ul> </li> <li>Permit action: tunnel ipsec-vpn ipv6-ike-vpn-chicago</li> <li>Permit action: tunnel pair-policy ipv6-vpn-untr-tr</li> </ul>
This security policy permits traffic from the untrust zone to the trust zone.	ipv6-vpn-untr-tr	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address chicago</li> <li>destination-address sunnyvale</li> <li>application any</li> </ul> </li> <li>Permit action: tunnel ipsec-vpn ipv6-ike-vpn-chicago</li> <li>Permit action: tunnel pair-policy ipv6-vpn-tr-untr</li> </ul>
<p>This security policy permits all traffic from the trust zone to the untrust zone.</p> <p><b>NOTE:</b> You must put the ipv6-vpn-tr-untr policy before the permit-any security policy. Junos OS performs a security policy lookup starting at the top of the list. If the permit-any policy comes before the ipv6-vpn-tr-untr policy, all traffic from the trust zone will match the permit-any policy and be permitted. Thus, no traffic will ever match the ipv6-vpn-tr-untr policy.</p>	permit-any	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address any</li> <li>source-destination any</li> <li>application any</li> </ul> </li> <li>Action: permit</li> </ul>

Table 62: TCP-MSS Configuration Parameters

Purpose	Configuration Parameters
<p>TCP-MSS is negotiated as part of the TCP three-way handshake and limits the maximum size of a TCP segment to better fit the MTU limits on a network. This is especially important for VPN traffic, as the IPsec encapsulation overhead, along with the IP and frame overhead, can cause the resulting ESP packet to exceed the MTU of the physical interface, thus causing fragmentation. Fragmentation results in increased use of bandwidth and device resources.</p> <p><b>NOTE:</b> We recommend a value of 1350 as the starting point for most Ethernet-based networks with an MTU of 1500 or greater. You might need to experiment with different TCP-MSS values to obtain optimal performance. For example, you might need to change the value if any device in the path has a lower MTU, or if there is any additional overhead such as PPP or Frame Relay.</p>	MSS value: 1350

## Configuration

### Configuring Basic Network, Security Zone, and Address Book Information

**CLI Quick Configuration** To quickly configure basic network, security zone, and address book information, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/14 unit 0 family inet6 address 1212::1111/64
set interfaces ge-0/0/15 unit 0 family inet6 address 1111::1111/64
set routing-options static route 0.0.0.0/0 next-hop 1.1.1.1
set security zones security-zone untrust interfaces ge-0/0/15.0
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust address-book address chicago 1111::abcd/128
set security zones security-zone trust interfaces ge-0/0/14.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust address-book address sunnyvale 1212::abcd/64
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure basic network, security zone, and address book information:

1. Configure Ethernet interface information.

```
[edit]
user@host# set interfaces ge-0/0/14 unit 0 family inet6 address 1212::1111/64
user@host# set interfaces ge-0/0/15 unit 0 family inet6 address 1111::1111/64
```

2. Configure static route information.

```
[edit]
user@host# set routing-options static route 0.0.0.0/0 next-hop 1.1.1.1
```

3. Configure the untrust security zone.

```
[edit]
user@host# edit security zones security-zone untrust
```

4. Assign an interface to the untrust security zone.

```
[edit security zones security-zone untrust]
```

- ```
user@host# set interfaces ge-0/0/15.0
```
5. Specify allowed system services for the untrust security zone.
 

```
[edit security zones security-zone untrust]
user@host# set host-inbound-traffic system-services ike
```
  6. Configure the address book entry for the untrust security zone.
 

```
[edit security zones security-zone untrust]
user@host# set address-book address chicago 1111::abcd/128
```
  7. Configure the trust security zone.
 

```
[edit]
user@host# edit security zones security-zone trust
```
  8. Assign an interface to the trust security zone.
 

```
[edit security zones security-zone trust]
user@host# set interfaces ge-0/0/14.0
```
  9. Specify allowed system services for the trust security zone.
 

```
[edit security zones security-zone trust]
user@host# set host-inbound-traffic system-services all
```
  10. Configure the address book entry for the trust security zone.
 

```
[edit security zones security-zone trust]
user@host# set address-book address sunnyvale 1212::abcd/64
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show routing-options**, and **show security zones** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/14 {
  unit 0 {
    family inet6 {
      address 1212::1111/64;
    }
  }
}
ge-0/0/15 {
  unit 0 {
    family inet6 {
      address 1111::1111/64;
    }
  }
}
[edit]
user@host# show routing-options
static {
  route 0.0.0.0/0 next-hop 1.1.1.1;
}
```

```
[edit]
user@host# show security zones
security-zone untrust {
  address-book {
    address chicago 1111::abcd/128{
  }
  host-inbound-traffic {
    system-services {
      ike;
    }
  }
  interfaces {
    ge-0/0/15.0;
  }
}
security-zone trust {
  address-book {
    address sunnyvale 1212::abcd/64{
  }
  host-inbound-traffic {
    system-services {
      all;
    }
  }
  interfaces {
    ge-0/0/14.0;
  }
}
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring IKE

#### CLI Quick Configuration

To quickly configure IKE, copy the following commands and paste them into the CLI:

```
[edit]
set security ike proposal ipv6-ike-phase1-proposal authentication-method pre-shared-keys
set security ike proposal ipv6-ike-phase1-proposal dh-group group2
set security ike proposal ipv6-ike-phase1-proposal authentication-algorithm sha1
set security ike proposal ipv6-ike-phase1-proposal encryption-algorithm aes-128-cbc
set security ike policy ipv6-ike-phase1-policy mode aggressive
set security ike policy ipv6-ike-phase1-policy proposals ipv6-ike-phase1-proposal
set security ike policy ipv6-ike-phase1-policy pre-shared-key ascii-text 1111111111111111
set security ike gateway gw-chicago external-interface ge-0/0/15.0
set security ike gateway gw-chicago ike-policy ipv6-ike-phase1-policy
set security ike gateway gw-chicago address 1111::1112/64
```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure IKE:

1. Create the IKE Phase 1 proposal.

```
[edit security ike]
```

- ```

user@host# set proposal ipv6-ike-phase1-proposal

```
2. Define the IKE proposal authentication method.

```

[edit security ike proposal ipv6-ike-phase1-proposal]
user@host# set authentication-method pre-shared-keys

```
  3. Define the IKE proposal Diffie-Hellman group.

```

[edit security ike proposal ipv6-ike-phase1-proposal]
user@host# set dh-group group2

```
  4. Define the IKE proposal authentication algorithm.

```

[edit security ike proposal ipv6-ike-phase1-proposal]
user@host# set authentication-algorithm sha1

```
  5. Define the IKE proposal encryption algorithm.

```

[edit security ike proposal ipv6-ike-phase1-proposal]
user@host# set encryption-algorithm aes-128-cbc

```
  6. Create an IKE Phase 1 policy.

```

[edit security ike]
user@host# set policy ipv6-ike-phase1-policy

```
  7. Set the IKE Phase 1 policy mode.

```

[edit security ike policy ipv6-ike-phase1-policy]
user@host# set mode aggressive

```
  8. Specify a reference to the IKE proposal.

```

[edit security ike policy ipv6-ike-phase1-policy]
user@host# set proposals ipv6-ike-phase1-proposal

```
  9. Define the IKE Phase 1 policy authentication method.

```

[edit security ike policy ipv6-ike-phase1-policy]
user@host# set pre-shared-key ascii-text 1111111111111111

```
  10. Create an IKE Phase 1 gateway and define its external interface.

```

[edit security ike]
user@host# set gateway gw-chicago external-interface ge-0/0/15.0

```
  11. Define the IKE Phase 1 policy reference.

```

[edit security ike gateway gw-chicago]
user@host# set ike-policy ipv6-ike-phase1-policy

```
  12. Assign an IP address to the IKE Phase 1 gateway.

```

[edit security ike gateway gw-chicago]
user@host# set address 1111::1112

```

**Results** From configuration mode, confirm your configuration by entering the **show security ike** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show security ike

```

```

proposal ipv6-ike-phase1-proposal {
  authentication-method pre-shared-keys;
  dh-group group2;
  authentication-algorithm sha1;
  encryption-algorithm aes-128-cbc;
}
policy ipv6-ike-phase1-policy {
  mode ;
  proposals ipv6-ike-phase1-proposal;
  pre-shared-key ascii-text "$9$jriHP5QFn/ApPfBIehr1Yg4aDik.P5z3Dj9Apu1l7—dbgoJGD";
  ## SECRET-DATA
}
gateway gw-chicago {
  ike-policy ipv6-ike-phase1-policy;
  address 1111::1112;
  external-interface ge-0/0/15.0;
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring IPsec

#### CLI Quick Configuration

To quickly configure IPsec, copy the following commands and paste them into the CLI:

```

[edit]
set security ipsec proposal ipv6-ipsec-phase2-proposal protocol esp
set security ipsec proposal ipv6-ipsec-phase2-proposal authentication-algorithm
  hmac-sha1-96
set security ipsec proposal ipv6-ipsec-phase2-proposal encryption-algorithm aes-128-cbc
set security ipsec policy ipv6-ipsec-phase2-policy proposals ipv6-ipsec-phase2-proposal
set security ipsec policy ipv6-ipsec-phase2-policy perfect-forward-secrecy keys group2
set security ipsec vpn ipv6-ike-vpn-chicago ike gateway gw-chicago
set security ipsec vpn ipv6-ike-vpn-chicago ike ipv6-ipsec-policy ipsec-phase2-policy

```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see [Using the CLI Editor in Configuration Mode](#).

To configure IPsec:

1. Create an IPsec Phase 2 proposal.

```

[edit]
user@host# set security ipsec proposal ipv6-ipsec-phase2-proposal

```

2. Specify the IPsec Phase 2 proposal protocol.

```

[edit security ipsec proposal ipv6- ipsec-phase2-proposal]
user@host# set protocol esp

```

3. Specify the IPsec Phase 2 proposal authentication algorithm.

```

[edit security ipsec proposal ipv6-ipsec-phase2-proposal]
user@host# set authentication-algorithm hmac-sha1-96

```

4. Specify the IPsec Phase 2 proposal encryption algorithm.

```

[edit security ipsec proposal ipv6-ipsec-phase2-proposal]

```



- ```
user@host# set encryption-algorithm aes-128-cbc
```
5. Create the IPsec Phase 2 policy.
 

```
[edit security ipsec]
user@host# set policy ipv6-ipsec-phase2-policy
```
  6. Specify the IPsec Phase 2 proposal reference.
 

```
[edit security ipsec policy ipv6-ipsec-phase2-policy]
user@host# set proposals ipv6-ipsec-phase2-proposal
```
  7. Specify IPsec Phase 2 PFS to use Diffie-Hellman group 2.
 

```
[edit security ipsec policy ipv6-ipsec-phase2-policy]
user@host# set perfect-forward-secrecy keys group2
```
  8. Specify the IKE gateway.
 

```
[edit security ipsec]
user@host# set vpn ipv6-ike-vpn-chicago ike gateway gw-chicago
```
  9. Specify the IPsec Phase 2 policy.
 

```
[edit security ipsec]
user@host# set vpn ipv6-ike-vpn-chicago ike ipsec-policy ipv6-ipsec-phase2-policy
```

**Results** From configuration mode, confirm your configuration by entering the **show security ipsec** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ipsec
proposal ipv6-ipsec-phase2-proposal {
  protocol esp;
  authentication-algorithm hmac-sha1-96;
  encryption-algorithm aes-128-cbc;
}
policy ipv6-ipsec-phase2-policy {
  perfect-forward-secrecy {
    keys group2;
  }
  proposals ipv6-ipsec-phase2-proposal;
}
vpn ipv6-ike-vpn-chicago {
  ike {
    gateway gw-chicago;
    ipsec-policy ipv6-ipsec-phase2-policy;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring Security Policies

**CLI Quick Configuration** To quickly configure security policies, copy the following commands and paste them into the CLI:

```
[edit]
```

```

set security policies from-zone trust to-zone untrust policy ipv6-vpn-tr-untr match
  source-address sunnyvale
set security policies from-zone trust to-zone untrust policy ipv6-vpn-tr-untr match
  destination-address chicago
set security policies from-zone trust to-zone untrust policy ipv6-vpn-tr-untr match
  application any
set security policies from-zone trust to-zone untrust policy ipv6-vpn-tr-untr then permit
  tunnel ipsec-vpn ipv6-ike-vpn-chicago
set security policies from-zone trust to-zone untrust policy ipv6-vpn-tr-untr then permit
  tunnel pair-policy ipv6-vpn-untr-tr
set security policies from-zone untrust to-zone trust policy ipv6-vpn-untr-tr match
  source-address chicago
set security policies from-zone untrust to-zone trust policy ipv6-vpn-untr-tr match
  destination-address sunnyvale
set security policies from-zone untrust to-zone trust policy ipv6-vpn-untr-tr match
  application any
set security policies from-zone untrust to-zone trust policy ipv6-vpn-untr-tr then permit
  tunnel ipsec-vpn ipv6-ike-vpn-chicago
set security policies from-zone untrust to-zone trust policy ipv6-vpn-untr-tr then permit
  tunnel pair-policy ipv6-vpn-tr-untr
set security policies from-zone trust to-zone untrust policy permit-any match
  source-address any
set security policies from-zone trust to-zone untrust policy permit-any match
  destination-address any
set security policies from-zone trust to-zone untrust policy permit-any match application
  any
set security policies from-zone trust to-zone untrust policy permit-any then permit
insert security policies from-zone trust to-zone untrust policy ipv6-vpn-tr-untr before
  policy permit-any

```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure security policies:

1. Create the security policy to permit traffic from the trust zone to the untrust zone.

```

[edit security policies from-zone trust to-zone untrust]
user@host# set policy ipv6-vpn-tr-untr match source-address sunnyvale
user@host# set policy ipv6-vpn-tr-untr match destination-address chicago
user@host# set policy ipv6-vpn-tr-untr match application any
user@host# set policy ipv6-vpn-tr-untr then permit tunnel ipsec-vpn
  ipv6-ike-vpn-chicago
user@host# set policy ipv6-vpn-tr-untr then permit tunnel pair-policy
  ipv6-vpn-untr-tr

```

2. Create the security policy to permit traffic from the untrust zone to the trust zone.

```

[edit security policies from-zone untrust to-zone trust]
user@host# set policy ipv6-vpn-untr-tr match source-address sunnyvale
user@host# set policy ipv6-vpn-untr-tr match destination-address chicago
user@host# set policy ipv6-vpn-untr-tr match application any
user@host# set policy ipv6-vpn-untr-tr then permit tunnel ipsec-vpn
  ipv6-ike-vpn-chicago

```

```
user@host# set policy ipv6-vpn-untr-tr then permit tunnel pair-policy
ipv6-vpn-tr-untr
```

3. Create the security policy to permit traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy permit-any match source-address any
user@host# set policy permit-any match destination-address any
user@host# set policy permit-any match application any
user@host# set policy permit-any then permit
```

4. Reorder the security policies so that the vpn-tr-untr security policy is placed above the permit-any security policy.

```
[edit security policies from-zone trust to-zone untrust]
user@host# insert policy ipv6-vpn-tr-untr before policy permit-any
```

**Results** From configuration mode, confirm your configuration by entering the `show security policies` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security policies
from-zone trust to-zone untrust {
  policy ipv6-vpn-tr-untr {
    match {
      source-address sunnyvale;
      destination-address chicago;
      application any;
    }
    then {
      permit {
        tunnel {
          ipsec-vpn ipv6-ike-vpn-chicago;
          pair-policy ipv6-vpn-untr-tr;
        }
      }
    }
  }
}
policy permit-any {
  match {
    source-address any;
    destination-address any;
    application any;
  }
  then {
    permit
  }
}
}
from-zone untrust to-zone trust {
  policy ipv6-vpn-untr-tr {
    match {
      source-address chicago;
      destination-address sunnyvale;
      application any;
    }
  }
}
```

```

    }
    then {
      permit {
        tunnel {
          ipsec-vpn ipv6-ike-vpn-chicago;
          pair-policy ipv6-vpn-tr-untr;
        }
      }
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring TCP-MSS

**CLI Quick Configuration** To quickly configure TCP-MSS information, copy the following commands and paste them into the CLI:

```

[edit]
set security flow tcp-mss ipsec-vpn mss 1350

```

**Step-by-Step Procedure** To configure TCP-MSS information:

1. Configure TCP-MSS information.

```

[edit]
user@host# set security flow tcp-mss ipsec-vpn mss 1350

```

**Results** From configuration mode, confirm your configuration by entering the **show security flow** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show security flow
tcp-mss {
  ipsec-vpn {
    mss 1350;
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying the IKE Phase 1 Status on page 558
- Verifying the IPsec Phase 2 Status on page 560

### Verifying the IKE Phase 1 Status

**Purpose** Verify the IKE Phase 1 status.

**Action**

**NOTE:** Before starting the verification process, you need to send traffic from a host in Sunnyvale to a host in Chicago. For policy-based VPNs, a separate host must generate the traffic; traffic initiated from the SRX Series device will not match the VPN policy. We recommend that the test traffic be from a separate device on one side of the VPN to a second device on the other side of the VPN. For example, initiate ping from 1212::abcd/64 to 1111::abcd/128.

From operational mode, enter the **show security ike security-associations** command. After obtaining an index number from the command, use the **show security ike security-associations index *index\_number* detail** command.

```
user@host> show security ike security-associations
Index  Remote Address  State  Initiator cookie  Responder cookie  Mode
5      1111::1112      UP     e48efd6a444853cf  0d09c59aafb720be  Aggressive
```

```
user@host> show security ike security-associations index 5 detail
IKE peer 1111::1112, Index 5,
  Role: Initiator, State: UP
  Initiator cookie: e48efd6a444853cf, Responder cookie: 0d09c59aafb720be
  Exchange type: Aggressive, Authentication method: Pre-shared-keys
  Local: 1111::1111:500, Remote: 1111::1112:500
  Lifetime: Expires in 19518 seconds
  Peer ike-id: not valid
  Xauth assigned IP: 0.0.0.0
  Algorithms:
    Authentication      : sha1
    Encryption          : aes-128-cbc
    Pseudo random function: hmac-sha1
  Traffic statistics:
    Input bytes   :           1568
    Output bytes  :           2748
    Input packets:             6
    Output packets:            23
  Flags: Caller notification sent
  IPSec security associations: 5 created, 0 deleted
  Phase 2 negotiations in progress: 1

  Negotiation type: Quick mode, Role: Initiator, Message ID: 2900338624
  Local: 1111::1111:500, Remote: 1111::1112:500
  Local identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Flags: Caller notification sent, Waiting for done
```

**Meaning** The **show security ike security-associations** command lists all active IKE Phase 1 security associations (SAs). If no SAs are listed, there was a problem with Phase 1 establishment. Check the IKE policy parameters and external interface settings in your configuration.

If SAs are listed, review the following information:

- Index—This value is unique for each IKE SA, which you can use in the **show security ike security-associations index *index\_number* detail** command to get more information about the SA.
- Remote Address—Verify that the remote IP address is correct.
- State
  - UP—The Phase 1 SA has been established.
  - DOWN—There was a problem establishing the Phase 1 SA.
- Mode—Verify that the correct mode is being used.

Verify that the following are correct in your configuration:

- External interfaces (the interface must be the one that receives IKE packets)
- IKE policy parameters
- Preshared key information
- Phase 1 proposal parameters (must match on both peers)

The **show security ike security-associations index 5 detail** command lists additional information about the security association with an index number of 5:

- Authentication and encryption algorithms used
- Phase 1 lifetime
- Traffic statistics (can be used to verify that traffic is flowing properly in both directions)
- Initiator and responder role information



**NOTE:** Troubleshooting is best performed on the peer using the responder role.

- Number of IPsec SAs created
- Number of Phase 2 negotiations in progress

### Verifying the IPsec Phase 2 Status

**Purpose** Verify the IPsec Phase 2 status.

**Action** From operational mode, enter the **show security ipsec security-associations** command. After obtaining an index number from the command, use the **show security ipsec security-associations index *index\_number* detail** command.

```
user@host> show security ipsec security-associations
total configured sa: 2
  ID   Algorithm      SPI      Life:sec/kb  Mon vsys Port  Gateway
  2    ESP:aes-128/sha1 14caf1d9 3597/ unlim  -  root 500  1111::1112
  2    ESP:aes-128/sha1 9a4db486 3597/ unlim  -  root 500  1111::1112
```

```

user@host> show security ipsec security-associations index 2 detail
Virtual-system: Root
Local Gateway: 111::1111, Remote Gateway: 1111::1112
Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
DF-bit: clear
Direction: inbound, SPI: 14caf1d9, AUX-SPI: 0
                    , VPN Monitoring: -
Hard lifetime: Expires in 3440 seconds
Lifeseize Remaining: Unlimited
Soft lifetime: Expires in 2813 seconds
Mode: tunnel, Type: dynamic, State: installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
Anti-replay service: counter-based enabled, Replay window size: 64

Direction: outbound, SPI: 9a4db486, AUX-SPI: 0
                    , VPN Monitoring: -
Hard lifetime: Expires in 3440 seconds
Lifeseize Remaining: Unlimited
Soft lifetime: Expires in 2813 seconds
Mode: tunnel, Type: dynamic, State: installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: aes-cbc (128 bits)
Anti-replay service: counter-based enabled, Replay window size: 64

```

**Meaning** The output from the `show security ipsec security-associations` command lists the following information:

- The ID number is 2. Use this value with the `show security ipsec security-associations index` command to get more information about this particular SA.
- There is one IPsec SA pair using port 500, which indicates that no NAT-traversal is implemented. (NAT-traversal uses port 4500 or another random high-number port.)
- The SPIs, lifetime (in seconds), and usage limits (or lifeseize in KB) are shown for both directions. The 3597/unlim value indicates that the Phase 2 lifetime expires in 3597 seconds, and that no lifeseize has been specified, which indicates that the lifetime is unlimited. Phase 2 lifetime can differ from Phase 1 lifetime, as Phase 2 is not dependent on Phase 1 after the VPN is up.
- VPN monitoring is not enabled for this SA, as indicated by a hyphen in the Mon column. If VPN monitoring is enabled, U (up) or D (down) is listed.
- The virtual system (vsys) is the root system, and it always lists 0.

The output from the `show security ipsec security-associations index 2 detail` command lists the following information:

- The local and remote identities make up the proxy ID for the SA.

A proxy ID mismatch is one of the most common reasons for a Phase 2 failure. For policy-based VPNs, the proxy ID is derived from the security policy. The local and remote addresses are derived from the address book entries, and the service is derived from the application configured for the policy. If Phase 2 fails because of a proxy ID mismatch, you can use the policy to confirm which address book entries are configured. Verify that the addresses match the information being sent. Check the service to ensure that the ports match the information being sent.



**NOTE:** For some third-party vendors, the proxy ID must be manually entered to match.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding IPv6 IKE and IPsec Packet Processing on page 538](#)
- [IPv6 IPsec Configuration Overview on page 543](#)
- [Example: Configuring an IPv6 IPsec Manual VPN on page 543](#)

## Global SPI and VPN Monitoring Features

- [Understanding Global SPI and VPN Monitoring Features on page 562](#)
- [Example: Configuring Global SPI and VPN Monitoring Features on page 563](#)

### Understanding Global SPI and VPN Monitoring Features

You can monitor and maintain the efficient operation of your VPN using the following global VPN features:

- **SPI—Peers in a security association (SA) can become unsynchronized when one of the peers fails.** For example, if one of the peers reboots, it might send an incorrect security parameter index (SPI). You can enable the device to detect such an event and resynchronize the peers by configuring the bad SPI response feature.
- **VPN monitoring—You can use the global VPN monitoring feature to periodically send Internet Control Message Protocol (ICMP) requests to the peer to determine if the peer is reachable.**

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [VPN Overview on page 451](#)
- [IPsec VPN Configuration Overview](#)
- [Example: Configuring Global SPI and VPN Monitoring Features \(CLI\)](#)



## Example: Configuring Global SPI and VPN Monitoring Features

- Requirements on page 563
- Overview on page 563
- Configuration on page 563

### Requirements

Before you begin, understand global SPI and VPN monitoring features. See “Understanding Global SPI and VPN Monitoring Features” on page 562.

### Overview

In this example, you configure the device to detect and respond five times to a bad IPsec SPI before deleting the SA and initiating a new one. You also configure the device to monitor the VPN by sending ICMP requests to the peer every 15 seconds, and to declare the peer unreachable after 15 unsuccessful pings.

### Configuration

#### Step-by-Step Procedure

To configure global VPN settings in the CLI editor:

1. Specify global VPN settings.
 

```
[edit]
user@host# set security ike respond-bad-spi 5
user@host# set security ipsec vpn-monitor-options interval 15 threshold 15
```

#### Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Example: Configuring a Policy-Based VPN on page 489
- Example: Configuring a Route-Based VPN on page 470

## Virtual Router Support for Route-Based VPNs

This feature includes routing-instance support for route-based VPNs. In previous releases, when an st0 interface was put in a nondefault routing instance, the VPN tunnels on this interface did not work properly. In the Junos OS 10.4 release, the support is enabled to place st0 interfaces in a routing instance, where each unit is configured in point-to-point mode or multipoint mode. Therefore, VPN traffic now works correctly in a nondefault VR. You can now configure different subunits of the st0 interface in different routing instances. The following functions are supported for nondefault routing instances:

- Manual key management
- Transit traffic
- Self-traffic
- VPN monitoring
- Hub-and-spoke VPNs

- Encapsulating Security Payload (ESP) protocol
- Authentication Header (AH) protocol
- Aggressive mode or main mode
- st0 anchored on the loopback (lo0) interface
- Maximum number of virtual routers (VRs) supported on an SRX Series device
- Applications such as Application Layer Gateway (ALG), Intrusion Detection and Prevention (IDP), and Unified Threat Management (UTM)
- Dead peer detection (DPD)
- Chassis cluster active/backup
- Open Shortest Path First (OSPF) over st0
- Routing Information Protocol (RIP) over st0



**NOTE:** For VPN traffic to work in a nondefault VR, the IKE listener must be placed in the default VR.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Virtual Router Limitations on page 568](#)

---

## Example: Configuring an st0 Interface in a Virtual Router

This example shows how to configure an st0 interface in a virtual router.

- [Requirements on page 564](#)
- [Overview on page 564](#)
- [Configuration on page 565](#)
- [Verification on page 568](#)

### Requirements

Before you begin, configure the interfaces and assign the interfaces to security zones. See “Security Zones and Interfaces Overview” on page 111.

### Overview

In this example, you perform the following operations:

- Configure the interfaces.
- Configure IKE Phase 1 proposals.
- Configure IKE policies, and reference the proposals.
- Configure an IKE gateway, and reference the policy.
- Configure Phase 2 proposals.

- Configure policies, and reference the proposals.
- Configure AutoKey IKE, and reference the policy and gateway.
- Configure the security policy.
- Configure the routing instance.
- Configure the VPN bind to tunnel interface.
- Configure the routing options.

## Configuration

**CLI Quick Configuration** To quickly configure an st0 interface in a virtual router, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet address 1.1.1.2/30
set interfaces ge-0/0/1 unit 0 family inet address 2.2.2.2/30
set interfaces st0 unit 0 family inet address 3.3.3.2/30
set security ike proposal first_ikeprop authentication-method pre-shared-keys
set security ike proposal first_ikeprop dh-group group2
set security ike proposal first_ikeprop authentication-algorithm md5
set security ike proposal first_ikeprop encryption-algorithm 3des-cbc
set security ike policy first_ikepol mode main
set security ike policy first_ikepol proposals first_ikeprop
set security ike policy first_ikepol pre-shared-key ascii-text
"$9$xFU-b2ZUH5Qn4aQn/CB17-V"
set security ike gateway first ike-policy first_ikepol
set security ike gateway first address 4.4.4.2
set security ike gateway first external-interface ge-0/0/0.0
set security ipsec proposal first_ipsecprop protocol esp
set security ipsec proposal first_ipsecprop authentication-algorithm hmac-md5-96
set security ipsec proposal first_ipsecprop encryption-algorithm 3des-cbc
set security ipsec policy first_ipsecpol perfect-forward-secrecy keys group1
set security ipsec policy first_ipsecpol proposals first_ipsecprop
set security ipsec vpn first_vpn bind-interface st0.0
set security ipsec vpn first_vpn ike gateway first
set security ipsec vpn first_vpn ike ipsec-policy first_ipsecpol
set security ipsec vpn first_vpn establish-tunnels immediately
set security policies default-policy permit-all
set routing-instances VR1 instance-type virtual-router
set routing-instances VR1 interface ge-0/0/1.0
set routing-instances VR1 interface st0.0
set routing-instances VR1 routing-options static route 6.6.6.0/24 next-hop st0.0
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure an st0 in a VR:

1. Configure the interfaces.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 1.1.1.2/30
```

- ```
user@host# set interfaces ge-0/0/1 unit 0 family inet address 2.2.2.2/30
user@host# set interfaces st0 unit 0 family inet address 3.3.3.2/30
```
2. Configure Phase 1 of the IPsec tunnel.

```
[edit security ike]
user@host# set proposal first_ikeprop authentication-method pre-shared-keys
user@host# set proposal first_ikeprop dh-group group2
user@host# set proposal first_ikeprop authentication-algorithm md5
user@host# set proposal first_ikeprop encryption-algorithm 3des-cbc
```
  3. Configure the IKE policies, and reference the proposals.

```
[edit security ike]
user@host# set policy first_ikepol mode main
user@host# set policy first_ikepol proposals first_ikeprop
user@host# set policy first_ikepol pre-shared-key ascii-text
"$9$xFU-b2ZUH5Qn4aQn/CB17-V"
```
  4. Configure the IKE gateway, and reference the policy.

```
[edit security ike]
user@host# set gateway first ike-policy first_ikepol
user@host# set gateway first address 4.4.4.2
user@host# set gateway first external-interface ge-0/0/0.0
```
  5. Configure Phase 2 of the IPsec tunnel.

```
[edit security ipsec]
user@host# set proposal first_ipsecprop protocol esp
user@host# set proposal first_ipsecprop authentication-algorithm hmac-md5-96
user@host# set proposal first_ipsecprop encryption-algorithm 3des-cbc
```
  6. Configure the policies, and reference the proposals.

```
[edit security ipsec]
user@host# set policy first_ipsecpol perfect-forward-secrecy keys group1
user@host# set policy first_ipsecpol proposals first_ipsecprop
```
  7. Configure AutoKey IKE, and reference the policy and gateway.

```
[edit security ipsec]
user@host# set vpn first_vpn ike gateway first
user@host# set vpn first_vpn ike ipsec-policy first_ipsecpol
user@host# set vpn first_vpn establish-tunnels immediately
```
  8. Configure the VPN bind to tunnel interface.

```
[edit security ipsec]
user@host# set vpn first_vpn bind-interface st0.0
```
  9. Configure the security policy.

```
[edit security policies]
user@host# set default-policy permit-all
```
  10. Configure the st0 in the routing instance.

```
[edit routing-instances]
user@host# set VR1 instance-type virtual-router
user@host# set VR1 interface ge-0/0/1.0
user@host# set VR1 interface st0.0
```

11. Configure the routing options.

```
[edit routing-instances VR1 routing-options]
user@host# set static route 6.6.6.0/24 next-hop st0.0
```

**Results** From configuration mode, confirm your configuration by entering the **show security** and **show routing-instances** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show security
ike {
  proposal first_ikeprop {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm md5;
    encryption-algorithm 3des-cbc;
  }
  policy first_ikepol {
    mode main;
    proposals first_ikeprop;
    pre-shared-key ascii-text "$9$xFU-b2ZUH5Qn4aQn/CB17-V"; ## SECRET-DATA
  }
  gateway first {
    ike-policy first_ikepol;
    address 4.4.4.2;
    external-interface ge-0/0/0.0;
  }
}
ipsec {
  proposal first_ipsecprop {
    protocol esp;
    authentication-algorithm hmac-md5-96;
    encryption-algorithm 3des-cbc;
  }
  policy first_ipsecpol {
    perfect-forward-secrecy {
      keys group;
    }
    proposals first_ipsecprop;
  }
  vpn first_vpn {
    bind-interface st0.0;
    ike {
      gateway first;
      ipsec-policy first_ipsecpol;
    }
    establish-tunnels immediately;
  }
}
policies {
  default-policy {
    permit-all;
  }
}
user@host# show routing-instances
```

```
VR1 {  
  instance-type virtual-router;  
  interface ge-0/0/1.0;  
  interface st0.0;  
  routing-options {  
    static {  
      route 6.6.6.0/24 next-hop st0.0;  
    }  
  }  
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform this task:

- Verifying an st0 interface in the Virtual Router on page 568

### [Verifying an st0 interface in the Virtual Router](#)

---

**Purpose** Verify the st0 interface in the virtual router.

**Action** From operational mode, enter the **show interfaces st0.0 detail** command. The number listed for routing table corresponds to the order that the routing tables in the **show route all** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

## Understanding Virtual Router Limitations

---

The following features are not supported in this release for virtual router (VR):

- Point-to-multipoint tunnel inside VR
- Dynamic endpoint VPN and remote access VPN inside VR
- NAT-Traversal (NAT-T) inside VR
- Policy-based VPN inside VR
- Public key infrastructure (PKI) inside VR
- Internet Key Exchange (IKE) inside VR
- Chassis cluster active/active with VPN inside VR

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Virtual Router Support for Route-Based VPNs on page 563

# Public Key Cryptography for Certificates

- Understanding Certificates and PKI on page 569
- Digital Certificates Configuration Overview on page 574
- Public-Private Key Pairs on page 575
- CA Profiles on page 577
- Certificate Enrollment on page 578
- Certificate Requests on page 583
- Certificate Loading on page 585
- CRLs on page 587
- Self-Signed Certificates on page 592
- Deleting Certificates (CLI Procedure) on page 595

## Understanding Certificates and PKI

---

A *digital certificate* is an electronic means for verifying your identity through a trusted third party, known as a certificate authority (CA). Alternatively, you can use a self-signed certificate to attest to your identity.

The CA server you use can be owned and operated by an independent CA or by your own organization, in which case you become your own CA. If you use an independent CA, you must contact them for the addresses of their CA and certificate revocation list (CRL) servers (for obtaining certificates and CRLs) and for the information they require when submitting personal certificate requests. When you are your own CA, you determine this information yourself.

The Public Key Infrastructure (PKI) provides an infrastructure for digital certificate management.

This topic includes the following sections:

- Certificate Signatures and Verification on page 570
- Public Key Infrastructure on page 570
- PKI Management and Implementation on page 572
- Internet Key Exchange on page 573

## Certificate Signatures and Verification

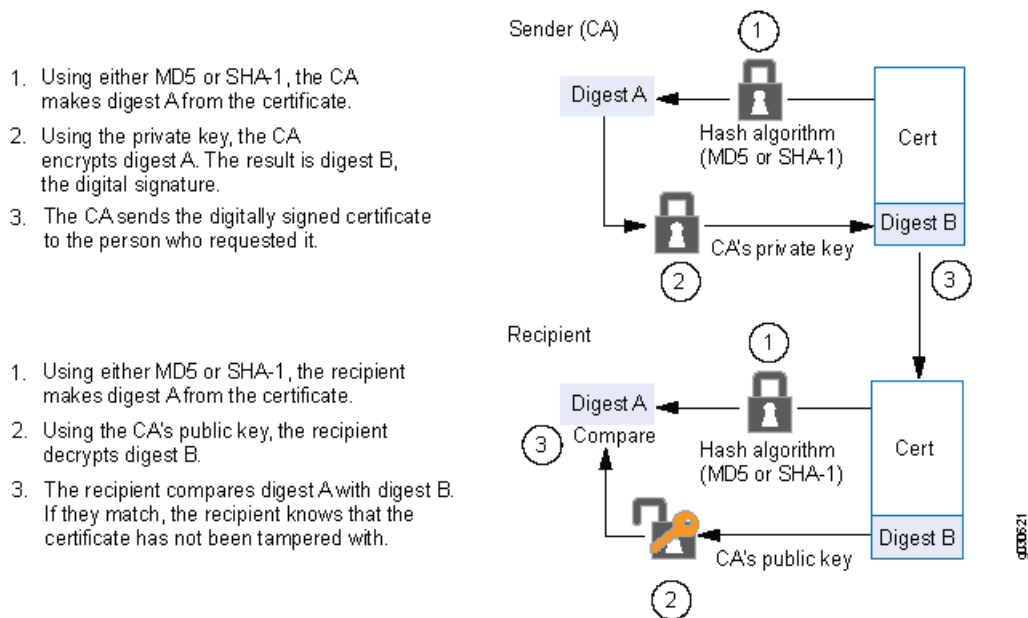
The CA that issues a certificate uses a Message Digest 5 (MD5) or Secure Hash Algorithm 1 (SHA-1) to generate a digest, and then “signs” the certificate by encrypting the digest with its private key. The result is a digital signature. The CA then makes the digitally signed certificate available for download to the person who requested it. Figure 55 on page 570 illustrates this process.

The recipient of the certificate generates another digest by applying the same MD5 or SHA-1 hash algorithm to the certificate file, then uses the CA's public key to decrypt the digital signature. By comparing the decrypted digest with the digest just generated, the recipient can confirm the integrity of the CA's signature and, by extension, the integrity of the accompanying certificate. Figure 55 on page 570 illustrates this process.



**NOTE:** If the issuer of the end-entity (EE) certificate is not a root certificate, up to eight levels are verified. Revocation status of each certificate in the verification chain is also verified. A certificate revocation status is considered “good” when its serial number is not in the CRL, which satisfies the refresh requirement per CA profile.

Figure 55: Digital Signature Verification



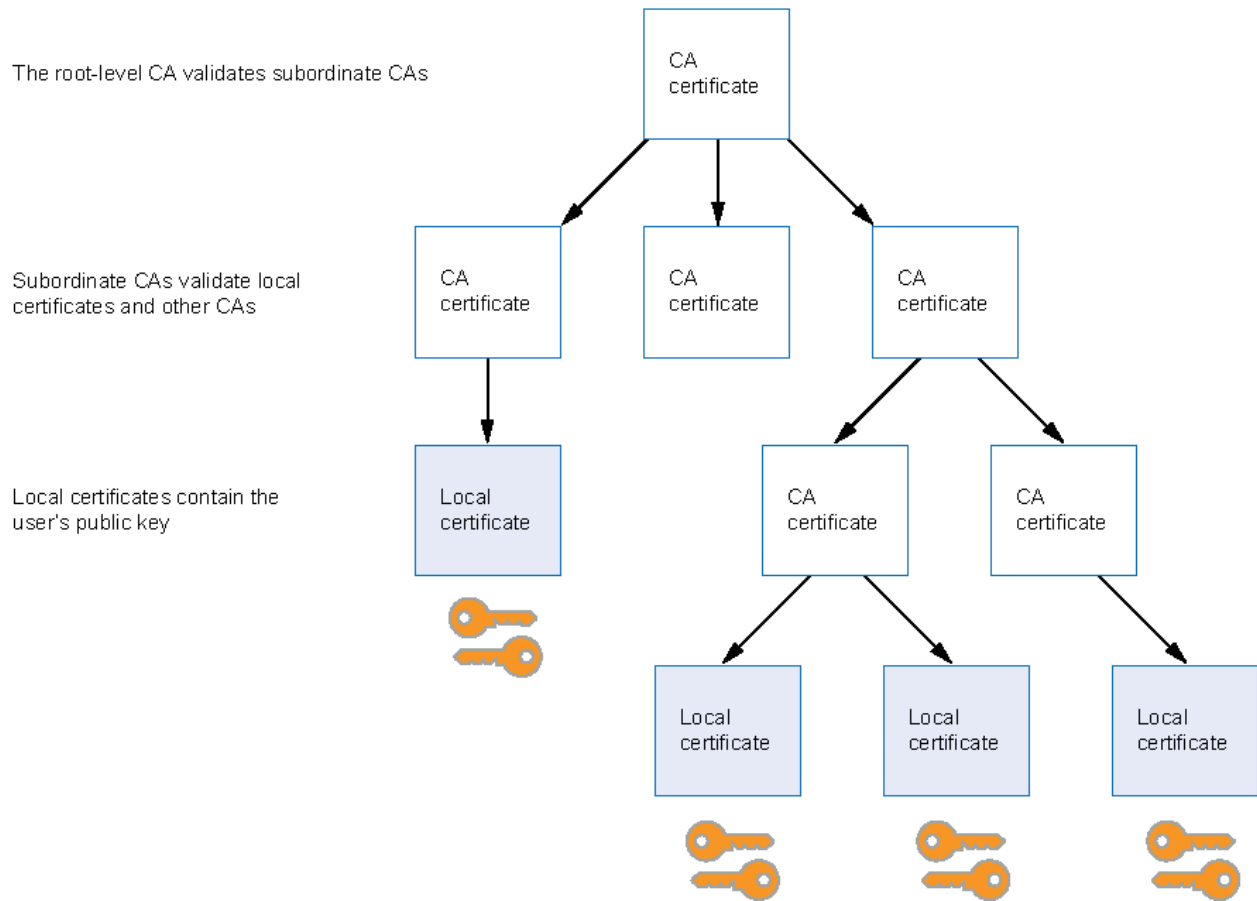
## Public Key Infrastructure

To verify the trustworthiness of a certificate, you must be able to track a path of certified certificate authorities (CAs) from the one issuing your local certificate to the root authority of a CA domain. *Public key infrastructure* (PKI) refers to the hierarchical structure of trust required for the successful implementation of public key cryptography.

Figure 56 on page 571 shows the structure of a single-domain certificate authority.

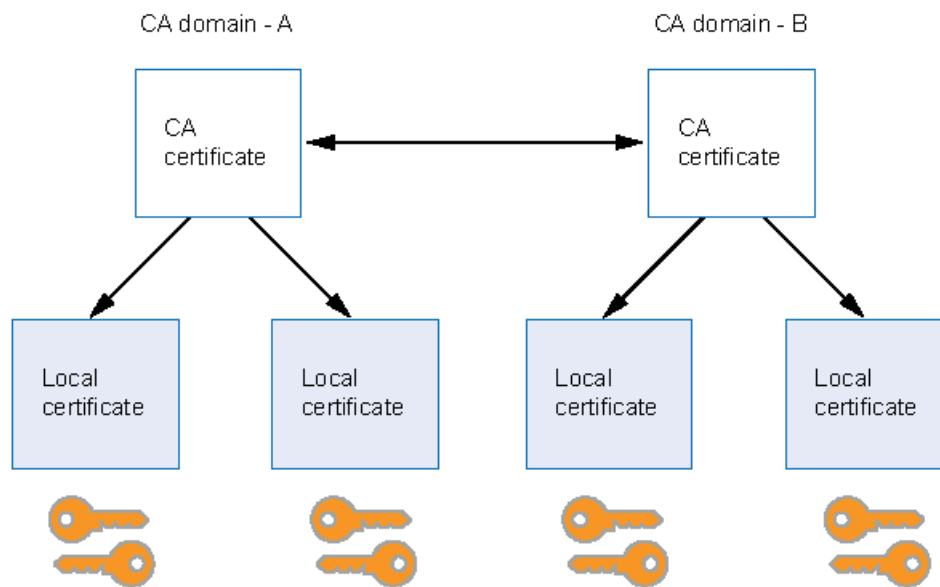


Figure 56: PKI Hierarchy of Trust—CA Domain



If certificates are used solely within an organization, that organization can have its own CA domain within which a company CA issues and validates certificates for its employees. If that organization later wants its employees to exchange their certificates with certificates from another CA domain (for example, with employees at another organization that has its own CA domain), the two CAs can develop cross-certification by agreeing to trust the authority of each other. In this case, the PKI structure does not extend vertically but does extend horizontally. See Figure 57 on page 572.

Figure 57: Cross-Certification



Users in the CA domain A can use their certificates and key pairs with users in CA domain B because the CA's have cross-certified each other.

## PKI Management and Implementation

The minimum PKI elements required for certificate-based authentication in Junos OS are:

- CA certificates and authority configuration.
- Local certificates including the devices identity (example: IKE ID type and value) and private and public keys
- Certificate validation through a CRL.

Junos OS supports three different types of PKI objects:

- Private/public key pair
- Certificates
  - Local certificate—The local certificate contains the public key and identity information for the Juniper Networks device. The Juniper Networks device owns the associated private key. This certificate is generated based on a certificate request from the Juniper Networks device.
  - Pending certificate — A pending certificate contains a key pair and identity information that is generated into a PKCS10 certificate request and manually sent to a certificate authority (CA). While the Juniper Networks device waits for the certificate from the CA, the existing object (key pair and the certificate request) is tagged as a certificate request or pending certificate.



**NOTE:** Junos OS Release 9.0 or later supports automatic sending of certificate requests through the Simple Certificate Enrollment Protocol (SCEP).

- CA certificate — When the certificate is issued by the CA and loaded into the Junos device, the pending certificate is replaced by the newly generated local certificate. All other certificates loaded into the device are considered CA certificates.

For convenience and practicality, PKI must be transparently managed and implemented. Toward this goal, Junos OS supports the following features:

- Generates a public-private key pair.
- Loads multiple local certificates from different CAs.
- Delivers a certificate when establishing an IPsec tunnel.
- Validates a certificate path upward through eight levels of CA authorities in the PKI hierarchy.
- Supports the Public-Key Cryptography Standards #7 (PKCS #7) cryptographic . As a result, the device can accept X.509 certificates and certificate revocation lists (CRLs) packaged within a PKCS #7 envelope.



**NOTE:** Junos OS supports a PKCS #7 file size of up to 7 KB.

- Retrieves CRLs online retrieval through Lightweight Directory Access Protocol (LDAP) or Hypertext Transfer Protocol (HTTP).

## Internet Key Exchange

The procedure for digitally signing messages sent between two participants in an Internet Key Exchange (IKE) session is similar to digital certificate verification, with the following differences:

- Instead of making a digest from the CA certificate, the sender makes it from the data in the IP packet payload.
- Instead of using the CA's public-private key pair, the participants use the sender's public-private key pair.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Digital Certificates Configuration Overview on page 574
- VPN Overview on page 451

## Digital Certificates Configuration Overview

---

You can obtain CA and local certificates manually, or online using the Simple Certificate Enrollment Protocol (SCEP). Certificates are verifiable and renewable, and you can delete them when they are no longer needed.

Junos OS Release 8.5 and earlier support only manual certificate requests. This process includes generation of a PKCS10 request, submission to the CA, retrieval of the signed certificate, and manually loading of the certificate into the Juniper Networks device.

Automatic sending of certificate requests through SCEP is supported only in Junos OS Release 9.0 or later.

To use a digital certificate to authenticate your identity when establishing a secure VPN connection, you must first do the following:

- Obtain a CA certificate from which you intend to obtain a local certificate, and then load the CA certificate onto the device. The CA certificate can contain a CRL to identify invalid certificates.
- Obtain a local certificate from the CA whose CA certificate you have previously loaded, and then load the local certificate in the device. The local certificate establishes the identity of the Juniper Networks device with each tunnel connection.

This topic includes the following sections:

- [Enabling Digital Certificates Online: Configuration Overview](#) on page 574
- [Manually Generating Digital Certificates: Configuration Overview](#) on page 574

### Enabling Digital Certificates Online: Configuration Overview

SCEP uses the online method to request digital certificates. To obtain a certificate online:

1. Generate a key pair on the device. See “[Example: Generating a Public-Private Key Pair](#)” on page 576.
2. Create a CA profile or profiles containing information specific to a CA. See “[Example: Configuring a CA Profile](#)” on page 577.
3. Enroll the CA certificate. See “[Enrolling a CA Certificate Online Using SCEP](#)” on page 579.
4. Enroll the local certificate from the CA whose CA certificate you have previously loaded. See “[Example: Enrolling a Local Certificate Online Using SCEP](#)” on page 580.
5. Configure automatic reenrollment. See “[Example: Using SCEP to Automatically Renew a Local Certificate](#)” on page 581.

### Manually Generating Digital Certificates: Configuration Overview

To obtain digital certificates manually:

1. Generate a key pair on the device. See “Example: Generating a Public-Private Key Pair” on page 576.
2. Create a CA profile or profiles containing information specific to a CA. See “Example: Configuring a CA Profile” on page 577.
3. Generate the CSR for the local certificate and send it to the CA server. See “Example: Manually Generating a CSR for the Local Certificate and Sending it to the CA Server” on page 584.
4. Load the certificate onto the device. See “Example: Loading CA and Local Certificates Manually” on page 585.
5. Configure automatic reenrollment. See “Example: Using SCEP to Automatically Renew a Local Certificate” on page 581.
6. If necessary, load the certificate's CRL on the device. See “Example: Manually Loading a CRL onto the Device” on page 587.
7. If necessary, configure the CA profile with CRL locations. See “Example: Configuring a Certificate Authority Profile with CRL Locations” on page 590

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Certificates and PKI on page 569](#)
- [Example: Verifying Certificate Validity on page 589](#)
- [Example: Configuring a Certificate Authority Profile with CRL Locations on page 590](#)
- [Deleting Certificates \(CLI Procedure\) on page 595](#)

## Public-Private Key Pairs

- [Understanding Public Key Cryptography on page 575](#)
- [Example: Generating a Public-Private Key Pair on page 576](#)

## Understanding Public Key Cryptography

The public-private key pairs used in public key cryptography play an important role in the use of digital certificates. A *public-private key pair* encrypts and decrypts data. Data encrypted with a public key, which the owner makes available to the public, can be decrypted with the corresponding private key only, which the owner keeps secret and protected. For example, if Alice wants to send Bob an encrypted message, Alice can encrypt it with Bob's public key and send it to him. Bob then decrypts the message with his private key.

The reverse process is also useful: encrypting data with a private key and decrypting it with the corresponding public key. This process is known as creating a digital signature. For example, if Alice wants to present her identity as the sender of a message, she can encrypt the message with her private key and send the message to Bob. Bob then decrypts the message with Alice's public key, thus verifying that Alice is indeed the sender.

When you generate a public-private key pair, the device automatically saves the key pair in a file in the certificate store, where it is subsequently used in certificate request commands. The generated key pair is saved as **certificate-id.priv**.



**NOTE:** The default RSA and DSA key size is 1024 bits. If you are using the Simple Certificate Enrollment Protocol (SCEP), Junos OS supports RSA only.



**NOTE:** If the device renews a great number of certificates at once, thus using up keys rapidly, it might run out of pregenerated keys and have to generate them promptly for each new request. In this case, the generation of keys might affect the performance of the device, especially in a high-availability environment where the performance of the device might slow down for a number of minutes.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Certificates and PKI on page 569](#)
- [Example: Generating a Public-Private Key Pair on page 576](#)
- [Digital Certificates Configuration Overview on page 574](#)

### Example: Generating a Public-Private Key Pair

This example shows how to generate a public-private key pair.

- [Requirements on page 576](#)
- [Overview on page 576](#)
- [Configuration on page 576](#)
- [Verification on page 577](#)

#### Requirements

No special configuration beyond device initialization is required before configuring this feature.

#### Overview

In this example, you generate a public-private key pair named ca-ipsec.

#### Configuration

#### Step-by-Step Procedure

To generate a public-private key pair:

1. Create a certificate key pair.

[edit]

```
user@host> request security pki generate-key-pair certificate-id ca-ipsec
```

## Verification

---

After the public-private key pair is generated, the Juniper Networks device displays the following:

```
generated key pair ca-ipsec, key size 1024 bits
```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Public Key Cryptography on page 575](#)
- [Example: Verifying Certificate Validity on page 589](#)
- [Digital Certificates Configuration Overview on page 574](#)

## CA Profiles

---

- [Understanding Certificate Authority Profiles on page 577](#)
- [Example: Configuring a CA Profile on page 577](#)

### Understanding Certificate Authority Profiles

A certificate authority (CA) profile configuration contains information specific to a CA. You can have multiple CA profiles on the device. For example, you might have one profile for Microsoft and one for Entrust. Each profile is associated with a CA certificate. If you want to load a new CA certificate without removing the older one, you must create a new CA profile (for example, Microsoft-2008).



**NOTE:** The following CAs are supported: Entrust, Microsoft, and Verisign. SCEP only supports the Microsoft CA.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Certificates and PKI on page 569](#)
- [Example: Configuring a CA Profile on page 577](#)

### Example: Configuring a CA Profile

This example shows how to configure a CA profile.

- [Requirements on page 577](#)
- [Overview on page 578](#)
- [Configuration on page 578](#)
- [Verification on page 578](#)

#### Requirements

---

No special configuration beyond device initialization is required before configuring this feature.

## Overview

---

In this example, you create a CA profile called `ca-profile-ipsec` with CA identity `microsoft-2008`. The configuration specifies that the CRL be refreshed every 48 hours, and the location to retrieve the CRL is `http://www.my-ca.com`. Within the example, you set the enrollment retry value to 20. (The default retry value is 10.)

Automatic certificate polling is set to every 30 minutes. If you configure retry only without configuring a retry interval, then the default retry interval is 900 seconds (or 15 minutes). If you do not configure retry or a retry interval, then there is no polling.

## Configuration

---

### Step-by-Step Procedure

To configure a CA profile:

1. Create a CA profile.

[edit]

```
user@host# set security pki ca-profile ca-profile-ipsec ca-identity microsoft-2008
revocation-check crl refresh-interval 48 url http://www.my-ca.com/my-crl.crl
```

2. Specify the enrollment retry value.

[edit]

```
user@host# set security pki ca-profile ca-profile-ipsec enrollment retry 20
```

3. Specify the time interval in seconds between attempts to automatically enroll the CA certificate online.

[edit]

```
user@host# set security pki ca-profile ca-profile-ipsec enrollment retry-interval
1800
```

4. If you are done configuring the device, commit the configuration.

[edit]

```
user@host# commit
```

## Verification

---

To verify the configuration is working properly, enter the `show security pki` command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Certificate Authority Profiles on page 577](#)
- [Digital Certificates Configuration Overview on page 574](#)

## Certificate Enrollment

---

- [Understanding Online CA Certificate Enrollment on page 579](#)
- [Enrolling a CA Certificate Online Using SCEP on page 579](#)
- [Example: Enrolling a Local Certificate Online Using SCEP on page 580](#)
- [Example: Using SCEP to Automatically Renew a Local Certificate on page 581](#)



## Understanding Online CA Certificate Enrollment

With Simple Certificate Enrollment Protocol (SCEP), you can configure your Juniper Networks device to obtain a certificate authority (CA) certificate online and start the online enrollment for the specified certificate ID. The CA public key verifies certificates from remote peers.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Public Key Cryptography on page 575
- Understanding Certificates and PKI on page 569
- Enrolling a CA Certificate Online Using SCEP on page 579
- Example: Enrolling a Local Certificate Online Using SCEP on page 580

## Enrolling a CA Certificate Online Using SCEP

Before you begin:

1. Generate a public and private key pair. See “Example: Generating a Public-Private Key Pair” on page 576.
2. Create a CA profile. See “Example: Configuring a CA Profile” on page 577.

To enroll a CA certificate online:

1. Retrieve the CA certificate online using SCEP. (The attributes required to reach the CA server are obtained from the defined CA profile.)

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile-ipsec
```

The command is processed synchronously to provide the fingerprint of the received CA certificate.

Fingerprint:

```
e6:fa:d6:da:e8:8d:d3:00:e8:59:12:e1:2c:b9:3c:c0:9d:6c:8f:8d (sha1)
```

```
82:e2:dc:ea:48:4c:08:9a:fd:b5:24:b0:db:c3:ba:59 (md5)
```

```
Do you want to load the above CA certificate ? [yes,no]
```

2. Confirm that the correct certificate is loaded. The CA certificate is loaded only when you type **yes** at the CLI prompt.

For more information on the certificate, such as the bit length of the key pair, use the command **show security pki ca-certificate** described in the *Junos OS CLI Reference*.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Online CA Certificate Enrollment on page 579
- Digital Certificates Configuration Overview on page 574
- Example: Enrolling a Local Certificate Online Using SCEP on page 580
- Example: Using SCEP to Automatically Renew a Local Certificate on page 581

## Example: Enrolling a Local Certificate Online Using SCEP

This example shows how to enroll a local certificate online.

- Requirements on page 580
- Overview on page 580
- Configuration on page 581
- Verification on page 581

### Requirements

---

Before you begin:

- Generate a public and private key pair. See “Example: Generating a Public-Private Key Pair” on page 576.
- Configure a certificate authority profile. See “Example: Configuring a CA Profile” on page 577.
- Enroll the CA certificate. See “Enrolling a CA Certificate Online Using SCEP” on page 579.

### Overview

---

In this example, you configure your Juniper Networks device to obtain a local certificate online and start the online enrollment for the specified certificate ID with SCEP. You specify the CA profile name as `ca-profile-ipsec` and the CA location as `http://10.155.8.1/certsrv/mscep/mscep.dll`.

You will use the `request security pki local-certificate enroll` command to start the online enrollment for the specified certificate ID. You must specify the CA profile name (for example, `ca-profile-ipsec`), the certificate ID corresponding to a previously generated key-pair (for example, `qqq`), and the following information:



**NOTE:** SCEP sends a PKCS #10 format certificate request enveloped in PKCS #7 format.

- The challenge CA password for certificate enrollment and revocation—for example, `aaa`. If the CA does not provide the challenge password, then choose your own password.
- At least one of the following values:
  - The domain name to identify the certificate owner in IKE negotiations—for example, `qqq.juniper.net`.
  - The identity of the certificate owner for IKE negotiation with the e-mail statement—for example, `qqq@juniper.net`.
  - The IP address if the device is configured for a static IP address—for example, `10.10.10.10`.

- Specify the subject name in the distinguished name format in quotation marks, including the domain component (DC), common name (CN), organizational unit name (OU), organization name (O), locality (L), state (ST), and country (C).

Once the device certificate is obtained and the online enrollment begins for the certificate ID. The command is processed asynchronously.

### Configuration

#### Step-by-Step Procedure

To enroll a local certificate online:

1. Specify the CA profile.

[edit]

```
user@host# set security pki ca-profile ca-profile-ipsec enrollment url
http://10.155.8.1/certsrv/mscep/mscep.dll
```

2. If you are done configuring the device, commit the configuration.

[edit]

```
user@host# commit
```

3. Initiate the enrollment process by running the operational mode command.

```
user@host> request security pki local-certificate enroll ca-profile ca-profile-ipsec
certificate-id qqj challenge-password aaa domain-name qqj.juniper.net email
qqj@juniper.net ip-address 10.10.10.10 subject DC=juniper, CN=rout3,
OU=marketing, O=juniper, L=sunnyvale, ST=california, C=us
```

### Verification

To verify the configuration is working properly, enter the **show security pki** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Online CA Certificate Enrollment on page 579](#)
- [Digital Certificates Configuration Overview on page 574](#)
- [Enrolling a CA Certificate Online Using SCEP on page 579](#)
- [Example: Using SCEP to Automatically Renew a Local Certificate on page 581](#)

### Example: Using SCEP to Automatically Renew a Local Certificate

This example shows how to renew the local certificates automatically using SCEP.

- [Requirements on page 581](#)
- [Overview on page 582](#)
- [Configuration on page 582](#)
- [Verification on page 583](#)

### Requirements

Before you begin:

- Obtain a certificate either on line or manually. See “Enabling Digital Certificates Online: Configuration Overview” on page 574.
- Obtain a local certificate. See “Example: Enrolling a Local Certificate Online Using SCEP” on page 580.

## Overview

You can enable the device to automatically renew certificates that were acquired by online enrollment or loaded manually. Automatic certificate renewal saves you from having to remember to renew certificates on the device before they expire, and helps to maintain valid certificates at all times.

Automatic certificate renewal is disabled by default. You can enable automatic certificate renewal and configure the device to automatically send out a request to renew a certificate before it expires. You can set the time when you want the device to send out the certificate renewal request, in number of days and minutes before the expiration date. By setting different times for the certificates, you prevent the device from having to renew all the certificates at the same time.

For this feature to work, the device must be able to reach the SCEP server, and the certificate must be present on the device during the renewal process. Furthermore, you must also ensure that the CA issuing the certificate can return the same DN. The CA must not modify the subject name or alternate subject name extension in the new certificate.

In this example, you can enable and disable automatic SCEP certificate renewal either for all SCEP certificates or on a per-certificate basis. You set the **security pki auto-re-enrollment** command to enable and configure certificate reenrollment. You specify the certificate ID of the CA certificate as `sm1` and set the CA profile name associated with the certificate to `aaa`. You set the challenge password for CA certificate to `abc`. This password must be the same one configured previously for the CA. You also set the trigger time for the reenrollment to 10. During automatic reenrollment, by default, the Juniper Networks device uses the existing key pair. To generate a new key pair, use the **re-generate-keypair** command.

## Configuration

### Step-by-Step Procedure

To enable and configure local certificate reenrollment:

1. To enable and configure certificate reenrollment.

```
[edit]
user@host# set security pki auto-re-enrollment certificate-id ca-ipsec
ca-profile-name ca-profile-ipsec challenge-password abc
re-enroll-trigger-time-percentage 10 re-generate-keypair
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

## Verification

To verify the configuration is working properly, enter the **show security pki local-certificate detail** operational mode command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Online CA Certificate Enrollment on page 579
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 590
- Enrolling a CA Certificate Online Using SCEP on page 579
- Example: Enrolling a Local Certificate Online Using SCEP on page 580

## Certificate Requests

- Understanding Local Certificate Requests on page 583
- Example: Manually Generating a CSR for the Local Certificate and Sending it to the CA Server on page 584

### Understanding Local Certificate Requests

When you create a local certificate request, the device generates a CA certificate in PKCS #10 format from a key pair you previously generated using the same certificate ID.

A subject name is associated with the local certificate request in the form of a common name (CN), organizational unit (OU), organization (O), locality (L), state (ST), country (C), and domain component (DC). Additionally, a subject alternative name is associated in the following form:

- IP address
- E-mail address
- Fully qualified domain name (FQDN)



**NOTE:** Some CAs do not support an e-mail address as the domain name in a certificate. If you do not include an e-mail address in the local certificate request, you cannot use an e-mail address as the local IKE ID when configuring the device as a dynamic peer. Instead, you can use a fully qualified domain name (if it is in the local certificate), or you can leave the local ID field empty. If you do not specify a local ID for a dynamic peer, enter the *hostname.domain-name* of that peer on the device at the other end of the IPsec tunnel in the peer ID field.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Certificates and PKI on page 569
- Example: Manually Generating a CSR for the Local Certificate and Sending it to the CA Server on page 584

## Example: Manually Generating a CSR for the Local Certificate and Sending it to the CA Server

This example shows how to generate a certificate signing request manually.

- Requirements on page 584
- Overview on page 584
- Configuration on page 584
- Verification on page 584

### Requirements

Before you begin:

1. Generate a public and private key. See “Example: Generating a Public-Private Key Pair” on page 576.

### Overview

In this example, you generate a certificate request using the certificate ID of a public-private key pair you previously generated (ca-ipsec). Then you specify the domain name (juniper.net) and the associated common name (abc). The certificate request is displayed in PEM format.

You copy the generated certificate request and paste it into the appropriate field at the CA website to obtain a local certificate. (Refer to the CA server documentation to determine where to paste the certificate request.) When the PKCS #10 content is displayed, the MD5 hash and SHA-1 hash of the PKCS #10 file is also displayed.

### Configuration

#### Step-by-Step Procedure

To generate a local certificate manually:

1. Specify certificate ID, domain name, and common name.

```
user@host> request security pki generate-certificate-request certificate-id ca-ipsec
domain-name juniper.net subject CN=abc
```

### Verification

To view the certificate signing request, enter the **show security pki certificate-request detail** command.

```
Certificate identifier: ca-ipsec
Certificate version: 1
Issued to: CN = abc
Public key algorithm: rsaEncryption(1024 bits)
30:81:89:02:81:81:00:da:ea:cd:3a:49:1f:b7:33:3c:c5:50:fb:57
de:17:34:1c:51:9b:7b:1c:e9:1c:74:86:69:a4:36:77:13:a7:10:0e
52:f4:2b:52:39:07:15:3f:39:f5:49:d6:86:70:4b:a6:2d:73:b6:68
39:d3:6b:f3:11:67:ee:b4:40:5b:f4:de:a9:a4:0e:11:14:3f:96:84
03:3c:73:c7:75:f5:c4:c2:3f:5b:94:e6:24:aa:e8:2c:54:e6:b5:42
c7:72:1b:25:ca:f3:b9:fa:7f:41:82:6e:76:8b:e6:d7:d2:93:9b:38
fe:fd:71:01:2c:9b:5e:98:3f:0c:ed:a9:2b:a7:fb:02:03:01:00:01
Fingerprint:
```

```
0f:e6:2e:fc:6d:52:5d:47:6e:10:1c:ad:a0:8a:4c:b7:cc:97:c6:01 (sha1)
f8:e6:88:53:52:c2:09:43:b7:43:9c:7a:a2:70:98:56 (md5)
```

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- `show security pki certificate-request` in the *Junos OS CLI Reference*
- Understanding Local Certificate Requests on page 583
- Digital Certificates Configuration Overview on page 574

## Certificate Loading

---

- Understanding Certificate Loading on page 585
- Example: Loading CA and Local Certificates Manually on page 585

## Understanding Certificate Loading

After you download certificates from a CA, you transfer them to the device (for example, using FTP), and then load them.

You can load the following certificate files onto a device running Junos OS:

- A local or end-entity (EE) certificate that identifies your local device. This certificate is your public key.
- A CA certificate that contains the CA's public key.
- A CRL that lists any certificates revoked by the CA.



**NOTE:** You can load multiple EE certificates onto the device.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Certificates and PKI on page 569
- Example: Loading CA and Local Certificates Manually on page 585

## Example: Loading CA and Local Certificates Manually

This example shows how to load CA and local certificates manually.

- Requirements on page 585
- Overview on page 586
- Configuration on page 586
- Verification on page 586

### Requirements

---

Before you begin:

- Generate a public-private key pair. See “Example: Generating a Public-Private Key Pair” on page 576.
- Create a CA profile. See “Understanding Certificate Authority Profiles” on page 577.



**NOTE:** CA Profile is only required for the CA certificate and not for the local certificate

- Generate a certificate request. See “Example: Manually Generating a CSR for the Local Certificate and Sending it to the CA Server” on page 584.

### Overview

In this example, you download the local.cert and ca.cert certificates and save them to the /var/tmp/ directory on the device.

### Configuration

#### Step-by-Step Procedure

To load the certificate files onto a device:

1. Load the local certificate.
 

```
[edit]
user@host> request security pki local-certificate load certificate-id local.cert
filename /var/tmp/local.cert
```
2. Load the CA certificate.
 

```
[edit]
user@host> request security pki ca-certificate load ca-profile ca-profile-ipsec
filename /var/tmp/ca.cert
```
3. Examine the fingerprint of the CA certificate, if it is correct for this CA certificate say yes to accept.

### Verification

To verify the certificates loaded properly, enter the **show security pki local-certificate** and **show security pki ca-certificate** commands in operational mode.

```
Fingerprint:
e8:bf:81:6a:cd:26:ad:41:b3:84:55:d9:10:c4:a3:cc:c5:70:f0:7f (sha1)
19:b0:f8:36:e1:80:2c:30:a7:31:79:69:99:b7:56:9c (md5)
Do you want to load this CA certificate ? [yes,no] (no) yes
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Certificate Loading on page 585
- Digital Certificates Configuration Overview on page 574
- Example: Using SCEP to Automatically Renew a Local Certificate on page 581
- Example: Verifying Certificate Validity on page 589
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 590



## CRLs

---

- Understanding Certificate Revocation Lists on page 587
- Example: Manually Loading a CRL onto the Device on page 587
- Example: Verifying Certificate Validity on page 589
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 590
- Deleting a Loaded CRL (CLI Procedure) on page 591

### Understanding Certificate Revocation Lists

In the normal course of business, certificates are revoked for various reasons. You might wish to revoke a certificate if you suspect that it has been compromised, for example, or when a certificate holder leaves the company.

You can manage certificate revocations and validations in two ways:

- Locally— This is a limited solution.
- By referencing a Certificate Authority (CA) certificate revocation list (CRL)— You can automatically access the CRL online at intervals you specify or at the default interval set by the CA.

In Phase 1 negotiations, participants check the CRL list to see if certificates received during an IKE exchange are still valid. If a CRL did not accompany a CA certificate and is not loaded on the device, the device tries to download it automatically from the CRL distribution point of the local certificate. If the device fails to connect to the URL in the certificate distribution point (CDP), it tries to retrieve the CRL from the URL configured in the CA profile.

If the certificate does not contain a certificate distribution point extension, and you cannot automatically retrieve the CRL through Lightweight Directory Access Protocol (LDAP) or Hypertext Transfer Protocol (HTTP), you can retrieve a CRL manually and load that in the device.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Certificates and PKI on page 569
- Example: Manually Loading a CRL onto the Device on page 587
- Example: Verifying Certificate Validity on page 589
- Deleting a Loaded CRL (CLI Procedure) on page 591
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 590

### Example: Manually Loading a CRL onto the Device

This example shows how to load a CRL manually onto the device.

- Requirements on page 588
- Overview on page 588

- Configuration on page 588
- Verification on page 588

### Requirements

Before you begin:

1. Generate a public and private key pair. See “Example: Generating a Public-Private Key Pair” on page 576.
2. Generate a certificate request. See “Example: Manually Generating a CSR for the Local Certificate and Sending it to the CA Server” on page 584.
3. Configure a certificate authority (CA) profile. See “Example: Configuring a CA Profile” on page 577.
4. Load your certificate onto the device. See “Example: Loading CA and Local Certificates Manually” on page 585.

### Overview

You can load a CRL manually, or you can have the device load it automatically, when you verify certificate validity. To load a CRL manually, you obtain the CRL from a CA and transfer it to the device (for example, using FTP).

In this example, you load a CRL certificate called **revoke.crl** from the `/var/tmp` directory on the device. The CA profile is called **ca-profile-ipsec**. (Maximum file size is 5 MB.)



**NOTE:** If a CRL is already loaded into the ca-profile the command `clear security pki crl ca-profile ca-profile-ipsec` must be run first to clear the old CRL.

### Configuration

#### Step-by-Step Procedure

To load a CRL certificate manually:

1. Load a CRL certificate.

[edit]

```
user@host> request security pki crl load ca-profile ca-profile-ipsec filename
/var/tmp/revoke.crl
```



**NOTE:** Junos OS supports loading of CA certificates in X509, PKCS #7, DER, or PEM formats.

### Verification

To verify the configuration is working properly, enter the **show security pki crl** operational mode command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

- Understanding Certificate Revocation Lists on page 587
- Digital Certificates Configuration Overview on page 574
- Example: Verifying Certificate Validity on page 589
- Example: Configuring a Certificate Authority Profile with CRL Locations on page 590
- Deleting a Loaded CRL (CLI Procedure) on page 591

## Example: Verifying Certificate Validity

This example shows how to verify the validity of a certificate.

- Requirements on page 589
- Overview on page 589
- Configuration on page 589
- Verification on page 590

### Requirements

No special configuration beyond device initialization is required before configuring this feature.

### Overview

In this example, you verify certificates manually to find out whether a certificate has been revoked or whether the CA certificate used to create a local certificate is no longer present on the device.

When you verify certificates manually, the device uses the CA certificate (**ca-cert**) to verify the local certificate (**local.cert**). If the local certificate is valid, and if **revocation-check** is enabled in the CA profile, the device verifies that the CRL is loaded and valid. If the CRL is not loaded and valid, the device downloads the new CRL.

For CA-issued certificates or CA certificates, a DNS must be configured in the device's configuration. The DNS must be able to resolve the host in the distribution CRL and in the CA cert/revocation list url in the ca-profile configuration. Additionally, you must have network reachability to the same host in order for the checks to receive.

### Configuration

#### Step-by-Step Procedure

To manually verify the validity of a certificate:

1. Verify the validity of a local certificate.
 

```
[edit]
user@host> request security pki local-certificate verify certificate-id local.cert
```
2. Verify the validity of a CA certificate.
 

```
[edit]
user@host> request security pki ca-certificate verify ca-profile ca-profile-ipsec
```



NOTE: The associated private key and the signature are also verified.

### Verification

To verify the configuration is working properly, enter the **show security pki ca-profile** command.



NOTE: If an error is returned instead of a positive verification the failure is logged in `pkid`.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Certificate Revocation Lists on page 587](#)
- [Example: Manually Loading a CRL onto the Device on page 587](#)
- [Example: Configuring a Certificate Authority Profile with CRL Locations on page 590](#)
- [Deleting a Loaded CRL \(CLI Procedure\) on page 591](#)

### Example: Configuring a Certificate Authority Profile with CRL Locations

This example shows how to configure a certificate authority profile with CRL locations.

- [Requirements on page 590](#)
- [Overview on page 591](#)
- [Configuration on page 591](#)
- [Verification on page 591](#)

#### Requirements

Before you begin:

1. Generate a key pair in the device. See “[Example: Generating a Public-Private Key Pair](#)” on page 576.
2. Create a CA profile or profiles containing information specific to a CA. See “[Example: Configuring a CA Profile](#)” on page 577.
3. Obtain a personal certificate from the CA. See “[Example: Manually Generating a CSR for the Local Certificate and Sending it to the CA Server](#)” on page 584.
4. Load the certificate onto the device. See “[Example: Loading CA and Local Certificates Manually](#)” on page 585.
5. Configure automatic reenrollment. See “[Example: Configuring SecurID User Authentication](#)” on page 411.
6. If necessary, load the certificate's CRL on the device. See “[Example: Manually Loading a CRL onto the Device](#)” on page 587.

## Overview

In Phase 1 negotiations, you check the CRL list to see if the certificate that you received during an IKE exchange is still valid. If a CRL did not accompany a CA certificate and is not loaded on the device, Junos OS tries to retrieve the CRL through the LDAP or HTTP CRL location defined within the CA certificate itself. If no URL address is defined in the CA certificate, the device uses the URL of the server that you define for that CA certificate. If you do not define a CRL URL for a particular CA certificate, the device gets the CRL from the URL in the CA profile configuration.



**NOTE:** The CRL distribution point extension (.cdp) in an X509 certificate can be added to either an HTTP URL or an LDAP URL.

In this example, you direct the device to check the validity of the CA profile called **my\_profile** and, if a CRL did not accompany a CA certificate and is not loaded on the device, to retrieve the CRL from the URL **http://abc/abc-crl.crl**.

## Configuration

### Step-by-Step Procedure

To configure certificate using CRL:

1. Specify the CA profile and URL.

```
[edit]
user@host# set security pki ca-profile my_profile revocation-check url
http://abc/abc-crl.crl
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

## Verification

To verify the configuration is working properly, enter the **show security pki** operational mode command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Certificate Revocation Lists on page 587](#)
- [Example: Manually Loading a CRL onto the Device on page 587](#)
- [Example: Verifying Certificate Validity on page 589](#)
- [Deleting a Loaded CRL \(CLI Procedure\) on page 591](#)
- [Deleting Certificates \(CLI Procedure\) on page 595](#)

## Deleting a Loaded CRL (CLI Procedure)

You can choose to delete a loaded CRL if you no longer need to use it to manage certificate revocations and validation.

Use the following command to delete a loaded certificate revocation list:

```
user@host> clear security pki crl ca-profile (ca-profile | all)
```

Specify a CA profile to delete a CRL associated with the CA identified by the profile, or use **all** to delete all CRLs.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Certificate Revocation Lists on page 587](#)
- [Example: Manually Loading a CRL onto the Device on page 587](#)
- [Example: Verifying Certificate Validity on page 589](#)
- [Example: Configuring a Certificate Authority Profile with CRL Locations on page 590](#)

## Self-Signed Certificates

---

- [Understanding Self-Signed Certificates on page 592](#)
- [Using Automatically Generated Self-Signed Certificates \(CLI Procedure\) on page 593](#)
- [Example: Manually Generating Self-Signed Certificates on page 594](#)

### Understanding Self-Signed Certificates

A self-signed certificate is a certificate that is signed by its creator rather than by a Certificate Authority (CA).

Self-signed certificates allow for use of SSL-based (Secure Sockets Layer) services without requiring that the user or administrator to undertake the considerable task of obtaining an identity certificate signed by a CA.



**NOTE:** Self-signed certificates do not provide additional security as do those generated by CAs. This is because a client cannot verify that the server he or she has connected to is the one advertised in the certificate.

This topic includes the following sections:

- [Generating Self-Signed Certificates on page 592](#)
- [Automatically Generating Self-Signed Certificates on page 593](#)
- [Manually Generating Self-Signed Certificates on page 593](#)

### Generating Self-Signed Certificates

---

Junos OS provides two methods for generating a self-signed certificate:

- Automatic generation

In this case, the creator of the certificate is the Juniper Networks device. An automatically generated self-signed certificate is configured on the device by default.

After the device is initialized, it checks for the presence of an automatically generated self-signed certificate. If it does not find one, the device generates one and saves it in the file system.

- Manual generation

In this case, you create the self-signed certificate for the device.

At any time, you can use the CLI to generate a self-signed certificate. These certificates are also used to gain access to SSL services.

Self-signed certificates are valid for five years from the time they were generated.

### Automatically Generating Self-Signed Certificates

An automatically generated self-signed certificate allows for use of SSL-based services without requiring that the administrator obtain an identity certificate signed by a CA.

A self-signed certificate that is automatically generated by the device is similar to a Secure Shell (SSH) host key. It is stored in the file system, not as part of the configuration. It persists when the device is rebooted, and it is preserved when a **request system snapshot** command is issued.

### Manually Generating Self-Signed Certificates

A self-signed certificate that you manually generate allows for use of SSL-based services without requiring that you obtain an identity certificate signed by a CA. A manually generated self-signed certificate is one example of a public key infrastructure (PKI) local certificate. As is true of all PKI local certificates, manually generated self-signed certificates are stored in the file system.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Certificates and PKI on page 569
- Using Automatically Generated Self-Signed Certificates (CLI Procedure) on page 593
- Example: Manually Generating Self-Signed Certificates on page 594

## Using Automatically Generated Self-Signed Certificates (CLI Procedure)

After the device is initialized, it checks for the presence of a self-signed certificate. If a self-signed certificate is not present, the device automatically generates one.

You can add the following statement to your configuration if you want to use the automatically generated self-signed certificate to provide access to HTTPS services:

```
system {
  services {
    web-management {
      http {
        interface [ ... ];
      } https {
        system-generated-certificate;
        interface [ ... ];
      }
    }
  }
}
```

```
}  
}
```

The device uses the following distinguished name for the automatically generated certificate:

**“CN=<device serial number>, CN=system generated, CN=self-signed”**

Use the following command to specify that the automatically generated self-signed certificate is to be used for Web management HTTPS services:

```
user@host# set system services web-management https system-generated-certificate
```

Use the following operational command to delete the automatically generated self-signed certificate:

```
user@host# clear security pki local-certificate system-generated
```

After you delete the system-generated self-signed certificate, the device automatically generates a new one and saves it in the file system.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Self-Signed Certificates on page 592](#)
- [Digital Certificates Configuration Overview on page 574](#)
- [Example: Manually Generating Self-Signed Certificates on page 594](#)

### Example: Manually Generating Self-Signed Certificates

This example shows how to generate self-signed certificates manually.

- [Requirements on page 594](#)
- [Overview on page 594](#)
- [Configuration on page 595](#)
- [Verification on page 595](#)

#### Requirements

---

Before you begin, generate a public private key pair. See “Example: Generating a Public-Private Key Pair” on page 576

#### Overview

---

For a manually generated self-signed certificate, you specify the DN when you create it. For an automatically generated self-signed certificate, the system supplies the DN, identifying itself as the creator.

In this example, you generate a self-signed certificate with the e-mail address as **mholmes@juniper.net**. You specify a certificate-id of **self-cert** to be referenced by web management, which refers a key-pair of the same certificate-id.



## Configuration

---

### Step-by-Step Procedure

To generate the self-signed certificate manually:

1. Create the self-signed certificate.

```
user@host> request security pki local-certificate generate-self-signed certificate-id
self-cert subject CN=abc domain-name Juniper.net ip-address 1.2.3.4 email
mholmes@juniper.net
```

## Verification

---

To verify the certificate was properly generated and loaded, enter the **show security pki local-certificate** operational mode command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Self-Signed Certificates on page 592
- Digital Certificates Configuration Overview on page 574
- Using Automatically Generated Self-Signed Certificates (CLI Procedure) on page 593

## Deleting Certificates (CLI Procedure)

---

You can delete a local or trusted CA certificate that is automatically or manually generated.

Use the following command to delete a local certificate:

```
user@host> clear security pki local certificate certificate-id (certificate-id | all |
system-generated )
```

Specify a certificate ID to delete a local certificate with a specific ID, use **all** to delete all local certificates, or specify **system-generated** to delete the automatically generated self-signed certificate.

When you delete an automatically generated self-signed certificate, the device generates a new one.

To delete a CA certificate:

```
user@host> clear security pki ca-certificate ca-profile (ca-profile-name | all)
```

Specify a CA profile to delete a specific CA certificate, or use **all** to delete all CA certificates present in the persistent store.



**NOTE:** You are asked for confirmation before a CA certificate can be deleted.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Digital Certificates Configuration Overview on page 574



## CHAPTER 20

# Dynamic VPNs

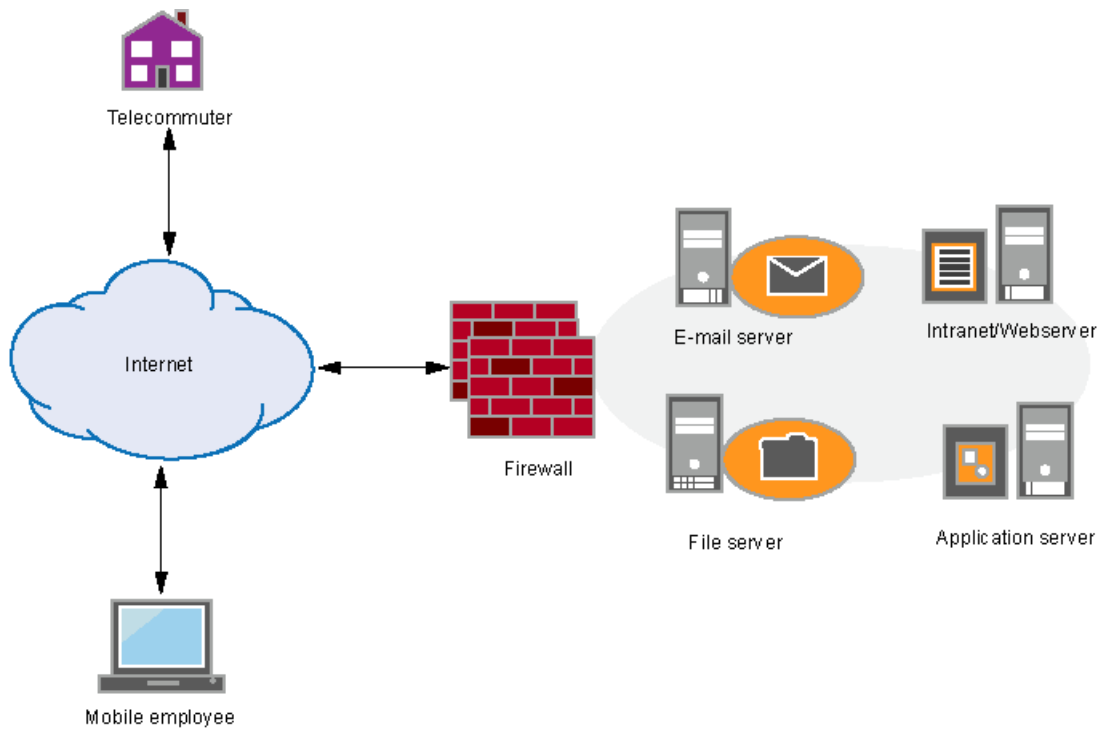
- [Dynamic VPN Overview on page 597](#)
- [Understanding Remote Client Access to the VPN on page 599](#)
- [Dynamic VPN Configuration on page 599](#)
- [Local Authentication and Address Assignment on page 612](#)
- [Dynamic VPN Proposal Sets on page 616](#)
- [Group and Shared IKE IDs on page 617](#)
- [Junos Pulse Client for Dynamic VPN Access on page 637](#)
- [Access Manager Client-Side Reference on page 646](#)

### Dynamic VPN Overview

---

Virtual private network (VPN) tunnels enable users to securely access assets such as e-mail servers and application servers that reside behind a firewall. End-to-site VPN tunnels are particularly helpful to remote users such as telecommuters because a single tunnel enables access to all of the resources on a network—the users do not need to configure individual access settings to each application and server. See Figure 58 on page 598.

Figure 58: Using a VPN Tunnel to Enable Remote Access to a Corporate Network



The dynamic VPN feature further simplifies remote access by enabling users to establish Internet Protocol Security (IPsec) VPN tunnels without having to manually configure VPN settings on their PCs or laptops. Instead, authenticated users can simply download the VPN client software to their computers. This Layer 3 remote access client uses client-side configuration settings that it receives from the server to create and manage a secure end-to-site VPN tunnel to the server.



**NOTE:** If more than two simultaneous user connections are required, a dynamic VPN license must be installed. The dynamic VPN feature is disabled by default on the device. To enable dynamic VPN, you must configure the feature using the `dynamic-vpn` configuration statement at the [edit security] hierarchy level. See the [Junos OS Administration Guide for Security Devices](#) for information about installing and managing licenses.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Dynamic VPN Configuration Overview on page 601
- Understanding Dynamic VPN Tunnels on page 600
- Understanding Remote Client Access to the VPN on page 599
- Access Manager Client-Side System Requirements on page 647

---

## Understanding Remote Client Access to the VPN

---

A common dynamic VPN deployment is to provide VPN access to remote clients connected through a public network such as the Internet. IPsec access is provided through a gateway on the Juniper Networks device. The VPN client software is distributed to remote clients through a Web portal. A remote client accesses the Web portal and, after being authenticated, downloads and installs the VPN client software.

The following describes the process for a remote client to access the VPN:

1. The remote client contacts the Web portal by establishing an HTTP or HTTPS connection to the interface on the SRX Series device that is configured to terminate the VPN tunnels.
2. The remote client is redirected to the Web portal for authentication, where users are prompted to enter their credentials.
3. Upon successful authentication, the server determines if client software is installed in the remote client and if the software is the most recent version. If the remote client does not have the client software installed or the installed software is an older version, new software is installed in the remote client. The client software is launched and a new authentication takes place.
4. Upon success authentication, the remote client downloads the latest configuration options from the server. This ensures that the remote client always has the most recent configuration when it attempts to build a tunnel.
5. A new authentication is performed using IPsec extended authentication (XAuth). An IP address is assigned to the remote client from a local address pool or through an external RADIUS server. Upon successful authentication and address assignment, a tunnel is established.

After VPN software is installed on the remote client, the user can access the VPN by either logging in to the Web portal or launching the client software directly. In either case, the remote client authenticates with the Juniper Networks device and downloads the latest available configuration for the client.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Dynamic VPN Overview on page 597](#)
- [Dynamic VPN Configuration Overview on page 601](#)
- [Understanding Dynamic VPN Tunnels on page 600](#)
- [Example: Configuring Dynamic VPN on page 603](#)

---

## Dynamic VPN Configuration

---

- [Understanding Dynamic VPN Tunnels on page 600](#)
- [Dynamic VPN Configuration Overview on page 601](#)
- [Example: Configuring Dynamic VPN on page 603](#)

## Understanding Dynamic VPN Tunnels

Dynamic VPN tunnels are configured in the same way as traditional IPsec VPN tunnels. However, not all IPsec VPN options are supported.

The following list describes the requirements and supported options when configuring dynamic VPN tunnels:

- Only policy-based VPNs are supported. Route-based VPNs are not supported with dynamic VPN tunnels. Traffic allowed from the VPN can be controlled by pushing routes to the remote client as part of the client's configuration.
- Dynamic VPN tunnels must be configured with extended authentication (XAuth). This can be done using local authentication or an external RADIUS server. XAuth is required to obtain username and password information during IPsec negotiation and to push an IP address to the remote client. For local authentication, the IP addresses assigned to remote clients can be drawn from a local pool. Optionally, DNS and WINS server addresses may also be pushed to the remote client.
- Only preshared keys are supported for Phase 1 authentication with dynamic VPN tunnels. The same preshared key can be used for all remote clients because a different username and password is assigned to each remote client.
- When a dynamic VPN client negotiates an AutoKey IKE tunnel with a preshared key, aggressive mode must be used. Therefore, you must always configure aggressive mode with dynamic VPN tunnels.
- Shared or group IKE IDs can be used to configure a single VPN that is shared by all remote clients. When a single VPN is shared, the total number of simultaneous connections to the gateway cannot be larger than the number of dynamic VPN licenses installed. When configuring a shared or group IKE ID gateway, you can configure the maximum number of connections to be larger than the number of installed dynamic VPN licenses. However, if a new connection will exceed the number of licensed connections, the connection will be denied.
- The dynamic VPN client supports the following algorithms: MD5, SHA-1, DES, 3DES, AES (with 96-bit, 128-bit, and 256-bit keys). The dynamic VPN client supports DH groups 1,2, and 5. Tunnel negotiations will fail if other values are configured on the Juniper Networks device.
- Either proposal sets or custom proposals may be configured for IKE and IPsec negotiations. If there is a list of custom proposals referenced from the IKE or IPsec policy, only the first proposal is sent to the client and other proposals in the list are ignored.
- The same access profile should be used for both IKE and dynamic VPN tunnels. Doing so avoids unpredictable behavior if the tunnel goes down unexpectedly or the client crashes.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Dynamic VPN Overview on page 597](#)
- [Dynamic VPN Configuration Overview on page 601](#)

- Example: Configuring Dynamic VPN on page 603
- Understanding IKE and IPsec Packet Processing on page 458

## Dynamic VPN Configuration Overview

A dynamic VPN allows administrators to provide IPsec access to a gateway on a Juniper Networks device while also providing a way to distribute the Dynamic VPN software to remote clients through the use of a Web portal.

The following procedure lists the tasks for configuring a dynamic VPN.

1. Configure authentication and address assignment for the remote clients:
  - a. Configure an XAuth profile to authenticate users and assign addresses. Either local authentication or an external RADIUS server may be used. Use the **profile** configuration statement at the [**edit access**] hierarchy level to configure the XAuth profile.  
  
To use the XAuth profile for Web authentication, use the **web-authentication** configuration statement at the [**edit access firewall-authentication**] hierarchy level.
  - b. Assign IP addresses from a local address pool if local authentication is used. Use the **address-assignment pool** configuration statement at the [**edit access**] hierarchy level. A subnet or a range of IP addresses can be specified. IP addresses for DNS and WINS servers may also be specified.
2. Configure the VPN tunnel:
  - a. Configure the IKE policy. The mode must be aggressive. Basic, compatible, or standard proposal sets may be used. Only preshared keys are supported for Phase 1 authentication. Use the **policy** configuration statement at the [**edit security ike**] hierarchy level.
  - b. Configure the IKE gateway. Either shared or group IKE IDs can be used. You can configure the maximum number of simultaneous connections to the gateway. Use the **gateway** configuration statement at the [**edit security ike**] hierarchy level.
  - c. Configure the IPsec VPN. Basic, compatible, or standard proposal sets may be specified with the **policy** configuration statement at the [**edit security ipsec**] hierarchy level. Use the **vpn** configuration statement at the [**edit security ipsec**] hierarchy level to configure the IPsec gateway and policy.
  - d. Configure a security policy to allow traffic from the remote clients to the IKE gateway. Use the **policy** configuration statement at the [**edit security policies from-zone zone to-zone zone**] hierarchy level.



**NOTE:** The placement of this security policy is important. It needs to be placed above more specific, non-VPN policies so that traffic that is intended to be sent over the VPN tunnel is processed correctly.

- e. Configure host inbound traffic to allow specific traffic to reach the device from systems that are connected to its interfaces. For example, IKE and HTTPS traffic must be allowed. See “Understanding How to Control Inbound Traffic Based on Traffic Types” on page 116.
  - f. (Optional) If the client address pool belongs to a subnet that is directly connected to the device, the device would need to respond to ARP requests to addresses in the pool from other devices in the same zone. Use the **proxy-arp** configuration statement at the [**edit security nat**] hierarchy level. Specify the interface that directly connects the subnet to the device and the addresses in the pool.
3. Associate the dynamic VPN with remote clients:
    - a. Specify the access profile for use with dynamic VPN. Use the **access-profile** configuration statement at the [**edit security dynamic-vpn**] hierarchy level.
    - b. Configure the clients who can use the dynamic VPN. Specify protected resources (traffic to the protected resource travels through the specified dynamic VPN tunnel and is therefore protected by the firewall's security policies) or exceptions to the protected resources list (traffic that does not travel through the dynamic VPN tunnel and is sent in cleartext). These options control the routes that are pushed to the client when the tunnel is up, therefore controlling the traffic that is sent through the tunnel. Use the **clients** configuration statement at the [**edit security dynamic-vpn**] hierarchy level.



**NOTE:** The Web portal requires that HTTPS is enabled on the Juniper Networks device. If HTTPS is already enabled for J-Web access, no further action is required. Otherwise, use the **https** configuration statement at the [**edit system services web-management**] hierarchy level to enable HTTPS. To enable HTTPS for dynamic VPN access without allowing J-Web access, do not specify an interface for J-Web access.



**NOTE:** If users will log in to the server by running the Access Manager Web client software instead of connecting to the server through HTTP/HTTPS, use the **force-upgrade** configuration statement at the [**edit security dynamic-vpn**] hierarchy level. This configuration automatically upgrades the client's software when a more recent version is available. If you do not enable this option, the user is given a choice to manually upgrade the client's software when a more recent version is available.

If users will run the Junos Pulse VPN client software, we do not recommend that you use the **force-upgrade** configuration statement.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Dynamic VPN Overview on page 597](#)



- Understanding Dynamic VPN Tunnels on page 600
- Example: Configuring Dynamic VPN on page 603

## Example: Configuring Dynamic VPN

This example shows how to configure a dynamic VPN on a Juniper Networks device to provide VPN access to remote clients.

- Requirements on page 603
- Overview on page 603
- Configuration on page 606
- Verification on page 611

### Requirements

---

Before you begin:

1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 111.
3. If there will be more than two simultaneous user connections, install a Dynamic VPN license in the device. See the *Junos OS Administration Guide for Security Devices*.
4. Read “Dynamic VPN Configuration Overview” on page 601.

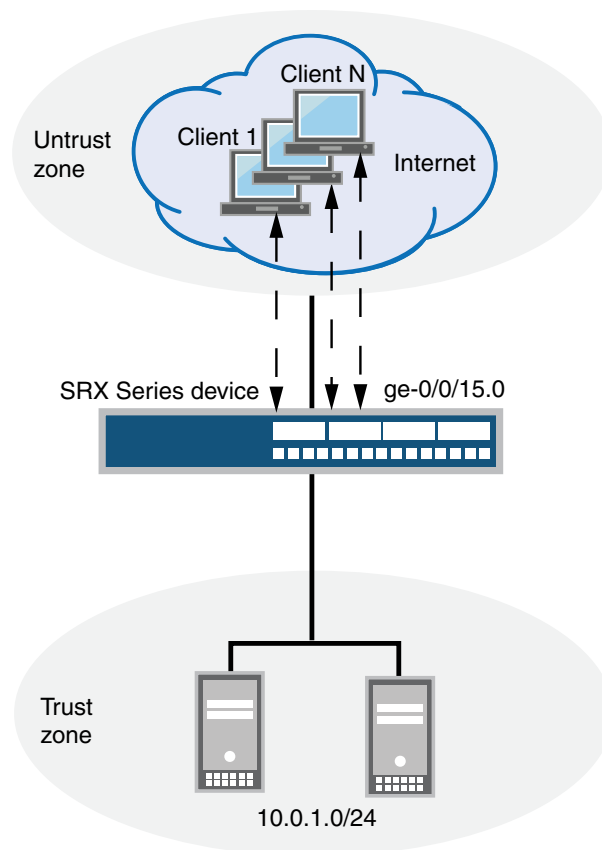
### Overview

---

A common deployment scenario for dynamic VPN is to provide VPN access to remote clients that are connected through a public network such as the Internet. A public IP address is assigned to one of the gateway’s interfaces; this interface is normally part of the untrust zone. Remote clients can access the VPN through a Web portal and, after being authenticated, can download and install the VPN client software. After the client software is installed, the remote user can access the VPN by either logging in to the Web portal or by launching the client directly. In either case, the remote client authenticates with the SRX Series device and downloads the latest configuration available.

Figure 59 on page 604 illustrates this deployment topology. The ge-0/0/15.0 interface on the SRX Series device is the termination point for the dynamic VPN tunnel. Remote clients in the untrust zone access the ge-0/0/15.0 interface through an HTTP or HTTPS connection.

Figure 59: Dynamic VPN Deployment Topology



In this example, XAuth client authentication is performed locally and client IP addresses are assigned from an address pool configured on the SRX Series device. See Table 63 on page 604.

Then, standard proposal sets are used for both IKE and IPsec negotiations. For dynamic VPN tunnels, aggressive mode must be configured and only preshared keys are supported for Phase 1 authentication. A group IKE ID is used and the maximum number of connections is set to 10. Because dynamic VPNs must be policy-based VPNs, a security policy must be configured to forward traffic to the tunnel. IKE and HTTPS traffic must be allowed for host inbound traffic. See Table 64 on page 605.

Finally, the XAuth profile configured for remote clients is specified for the dynamic VPN. Remote users are associated with the configured IPsec VPN. Also configured are remote protected resources (the destination addresses of traffic that is always sent through the tunnel) and remote exceptions (the destination addresses of traffic that is sent in cleartext instead of through the tunnel). See Table 65 on page 605.

Table 63: Remote Client Authentication and Address Assignment Configuration

Feature	Name	Configuration Parameters
IP address pool	dyn-vpn-address-pool	<ul style="list-style-type: none"> <li>Addresses: 10.10.10.0/24</li> <li>DNS server address: 4.2.2.2/32.</li> </ul>

Table 63: Remote Client Authentication and Address Assignment Configuration (*continued*)

Feature	Name	Configuration Parameters
XAuth profile	dyn-vpn-access-profile	<ul style="list-style-type: none"> <li>Remote client username: 'client1' with password \$9\$uY4o0EyMWxdwgX7</li> <li>Remote client username: 'client2' with password \$9\$neNM9CuB1hyrv5Q39</li> <li>IP address pool reference: dyn-vpn-address-pool</li> <li>This profile is the default profile for web authentication.</li> </ul>

Table 64: VPN Tunnel Configuration Parameters

Feature	Name	Configuration Parameters
IKE policy (Phase 1)	ike-dyn-vpn-policy	<ul style="list-style-type: none"> <li>Mode: aggressive</li> <li>Proposal set: standard</li> <li>Preshared key: (ASCII) \$9\$KHxWXNs2aikPdbkP5Q9CKM8</li> </ul>
IKE gateway (Phase 1)	dyn-vpn-local-gw	<ul style="list-style-type: none"> <li>IKE policy reference: ike-dyn-vpn-policy</li> <li>Dynamic hostname: dynvpn</li> <li>IKE user type: group IKE ID</li> <li>Maximum number of concurrent connections: 10</li> <li>External interface: ge-0/0/15.0</li> <li>Access profile reference: dyn-vpn-access-profile</li> </ul>
IPsec policy (Phase 2)	ipsec-dyn-vpn-policy	Proposal set: standard
IPsec VPN (Phase 2)	dyn-vpn	<ul style="list-style-type: none"> <li>IKE gateway reference: dyn-vpn-local-gw</li> <li>IPsec policy reference: ipsec-dyn-vpn-policy</li> </ul>
Security policy (permits traffic from the untrust zone to the trust zone)	dyn-vpn-policy	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source address any</li> <li>destination address any</li> <li>application any</li> </ul> </li> <li>Permit action: tunnel ipsec-vpn dyn-vpn</li> </ul>
Host inbound traffic		<p>Allow the following types of traffic to the ge-0/0/15.0 interface in the untrust zone:</p> <ul style="list-style-type: none"> <li>IKE</li> <li>HTTPS</li> <li>ping</li> <li>SSH</li> </ul>

Table 65: Dynamic VPN Configuration for Remote Clients

Feature	Name	Configuration Parameters
Access profile for remote clients		Access profile reference: dyn-vpn-access-profile

Table 65: Dynamic VPN Configuration for Remote Clients (*continued*)

Feature	Name	Configuration Parameters
Remote clients	all	<ul style="list-style-type: none"> <li>IPsec VPN reference: dyn-vpn</li> <li>User name reference: client1 and client2</li> <li>Remote protected resources: 10.0.0.0/8</li> <li>Remote exceptions: 0.0.0.0/0</li> </ul>

### Configuration

- Configuring the Remote User Authentication and Address Assignment on page 606
- Configuring the VPN Tunnel on page 607
- Associate the Dynamic VPN with Remote Clients on page 610

#### *Configuring the Remote User Authentication and Address Assignment*

#### CLI Quick Configuration

To quickly configure remote user authentication and address assignment, copy the following commands and paste them into the CLI:

```
[edit]
set access profile dyn-vpn-access-profile client client1 firewall-user password
"$9$uY4o0EyMWxdwgX7"
set access profile dyn-vpn-access-profile client client2 firewall-user password
"$9$neNM9CuB1hyrv5Q39"
set access profile dyn-vpn-access-profile address-assignment pool dyn-vpn-address-pool
set access address-assignment pool dyn-vpn-address-pool family inet network
10.10.10.0/24
set access address-assignment pool dyn-vpn-address-pool family inet xauth-attributes
primary-dns 4.2.2.2/32
set access firewall-authentication web-authentication default-profile
dyn-vpn-access-profile
```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure remote user authentication and address assignment:

- Create the address assignment pool.

```
[edit access address-assignment]
user@host# set pool dyn-vpn-address-pool family inet network 10.10.10.0/24
user@host# set pool dyn-vpn-address-pool family inet xauth-attributes primary-dns
4.2.2.2/32
```

- Configure the XAuth profile.

```
[edit access]
user@host# set profile dyn-vpn-access-profile client client1 firewall-user password
"$9$uY4o0EyMWxdwgX7"
user@host# set profile dyn-vpn-access-profile client client2 firewall-user password
"$9$neNM9CuB1hyrv5Q39"
user@host# set profile dyn-vpn-access-profile address-assignment pool
dyn-vpn-address-pool
```

- Configure Web authentication using the XAuth profile.

```
[edit access firewall-authentication]
user@host# set web-authentication default-profile dyn-vpn-access-profile
```

**Results** From configuration mode, confirm your configuration by entering the **show access** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show access
profile dyn-vpn-access-profile {
  client client1 {
    firewall-user {
      password "$9$uY4o0EyMWxdwgX7"; ## SECRET-DATA
    }
  }
  client client2 {
    firewall-user {
      password "$9$neNM9CuB1hyrv5Q39"; ## SECRET-DATA
    }
  }
  address-assignment {
    pool dyn-vpn-address-pool;
  }
}
address-assignment {
  pool dyn-vpn-address-pool {
    family inet {
      network 10.10.10.0/24;
      xauth-attributes {
        primary-dns 4.2.2.2/32;
      }
    }
  }
}
firewall-authentication {
  web-authentication {
    default-profile dyn-vpn-access-profile;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### *Configuring the VPN Tunnel*

**CLI Quick Configuration** To quickly configure the VPN tunnel, copy the following commands and paste them into the CLI:

```
[edit]
set security ike policy ike-dyn-vpn-policy mode aggressive
set security ike policy ike-dyn-vpn-policy proposal-set standard
set security ike policy ike-dyn-vpn-policy pre-shared-key ascii-text
"$9$KHxWXNs2aikPdbkP5Q9CKM8"
set security ike gateway dyn-vpn-local-gw ike-policy ike-dyn-vpn-policy
set security ike gateway dyn-vpn-local-gw dynamic hostname dynvpn
```

```

set security ike gateway dyn-vpn-local-gw dynamic connections-limit 10
set security ike gateway dyn-vpn-local-gw dynamic ike-user-type group-ike-id
set security ike gateway dyn-vpn-local-gw external-interface ge-0/0/15.0
set security ike gateway dyn-vpn-local-gw xauth access-profile dyn-vpn-access-profile
set security ipsec policy ipsec-dyn-vpn-policy proposal-set standard
set security ipsec vpn dyn-vpn ike gateway dyn-vpn-local-gw
set security ipsec vpn dyn-vpn ike ipsec-policy ipsec-dyn-vpn-policy
set security policies from-zone untrust to-zone trust policy dyn-vpn-policy match
  source-address any
set security policies from-zone untrust to-zone trust policy dyn-vpn-policy match
  destination-address any
set security policies from-zone untrust to-zone trust policy dyn-vpn-policy match
  application any
set security policies from-zone untrust to-zone trust policy dyn-vpn-policy then permit
  tunnel ipsec-vpn dyn-vpn
set security zones security-zone untrust interfaces ge-0/0/15.0 host-inbound-traffic
  system-services ike
set security zones security-zone untrust interfaces ge-0/0/15.0 host-inbound-traffic
  system-services https
set security zones security-zone untrust interfaces ge-0/0/15.0 host-inbound-traffic
  system-services ping
set security zones security-zone untrust interfaces ge-0/0/15.0 host-inbound-traffic
  system-services ssh

```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see [Using the CLI Editor in Configuration Mode](#).

To configure the VPN tunnel:

1. Configure the IKE policy.

```

[edit security ike]
user@host# set policy ike-dyn-vpn-policy mode aggressive
user@host# set policy ike-dyn-vpn-policy proposal-set standard
user@host# set policy ike-dyn-vpn-policy pre-shared-key ascii-text
"$9$KHxWXNs2aikPdbkP5Q9CKM8"

```

2. Configure the IKE gateway.

```

[edit security ike]
user@host# set gateway dyn-vpn-local-gw ike-policy ike-dyn-vpn-policy
user@host# set gateway dyn-vpn-local-gw dynamic hostname dynvpn
user@host# set gateway dyn-vpn-local-gw dynamic ike-user-type group-ike-id
user@host# set gateway dyn-vpn-local-gw dynamic connections-limit 10
user@host# set gateway dyn-vpn-local-gw external-interface ge-0/0/15.0
user@host# set gateway dyn-vpn-local-gw xauth access-profile
  dyn-vpn-access-profile

```

3. Configure IPsec.

```

[edit security ipsec]
user@host# set policy ipsec-dyn-vpn-policy proposal-set standard
user@host# set vpn dyn-vpn ike gateway dyn-vpn-local-gw
user@host# set vpn dyn-vpn ike ipsec-policy ipsec-dyn-vpn-policy

```

4. Configure the security policy.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy dyn-vpn-policy match source-address any destination-address
any application any
user@host# set policy dyn-vpn-policy then permit tunnel ipsec-vpn dyn-vpn
```

5. Configure host inbound traffic.

```
[edit security zones security-zone untrust interfaces ge-0/0/15.0]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services https
user@host# set host-inbound-traffic system-services ping
user@host# set host-inbound-traffic system-services ssh
```

**Results** From configuration mode, confirm your configuration by entering the **show security ike**, **show security ipsec**, **show security policies**, and **show security zones** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security ike
policy ike-dyn-vpn-policy {
  mode aggressive;
  proposal-set standard;
  pre-shared-key ascii-text "$9$KHxWXNs2aikPdbkP5Q9CKM8"; ## SECRET-DATA
}
gateway dyn-vpn-local-gw {
  ike-policy ike-dyn-vpn-policy;
  dynamic {
    hostname dynvpn;
    connections-limit 10;
    ike-user-type group-ike-id;
  }
  external-interface ge-0/0/15.0;
  xauth access-profile dyn-vpn-access-profile;
}

[edit]
user@host# show security ipsec
policy ipsec-dyn-vpn-policy {
  proposal-set standard;
}
vpn dyn-vpn {
  ike {
    gateway dyn-vpn-local-gw;
    ipsec-policy ipsec-dyn-vpn-policy;
  }
}

[edit]
user@host# show security policies
policy dyn-vpn-policy {
  match {
    source-address any;
    destination-address any;
    application any;
  }
}
```

```

then {
  permit {
    tunnel {
      ipsec-vpn dyn-vpn;
    }
  }
}
[edit]
user@host# show security zones
security-zone untrust {
  interfaces {
    ge-0/0/15.0 {
      host-inbound-traffic {
        system-services {
          ike;
          https;
          ping;
          ssh;
        }
      }
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### *Associate the Dynamic VPN with Remote Clients*

#### **CLI Quick Configuration**

To quickly associate the dynamic VPN with remote clients, copy the following commands and paste them into the CLI:

```

[edit]
set security dynamic-vpn access-profile dyn-vpn-access-profile
set security dynamic-vpn clients all remote-protected-resources 10.0.0.0/8
set security dynamic-vpn clients all remote-exceptions 0.0.0.0/0
set security dynamic-vpn clients all ipsec-vpn dyn-vpn
set security dynamic-vpn clients all user client1
set security dynamic-vpn clients all user client2

```

#### **Step-by-Step Procedure**

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To associate the dynamic VPN with remote clients:

1. Specify the access profile to use with dynamic VPN.

```

[edit security dynamic-vpn]
user@host# set access-profile dyn-vpn-access-profile

```

2. Configure the clients who can use the dynamic VPN.

```

[edit security dynamic-vpn]
user@host# set clients all ipsec-vpn dyn-vpn
user@host# set clients all user client1
user@host# set clients all user client2

```



```

user@host# set clients all remote-protected-resources 10.0.0.0/8
user@host# set clients all remote-exceptions 0.0.0.0/0

```

**Results** From configuration mode, confirm your configuration by entering the **show security dynamic-vpn** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show security dynamic-vpn
access-profile dyn-vpn-access-profile;
clients {
  all {
    remote-protected-resources {
      10.0.0.0/8;
    }
    remote-exceptions {
      0.0.0.0/0;
    }
    ipsec-vpn dyn-vpn;
    user {
      client1;
      client2;
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

Dynamic VPN tunnels can be monitored with the same commands used to monitor traditional IPsec VPN tunnels. To confirm that the configuration is working properly, perform these tasks:

- Verifying IKE Phase 1 Status on page 611
- Verifying Connected Clients and Assigned Addresses on page 611
- Verifying IPsec Phase 2 Status on page 612
- Verifying Concurrent Connections and Parameters for Each User on page 612

#### Verifying IKE Phase 1 Status

**Purpose** Verify the IKE Phase 1 status of the security associations.

**Action** From operational mode, enter the **show security ike security-associations** command.

```

user@host> show security ike security-associations

```

Index	Remote Address	State	Initiator cookie	Responder cookie	Mode
18	172.19.100.99	UP	37b45aa1469e488b	7d4454404002e2e6	Aggressive

#### Verifying Connected Clients and Assigned Addresses

**Purpose** Verify that the remote clients and the IP addresses assigned to them are using XAuth.

**Action** From operational mode, enter the **show security ike active-peer** command.

```
user@host> show security ike active-peer
Remote Address      Port    Peer IKE-ID      XAUTH username    Assigned
IP
172.19.100.99      500     testdynvpn       test               10.10.10.2
```

#### *Verifying IPsec Phase 2 Status*

**Purpose** Verify the IPsec Phase 2 status of the security associations.

**Action** From operational mode, enter the **show security ipsec security-associations** command.

```
user@host> show security ipsec security-associations
Total active tunnels: 1
ID      Gateway      Port Algorithm      SPI      Life:sec/kb Mon vsys
<133955586 172.19.100.99 500 ESP:aes-128/sha1 9c23b7a9 2862/ 449996 - root
>133955586 172.19.100.99 500 ESP:aes-128/sha1 c72c8f88 2862/ 449996 - root
```

#### *Verifying Concurrent Connections and Parameters for Each User*

**Purpose** Verify the number of concurrent connections and the negotiated parameters for each user.

**Action** From operational mode, enter the **show security dynamic-vpn users** command.

```
user@host> show security dynamic-vpn users
User: test , Number of connections: 1
Remote IP: 172.19.100.99
IPSEC VPN: dyn-vpn
IKE gateway: dyn-vpn-local-gw
IKE ID : testdynvpn
IKE Lifetime: 28800
IPSEC Lifetime: 3600
Status: CONNECTED
```

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Dynamic VPN Overview on page 597](#)
  - [Understanding Dynamic VPN Tunnels on page 600](#)
  - [Dynamic VPN Configuration Overview on page 601](#)

## Local Authentication and Address Assignment

- [Understanding Local Authentication and Address Assignment on page 612](#)
- [Example: Configuring Local Authentication and Address Pool on page 613](#)

### Understanding Local Authentication and Address Assignment

A client application can request an IP address on behalf of a client. This request is made at the same time as the client authentication request. Upon successful authentication of the client, an IP address can be assigned to the client from a predefined address pool

or a specific IP address can be assigned. Other attributes, such as WINS or DNS server IP addresses, can also be provided to the client.

Address pools are defined with the **pool** configuration statement at the **[edit access address-assignment]** hierarchy level. An address pool definition contains network information (IP address with optional netmask), optional range definitions, and DHCP or XAuth attributes that can be returned to the client. If all addresses in a pool are assigned, a new request for a client address will fail even if the client is successfully authenticated.

Access profiles are defined with the **profile** configuration statement at the **[edit access]** hierarchy. A defined address pool can be referenced in an access profile configuration.

You can also bind a specific IP address to a client in an access profile with the **xauth ip-address address** option. The IP address must be in the range of addresses specified in the address pool. It must also be a different from the IP address specified with the **host** configuration statement at the **[edit access profile address-assignment pool pool-name family inet]** hierarchy level. If a client that is currently bound to a specific IP address logs in again with the same credentials, it is assigned an IP address from the address pool.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Configuring Local Authentication and Address Pool on page 613](#)
- [Dynamic VPN Overview on page 597](#)
- [Understanding Dynamic VPN Tunnels on page 600](#)
- [Dynamic VPN Configuration Overview on page 601](#)
- [Example: Configuring Dynamic VPN on page 603](#)

### Example: Configuring Local Authentication and Address Pool

This example shows how to create an address pool and how to assign client IP addresses in an access profile.

#### Requirements

Before you begin, configure primary and secondary DNS and WINS servers and assign IP addresses to them.

#### Overview

This example creates an address pool **xauth1** that consists of the IP addresses in the 40.0.0.0/24 subnet. The **xauth1** pool also assigns IP addresses for primary and secondary DNS and WINS servers.

The access profile **dvpn-auth** references the **xauth1** pool. The **dvpn-auth** access profile configures two clients:

- **jason**: The IP address 40.0.0.1 is bound to this client. Upon successful authentication, the client is assigned the IP address 40.0.0.1. If the client logs in again before logging out, the client is assigned an IP address from the **xauth1** pool.
- **jacky**: Upon successful authentication, the client is assigned an IP address from the **xauth1** pool.

In addition, the **dvpn-auth** access profile specifies that password authentication is used to verify clients at login. Additional authentication methods may be specified; the software tries the authentication methods in order, from first to last, for each client login attempt.

### Configuration

**CLI Quick Configuration** To quickly configure an address pool and an access profile that uses the address pool, copy the following commands and paste them into the CLI.

```
[edit]
set access profile dvpn-auth authentication-order password
set access profile dvpn-auth client jacky firewall-user password "$9$VusgJGUHmPQ249p"
set access profile dvpn-auth client jason xauth ip-address 40.0.0.1/32
set access profile dvpn-auth client jason firewall-user password "$9$Q1Y53/tu0lcrvp0vL"
set access profile dvpn-auth address-assignment pool xauth1
set access address-assignment pool xauth1 family inet network 40.0.0.0/24
set access address-assignment pool xauth1 family inet xauth-attributes primary-dns
  40.0.0.250/32
set access address-assignment pool xauth1 family inet xauth-attributes secondary-dns
  40.0.0.251/32
set access address-assignment pool xauth1 family inet xauth-attributes primary-wins
  40.0.0.253/32
set access address-assignment pool xauth1 family inet xauth-attributes secondary-wins
  40.0.0.254/32
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure an address pool and an access profile that uses the address pool:

1. Create the address pool.

```
[edit access address-assignment]
user@host# set pool xauth1 family inet network 40.0.0.0/24 xauth-attributes
  primary-dns 40.0.0.250 secondary-dns 40.0.0.251 primary-wins 40.0.0.253
  secondary-wins 40.0.0.254
```

2. Configure the access profile.

```
[edit access]
user@host# set profile dvpn-auth address-assignment pool xauth1
user@host# set profile dvpn-auth authentication-order password
user@host# set profile dvpn-auth client jason xauth ip-address 40.0.0.1
user@host# set profile dvpn-auth client jason firewall-user password jason
user@host# set profile dvpn-auth client jacky firewall-user password jacky
```

**Results** From configuration mode, confirm your configuration by entering the **show access** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

user@host# show access
profile dvpn-auth {
  authentication-order password;
  client jacky {
    firewall-user {
      password "$9$VusgJGUHmPQ249p"; ## SECRET-DATA
    }
  }
  client jason {
    xauth {
      ip-address 40.0.0.1/32;
    }
    firewall-user {
      password "$9$Q1Y53/tu0lcrvp0vL"; ## SECRET-DATA
    }
  }
  address-assignment {
    pool xauth1;
  }
}
address-assignment {
  pool xauth1 {
    family inet {
      network 40.0.0.0/24;
      xauth-attributes {
        primary-dns 40.0.0.250/32;
        secondary-dns 40.0.0.251/32;
        primary-wins 40.0.0.253/32;
        secondary-wins 40.0.0.254/32;
      }
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Address Assignment on page 615

#### **Verifying Address Assignment**

**Purpose** Verify address assignment. For XAuth, the hardware address is always shown as NA. If a static IP address is assigned to a specific user, the user name and profile name (in the format `user@profile`) is displayed in the "Host/User" column. If a client is assigned an IP address from the pool, the username is displayed; if the username does not exist, NA is displayed. For other applications (for example, DHCP), the hostname is displayed if configured; if the hostname is not configured, NA is displayed.

**Action** From operational mode, enter the **show network-access address-assignment pool** command.

```

user
user@host> show network-access address-assignment pool xauth1
IP address      Hardware address      Host/User              Type
40.0.0.1        NA                    jason@dvpn-auth      XAUTH
40.0.0.2        NA                    jacky                  XAUTH

```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Local Authentication and Address Assignment on page 612
- Dynamic VPN Overview on page 597
- Understanding Dynamic VPN Tunnels on page 600
- Dynamic VPN Configuration Overview on page 601
- Example: Configuring Dynamic VPN on page 603

## Dynamic VPN Proposal Sets

Configuring custom Internet Key Exchange (IKE) and IP Security (IPsec) proposals for IKE and IPsec policies can be tedious and time-consuming when there are many dynamic VPN clients. The administrator can select basic, compatible, or standard proposal sets for dynamic VPN clients. Each proposal set consists of two or more predefined proposals. The server selects one predefined proposal from the set and pushes it to the client in the client configuration. The client uses this proposal in negotiations with the server to establish the connection.

The default values for IKE and IPsec security association (SA) rekey timeout are as follows:

- For IKE SAs, the rekey timeout is 28,800 seconds.
- For IPsec SAs, the rekey timeout is 3600 seconds.



**NOTE:** Because proposal-set configuration does not allow for configuration of rekey timeout, these values are included in the client configuration that is sent to the client at client download time.

The basic use cases for proposals are as follows:

- IKE and IPsec both use proposal sets.

The server selects a predefined proposal from the proposal set and sends it to the client, along with the default rekey timeout value.

- IKE uses a proposal set, and IPsec uses a custom proposal.

The server sends a predefined IKE proposal from the configured IKE proposal set to the client, along with the default rekey timeout value. For IPsec, the server sends the setting that is configured in the IPsec proposal.

- IKE uses a custom proposal, and IPsec uses a proposal set.

The server sends a predefined IPsec proposal from the configured IPsec proposal set to the client, along with the default rekey timeout value. For IKE, the server sends the setting that is configured in the IKE proposal.



**NOTE:** If IPsec uses a standard proposal set and perfect forward secrecy (PFS) is not configured, then the default PFS is set as group2. For other proposal sets, PFS will not be set, because it is not configured. Also, for the IPsec proposal set, the group configuration in ipsec policy **perfect-forward-secrecy keys** overrides the Diffie-Hellman (DH) group setting in the proposal sets.

Because the client accepts only one proposal for negotiating tunnel establishment with the server, the server internally selects one proposal from the proposal set to send to the client. The selected proposal for each set is listed as follows:

For IKE

- Sec-level basic: preshared key, g1, des, sha1
- Sec-level compatible: preshared key, g2, 3des, sha1
- Sec-level standard: preshared key, g2, aes128, sha1

For IPsec

- Sec-level basic: esp, no pfs (if not configured) or groupx (if configured), des, sha1
- Sec-level compatible: esp, no pfs (if not configured) or groupx (if configured), 3des, sha1
- Sec-level standard: esp, g2 (if not configured) or groupx (if configured), aes128, sha1

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Dynamic VPN Overview on page 597](#)

## Group and Shared IKE IDs

- [Understanding Group and Shared IKE IDs on page 618](#)
- [Example: Configuring a Group IKE ID for Multiple Users on page 619](#)
- [Example: Configuring Individual IKE IDs for Multiple Users on page 626](#)

## Understanding Group and Shared IKE IDs

With dynamic VPN, a unique Internet Key Exchange (IKE) ID is used for each user connection. When there are a large number of users who need to access the VPN, configuring an individual IKE gateway, IPsec VPN, and a security policy for each user can be cumbersome. The group IKE ID and shared IKE ID features allow a number of users to share an IKE gateway configuration, thus reducing the number of VPN configurations required.



**NOTE:** We recommend that you configure group IKE IDs for dynamic VPN deployments because group IKE IDs provide a unique preshared key and IKE ID for each user.

This topic includes the following sections:

- Group IKE IDs on page 618
- Shared IKE IDs on page 619

### Group IKE IDs

When group IKE IDs are configured, the IKE ID of each user is a concatenation of a user-specific part and a part that is common to all group IKE ID users. For example, the user Bob might use "Bob.juniper.net" as his full IKE ID, where ".juniper.net" is common to all users. The full IKE ID is used to uniquely identify each user connection.

Although group IKE IDs do not require XAuth, XAuth is required by dynamic VPN to retrieve network attributes like client IP addresses. A warning is displayed if XAuth is not configured for a dynamic VPN that uses group IKE IDs.



**NOTE:** We recommend that users use the same credentials for both WebAuth and XAuth authentication when group IKE IDs are configured.

Multiple users can use the same group IKE ID, but a single user cannot use the same group IKE ID for different connections. If a user needs to have connections from different remote clients, they need to have different group IKE IDs configured, one for each connection. If a user only has one group IKE ID configured and attempts a second connection from another PC, the first connection will be terminated to allow the second connection to go through.

To configure a group IKE ID:

- Configure **ike-user-type group-ike-id** at the [edit security ike gateway *gateway-name dynamic*] hierarchy level.
- Configure the **hostname** configuration statement at the [edit security ike gateway *gateway-name dynamic*] hierarchy level. This configuration is the common part of the full IKE ID for all users.



- Configure the **pre-shared-key** configuration statement at the [edit security ike policy *policy-name*] hierarchy level. The configured preshared key is used to generate the actual preshared key.

### Shared IKE IDs

When a shared IKE ID is configured, all users share a single IKE ID and a single IKE preshared key. Each user is authenticated through the mandatory XAuth phase, where the credentials of individual users are verified either with an external RADIUS server or with a local access database. XAuth is required for shared IKE IDs.

The XAuth user name together with the configured shared IKE ID is used to distinguish between different user connections. Because the user name is used to identify each user connection, both the WebAuth user name and XAuth user name must be the same.

Multiple users can use the same shared IKE ID, but a single user cannot use the same shared IKE ID for different connections. If a user needs to have connections from different remote clients, they need to have different shared IKE IDs configured, one for each connection. If a user has only one shared IKE ID configured and attempts a second connection from another client, the first connection will be terminated to allow the second connection to go through. Also, because the user name is needed to identify each user connection along with the IKE ID, the user must use the same credentials for both WebAuth and XAuth authentication.

To configure a shared IKE ID:

- Configure **ike-user-type shared-ike-id** at the [edit security ike gateway *gateway-name dynamic*] hierarchy level.
- Configure the **hostname** configuration statement at the [edit security ike gateway *gateway-name dynamic*] hierarchy level. The configured hostname is shared by all users configured in the dynamic VPN access profile.
- Configure the **pre-shared-key** configuration statement at the [edit security ike policy *policy-name*] hierarchy level. The configured preshared key is shared by all users configured in the dynamic VPN access profile.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Dynamic VPN Tunnels on page 600
- Dynamic VPN Configuration Overview on page 601
- Example: Configuring a Group IKE ID for Multiple Users on page 619
- Example: Configuring Individual IKE IDs for Multiple Users on page 626

### Example: Configuring a Group IKE ID for Multiple Users

This example shows how to configure a group IKE ID that is used by multiple users.

- Requirements on page 620
- Overview on page 620

- Configuration on page 621
- Verification on page 625

### Requirements

Before you begin:

1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 113.
3. If there will be more than two simultaneous user connections, install a Dynamic VPN license in the device. See the *Junos OS Administration Guide for Security Devices*.
4. Read “Dynamic VPN Configuration Overview” on page 601.

### Overview

In this example, you configure two remote dynamic VPN users who use a single IKE ID and a single IKE preshared key (see Table 67 on page 620 and Table 68 on page 621). An external RADIUS server is used to authenticate users and assign IP addresses to clients (see Table 66 on page 620).

**Table 66: RADIUS Server User Authentication (Group IKE ID)**

Feature	Name	Configuration Parameters
XAuth profile	radius-profile	<ul style="list-style-type: none"> <li>• RADIUS is the authentication method used to verify user credentials.</li> <li>• The RADIUS server IP address is 10.100.100.250 and the password is secret.</li> <li>• This profile is the default profile for Web authentication.</li> </ul>

**Table 67: Group IKE ID VPN Tunnel Configuration Parameters**

Feature	Name	Configuration Parameters
IKE policy (Phase 1)	clientpol-group	<ul style="list-style-type: none"> <li>• Mode: aggressive</li> <li>• Proposal set: compatible</li> <li>• Preshared key: (ASCII) for-everyone-in-access-profile</li> </ul>
IKE gateway (Phase 1)	groupgw	<ul style="list-style-type: none"> <li>• IKE policy reference: clientpol-group</li> <li>• Dynamic hostname: juniper.net</li> <li>• IKE user type: group IKE ID</li> <li>• Maximum number of concurrent connections: 50</li> <li>• External interface: ge-0/0/0.0</li> <li>• Access profile reference: radius-profile</li> </ul>
IPsec policy (Phase 2)	clientlvpnPol	Proposal set: compatible
IPsec VPN (Phase 2)	groupvpn	<ul style="list-style-type: none"> <li>• IKE gateway reference: groupgw</li> <li>• IPsec policy reference: clientlvpnPol</li> </ul>

Table 67: Group IKE ID VPN Tunnel Configuration Parameters (*continued*)

Feature	Name	Configuration Parameters
Security policy (permits traffic from the untrust zone to the trust zone)	group-sec-policy	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source address any</li> <li>destination address any</li> <li>application any</li> </ul> </li> <li>Permit action: tunnel ipsec-vpn groupvpn</li> </ul>
Host inbound traffic		<p>Allow the following types of traffic to the ge-0/0/0.0 interface in the untrust zone:</p> <ul style="list-style-type: none"> <li>IKE</li> <li>HTTPS</li> <li>ping</li> <li>SSH</li> </ul>

Table 68: Group IKE ID Dynamic VPN Configuration for Remote Clients

Feature	Name	Configuration Parameters
Access profile for remote clients		Access profile reference: radius-profile
Remote clients	groupcfg	<ul style="list-style-type: none"> <li>IPsec VPN reference: groupvpn</li> <li>User name reference: derek and chris</li> <li>Remote protected resources: 10.100.100.0/24</li> <li>Remote exceptions: 0.0.0.0/0, 1.1.1.1/24, 0.0.0.0/32</li> </ul>

### Configuration

#### CLI Quick Configuration

To quickly configure a group IKE ID for multiple users, copy the following commands and paste them into the CLI.

```
[edit]
set access profile radius-profile authentication-order radius
set access profile radius-profile radius-server 10.100.100.250 secret
"$9$UYHPQ/9pB1h/COREcMW"
set access firewall-authentication web-authentication default-profile radius-profile
set security ike policy clientpol-group mode aggressive
set security ike policy clientpol-group proposal-set compatible
set security ike policy clientpol-group pre-shared-key ascii-text
"$9$w50BESMpxe7WgaDKrCRSvURSeNdyGQZ67aUHmF3JK8Nj0PQGDnQDnc87NbzG"
set security ike gateway groupgw ike-policy clientpol-group
set security ike gateway groupgw dynamic hostname juniper.net
set security ike gateway groupgw dynamic connections-limit 50
set security ike gateway groupgw dynamic ike-user-type group-ike-id
set security ike gateway groupgw external-interface ge-0/0/0.0
set security ike gateway groupgw xauth access-profile radius-profile
set security ipsec policy client1vpnPol proposal-set compatible
set security ipsec vpn groupvpn ike gateway groupgw
set security ipsec vpn groupvpn ike ipsec-policy client1vpnPol
```

```

set security policies from-zone untrust to-zone trust policy group-sec-policy match
  source-address any
set security policies from-zone untrust to-zone trust policy group-sec-policy match
  destination-address any
set security policies from-zone untrust to-zone trust policy group-sec-policy match
  application any
set security policies from-zone untrust to-zone trust policy group-sec-policy then permit
  tunnel ipsec-vpn groupvpn
set security dynamic-vpn access-profile radius-profile
set security dynamic-vpn clients groupcfg remote-protected-resources 10.100.100.0/24
set security dynamic-vpn clients groupcfg remote-exceptions 0.0.0.0/0
set security dynamic-vpn clients groupcfg remote-exceptions 1.1.1.1/24
set security dynamic-vpn clients groupcfg remote-exceptions 0.0.0.0/32
set security dynamic-vpn clients groupcfg ipsec-vpn groupvpn
set security dynamic-vpn clients groupcfg user chris
set security dynamic-vpn clients groupcfg user derek
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services ike
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services https
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services ping
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services ssh

```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see [Using the CLI Editor in Configuration Mode](#).

To configure a group IKE ID for multiple users:

1. Configure the XAuth profile.

```

[edit access]
user@host# set profile radius-profile authentication-order radius
user@host# set profile radius-profile radius-server 10.100.100.250 secret secret
user@host# set firewall-authentication web-authentication default-profile
  radius-profile

```

2. Configure the IKE policy.

```

[edit security ike]
user@host# set policy clientpol-group mode aggressive
user@host# set policy clientpol-group proposal-set compatible
user@host# set policy clientpol-group pre-shared-key ascii-text
  for-everyone-in-access-profile

```

3. Configure the IKE gateway.

```

[edit security ike]
user@host# set gateway groupgw ike-policy clientpol-group
user@host# set gateway groupgw dynamic hostname juniper.net
user@host# set gateway groupgw dynamic ike-user-type group-ike-id
user@host# set gateway groupgw dynamic connections-limit 50
user@host# set gateway groupgw external-interface ge-0/0/0.0
user@host# set gateway groupgw xauth access-profile radius-profile

```

4. Configure IPsec.
 

```
[edit security ipsec]
user@host# set policy client1vpnPol proposal-set compatible
user@host# set vpn groupvpn ike gateway groupgw
user@host# set vpn groupvpn ike ipsec-policy client1vpnPol
```
5. Configure the security policy.
 

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy group-sec-policy match source-address any
destination-address any application any
user@host# set policy group-sec-policy then permit tunnel ipsec-vpn groupvpn
```
6. Configure host inbound traffic.
 

```
[edit security zones security-zone untrust interfaces ge-0/0/0.0]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services https
user@host# set host-inbound-traffic system-services ping
user@host# set host-inbound-traffic system-services ssh
```
7. Specify the access profile to use with dynamic VPN.
 

```
[edit security dynamic-vpn]
user@host# set access-profile radius-profile
```
8. Configure the clients who can use the dynamic VPN.
 

```
[edit security dynamic-vpn]
user@host# set clients groupcfg ipsec-vpn groupvpn
user@host# set clients groupcfg user derek
user@host# set clients groupcfg user chris
user@host# set clients groupcfg remote-protected-resources 10.100.100.0/24
user@host# set clients groupcfg remote-exceptions 0.0.0.0/0
user@host# set clients groupcfg remote-exceptions 1.1.1./24
user@host# set clients groupcfg remote-exceptions 0.0.0.0/32
```

**Results** From configuration mode, confirm your configuration by entering the `show security ike`, `show security ipsec`, `show security policies`, `show security zones`, and `show security dynamic-vpn` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show access
profile radius-profile {
  authentication-order radius;
  radius-server {
    10.100.100.250 secret "$9$UYHPQ/9pB1h/COREcMW"; ## SECRET-DATA
  }
}
firewall-authentication {
  web-authentication {
    default-profile radius-profile;
  }
}
user@host# show security ike
ike {
  policy clientpol-group {
```

```

mode aggressive;
proposal-set compatible;
pre-shared-key ascii-text
"$0$w50BESMpbx7WgDKFrCPSuIPSeNdy6GQ2674dUhmF3yK8Nj0PQGDnc0hc8X7NbzG";
  ## SECRET-DATA
}
gateway groupgw {
ike-policy clientpol-group;
dynamic {
  hostname juniper.net;
  connections-limit 50;
  ike-user-type group-ike-id;
}
external-interface ge-0/0/0.0;
xauth access-profile radius-profile;
}
}
user@host# show security ipsec
ipsec {
  policy clientlvpnPol {
    proposal-set compatible;
  }
  vpn groupvpn {
    ike {
      gateway groupgw;
      ipsec-policy clientlvpnPol;
    }
  }
}
}
user@host# show security policies
policies {
  from-zone untrust to-zone trust {
    policy group-sec-policy {
      match {
        source-address any;
        destination-address any;
        application any;
      }
      then {
        permit {
          tunnel {
            ipsec-vpn groupvpn;
          }
        }
      }
    }
  }
}
}
}
user@host# show security zones
security-zone untrust {
  interfaces {
    ge-0/0/0.0 {
      host-inbound-traffic {
        system-services {
          ike;
          https;
        }
      }
    }
  }
}

```

```

        ping;
        ssh;
    }
}
}
}
}
user@host# show security dynamic-vpn
dynamic-vpn {
  access-profile radius-profile;
  clients {
    groupcfg {
      remote-protected-resources {
        10.100.100.0/24;
      }
      remote-exceptions {
        0.0.0.0/0;
        1.1.1.1/24;
        0.0.0.0/32;
      }
      ipsec-vpn groupvpn;
      user {
        chris;
        derek;
      }
    }
  }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

Dynamic VPN tunnels can be monitored with the same commands used to monitor traditional IPsec VPN tunnels. To confirm that the configuration is working properly, perform these tasks:

- Verifying IKE Phase 1 Status on page 625
- Verifying Connected Clients and Assigned Addresses on page 625
- Verifying IPsec Phase 2 Status on page 626
- Verifying Concurrent Connections and Parameters for Each User on page 626

#### *Verifying IKE Phase 1 Status*

**Purpose** Verify the IKE Phase 1 status of the security associations.

**Action** From operational mode, enter the **show security ike security-associations** command.

#### *Verifying Connected Clients and Assigned Addresses*

**Purpose** Verify that the remote clients and the IP addresses assigned to them are using XAuth.

**Action** From operational mode, enter the **show security ike active-peer** command.

**Verifying IPsec Phase 2 Status**

**Purpose** Verify the IPsec Phase 2 status of the security associations.

**Action** From operational mode, enter the `show security ipsec security-associations` command.

**Verifying Concurrent Connections and Parameters for Each User**

**Purpose** Verify the number of concurrent connections and the negotiated parameters for each user.

**Action** From operational mode, enter the `show security dynamic-vpn users` command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding Dynamic VPN Tunnels](#) on page 600
  - [Dynamic VPN Configuration Overview](#) on page 601
  - [Understanding Group and Shared IKE IDs](#) on page 618

**Example: Configuring Individual IKE IDs for Multiple Users**

This example shows how to configure individual IKE IDs for multiple users.



**NOTE:** When there are a large number of users who need to access the VPN, configuring an individual IKE gateway, IPsec VPN, and a security policy for each user can be cumbersome. The group IKE ID feature allows a number of users to share an IKE gateway configuration, thus reducing the number of VPN configurations required. See “Understanding Group and Shared IKE IDs” on page 618.

- [Requirements](#) on page 626
- [Overview](#) on page 627
- [Configuration](#) on page 629
- [Verification](#) on page 636

**Requirements**

Before you begin:

1. Configure network interfaces on the device. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 113.
3. If there will be more than two simultaneous user connections, install a Dynamic VPN license in the device. See the [Junos OS Administration Guide for Security Devices](#).
4. Read “Dynamic VPN Configuration Overview” on page 601.



## Overview

The following example shows the configuration for two remote dynamic VPN users. For each user, an IKE policy and gateway, IPsec policy and VPN, and a security policy must be configured (see Table 70 on page 627 and Table 71 on page 628). An external RADIUS server is used to authenticate users and assign IP addresses to clients (see Table 69 on page 627).

**Table 69: RADIUS Server User Authentication (Individual IKE ID)**

Feature	Name	Configuration Parameters
XAuth profile	radius-profile	<ul style="list-style-type: none"> <li>RADIUS is the authentication method used to verify user credentials.</li> <li>RADIUS server IP address is 10.100.100.250 and the password is secret.</li> <li>This profile is the default profile for Web authentication.</li> </ul>

**Table 70: Client 1 Configuration Parameters**

Feature	Name	Configuration Parameters
IKE policy (Phase 1)	client1pol	<ul style="list-style-type: none"> <li>Mode: aggressive</li> <li>Proposal set: compatible</li> <li>Preshared key: (ASCII) for-client1</li> </ul>
IKE gateway (Phase 1)	client1gw	<ul style="list-style-type: none"> <li>IKE policy reference: client1pol</li> <li>Dynamic hostname: juniper.net</li> <li>External interface: ge-0/0/0.0</li> <li>Access profile reference: radius-profile</li> </ul>
IPsec policy (Phase 2)	client1vpnPol	Proposal set: compatible
IPsec VPN (Phase 2)	client1vpn	<ul style="list-style-type: none"> <li>IKE gateway reference: client1gw</li> <li>IPsec policy reference: client1vpnPol</li> </ul>
Security policy (permits traffic from the untrust zone to the trust zone)	client1-policy	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source address any</li> <li>destination address any</li> <li>application any</li> </ul> </li> <li>Permit action: tunnel ipsec-vpn client1vpn</li> </ul>
Host inbound traffic		<p>Allow the following types of traffic to the ge-0/0/0.0 interface in the untrust zone:</p> <ul style="list-style-type: none"> <li>IKE</li> <li>HTTPS</li> <li>ping</li> <li>SSH</li> </ul>
Access profile for remote clients		Access profile reference: radius-profile

Table 70: Client 1 Configuration Parameters (*continued*)

Feature	Name	Configuration Parameters
Remote clients	cfg1	<ul style="list-style-type: none"> <li>IPsec VPN reference: client1vpn</li> <li>User name reference: derek</li> <li>Remote protected resources: 10.100.100.0/24</li> <li>Remote exceptions: 0.0.0.0/0, 1.1.1.1/24, 0.0.0.0/32</li> </ul>

Table 71: Client 2 Configuration Parameters

Feature	Name	Configuration Parameters
IKE policy (Phase 1)	client2pol	<ul style="list-style-type: none"> <li>Mode: aggressive</li> <li>Proposal set: compatible</li> <li>Preshared key: (ASCII) for-client2</li> </ul>
IKE gateway (Phase 1)	client2gw	<ul style="list-style-type: none"> <li>IKE policy reference: client2pol</li> <li>Dynamic hostname: juniper.net</li> <li>External interface: ge-0/0/0.0</li> <li>Access profile reference: radius-profile</li> </ul>
IPsec policy (Phase 2)	client2vpnPol	Proposal set: compatible
IPsec VPN (Phase 2)	client2vpn	<ul style="list-style-type: none"> <li>IKE gateway reference: client2gw</li> <li>IPsec policy reference: client2vpnPol</li> </ul>
Security policy (permits traffic from the untrust zone to the trust zone)	client2-policy	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source address any</li> <li>destination address any</li> <li>application any</li> </ul> </li> <li>Permit action: tunnel ipsec-vpn client2vpn</li> </ul>
Host inbound traffic		<p>Allow the following types of traffic to the ge-0/0/0.0 interface in the untrust zone:</p> <ul style="list-style-type: none"> <li>IKE</li> <li>HTTPS</li> <li>ping</li> <li>SSH</li> </ul>
Access profile for remote clients		Access profile reference: radius-profile
Remote clients	cfg2	<ul style="list-style-type: none"> <li>IPsec VPN reference: client2vpn</li> <li>User name reference: chris</li> <li>Remote protected resources: 10.100.100.0/24</li> <li>Remote exceptions: 0.0.0.0/0, 1.1.1.1/24</li> </ul>

## Configuration

- Configuring the XAuth Profile on page 629
- Configuring Client 1 on page 630
- Configuring Client 2 on page 633

### Configuring the XAuth Profile

**CLI Quick Configuration** To quickly configure the XAuth profile for all users, copy the following commands and paste them into the CLI.

```
[edit]
set access profile radius-profile authentication-order radius
set access profile radius-profile radius-server 10.100.100.250 secret
"$9$/or0tBEleWx7VlKX-dbaJ"
set access firewall-authentication web-authentication default-profile radius-profile
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the XAuth profile:

1. Configure the access profile.

```
[edit access]
user@host# set profile radius-profile authentication-order radius
user@host# set profile radius-profile radius-server 10.100.100.250 secret secret
```

2. Configure Web authentication using the XAuth profile.

```
[edit access]
user@host# set firewall-authentication web-authentication default-profile
radius-profile
```

**Results** From configuration mode, confirm your configuration by entering the **show access** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show access
profile radius-profile {
  authentication-order radius;
  radius-server {
    10.100.100.250 secret "$9$/or0tBEleWx7VlKX-dbaJ"; ## SECRET-DATA
  }
}
firewall-authentication {
  web-authentication {
    default-profile radius-profile;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring Client 1

**CLI Quick Configuration** To quickly configure the first user, copy the following commands and paste them into the CLI.

```
[edit]
set security ike policy client1pol mode aggressive
set security ike policy client1pol proposal-set compatible
set security ike policy client1pol pre-shared-key ascii-text
  "$9$AoVYuORx7VsgJW87Vb2GUfTz36ArIMx-weK"
set security ike gateway client1gw ike-policy client1pol
set security ike gateway client1gw dynamic hostname juniper.net
set security ike gateway client1gw external-interface ge-0/0/0.0
set security ike gateway client1gw xauth access-profile radius-profile
set security ipsec policy client1vpnPol proposal-set compatible
set security ipsec vpn client1vpn ike gateway client1gw
set security ipsec vpn client1vpn ike ipsec-policy client1vpnPol
set security policies from-zone untrust to-zone trust policy client1-sec-policy match
  source-address any
set security policies from-zone untrust to-zone trust policy client1-sec-policy match
  destination-address any
set security policies from-zone untrust to-zone trust policy client1-sec-policy match
  application any
set security policies from-zone untrust to-zone trust policy client1-sec-policy then permit
  tunnel ipsec-vpn client1vpn
set security dynamic-vpn access-profile radius-profile
set security dynamic-vpn clients cfg1 remote-protected-resources 10.100.100.0/24
set security dynamic-vpn clients cfg1 remote-exceptions 0.0.0.0/0
set security dynamic-vpn clients cfg1 remote-exceptions 1.1.1.1/24
set security dynamic-vpn clients cfg1 remote-exceptions 0.0.0.0/32
set security dynamic-vpn clients cfg1 ipsec-vpn client1vpn
set security dynamic-vpn clients cfg1 user derek
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services ike
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services https
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services ping
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
  system-services ssh
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure dynamic VPN for a single user:

1. Configure the IKE policy.

```
[edit security ike]
user@host# set policy client1pol mode aggressive
user@host# set policy client1pol proposal-set compatible
user@host# set policy client1pol pre-shared-key ascii-text for-client1
```

2. Configure the IKE gateway.

```
[edit security ike]
user@host# set gateway client1gw ike-policy client1pol
user@host# set gateway client1gw dynamic hostname juniper.net
user@host# set gateway client1gw external-interface ge-0/0/0.0
user@host# set gateway client1gw xauth access-profile radius-profile
```

3. Configure IPsec.

```
[edit security ipsec]
user@host# set policy client1vpnPol proposal-set compatible
user@host# set vpn client1vpn ike gateway client1gw
user@host# set vpn client1vpn ike ipsec-policy client1vpnPol
```

4. Configure the security policy.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy client1-sec-policy match source-address any
destination-address any application any
user@host# set policy client1-sec-policy then permit tunnel ipsec-vpn client1vpn
```

5. Configure host inbound traffic.

```
[edit security zones security-zone untrust interfaces ge-0/0/0.0]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services https
user@host# set host-inbound-traffic system-services ping
user@host# set host-inbound-traffic system-services ssh
```

6. Specify the access profile to use with dynamic VPN.

```
[edit security dynamic-vpn]
user@host# set access-profile radius-profile
```

7. Configure the clients who can use the dynamic VPN.

```
[edit security dynamic-vpn]
user@host# set clients cfg1 ipsec-vpn client1vpn
user@host# set clients cfg1 user derek
user@host# set clients cfg1 remote-protected-resources 10.100.100.0/24
user@host# set clients cfg1 remote-exceptions 0.0.0.0/0
user@host# set clients cfg1 remote-exceptions 1.1.1.1/24
user@host# set clients cfg1 remote-exceptions 0.0.0.0/32
```

**Results** From configuration mode, confirm your configuration by entering the **show security ike**, **show security ipsec**, **show security policies**, **show security zones**, and **show security dynamic-vpn** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show security ike
policy client1pol {
  mode aggressive;
  proposal-set compatible;
  pre-shared-key ascii-text "$9$AoVYuORx7VsgJW87Vb2GUfTz36ArlMx-weK"; ##
  SECRET-DATA
}
gateway client1gw {
  ike-policy client1pol;
  dynamic hostname juniper.net;
  external-interface ge-0/0/0.0;
```

```
        xauth access-profile radius-profile;
    }
user@host# show security ipsec
policy client1vpnPol {
    proposal-set compatible;
}
vpn client1vpn {
    ike {
        gateway client1gw;
        ipsec-policy client1vpnPol;
    }
}
user@host# show security policies
from-zone untrust to-zone trust {
    policy client1-sec-policy {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit {
                tunnel {
                    ipsec-vpn client1vpn;
                }
            }
        }
    }
}
}
user@host# show security zones
security-zone untrust {
    interfaces {
        ge-0/0/0.0 {
            host-inbound-traffic {
                system-services {
                    ike;
                    https;
                    ping;
                    ssh;
                }
            }
        }
    }
}
}
user@host# show security dynamic-vpn
access-profile radius-profile;
clients {
    cfg1 {
        remote-protected-resources {
            10.100.100.0/24;
        }
        remote-exceptions {
            0.0.0.0/0;
            1.1.1.1/24;
            0.0.0.0/32;
        }
    }
}
```

```

ipsec-vpn client1vpn;
user {
    derek;
}
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring Client 2

#### CLI Quick Configuration

To quickly configure the second user, copy the following commands and paste them into the CLI.

```

[edit]
set security ike policy client2pol mode aggressive
set security ike policy client2pol proposal-set compatible
set security ike policy client2pol pre-shared-key ascii-text
"$9$AoVYuORx7VsgJW87Vb2GUfTz36ArIMx-weK"
set security ike gateway client2gw ike-policy client2pol
set security ike gateway client2gw dynamic hostname juniper.net
set security ike gateway client2gw external-interface ge-0/0/0.0
set security ike gateway client2gw xauth access-profile radius-profile
set security ipsec policy client2vpnPol proposal-set compatible
set security ipsec vpn client2vpn ike gateway client2gw
set security ipsec vpn client2vpn ike ipsec-policy client2vpnPol
set security policies from-zone untrust to-zone trust policy client2-sec-policy match
source-address any
set security policies from-zone untrust to-zone trust policy client2-sec-policy match
destination-address any
set security policies from-zone untrust to-zone trust policy client2-sec-policy match
application any
set security policies from-zone untrust to-zone trust policy client2-sec-policy then permit
tunnel ipsec-vpn client1vpn
set security dynamic-vpn access-profile radius-profile
set security dynamic-vpn clients cfg2 remote-protected-resources 10.100.100.0/24
set security dynamic-vpn clients cfg2 remote-exceptions 1.1.1.1/24
set security dynamic-vpn clients cfg2 remote-exceptions 0.0.0.0/32
set security dynamic-vpn clients cfg2 ipsec-vpn client2vpn
set security dynamic-vpn clients cfg2 user chris
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
system-services ike
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
system-services https
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
system-services ping
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic
system-services ssh

```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure dynamic VPN for a single user:

1. Configure the IKE policy.

```
[edit security ike]
user@host# set policy client2pol mode aggressive
user@host# set policy client2pol proposal-set compatible
user@host# set policy client2pol pre-shared-key ascii-text for-client2
```

2. Configure the IKE gateway.

```
[edit security ike]
user@host# set gateway client2gw ike-policy client2pol
user@host# set gateway client2gw dynamic hostname juniper.net
user@host# set gateway client2gw external-interface ge-0/0/0.0
user@host# set gateway client2gw xauth access-profile radius-profile
```

3. Configure IPsec.

```
[edit security ipsec]
user@host# set policy client2vpnPol proposal-set compatible
user@host# set vpn client2vpn ike gateway client2gw
user@host# set vpn client2vpn ike ipsec-policy client2vpnPol
```

4. Configure the security policy.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy client2-sec-policy match source-address any
destination-address any application any
user@host# set policy client2-sec-policy then permit tunnel ipsec-vpn client2vpn
```

5. Configure host inbound traffic.

```
[edit security zones security-zone untrust interfaces ge-0/0/0.0]
user@host# set host-inbound-traffic system-services ike
user@host# set host-inbound-traffic system-services https
user@host# set host-inbound-traffic system-services ping
user@host# set host-inbound-traffic system-services ssh
```

6. Specify the access profile to use with dynamic VPN.

```
[edit security dynamic-vpn]
user@host# set access-profile radius-profile
```

7. Configure the clients who can use the dynamic VPN.

```
[edit security dynamic-vpn]
user@host# set clients cfg2 ipsec-vpn client1vpn
user@host# set clients cfg2 user chris
user@host# set clients cfg2 remote-protected-resources 10.100.100.0/24
user@host# set clients cfg2 remote-exceptions 1.1.1.1/24
user@host# set clients cfg2 remote-exceptions 0.0.0.0/32
```

**Results** From configuration mode, confirm your configuration by entering the **show security ike**, **show security ipsec**, **show security policies**, **show security zones**, and **show security dynamic-vpn** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show security ike
policy client2pol {
  mode aggressive;
  proposal-set compatible;
```



```

pre-shared-key ascii-text "$9$AoVYuORx7VsgJW87Vb2GUfTz36ArlMx-weK"; ##
SECRET-DATA
}
gateway client2gw {
    ike-policy client2pol;
    dynamic hostname juniper.net;
    external-interface ge-0/0/0.0;
    xauth access-profile radius-profile;
}
user@host# show security ipsec
policy client2vpnPol {
    proposal-set compatible;
}
}
vpn client2vpn {
    ike {
        gateway client2gw;
        ipsec-policy client2vpnPol;
    }
}
user@host# show security policies
from-zone untrust to-zone trust {
    policy client2-sec-policy {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit {
                tunnel {
                    ipsec-vpn client2vpn;
                }
            }
        }
    }
}
}
user@host# show security zones
security-zone untrust {
    interfaces {
        ge-0/0/0.0 {
            host-inbound-traffic {
                system-services {
                    ike;
                    https;
                    ping;
                    ssh;
                }
            }
        }
    }
}
}
user@host# show security dynamic-vpn
access-profile radius-profile;
clients {
    cfg2 {
        remote-protected-resources {

```

```
        10.100.100.0/24;
    }
    remote-exceptions {
        1.1.1.1/24;
        0.0.0.0/32;
    }
    ipsec-vpn client2vpn;
    user {
        chris;
    }
}
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

---

Dynamic VPN tunnels can be monitored with the same commands used to monitor traditional IPsec VPN tunnels. To confirm that the configuration is working properly, perform these tasks:

- Verifying IKE Phase 1 Status on page 636
- Verifying Connected Clients and Assigned Addresses on page 636
- Verifying IPsec Phase 2 Status on page 636
- Verifying Concurrent Connections and Parameters for Each User on page 636

#### *Verifying IKE Phase 1 Status*

**Purpose** Verify the IKE Phase 1 status of the security associations.

**Action** From operational mode, enter the **show security ike security-associations** command.

#### *Verifying Connected Clients and Assigned Addresses*

**Purpose** Verify that the remote clients and the IP addresses assigned to them are using XAuth.

**Action** From operational mode, enter the **show security ike active-peer** command.

#### *Verifying IPsec Phase 2 Status*

**Purpose** Verify the IPsec Phase 2 status of the security associations.

**Action** From operational mode, enter the **show security ipsec security-associations** command.

#### *Verifying Concurrent Connections and Parameters for Each User*

**Purpose** Verify the number of concurrent connections and the negotiated parameters for each user.

**Action** From operational mode, enter the **show security dynamic-vpn users** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding Dynamic VPN Tunnels on page 600](#)
  - [Dynamic VPN Configuration Overview on page 601](#)
  - [Understanding Group and Shared IKE IDs on page 618](#)
  - [Example: Configuring a Group IKE ID for Multiple Users on page 619](#)

## Junos Pulse Client for Dynamic VPN Access

---

- [Understanding Junos Pulse Client on page 637](#)
- [Junos Pulse Client Installation Requirements on page 637](#)
- [Deploying Junos Pulse Client Software on page 638](#)
- [Junos Pulse Interface and Connections on page 639](#)
- [Managing Junos Pulse Connections on page 641](#)

### Understanding Junos Pulse Client

Junos Pulse enables secure authenticated network connections to protected resources and services over local and wide area networks. It is a remote access client developed to replace the earlier access client called Juniper Networks Access Manager. You must uninstall the access client before you install the Junos Pulse client.

Junos Pulse supports remote virtual private network (VPN) tunnel connectivity to SRX Series gateways that are running Junos OS. To configure a firewall access environment for Junos Pulse clients, you must configure the VPN settings on the SRX Series gateway and create and deploy a firewall connection on the Junos Pulse client.

For SRX Series devices running Junos OS 10.2 through 10.4, Junos Pulse is supported but must be deployed separately. In Junos OS Release 11.1 and later releases, if the Pulse client does not exist on the client machine, the Pulse client is automatically downloaded and installed when you log into an SRX Series device. If the Pulse client exists on the client machine, you must launch the Pulse client.

- Related Documentation**
- [Dynamic VPN Overview on page 597](#)
  - [Junos Pulse Client Installation Requirements on page 637](#)
  - [Deploying Junos Pulse Client Software on page 638](#)

### Junos Pulse Client Installation Requirements

The Junos Pulse Release 1.0 client software is supported on computers that run Microsoft Windows. Table 1 on page 4 lists the minimum hardware and software requirements to support the Junos Pulse client software.

Table 72: Junos Pulse Client Hardware and Software Requirements

Component	Requirement
Operating system and browser	<ul style="list-style-type: none"> <li>Windows 7 Enterprise 64 bit; Internet Explorer 8.0 (32 bit) and Firefox 3.5</li> <li>Vista Enterprise XP 64 bit; Internet Explorer 7.0, Internet Explorer 8.0, and Firefox 3.0</li> <li>XP Professional XP 32 bit; Internet Explorer 7.0, Internet Explorer 8.0, and Firefox 3.5</li> <li>Windows Vista 32 bit; Internet Explorer 7.0, Internet Explorer 8.0</li> </ul>
CPU	500 MHz
Memory	512 MB of RAM
Available disk space	30 MB minimum of free space



**NOTE:** For increased security, we recommend that you disable the Fast User Switching feature on Windows endpoints. The Fast User Switching feature allows more than one user to log on simultaneously at a single computer. The feature is enabled by default for Windows 7 and Windows Vista and for domain users on Windows XP. With the Fast User Switching feature enabled, all concurrent user sessions on a system can access the current desktop connections to networks and Infranet Controllers. Thus, if one user has a current network connection, other users logged in on the same computer can access the same network connections, which creates a security risk.

#### Related Documentation

- Dynamic VPN Overview on page 597
- Understanding Junos Pulse Client on page 637
- Deploying Junos Pulse Client Software on page 638

## Deploying Junos Pulse Client Software

You must configure the dynamic VPN feature, which is disabled by default on the SRX Series device. You must enable and configure it before you can use it. The dynamic VPN feature secures traffic through your network by passing it through IPsec VPN tunnels. As part of the VPN configuration, you define the client configuration. The client and the settings are downloaded to your users' computers. The users must uninstall the Access Manager before installing the Junos Pulse client. See "Dynamic VPN Configuration Overview" on page 601.

**Junos Pulse Client Installation Overview**-This section describes how to deploy Junos Pulse client software from SRX Series Gateways.

You can deploy Junos Pulse to endpoints from SRX Series devices in the following way:

- **Web Install**—With a Web install, when you log into the access gateway's Web portal using the Dynamic VPN URL, the Pulse client gets downloaded on the client machine. After the Pulse client is downloaded on the client machine, you need to create a firewall connection.



**NOTE:** A Junos Pulse installation causes a restart of active network connections on a Windows endpoint. When a user initiates a Junos Pulse installation through a WAN connection to the Web interface of an access gateway, the user might need to log in to their service provider again to reestablish network connectivity. Users need to be aware of this issue before they begin the installation.

#### Related Documentation

- Dynamic VPN Configuration Overview on page 601
- Understanding Junos Pulse Client on page 637
- Junos Pulse Interface and Connections on page 639
- Managing Junos Pulse Connections on page 641

## Junos Pulse Interface and Connections

- Junos Pulse Interface on page 639
- Junos Pulse Connection Type on page 640
- Junos Pulse Connection Status on page 640
- Junos Pulse Log Files on page 640

### Junos Pulse Interface

The Junos Pulse interface shows your network connections and provides status about your endpoint's connectivity, security, and acceleration. For SRX Series devices, you have the optional WAN acceleration software installed. WAN acceleration software interacts with network devices to optimize application performance when you are connected over wide area networks.

If your Junos Pulse interface shows the Acceleration pane, it means that you are connected to a network device that can improve your application performance over wide area networks through WAN optimization. The acceleration service of Junos Pulse requires no configuration. Junos Pulse automatically discovers Juniper Networks WXC Series Application Acceleration Platforms in the data center and then negotiates a level of service that can be supported by both client and server. If the service is active, a check mark icon appears.

In some circumstances, you might find that you have better network performance for a particular application with WAN optimization turned off.

To enable or disable WAN optimization, on the Junos Pulse Acceleration pane, click Enable or Disable.

- **Connections**—Act on a selected connection: disconnect, edit, or delete. Add a new connection. Forget saved settings for all connections.
- **Logs**—Annotate, set logging level, or save log files.
- **About**—Display version and copyrights.
- **Help**—Display the Help file.
- **Close**—Close the Junos Pulse interface. Note that the program does not disconnect active network connections.

### Junos Pulse Connection Type

---

The connection type you choose when you define a new connection relates to the type of device that provides access to protected network resources. For SRX Series Services Gateway, use Firewall connection type.

If you want to access the SRX Series device through a wireless connection, use the Windows supplicant to a connection to the Junos wireless network and then to connect the SRX Series firewall through Junos Pulse.

### Junos Pulse Connection Status

---

Junos Pulse displays the status of a connection in the Connections pane and in the system tray. A Connections pane icon shows the state of each connection. The connection status is also indicated by the system tray icon.



**NOTE:** You can right-click the system tray icon to open the Junos Pulse interface or to close Junos Pulse.

---

A connection can be in any of the following states:

- No connection.
- Connecting. A connection stays in this state until it fails or succeeds.
- Connected with issues.
- Connection failed.
- Connected.

### Junos Pulse Log Files

---

A Junos Pulse log file tracks information that can help solve connection issues. Logging is a background operation. You do not need to make any changes to your logging environment unless instructed to do so as part of a troubleshooting effort.

To help in a troubleshooting effort, you might be asked to do the following tasks:

- **Annotate the logs**—When you annotate a log file, you insert text that marks the log file at a specific location. For example, to troubleshoot a connection problem, you might be asked to annotate the log file with specific text, attempt the connection that has been failing, and then annotate the log file again. This sequence of events allows

a support representative to search the log file for the text you inserted. The text brackets the entries of the log file that track your connection issue.

- **Set the log level**—The default logging level is Normal. For a troubleshooting operation, you might be asked to change the logging level to Detailed.
- **Save the logs**—The Save As operation gathers all of the log files into a single .zip file and lets you specify where to place the file.

#### Related Documentation

- Understanding Junos Pulse Client on page 637
- Deploying Junos Pulse Client Software on page 638
- Managing Junos Pulse Connections on page 641

## Managing Junos Pulse Connections

- Add a connection on page 641
- Connect to a Network on page 642
- Disconnect from an Active Network on page 643
- View Connection Properties on page 643
- Edit Connection Properties on page 643
- Forget Saved Settings on page 644
- Delete a Connection on page 644
- Troubleshoot a Junos Pulse Connection Issue on page 644
- Annotate Log Files on page 645
- Set Log Level on page 646
- Save Log Files on page 646
- View Component Version Information on page 646

### Add a connection

Each connection in Junos Pulse represents a protected network. Typically, your network administrator defines the connections for you and might disable the Add a Connection feature. If you are required to create new connections, your network administrator will tell you the settings that you must use.

To add a new connection:

- On the Connections pane, click the Add a Connection button.

The Add Connection dialog box appears.

- For Type, choose one of the following:

- **Firewall**—Use this network type if you are connecting to a Juniper Networks SRX Series device.
- For Name, specify a descriptive name for this connection. The name you specify will appear in the Connections pane of the Junos Pulse interface.
- For Server URL, specify the network that you want to connect to. You can enter the Server URL in any of the following formats:
  - An IP address, for example 10.204.71.86
  - A DNS name, for example server.mycompany.net
- Click **Add** to save your new connection and close the dialog box. Click **Connect** to save your new connection and initiate a connection to the network.

### Connect to a Network

---

You must have at least one connection listed in the Connections pane before you can connect to a network. The connection prompts you see depend on your network access environment.



**NOTE:** To use Junos Pulse with a wireless network, you might need to first configure your endpoint's wireless network settings through Windows or the wireless device software installed on your endpoint. For example, in Windows XP, use **Start > Control Panel > Network Connections** to access Windows network setup options. Or your network administrator can define your wireless connections and scan lists and include them in your Junos Pulse installation.

To use a defined connection to connect to a network:

1. In the Connections list, click **Connect** for the connection you want to establish.
2. Respond to the prompts for information such as username and password.

After you click the **Connect** button, you might need to respond to the following prompts:

- **Certificate**—If Junos Pulse needs to communicate with a certificate server, and your network administrator has configured more than one server, you are prompted to choose a server. A certificate issued by a certificate authority verifies that the network resource you are connecting to is valid. If the certificate is from a trusted source, it is automatically accepted and you do not see the certificate prompt. If there is a problem with the certificate, you might be asked if you want to accept the certificate and proceed with the connection.
- **Credentials**—Your username and password or username and token code, establish your identity to the access device. You might also be prompted for a secondary username and password and a username and password to a proxy server. Your authentication environment might periodically prompt you to change your password or your token PIN number.



A Save Settings check box might appear on each login screen. (Your administrator can disable this feature.) If you enable the check box, you are not prompted for that information the next time you login. If you save settings, you can use the Forget Saved Settings feature to return to being prompted for log in information. The Save Settings feature enables you to save the following information:

- Certificate acceptance
- Certificate selection
- Username and password

The steps that take place after you respond to all of the connection prompts depend on the access policies that your network administrator has configured and on the type of network access device. The connection process can include the following tasks:

- **Software updates**—Junos Pulse can be automatically updated at connect time. Your system might receive updated Junos Pulse software or you might receive additional software modules to support expanded services such as when you connect to a new type of network access device for the first time.

### Disconnect from an Active Network

---

To disconnect from a network:

1. On the Junos Pulse Connections pane, click **Disconnect** for the connection you want to disconnect.

Or

1. On the Junos Pulse Connections pane, right-click the connection to display the pop-up menu.
2. Click Disconnect.

### View Connection Properties

---

To view the properties for a connection:

1. In the Connections list, click the expand icon next to the connection name. Connection details appear beneath the connection.

### Edit Connection Properties

---

After you create a connection, you can edit the URL and the name that appears in the Connections pane. You can edit a connection only if that connection is not currently active.

To edit a connection:

1. In the Connections pane, right-click the connection to display the pop-up menu, and then click **Edit**.

The Edit Connection dialog box appears.

2. Edit the connection, and then click **Save** to save your changes and close the dialog box, or click **Connect** to initiate a connection and close the dialog box.

Or

1. Click the connection to select it.
2. Click the Edit Connection button.

The Edit Connection dialog box opens.

3. Edit the connection, and then click **Save** to save your changes and close the dialog box, or click **Connect** to initiate a connection and close the dialog box.

### Forget Saved Settings

---

When you connect to a network, you can check the Save Settings check box to have Junos Pulse remember your login credentials. (Note that your network administrator can disable this feature.) Each different screen where you are prompted for a response has its own Save Settings check box. If you save settings, you are not prompted to provide that information on subsequent login attempts. If your login credentials change, you must clear the saved settings in Junos Pulse and you are prompted to provide them again.

To remove saved login credentials:

1. Right-click anywhere in the Connections list to display the pop-up menu.
2. Click **Forget Saved Settings**. Junos Pulse clears the saved settings for all configured Connections.

Or

1. Click the **Forget Saved Settings** button:

### Delete a Connection

---

To delete a connection:

1. In the Connections list, click the connection you want to delete to select it.
2. Click the Delete button:
3. You are prompted to verify your decision before the connection is deleted.

Or

1. Right-click the connection you want to delete to display the pop-up menu, and then click **Delete**.
2. You are prompted to verify your decision before the connection is deleted.

### Troubleshoot a Junos Pulse Connection Issue

---

You can use the following troubleshooting information to help resolve connection issues. Table 73 on page 645 lists issues, descriptions and resolution suggestions.

Table 73: Junos Pulse Troubleshooting Information

Issue	Description and Resolution Suggestions
<p>During login, the following message appears:</p> <p><b>You are about to authenticate to an untrusted server. Do you accept the certificate for this connection?</b></p>	<p>Junos Pulse cannot verify the identity of the server. This error or any certificate error means that Junos Pulse cannot ensure that you are connecting to a trusted server. The server certificate might have been revoked or it might have expired. It could have been issued by a certificate authority that is not recognized by Junos Pulse, such as when your organization uses a self-signed certificate. If your network administrator has enabled this permission, you can choose to proceed and connect to the server, but you should do so only if your network administrator has advised you to ignore the certificate error.</p>
<p>During login, the following message appears:</p> <p><b>Credentials were invalid. Please try again.</b></p>	<p>If you selected the Save Settings check box when you activated a Junos Pulse connection, the credentials you provided are used every time you activate that connection without prompting you to enter login information. However, Junos Pulse cannot detect when you change your network password—it continues to use the saved settings, and that can result in the “Credentials were invalid” error. To resolve this issue, click the <b>Forget Saved Settings</b> button.</p> <p>The next time you connect, you are prompted for your login credentials, and you can specify your new login information. If you select the Save Settings check box, the new credential information will be saved.</p>
<p>The system tray icon and a Connections pane icon change to the failed” state:</p> <p><b>Connection failed</b></p>	<p>When you see the connection failed icon in the Connections pane, click the connection to display the connection status. The Details section shows the specific error, which you can click to open a window that shows a detailed description of the error.</p>

### Annotate Log Files

When you annotate a log file, you insert text that marks the log file at a specific location. For example, to troubleshoot a connection problem, you might be asked to annotate the log file with specific text, attempt the connection that has been failing, and then annotate the log file again. This sequence of events enables a support technician to easily find the log file entries that track your connection issue.

To annotate Junos Pulse log files:

1. Click the program icon at the top of the Junos Pulse window to display the pop-up menu.
2. Click **Logs>Annotate** to open the Annotate Logs dialog box.
3. Type your annotation text, and then click **OK**.

### Set Log Level

---

To set the log level:

1. Click the program icon at the top of the Junos Pulse window to display the pop-up menu.
2. Click **Logs>Log Level>Detailed** or **Logs>Log Level>Normal**. A check mark indicates the option that is currently enabled.

Normal logging is the default. You should set logging to Normal unless you are troubleshooting a connection issue.

Detailed logging creates a greater number of log entries and increases the size of the log files. Detailed logging is typically enabled only when troubleshooting an issue.

### Save Log Files

---

As part of a troubleshooting process, you might be asked to save the Junos Pulse log files. Pulse includes a number of possible components and each component generates one or more log files. The Save As operation combines all of the Pulse log files and other diagnostic files into a single file named LogsAndDiagnostics.zip.

To save Junos Pulse log files:

1. Click the program icon at the top of the Junos Pulse window to display the pop-up menu.
2. Click **Logs>Save As**. The Save As dialog box appears.
3. Either accept the default values for location and filename or specify new values, and then click **Save**.

### View Component Version Information

---

In a troubleshooting operation, your network administrator might ask you to verify the version numbers of the Junos Pulse programs. To view Junos Pulse version information:

1. Click **Pulse>About** to open the About dialog box.
2. Click **version details** to open the Pulse Version Details dialog box.

#### Related Documentation

- Understanding Junos Pulse Client on page 637
- Deploying Junos Pulse Client Software on page 638
- Junos Pulse Interface and Connections on page 639

### Access Manager Client-Side Reference

---

- Access Manager Client-Side System Requirements on page 647
- Access Manager Client-Side Files on page 647
- Access Manager Client-Side Registry Changes on page 650

- Access Manager Client-Side Error Messages on page 650
- Troubleshooting Access Manager Client-Side Problems on page 654

## Access Manager Client-Side System Requirements

The user can install Access Manager on Windows XP 32-bit, Windows Vista 64/32-bit, and Windows 7 64/32-bit machines with an Internet connection. The user must have administrator privileges to install the client, but not to run it.

Access Manager can run simultaneously on the same computer with other Juniper Networks clients, including the Odyssey Access Client (OAC), Network Connect client, Windows Secure Application Manager (WSAM) client, Host Checker client, and WX client.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Remote Client Access to the VPN on page 599
- Access Manager Client-Side Files on page 647
- Access Manager Client-Side Registry Changes on page 650
- Access Manager Client-Side Error Messages on page 650
- Troubleshooting Access Manager Client-Side Problems on page 654

## Access Manager Client-Side Files

Table 74 on page 647 lists the directories where Access Manager installs files on a user's computer, the files it installs, and the files that remain after the user uninstalls the client.

**Table 74: Access Manager Client-Side Files**

Installation Directory	Files Installed in Directory	Files Remaining After Uninstall
%COMMONFILES%\Juniper Networks\Connection Manager	<ul style="list-style-type: none"> <li>• ConnectionManagerService.dll</li> <li>• install.log</li> <li>• Uninstall.exe</li> <li>• Uninstall.exe.manifest</li> <li>• versionInfo.ini</li> </ul>	install.log
%COMMONFILES%\Juniper Networks\ConnectionStore	<ul style="list-style-type: none"> <li>• ConnectionStoreService.dll</li> <li>• dcfDOM.dll</li> <li>• install.log</li> <li>• Uninstall.exe</li> <li>• Uninstall.exe.manifest</li> <li>• versionInfo.ini</li> </ul>	install.log
%COMMONFILES%\Juniper Networks\IPSecMgr	<ul style="list-style-type: none"> <li>• install.log</li> <li>• ipsecmgr.dll</li> <li>• Uninstall.exe</li> <li>• Uninstall.exe.manifest</li> <li>• versionInfo.ini</li> </ul>	install.log

Table 74: Access Manager Client-Side Files (*continued*)

Installation Directory	Files Installed in Directory	Files Remaining After Uninstall
PROGRAMFILES%\Juniper Networks\Juniper Access Manager	<ul style="list-style-type: none"> <li>• AccessServiceComponent.x86.exe</li> <li>• ConnectionMgrComponent.x86.exe</li> <li>• ConnectionStoreComponent.x86.exe</li> <li>• install.log</li> <li>• IPSecMgrComponent.x86.exe</li> <li>• JamGUIComponent.x86.exe</li> <li>• JamInstaller.dep</li> <li>• jnprnaInstall.exe</li> <li>• TunnelManagerComponent.x86.exe</li> <li>• Uninstall.exe</li> <li>• Uninstall.exe.manifest</li> <li>• versionInfo.ini</li> <li>• vpnAccessMethodComponent.x86.exe</li> </ul>	<p>install.log</p> <p>Log file location: C:\Documents and Settings\All Users\Application Data\Juniper Networks\Logging</p>
%COMMONFILES%\Juniper Networks\JamUI	<ul style="list-style-type: none"> <li>• install.log</li> <li>• jamCommand.exe</li> <li>• jamTray.exe</li> <li>• jamUI.exe</li> <li>• jamUIResource_EN.dll</li> <li>• uiPlugin.dll</li> <li>• Uninstall.exe</li> <li>• Uninstall.exe.manifest</li> <li>• versionInfo.ini</li> </ul>	install.log
%COMMONFILES%\Juniper Networks\JUNS	<ul style="list-style-type: none"> <li>• access.ini</li> <li>• dsAccessService.exe</li> <li>• dsInstallerService.dll</li> <li>• dsLogService.dll</li> <li>• install.log</li> <li>• Uninstall.exe</li> <li>• Uninstall.exe.manifest</li> <li>• versionInfo.ini</li> </ul>	install.log

Table 74: Access Manager Client-Side Files (*continued*)

Installation Directory	Files Installed in Directory	Files Remaining After Uninstall
%COMMONFILES%\Juniper Networks\JNPRNA	<ul style="list-style-type: none"> <li>• install.log</li> <li>• jnprna.cat</li> <li>• jnprna.inf</li> <li>• jnprna.sys</li> <li>• jnprnaapi.dll</li> <li>• jnprnaNetInstall.dll</li> <li>• jnprnaNetInstall.log</li> <li>• jnprna_m.cat</li> <li>• jnprna_m.inf</li> <li>• jnprva.cat</li> <li>• jnprva.inf</li> <li>• jnprva.sys</li> <li>• jnprvamgr.cat</li> <li>• jnprvamgr.dll</li> <li>• jnprvamgr.inf</li> <li>• jnprvamgr.sys</li> <li>• nsStatsDump.exe</li> <li>• uninst.exe</li> <li>• versionInfo.ini</li> <li>• %WINDIR%\system32\drivers\jnprna.sys</li> <li>• %WINDIR%\system32\drivers\jnprva.sys</li> <li>• %WINDIR%\system32\drivers\jnprvamgr.sys</li> </ul>	<ul style="list-style-type: none"> <li>• install.log</li> <li>• jnprnaNetInstall.log</li> </ul>
COMMONFILES%\Juniper Networks\Tunnel Manager	<ul style="list-style-type: none"> <li>• dsTMClient.dll</li> <li>• dsTMService.dll</li> <li>• dsTunnelManager.dll</li> <li>• install.log</li> <li>• TM.dep</li> <li>• Uninstall.exe</li> <li>• Uninstall.exe.manifest</li> <li>• versionInfo.ini</li> </ul>	install.log
%COMMONFILES%\Juniper Networks\vpnAccessMethod	<ul style="list-style-type: none"> <li>• install.log</li> <li>• Uninstall.exe</li> <li>• Uninstall.exe.manifest</li> <li>• versionInfo.ini</li> <li>• vpnAccessMethod.dll</li> <li>• vpnAccessMethod_EN.dll</li> </ul>	install.log

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding Remote Client Access to the VPN on page 599
  - Access Manager Client-Side System Requirements on page 647

- Access Manager Client-Side Registry Changes on page 650
- Access Manager Client-Side Error Messages on page 650
- Troubleshooting Access Manager Client-Side Problems on page 654

## Access Manager Client-Side Registry Changes

Table 75 on page 650 lists the Windows Registry changes that the Access Manager client and components make to your users' computers when creating dynamic VPN tunnels.

**Table 75: Access Manager Client-Side Registry Changes**

Registry Key Location	Registry Key Changes
HKEY_LOCAL_MACHINE\SOFTWARE\Juniper Networks\Common Files	<ul style="list-style-type: none"> <li>• jnprnaapi="C:\Program Files\Common Files\Juniper Networks\JNPRNA\jnprnaapi.dll</li> <li>• jnprvamgr="C:\Program Files\Common Files\Juniper Networks\JNPRNA\jnprvamgr.dll</li> <li>• nsStatsDump="C:\Program Files\Common Files\Juniper Networks\JNPRNA\nsStatsDump.exe</li> <li>• dsLogService="C:\Program Files\Common Files\Juniper Networks\JUNS\dsLogService.dll</li> <li>• dsTMClient="C:\Program Files\Common Files\Juniper Networks\Tunnel Manager\dsTMClient.dll</li> <li>• dsTunnelManager="C:\Program Files\Common Files\Juniper Networks\Tunnel Manager\dsTunnelManager.dll</li> </ul>
HKEY_LOCAL_MACHINE\SOFTWARE\Juniper Networks\Logging	<ul style="list-style-type: none"> <li>• LogFileName="C:\Documents and Settings\All Users\Application Data\Juniper Networks\Logging\debuglog.log</li> <li>• "Level"="3"</li> <li>• "LogSizeInMB"="10"</li> </ul>
HKCU\Software\Juniper Networks\Access Manager\	(Content varies. Contains client configuration data downloaded from the server.)

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Remote Client Access to the VPN on page 599
- Access Manager Client-Side System Requirements on page 647
- Access Manager Client-Side Files on page 647
- Access Manager Client-Side Error Messages on page 650
- Troubleshooting Access Manager Client-Side Problems on page 654

## Access Manager Client-Side Error Messages

Table 76 on page 651 lists possible errors that end users might see when installing or running Access Manager, the possible causes for the messages, and suggested actions.



Table 76: Dynamic VPN Client-Side Errors

Error Message	Possible Causes	Suggested User Action
Component instance already in use	Internal error.	Try to reconnect to the firewall.
Memory allocation failure	Internal error.	Try to reconnect to the firewall. If the problem persists, exit and restart Access Manager.
Failed to load connection store	Internal error.	Try to reconnect to the firewall. If the problem persists, exit and restart Access Manager.
Cannot get connection information for firewall	Internal error. Could not retrieve connection information for the specified firewall.	Try to reconnect to the firewall. If the problem persists, exit and restart Access Manager.
Authentication failure: Unknown HTTP response code	Internal error. Could not decipher the HTTP response.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
Authentication failure: Incorrect username or password	The user entered an invalid username or password.	Reenter your credentials.
Authentication failure: Firewall is out of licenses	All available licenses are currently being used for other dynamic VPN sessions or no licenses are installed for the feature.	Try to reconnect to the firewall once a license has been freed by another user. If the problem persists, contact your system administrator.
Authentication failure: No configuration available	No configuration is currently available for the specified user account.	Contact your system administrator.
Cannot create IPsec route entry	Internal error. Failed to read route entry from the connection store.	Try to reconnect to the firewall. If the problem persists, exit and restart Access Manager.
Failed to read route entry from connection store	Internal error. Failed to read the route entry from the connection store.	Try to reconnect to the firewall. If the problem persists, exit and restart Access Manager.
Failed to add route entry to policy	Internal error. Failed to add the route entry to the connection store.	Try to reconnect to the firewall. If the problem persists, exit and restart Access Manager.
Failed to initialize IPsec Manager	Internal error. Failed to initialize the IPsec Manager.	Try to reconnect to the firewall. If the problem persists, exit and restart Access Manager.
IPsec authentication failed	Phase 1 negotiations, Extended Authentication (XAuth), or Phase 2 negotiations failed.	Try to reconnect to the firewall.
IPsec configuration failed	Internal error or policy configuration error. The Tunnel Manager was unable to configure the local IP settings.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.

Table 76: Dynamic VPN Client-Side Errors (*continued*)

Error Message	Possible Causes	Suggested User Action
IKE negotiations failed	The components cannot agree on security parameters during the IKE exchange. The administrator probably needs to reconfigure the Phase 1 proposal.	Contact your system administrator.
Failed to initialize authentication	Failed to authenticate when connecting to the firewall, possibly because the specified hostname did not resolve against the distinguished name server (DNS).	Try to reconnect to the firewall.
Failed to connect to server	The TCP connection to the webserver failed during authentication, possibly because of network connectivity issues.	Try to reconnect to the firewall.
Failed to send initial HTTP request	Webserver authentication failed, possibly because of network connectivity issues.	Try to reconnect to the firewall.
Failed to get HTTP response	Webserver authentication failed.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
Firewall refused authentication request	Webserver authentication failed.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
Client failed to provide login page	Webserver authentication failed.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
Server failed to send authentication request	Webserver authentication failed.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
Server failed to respond to authentication request	The client sent the user's credentials to the webserver, but the server failed to respond in a useful manner.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
Authentication negotiation failed	Webserver authentication failed.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
Failed to get configuration from firewall	Webserver authentication failed.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
The user cancelled authentication.	User canceled authentication	Try to reconnect to the firewall and reenter your credentials.
Failed to enter username or password	Authentication request timed out.	Try to reconnect to the firewall and reenter your credentials.
Server failed to request username and password	The client failed to display the user interface asking the user for credentials.	Exit and restart Access Manager. If the problem persists, contact your system administrator.

Table 76: Dynamic VPN Client-Side Errors (*continued*)

Error Message	Possible Causes	Suggested User Action
Your client state is preventing the connection	The user's client is in an inoperable state.	Try to reconnect to the firewall. If the problem persists, exit and restart Access Manager.
Cannot open connection store	The client could not contact the connection store.	Exit and restart Access Manager. If the problem persists, reinstall Access Manager.
Cannot process configuration provided by firewall	The script provided by the firewall was in some way unusable. The configuration might need to be updated on the server.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
Access Manager is not running	The Access Manager service is not running.	Exit and restart Access Manager.
Please select a connection	The user chose <b>Start Connection</b> without selecting a connection first.	Select the firewall you want to connect to and then choose <b>Start Connection</b> .
Are you sure you want to delete the selected connection?	The user chose <b>Delete Connection</b> .	Specify whether or not you want to delete the selected connection profile.
Cannot add new connection. Service is not running.	The Access Manager service is not running; therefore it cannot create a new connection profile.	Exit and restart Access Manager. If the problem persists, reinstall Access Manager.
Cannot add new connection	The Access Manager failed to add the new connection profile.	Try again. If the problem persists, exit and restart Access Manager.
Connection name is already in use	Unable to add connection profile because the specified connection name already exists.	Specify a unique name for the connection profile.
Please reinstall Access Manager	Files could not be found when trying to finish the operation.	Reinstall Access Manager.
Invalid server certificate	Certificate validation failed.	Check client-side logs to determine why the certificate failed.
Initializing service...	Initializing one of the client's core components. If the component does not initialize, the client cannot function.	Wait for the service to finish initializing.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Remote Client Access to the VPN on page 599
- Access Manager Client-Side System Requirements on page 647
- Access Manager Client-Side Files on page 647
- Access Manager Client-Side Registry Changes on page 650
- Troubleshooting Access Manager Client-Side Problems on page 654

## Troubleshooting Access Manager Client-Side Problems

**Problem** Users are having problems connecting to the remote access server using Access Manager.

**Solution** Use the following tools to troubleshoot client-side issues:

- Client-side logs—To view client-side logs, open Access Manager and choose **Save logs and diagnostics** from the File menu. Select a location on your computer to save the zipped log files and click **Save**.
- Detailed logs—To create more detailed client-side logs, open Access Manager and choose **Enable Detailed Logging** from the File menu.
- Firewall connection information—To view connection information for a given firewall, open Access Manager, right-click to select the firewall, and choose **Status**.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Remote Client Access to the VPN on page 599](#)
- [Access Manager Client-Side System Requirements on page 647](#)
- [Access Manager Client-Side Files on page 647](#)
- [Access Manager Client-Side Registry Changes on page 650](#)
- [Access Manager Client-Side Error Messages on page 650](#)

## CHAPTER 21

# Group VPNs

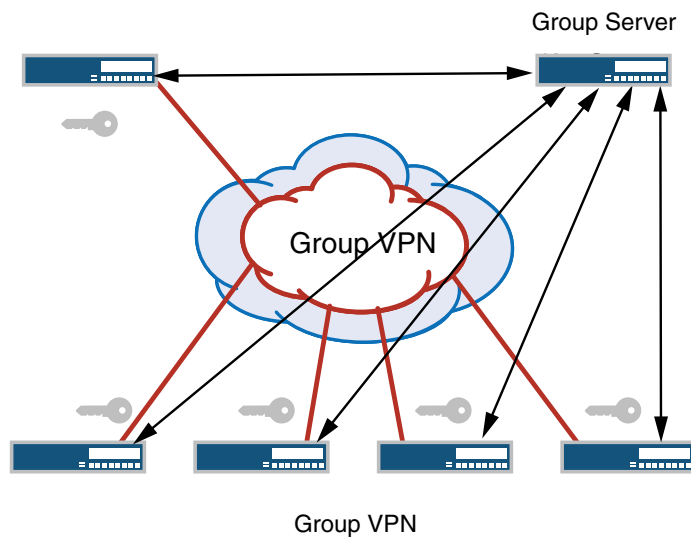
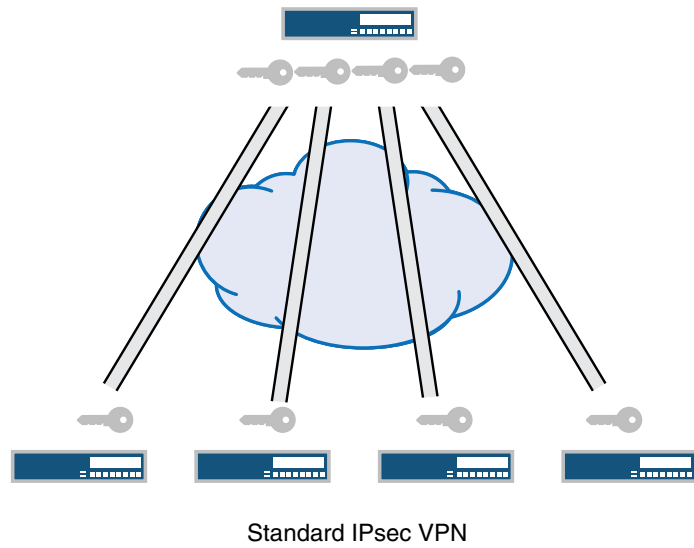
- Group VPN Overview on page 655
- Group VPNs on page 657
- Colocation Mode on page 679
- Server-Group Communications on page 689
- Understanding Group VPN Limitations on page 697
- Understanding Interoperability with Cisco GET VPN on page 698

### Group VPN Overview

---

An IPsec security association (SA) is a unidirectional agreement between virtual private network (VPN) participants that defines the rules to use for authentication and encryption algorithms, key exchange mechanisms, and secure communications. With current VPN implementations, the SA is a point-to-point tunnel between two security devices. A group VPN extends IPsec architecture to support SAs that are shared by a group of security devices (see Figure 60 on page 656).

Figure 60: Standard IPsec VPN and Group VPN



Server distributes IPsec SA. All members that belong to the group share the same IPsec SA.

With group VPNs, any-to-any connectivity is achieved by preserving the original source and destination IP addresses in the outer header. Secure multicast packets are replicated in the same way as cleartext multicast packets in the core network.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [VPN Overview on page 451](#)
- [Understanding IKE and IPsec Packet Processing on page 458](#)
- [Understanding the GDOI Protocol on page 657](#)
- [Understanding Group Servers and Members on page 658](#)
- [Group VPN Configuration Overview on page 663](#)

## Group VPNs

---

- [Understanding the GDOI Protocol on page 657](#)
- [Understanding Group Servers and Members on page 658](#)
- [Understanding IKE Phase 1 Configuration for Group VPN on page 659](#)
- [Understanding IPsec SA Configuration for Group VPN on page 659](#)
- [Understanding Dynamic Policies on page 660](#)
- [Understanding Antireplay on page 662](#)
- [Understanding VPN Group Configuration on page 662](#)
- [Group VPN Configuration Overview on page 663](#)
- [Example: Configuring Group VPNs on page 663](#)

## Understanding the GDOI Protocol

Group VPN is based on RFC 3547, *The Group Domain of Interpretation (GDOI)*. This RFC describes the protocol between group members and a group server to establish SAs among group members. GDOI messages create, maintain, or delete SAs for a group of devices. The GDOI protocol runs on port 848.

The Internet Security Association and Key Management Protocol (ISAKMP) defines two negotiation phases to establish SAs for an AutoKey IKE IPsec tunnel. Phase 1 allows two devices to establish an ISAKMP SA. Phase 2 establishes SAs for other security protocols, such as GDOI.

With group VPN, Phase 1 ISAKMP SA negotiation is performed between a group server and a group member. The server and member must use the same ISAKMP policy. In Phase 2, GDOI exchanges between the server and member establish the SAs that are shared with other group members. A group member does not need to negotiate IPsec with other group members. GDOI exchanges in Phase 2 must be protected by ISAKMP Phase 1 SAs.

There are two types of GDOI exchanges:

- The **groupkey-pull** exchange allows a member to request SAs and keys shared by the group from the server.
- The **groupkey-push** exchange is a single rekey message that allows the server to send group SAs and keys to members before existing group SAs expire. Rekey messages are unsolicited messages sent from the server to members.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Group VPN Overview on page 655](#)
- [Understanding IKE and IPsec Packet Processing on page 458](#)
- [Understanding Group Servers and Members on page 658](#)
- [Understanding Group Key Operations on page 690](#)

## Understanding Group Servers and Members

The center of a group VPN is the group server. The group server performs the following tasks:

- Controls group membership
- Generates encryption keys
- Manages group SAs and keys and distributes them to group members

Group members encrypt traffic based on the group SAs and keys provided by the group server.

A group server can service multiple groups. A single security device can be a member of multiple groups.

Each group is represented by a group identifier, which is a number between 1 and 65,535. The group server and group members are linked together by the group identifier. There can be only one group identifier per group, and multiple groups cannot use the same group identifier.

The following is a high-level view of group VPN server and member actions:

1. The group server listens on UDP port 848 for members to register. A member device must provide correct IKE Phase 1 authentication to join the group. Preshared key authentication on a per-member basis is supported.
2. Upon successful authentication and registration, the member device retrieves group SAs and keys from the server with a GDOI **groupkey-pull** exchange.
3. The server adds the member to the membership for the group.
4. Group members exchange packets encrypted with group SA keys.

The server periodically sends SA and key refreshes to group members with rekey (GDOI **groupkey-push**) messages. Rekey messages are sent before SAs expire; this ensures that valid keys are available for encrypting traffic between group members.



The server also sends rekey messages to provide new keys to members when there is a change in group membership or when the group SA has changed.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Group VPN Overview on page 655](#)
- [Understanding the GDOI Protocol on page 657](#)
- [Understanding Colocation Mode on page 679](#)
- [Understanding Dynamic Policies on page 660](#)
- [Understanding Antireplay on page 662](#)
- [Group VPN Configuration Overview on page 663](#)

## Understanding IKE Phase 1 Configuration for Group VPN

An IKE Phase 1 SA between the group server and a group member establishes a secure channel in which to negotiate IPsec SAs that are shared by a group. For standard IPsec VPNs on Juniper Networks security devices, Phase 1 SA configuration consists of specifying an IKE proposal, policy, and gateway. For group VPN, the IKE Phase 1 SA configuration is similar to the configuration for standard IPsec VPNs, but is performed at the [**edit security group-vpn**] hierarchy.

In the IKE proposal configuration, you set the authentication method and the authentication and encryption algorithms that will be used to open a secure channel between participants. In the IKE policy configuration, you set the mode (main or aggressive) in which the Phase 1 channel will be negotiated, specify the type of key exchange to be used, and reference the Phase 1 proposal. In the IKE gateway configuration, you reference the Phase 1 policy.

The IKE Phase 1 configuration on the group server must match the IKE Phase 1 configuration on group members. On the server, use the [**edit security group-vpn server ike**] hierarchy to configure IKE Phase 1 SA. On a group member, use the [**edit security group-vpn member ike**] hierarchy to configure IKE Phase 1 SA.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Group VPN Overview on page 655](#)
- [Understanding the GDOI Protocol on page 657](#)
- [Understanding Group Servers and Members on page 658](#)
- [Group VPN Configuration Overview on page 663](#)
- [Understanding IPsec SA Configuration for Group VPN on page 659](#)

## Understanding IPsec SA Configuration for Group VPN

After the server and member have established a secure and authenticated channel in Phase 1 negotiation, they proceed through Phase 2. Phase 2 negotiation establishes the IPsec SAs that are shared by group members to secure data that is transmitted among

members. While the IPsec SA configuration for group VPN is similar to the configuration for standard VPNs, a group member does not need to negotiate the SA with other group members.

Phase 2 IPsec configuration for group VPN consists of the following information:

- A proposal for the security protocol, authentication, and encryption algorithm to be used for the SA. The IPsec SA proposal is configured on the group server with the **proposal** configuration statement at the [edit security group-vpn server ipsec] hierarchy.
- A group policy that references the proposal. A group policy specifies the traffic (protocol, source address, source port, destination address, and destination port) to which the SA and keys apply. The group policy is configured on the server with the **ipsec-sa** configuration statement at the [edit security group-vpn server group] hierarchy.
- An Autokey IKE that references the group identifier, the group server (configured with the **ike-gateway** configuration statement), and the interface used by the member to connect to the group. The Autokey IKE is configured on the member with the **ipsec vpn** configuration statement at the [edit security group-vpn member] hierarchy.



**NOTE:** To prevent packet fragmentation issues, we recommend that the interface used by the group member to connect to the MPLS network be configured for a maximum transmission unit (MTU) size no larger than 1400 bytes. Use the set *interface* mtu configuration statement to set the MTU size.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Group VPN Overview on page 655](#)
- [Understanding the GDOI Protocol on page 657](#)
- [Understanding Group Servers and Members on page 658](#)
- [Group VPN Configuration Overview on page 663](#)
- [Understanding IKE Phase 1 Configuration for Group VPN on page 659](#)

## Understanding Dynamic Policies

The group server distributes group SAs and keys to members of a specified group. All members that belong to the same group can share the same set of IPsec SAs. But not all SAs configured for a group are installed on every group member. The SA installed on a specific member is determined by the policy associated with the group SA and the security policies configured on the member.

In a VPN group, each group SA and key that the server pushes to a member is associated with a *group policy*. The group policy describes the traffic on which the key should be used, including protocol, source address, source port, destination address, and destination port.



**NOTE:** Group policies that are identical (configured with the same source address, destination address, source port, destination port, and protocol values) cannot exist for a single group. An error is returned if you attempt to commit a configuration that contains identical group policies for a group. If this is the case, you must delete one of the identical group policies.

On a group member, a *scope policy* must be configured that defines the scope of the group policy downloaded from the server. A group policy distributed from the server is compared against the scope policies configured on the member. For a group policy to be installed on the member, the following conditions must be met:

- Any addresses specified in the group policy must be within the range of addresses specified in the scope policy.
- The source port, destination port, and protocol specified in the group policy must match those configured in the scope policy.

A group policy that is installed on a member is called a *dynamic policy*.

A scope policy can be part of an ordered list of security policies for a specific from-zone and to-zone context. Junos OS performs a security policy lookup on incoming packets starting from the top of the ordered list.

Depending on the position of the scope policy within the ordered list of security policies, there are several possibilities for dynamic policy lookup:

- If an incoming packet matches a scope policy, the search process continues for a matching dynamic policy. If there is a matching dynamic policy, that policy action (permit) is performed. If there is no matching dynamic policy, then the packet is dropped.



**NOTE:** In this release, only the tunnel action is allowed for a scope policy. Other actions are not supported.

- If the incoming packet matches a security policy before the scope policy is considered, dynamic policy lookup does not occur.

You configure a scope policy on a group member by using the **policies** configuration statement at the **[edit security]** hierarchy. Use the **ipsec-group-vpn** configuration statement in the permit tunnel rule to reference the group VPN; this allows group members to share a single SA.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Security Policies Overview on page 145
- Understanding Security Policy Ordering on page 173
- Example: Configuring a Security Policy to Permit or Deny All Traffic on page 152
- Understanding the GDOI Protocol on page 657

- [Understanding Group Servers and Members](#) on page 658
- [Group VPN Configuration Overview](#) on page 663

## Understanding Antireplay

Antireplay is an IPsec feature that can detect when a packet is intercepted and then replayed by attackers. Antireplay is enabled by default for group VPNs but can be disabled for a group with the **no-anti-replay** configuration statement.

When antireplay is enabled, the group server synchronizes the time between the group members. Each IPsec packet contains a timestamp. The group member checks whether the packet's timestamp falls within the configured **anti-replay-time-window** value (the default is 100 seconds). A packet is dropped if the timestamp exceeds the value.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [VPN Overview](#) on page 451
- [Understanding IKE and IPsec Packet Processing](#) on page 458
- [Understanding the GDOI Protocol](#) on page 657
- [Understanding Group Servers and Members](#) on page 658
- [Understanding VPN Group Configuration](#) on page 662

## Understanding VPN Group Configuration

The VPN group is configured on the server with the **group** configuration statement at the `[edit security group-vpn server]` hierarchy.

The group information consists of the following information:

- **Group identifier**—A value between 1 and 65,535 that identifies the VPN group. The same group identifier must be configured on the group member for Autokey IKE.
- **Group members**, as configured with the **ike-gateway** configuration statement. There can be multiple instances of this configuration statement, one for each member of the group.
- **IP address of the server** (the loopback interface address is recommended).
- **Group policies**—Policies that are to be downloaded to members. Group policies describe the traffic to which the SA and keys apply. See “Understanding Dynamic Policies” on page 660.
- **Server-member communication**—Optional configuration that allows the server to send rekey messages to members. See “Understanding Server-Member Communication” on page 689.
- **Antireplay**—Optional configuration that detects packet interception and replay. See “Understanding Antireplay” on page 662.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Group VPN Overview on page 655](#)
  - [Understanding the GDOI Protocol on page 657](#)
  - [Understanding Group Servers and Members on page 658](#)
  - [Group VPN Configuration Overview on page 663](#)

## Group VPN Configuration Overview

This topic describes the main tasks for configuring group VPN.

On the group server, configure the following:

1. IKE Phase 1 negotiation. See “Understanding IKE Phase 1 Configuration for Group VPN” on page 659.
2. Phase 2 IPsec SA. See “Understanding IPsec SA Configuration for Group VPN” on page 659.
3. VPN group. See “Understanding VPN Group Configuration” on page 662.

On the group member, configure the following:

1. IKE Phase 1 negotiation. See “Understanding IKE Phase 1 Configuration for Group VPN” on page 659.
2. Phase 2 IPsec SA. See “Understanding IPsec SA Configuration for Group VPN” on page 659.
3. Scope policy that determines which group policies are installed on the member. See “Understanding Dynamic Policies” on page 660.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding Group Servers and Members on page 658](#)
  - [Understanding Server-Member Communication on page 689](#)
  - [Example: Configuring Group VPNs on page 663](#)
  - [Example: Configuring Group VPN with Server-Member Colocation on page 680](#)

## Example: Configuring Group VPNs

This example shows how to configure group VPNs to extend IPsec architecture to support SAs that are shared by a group of security devices.

- [Requirements on page 664](#)
- [Overview on page 664](#)
- [Configuration on page 665](#)
- [Verification on page 677](#)

## Requirements

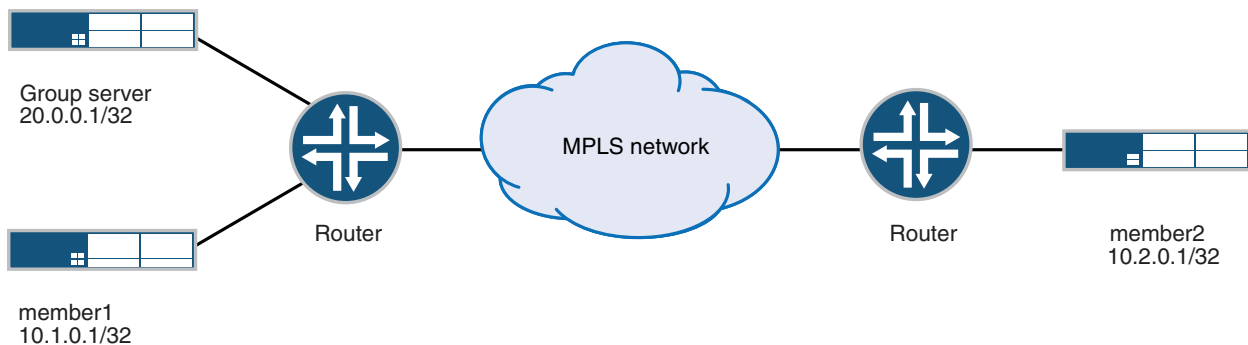
Before you begin:

- Configure the Juniper Networks security devices for network communication.
- Configure network interfaces on server and member devices. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

## Overview

In Figure 61 on page 664, a group VPN consists of two member devices (member1 and member2) and a group server (the IP address of the loopback interface on the server is 20.0.0.1). The group identifier is 1.

Figure 61: Server-Member Configuration Example



The Phase 2 group VPN SAs must be protected by a Phase 1 SA. Therefore, the group VPN configuration must include configuring IKE Phase 1 negotiations on both the group server and the group members. In addition, the same group identifier must be configured on both the group server and the group members.

Group policies are configured on the group server. All group policies configured for a group are downloaded to group members. Scope policies configured on a group member determine which group policies are actually installed on the member. In this example, the following group policies are configured on the group server for downloading to all group members:

- p1—Allows all traffic from 10.1.0.0/16 to 10.2.0.0/16
- p2—Allows all traffic from 10.2.0.0/16 to 10.1.0.0/16
- p3—Allows multicast traffic from 10.1.1.1/32

The member1 device is configured with scope policies that allow all unicast traffic to and from the 10.0.0.0/8 subnetwork. There is no scope policy configured on member1 to allow multicast traffic; therefore, the SA policy p3 is not installed on member1.

The member2 device is configured with scope policies that drop traffic from 10.1.0.0/16 from the trust zone to the untrust zone and to 10.1.0.0/16 from the untrust zone to the trust zone. Therefore the SA policy p2 is not installed on member2.

## Configuration

### Configuring the Group Server

**CLI Quick Configuration** To quickly configure the group server, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces lo0 unit 0 family inet address 20.0.0.1/32
set security group-vpn server ike proposal srv-prop authentication-method pre-shared-keys
set security group-vpn server ike proposal srv-prop dh-group group2
set security group-vpn server ike proposal srv-prop authentication-algorithm sha1
set security group-vpn server ike proposal srv-prop encryption-algorithm 3des-cbc
set security group-vpn server ike policy srv-pol mode main
set security group-vpn server ike policy srv-pol proposals srv-prop
set security group-vpn server ike policy srv-pol pre-shared-key ascii-text
  "$9$sc1grK8-VYZUHX7UHqmF3Sre"
set security group-vpn server ike gateway gw1 ike-policy srv-pol
set security group-vpn server ike gateway gw1 address 10.1.0.1
set security group-vpn server ike gateway gw2 ike-policy srv-pol
set security group-vpn server ike gateway gw2 address 10.2.0.1
set security group-vpn server ipsec proposal group-prop authentication-algorithm
  hmac-sha1-96
set security group-vpn server ipsec proposal group-prop encryption-algorithm 3des-cbc
set security group-vpn server ipsec proposal group-prop lifetime-seconds 3600
set security group-vpn server group group1 group-id 1
set security group-vpn server group group1 ike-gateway gw1
set security group-vpn server group group1 ike-gateway gw2
set security group-vpn server group group1 anti-replay-time-window 120
set security group-vpn server group group1 server-address 20.0.0.1
set security group-vpn server group group1 server-member-communication
  communication-type unicast
set security group-vpn server group group1 server-member-communication
  encryption-algorithm aes-128-cbc
set security group-vpn server group group1 server-member-communication
  sig-hash-algorithm md5
set security group-vpn server group group1 server-member-communication certificate
  srv-cert
set security group-vpn server group grp1 group-id 1
set security group-vpn server group grp1 ike-gateway gw1
set security group-vpn server group grp1 ike-gateway gw2
set security group-vpn server group grp1 anti-replay-time-window 120
set security group-vpn server group grp1 server-address 20.0.0.1
set security group-vpn server group grp1 ipsec-sa group-sa proposal group-prop
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 source
  10.1.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 destination
  10.2.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 source-port
  0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 destination-port
  0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 protocol 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 source
  10.2.0.0/16
```

```

set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 destination
  10.1.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 source-port
  0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2
  destination-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 protocol 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 source
  10.1.1.1/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 destination
  239.1.1.1/32
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 source-port
  0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3
  destination-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 protocol 0

```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the group server:

1. Configure the loopback address on the device.

```

[edit]
user@host# edit interfaces
user@host# set lo0 unit 0 family inet address 20.0.0.1/32

```

2. Configure IKE Phase 1 SA (this configuration must match the Phase 1 SA configured on the group members).

```

[edit security group-vpn server ike proposal srv-prop]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm 3des-cbc

```

3. Define the IKE policy and set the remote gateways.

```

[edit security group-vpn server ike]
user@host# set policy srv-pol mode main proposals srv-prop pre-shared-key
  ascii-text "$9$c1grK8-VYZUHX7UHqmF3Sre"
user@host# set gateway gw1 ike-policy srv-pol address 10.1.0.1
user@host# set gateway gw2 ike-policy srv-pol address 10.2.0.1

```

4. Configure the Phase 2 SA exchange.

```

[edit security group-vpn server ipsec proposal group-prop]
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm 3des-cbc
user@host# set lifetime-seconds 3600

```

5. Configure the group identifier, IKE gateway, antireplay time, and server address.

```

[edit security group-vpn server group grp1]
user@host# set group-id 1 anti-replay-time-window 120 server-address 20.0.0.1
user@host# set ike-gateway gw1
user@host# set ike-gateway gw2

```



6. Configure server-to-member communications.

```
[edit security group-vpn server group grp1]
user@host# set server-member-communication communication-type unicast
encryption-algorithm aes-128-cbc sig-hash-algorithm md5 certificate "srv-cert"
```

7. Configure the group policies to be downloaded to group members.

```
[edit security group-vpn server group grp1 ipsec-sa group-sa]
user@host# set proposal group-prop match-policy p1 source 10.1.0.0/16 destination
10.2.0.0/16 source-port 0 destination-port 0 protocol 0
user@host# set proposal group-prop match-policy p2 source 10.2.0.0/16 destination
10.1.0.0/16 source-port 0 destination-port 0 protocol 0
user@host# set proposal group-prop match-policy p3 source 10.1.1.1/16 destination
239.1.1.1/32 source-port 0 destination-port 0 protocol 0
```

**Results** From configuration mode, confirm your configuration by entering the **show security group-vpn server** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security group-vpn server
ike {
  proposal srv-prop {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
  }
  policy srv-pol {
    mode main;
    proposals srv-prop;
    pre-shared-key ascii-text "$9$c1grK8-VYZUHX7UHqmF3Sre"; ## SECRET-DATA
  }
  gateway gw1 {
    ike-policy srv-pol;
    address 10.1.0.1;
  }
  gateway gw2 {
    ike-policy srv-pol;
    address 10.2.0.1;
  }
}
ipsec {
  proposal group-prop {
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
    lifetime-seconds 3600;
  }
}
group group1 {
  group-id 1;
  ike-gateway gw1;
  ike-gateway gw2;
  anti-replay-time-window 120;
  server-address 20.0.0.1;
```

```

server-member-communication {
  communication-type unicast;
  encryption-algorithm aes-128-cbc;
  sig-hash-algorithm md5;
  certificate srv-cert;
}
}
group grp1 {
  group-id 1;
  ike-gateway gw1;
  ike-gateway gw2;
  anti-replay-time-window 120;
  server-address 20.0.0.1;
  ipsec-sa group-sa {
    proposal group-prop;
    match-policy p1 {
      source 10.1.0.0/16;
      destination 10.2.0.0/16;
      source-port 0;
      destination-port 0;
      protocol 0;
    }
    match-policy p2 {
      source 10.2.0.0/16;
      destination 10.1.0.0/16;
      source-port 0;
      destination-port 0;
      protocol 0;
    }
    match-policy p3 {
      source 10.1.1.1/16;
      destination 239.1.1.1/32;
      source-port 0;
      destination-port 0;
      protocol 0;
    }
  }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring Member1

**CLI Quick Configuration** To quickly configure member1, copy the following commands and paste them into the CLI:

```

[edit]
set security group-vpn member ike proposal prop1 authentication-method pre-shared-keys
set security group-vpn member ike proposal prop1 dh-group group2
set security group-vpn member ike proposal prop1 authentication-algorithm sha1
set security group-vpn member ike proposal prop1 encryption-algorithm 3des-cbc
set security group-vpn member ike policy pol1 mode main
set security group-vpn member ike policy pol1 proposals prop1
set security group-vpn member ike policy pol1 pre-shared-key ascii-text "$9$c1gr
  K8-VYZUHX7UHqmF3Sre"
set security group-vpn member ike gateway g1 ike-policy pol1

```

```

set security group-vpn member ike gateway g1 address 20.0.0.1
set security group-vpn member ike gateway g1 local-address 10.1.0.1
set security group-vpn member ipsec vpn v1 ike-gateway g1
set security group-vpn member ipsec vpn v1 group-vpn-external-interface ge-0/1/0
set security zones security-zone trust address-book address 10_subnet 10.0.0.0/8
set security zones security-zone untrust address-book address 10_subnet 10.0.0.0/8
set security policies from-zone trust to-zone untrust policy scope1 match source-address
  10_subnet
set security policies from-zone trust to-zone untrust policy scope1 match
  destination-address 10_subnet
set security policies from-zone trust to-zone untrust policy scope1 match application any
set security policies from-zone trust to-zone untrust policy scope1 then permit tunnel
  ipsec-group-vpn v1
set security policies from-zone untrust to-zone trust policy scope1 match source-address
  10_subnet
set security policies from-zone untrust to-zone trust policy scope1 match
  destination-address 10_subnet
set security policies from-zone untrust to-zone trust policy scope1 match application any
set security policies from-zone untrust to-zone trust policy scope1 then permit tunnel
  ipsec-group-vpn v1

```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure member1:

1. Configure Phase 1 SA (this configuration must match the Phase 1 SA configured on the group server).

```

[edit security group-vpn member ike proposal prop1]
user@member1# set authentication-method pre-shared-keys
user@member1# set dh-group group2
user@member1# set authentication-algorithm sha1
user@member1# set encryption-algorithm 3des-cbc

```

2. Define the IKE policy and set the remote gateways.

```

[edit security group-vpn member ike]
user@member1# set policy pol1 mode main proposals prop1 pre-shared-key ascii-text
"$9$ç1grK8-VYZUHX7UHqmF3Sre"
user@member1# set gateway g1 ike-policy pol1 address 20.0.0.1 local-address
10.1.0.1

```

3. Configure the group identifier, IKE gateway, and interface for member1.

```

[edit security group-vpn member ipsec]
user@member1# set vpn v1 group 1 ike-gateway g1 group-vpn-external-interface
ge-0/1/0

```



**NOTE:** To prevent packet fragmentation issues, we recommend that the interface used by the group members to connect to the MPLS network be configured for an MTU size no larger than 1400 bytes. Use the `set interface mtu` configuration statement to set the MTU size.

- Configure address book entries for the 10.0.0.0/8 subnet.

```
[edit security zones]
user@member1# set security-zone trust address-book address10_subnet10.0.0.0/8
user@member1# set security-zone untrust address-book address10_subnet
10.0.0.0/8
```

- Configure a scope policy from the trust zone to the untrust zone that allows unicast traffic to and from the 10.0.0.0/8 subnetwork.

```
[edit security policies from-zone trust to-zone untrust]
user@member1# set policy scope1 match source-address10_subnet
destination-address10_subnet application any
user@member1# set policy scope1 then permit tunnel ipsec-group-vpn v1
```

- Configure a scope policy from the untrust zone to the trust zone that allows unicast traffic to and from the 10.0.0.0/8 subnetwork.

```
[edit security policies from-zone untrust to-zone trust]
user@member1# set policy scope1 match source-address10_subnet
destination-address10_subnet application any
user@member1# set policy scope1 then permit tunnel ipsec-group-vpn v1
```

**Results** From configuration mode, confirm your configuration by entering the **show security group-vpn member** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@member1# show security group-vpn member
ike {
  proposal prop1 {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
  }
  policy pol1 {
    mode main;
    proposals prop1;
    pre-shared-key ascii-text "$9$CeS6uBEleWLNb"; ## SECRET-DATA
  }
  gateway g1 {
    ike-policy pol1;
    address 20.0.0.1;
    local-address 10.1.0.1;
  }
}
ipsec {
  vpn v1 {
    ike-gateway g1;
    group-vpn-external-interface ge-0/1/0;
    group 1;
  }
}
[edit]
```

```
user@member1# show security policies
from-zone trust to-zone trust {
  policy default-permit {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone trust to-zone untrust {
  policy scope1 {
    match {
      source-address 10_subnet;
      destination-address 10_subnet;
      application any;
    }
    then {
      permit {
        tunnel {
          ipsec-group-vpn v1;
        }
      }
    }
  }
  policy default-permit {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone untrust to-zone trust {
  policy scope1 {
    match {
      source-address 10_subnet;
      destination-address 10_subnet;
      application any;
    }
    then {
      permit {
        tunnel {
          ipsec-group-vpn v1;
        }
      }
    }
  }
  policy default-deny {
    match {
```

```

        source-address any;
        destination-address any;
        application any;
    }
    then {
        deny;
    }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Configuring Member2

**CLI Quick Configuration** To quickly configure Member2, copy the following commands and paste them into the CLI:

```

[edit]
set security group-vpn member ike proposal prop2 authentication-method pre-shared-keys
set security group-vpn member ike proposal prop2 authentication-method pre-shared-keys
set security group-vpn member ike proposal prop2 dh-group group2
set security group-vpn member ike proposal prop2 authentication-algorithm sha1
set security group-vpn member ike proposal prop2 encryption-algorithm 3des-cbc
set security group-vpn member ike policy pol2 mode main
set security group-vpn member ike policy pol2 proposals prop2
set security group-vpn member ike policy pol2 pre-shared-key ascii-text "$9$clgr
  K8-VYZUHX7UHqmF3Sre"
set security group-vpn member ike gateway g2 ike-policy pol2
set security group-vpn member ike gateway g2 address 20.0.0.1
set security group-vpn member ike gateway g2 local-address 10.2.0.1
set security group-vpn member ipsec vpn v2 ike-gateway g2
set security group-vpn member ipsec vpn v2 group-vpn-external-interface ge-0/1/0
set security group-vpn member ipsec vpn v2 group 1
set security zones security-zone trust address-book address 10_subnet 10.0.0.0/8
set security zones security-zone trust address-book address 10_1_0_0_16 10.1.0.0 /16
set security zones security-zone trust address-book address multicast_net 239.0. 0.0/8
set security zones security-zone untrust address-book address 10_subnet 10.0.0.0/8
set security zones security-zone untrust address-book address 10_1_0_0_16 10.1.0.0 /16
set security zones security-zone untrust address-book address multicast_net 239.0. 0.0/8
set security policies from-zone trust to-zone untrust policy deny2 match source-address
  10_1_0_0_16
set security policies from-zone trust to-zone untrust policy deny2 match
  destination-address any
set security policies from-zone trust to-zone untrust policy deny2 match application any
set security policies from-zone trust to-zone untrust policy deny2 then reject
set security policies from-zone trust to-zone untrust policy scope2 match source -address
  10_subnet
set security policies from-zone trust to-zone untrust policy scope2 match
  destination-address 10_subnet
set security policies from-zone trust to-zone untrust policy scope2 match application any
set security policies from-zone trust to-zone untrust policy scope2 then permit tunnel
  ipsec-group-vpn v2
set security policies from-zone trust to-zone untrust policy multicast-scope2 match
  source-address 10_subnet
set security policies from-zone trust to-zone untrust policy multicast-scope2 match
  destination-address multicast-net

```

```

set security policies from-zone trust to-zone untrust policy multicast-scope2 match
  application any
set security policies from-zone trust to-zone untrust policy multicast-scope2 then permit
  tunnel ipsec-group-vpn v2
set security policies from-zone untrust to-zone trust policy deny2 match source-address
  any set security policies from-zone untrust to-zone trust policy multicast-scope2 ma
  tch application any set security policies from-zone untr
set security policies from-zone untrust to-zone trust policy deny2 match
  destination-address 10_1_0_0_16
set security policies from-zone untrust to-zone trust policy deny2 match application any
set security policies from-zone untrust to-zone trust policy deny2 then reject
set security policies from-zone untrust to-zone trust policy scope2 match source-address
  10_subnet
set security policies from-zone untrust to-zone trust policy scope2 match
  destination-address 10_subnet
set security policies from-zone untrust to-zone trust policy scope2 match application any
set security policies from-zone untrust to-zone trust policy scope2 then permit tunnel
  ipsec-group-vpn v2
set security policies from-zone untrust to-zone trust policy multicast-scope2 match
  source-address 10_subnet
set security policies from-zone untrust to-zone trust policy multicast-scope2 match
  destination-address multicast-net
set security policies from-zone untrust to-zone trust policy multicast-scope2 match
  application any
set security policies from-zone untrust to-zone trust policy multicast-scope2 then permit
  tunnel ipsec-group-vpn v2

```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure member2:

1. Configure Phase 1 SA (this configuration must match the Phase 1 SA configured on the group server).

```

[edit security group-vpn member ike proposal prop2]
user@member2# set authentication-method pre-shared-keys
user@member2# set dh-group group2
user@member2# set authentication-algorithm sha1
user@member2# set encryption-algorithm 3des-cbc

```

2. Define the IKE policy and set the remote gateway.

```

[edit security group-vpn member ike]
user@member2# set policy pol2 mode main proposals prop2 pre-shared-key
  ascii-text "$9$Sc1grK8-VYZUHX7UHqmF3Sre"
user@member2# set gateway g2 ike-policy pol2 address 20.0.0.1 local-address
  10.2.0.1

```

3. Configure the group identifier, IKE gateway, and interface for member2.

```

[edit security group-vpn member ipsec]
user@member2# set vpn v2 group 1 ike-gateway g2 group-vpn-external-interface
  ge-0/1/0

```



**NOTE:** To prevent packet fragmentation issues, we recommend that the interface used by the group members to connect to the MPLS network be configured for an MTU size no larger than 1400 bytes. Use the `set interface mtu` configuration statement to set the MTU size.

4. Configure address book entries for the trust zone.

```
[edit security zones security-zone trust]
user@member2# set address-book address 10_subnet 10.0.0.0/8
user@member2# set address-book address 10_1_0_0_16 10.1.0.0/16
user@member2# set address-book address multicast_net 239.0.0.0/8
```

5. Configure address book entries for the untrust zone.

```
[edit security zones security-zone untrust]
user@member2# set address-book address 10_subnet 10.0.0.0/8
user@member2# set address-book address 10_1_0_0_16 10.1.0.0/16
user@member2# set address-book address multicast_net 239.0.0.0/8
```

6. Configure a scope policy from the trust zone to the untrust zone that blocks traffic from 10.1.0.0/16.

```
[edit security policies from-zone trust to-zone untrust]
user@member2# set policy deny2 match source-address 10_1_0_0_16
destination-address any application any
user@member2# set policy deny2 then reject
user@member2# set policy scope2 match source-address 10_subnet
destination-address 10_subnet application any
user@member2# set policy scope2 then permit tunnel ipsec-group-vpn v2
user@member2# set policy multicast-scope2 match source-address 10_subnet
destination-address multicast-net application any
user@member2# set policy multicast-scope2 then permit tunnel ipsec-group-vpn
v2
```

7. Configure a scope policy from the untrust zone to the trust zone that blocks traffic to 10.1.0.0/16.

```
[edit security policies from-zone untrust to-zone trust]
user@member2# set policy deny2 match source-address any destination-address
10_1_0_0_16 application any
user@member2# set policy deny2 then reject
user@member2# set policy scope2 match source-address 10_subnet
destination-address 10_subnet application any
user@member2# set policy scope2 then permit tunnel ipsec-group-vpn v2
user@member2# set policy multicast-scope2 match source-address 10_subnet
destination-address multicast-net application any
user@member2# set policy multicast-scope2 then permit tunnel ipsec-group-vpn
v2
```

**Results** From configuration mode, confirm your configuration by entering the `show security group-vpn member` and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.



```
[edit]
user@member2# show security group-vpn member
ike {
  proposal prop2 {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
  }
  policy pol2 {
    mode main;
    proposals prop2;
    pre-shared-key ascii-text "$9$Hm5FCA0BEy"; ## SECRET-DATA
  }
  gateway g2 {
    ike-policy pol2;
    address 20.0.0.1;
    local-address 10.2.0.1;
  }
}
ipsec {
  vpn v2 {
    ike-gateway g2;
    group-vpn-external-interface ge-0/1/0;
    group 1;
  }
}

[edit]
user@member2# show security policies
from-zone trust to-zone trust {
  policy default-permit {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone trust to-zone untrust {
  policy deny2 {
    match {
      source-address 10_1_0_0_16;
      destination-address any;
      application any;
    }
    then {
      reject;
    }
  }
  policy scope2 {
    match {
      source-address 10_subnet;
    }
  }
}
```

```
        destination-address 10_subnet;
        application any;
    }
    then {
        permit {
            tunnel {
                ipsec-group-vpn v2;
            }
        }
    }
}
policy multicast-scope2 {
    match {
        source-address 10_subnet;
        destination-address multicast-net;
        application any;
    }
    then {
        permit {
            tunnel {
                ipsec-group-vpn v2;
            }
        }
    }
}
policy default-permit {
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
}
from-zone untrust to-zone trust {
    policy deny2 {
        match {
            source-address any;
            destination-address 10_1_0_0_16;
            application any;
        }
        then {
            reject;
        }
    }
    policy scope2 {
        match {
            source-address 10_subnet;
            destination-address 10_subnet;
            application any;
        }
        then {
            permit {
                tunnel {
```

```

        ipsec-group-vpn v2;
    }
}
}
policy multicast-scope2 {
    match {
        source-address 10_subnet;
        destination-address multicast-net;
        application any;
    }
    then {
        permit {
            tunnel {
                ipsec-group-vpn v2;
            }
        }
    }
}
policy default-deny {
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        deny;
    }
}
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Dynamic Policies for Member1 on page 677
- Verifying Dynamic Policies for Member2 on page 678

#### *Verifying Dynamic Policies for Member1*

**Purpose** View the dynamic policies installed on member1.

**Action** After the group server downloads keys to member1, enter the **show security dynamic-policies** command from operational mode.

```

user@member1> show security dynamic-policies
Policy: scope1-0001, action-type: permit, State: enabled, Index: 1048580, AI: disabled,
Scope Policy: 4
Policy Type: Dynamic
Sequence number: 1
From zone: untrust, To zone: trust
Source addresses: 10.1.0.0/16
Destination addresses: 10.2.0.0/16
Application: Unknown

```

```

IP protocol: 0, ALG: 0, Inactivity timeout: 0
Source port range: [0-0]
Destination port range: [0-0]
Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
Policy: scope1-0001, action-type: permit, State: enabled, Index: 1048581, AI: disabled,
Scope Policy: 5
Policy Type: Dynamic
Sequence number: 2
From zone: trust, To zone: untrust
Source addresses: 10.1.0.0/16
Destination addresses: 10.2.0.0/16
Application: Unknown
IP protocol: 0, ALG: 0, Inactivity timeout: 0
Source port range: [0-0]
Destination port range: [0-0]
Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586

```

**Meaning** The multicast policy p3 from the server is not installed on member1 because there is no scope policy configured on member1 that allows multicast traffic.

#### *Verifying Dynamic Policies for Member2*

**Purpose** View the dynamic policies installed on member 2.

**Action** After the group server downloads keys to member2, enter the **show security dynamic-policies** command from operational mode.

```

user@member2> show security dynamic-policies
Policy: scope2-0001, action-type: permit, State: enabled, Index: 1048580, AI: disabled,
Scope Policy: 4
Policy Type: Dynamic
Sequence number: 1
From zone: untrust, To zone: trust
Source addresses: 10.1.0.0/16
Destination addresses: 10.2.0.0/16
Application: Unknown
IP protocol: 0, ALG: 0, Inactivity timeout: 0
Source port range: [0-0]
Destination port range: [0-0]
Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
Policy: scope2-0001, action-type: permit, State: enabled, Index: 1048580, AI: disabled,
Scope Policy: 4
Policy Type: Dynamic
Sequence number: 1
From zone: untrust, To zone: trust
Source addresses: 10.1.1.1/32
Destination addresses: 239.1.1.1/32
Application: Unknown
IP protocol: 0, ALG: 0, Inactivity timeout: 0
Source port range: [0-0]
Destination port range: [0-0]
Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
Policy: scope2-0001, action-type: permit, State: enabled, Index: 1048581, AI: disabled,
Scope Policy: 5
Policy Type: Dynamic

```

Sequence number: 2  
 From zone: trust, To zone: untrust  
 Source addresses: 10.2.0.0/16/0  
 Destination addresses: 10.1.0.0/16  
 Application: Unknown  
 IP protocol: 0, ALG: 0, Inactivity timeout: 0  
 Source port range: [0-0]  
 Destination port range: [0-0]  
 Tunnel: INSTANCE-gvpn\_133955586, Type: IPSec, Index: 133955586  
 Policy: scope2-0001, action-type: permit, State: enabled, Index: 1048581, AI: disabled,  
 Scope Policy: 5  
 Policy Type: Dynamic  
 Sequence number: 2  
 From zone: trust, To zone: untrust  
 Source addresses: 10.1.1.1/32  
 Destination addresses: 239.1.1.1/32  
 Application: Unknown  
 IP protocol: 0, ALG: 0, Inactivity timeout: 0  
 Source port range: [0-0]  
 Destination port range: [0-0]  
 Tunnel: INSTANCE-gvpn\_133955586, Type: IPSec, Index: 133955586

**Meaning** The policy p2 (for traffic from 10.1.0.0/16 to 10.2.0.0/16) from the server is not installed on member2, because it matches the deny2 security policy configured on member2.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Group VPN Overview on page 655](#)
  - [Group VPN Configuration Overview on page 663](#)
  - [Example: Configuring Group VPN with Server-Member Colocation on page 680](#)

## Colocation Mode

---

- [Understanding Colocation Mode on page 679](#)
- [Example: Configuring Group VPN with Server-Member Colocation on page 680](#)

## Understanding Colocation Mode

Group server and group member functions are separate and do not overlap. The server and member functions can coexist in the same physical device, which is referred as *colocation mode*. In colocation mode, there is no change in terms of functionality and behavior of the server or a member, but the server and member each need to be assigned different IP addresses so that packets can be delivered properly. In colocation mode, there can be only one IP address assigned to the server and one IP address assigned to the member across groups.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Group VPN Overview on page 655](#)
  - [Understanding Group Servers and Members on page 658](#)

- Understanding the GDOI Protocol on page 657
- Understanding Dynamic Policies on page 660
- Group VPN Configuration Overview on page 663
- Example: Configuring Group VPNs on page 663
- Example: Configuring Group VPN with Server-Member Colocation on page 680

## Example: Configuring Group VPN with Server-Member Colocation

This example shows how to configure a device for colocation mode, which allows server and member functions to coexist on the same physical device.

- Requirements on page 680
- Overview on page 680
- Configuration on page 681
- Verification on page 688

### Requirements

Before you begin:

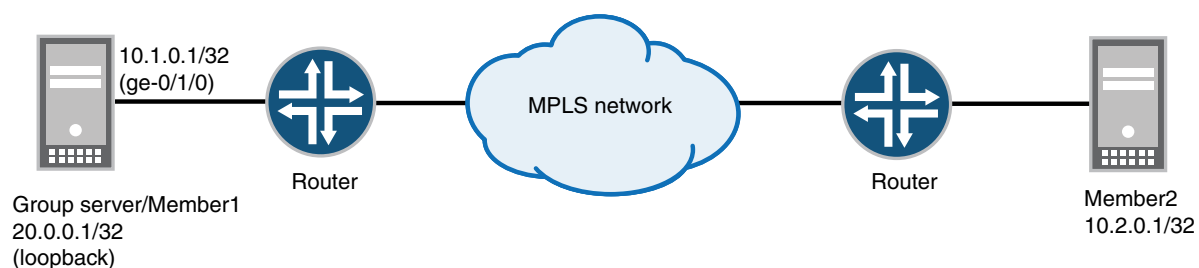
- Configure the Juniper Networks security devices for network communication.
- Configure network interfaces on server and member devices. See the *Junos OS Interfaces Configuration Guide for Security Devices*

### Overview

When colocation mode is configured, group server and group member functions can coexist in the same device. In colocation mode, the server and member must have different IP addresses so that packets are delivered properly.

In Figure 62 on page 680, a group VPN (group identifier is 1) consists of two members (member1 and member2) and a group server (the IP address of the loopback interface is 20.0.0.1). Note that member1 coexists in the same device as the group server. In this example, the interface that member1 uses to connect to the MPLS network (ge-0/1/0) is assigned the IP address 10.1.0.1/32.

Figure 62: Server-Member Colocation Example



g031042



**NOTE:** The configuration instructions in this topic describe how to configure the group server-member1 device for colocation mode. To configure member2, see “Example: Configuring Group VPNs” on page 663.



**NOTE:** To prevent packet fragmentation issues, we recommend that the interface used by the group member to connect to the MPLS network be configured for an MTU size no larger than 1400 bytes. Use the `set interface mtu` configuration statement to set the MTU size.

### Configuration

#### CLI Quick Configuration

To quickly configure group VPN with server-member colocation, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces lo0 unit 0 family inet address 20.0.0.1/32
set interfaces ge-0/1/0 unit 0 family inet address 10.1.0.1/32
set security group-vpn member ike proposal prop1 authentication-method pre-shared-keys
set security group-vpn member ike proposal prop1 dh-group group2
set security group-vpn member ike proposal prop1 authentication-algorithm sha1
set security group-vpn member ike proposal prop1 encryption-algorithm 3des-cbc
set security group-vpn member ike policy pol1 mode main
set security group-vpn member ike policy pol1 proposals prop1
set security group-vpn member ike policy pol1 pre-shared-key ascii-text "$9$c1gr
  K8-VYZUHX7UHqmF3Sre"
set security group-vpn member ike gateway g1 ike-policy pol1
set security group-vpn member ike gateway g1 address 20.0.0.1
set security group-vpn member ike gateway g1 local-address 10.1.0.1
set security group-vpn member ipsec vpn v1 ike-gateway g1
set security group-vpn member ipsec vpn v1 group-vpn-external-interface ge-0/1/0
set security group-vpn member ipsec vpn v1 group 1
set security group-vpn server ike proposal srv-prop authentication-method pre-shared-keys
set security group-vpn server ike proposal srv-prop dh-group group2
set security group-vpn server ike proposal srv-prop authentication-algorithm sha1
set security group-vpn server ike proposal srv-prop encryption-algorithm 3des-cbc
set security group-vpn server ike policy srv-pol mode main
set security group-vpn server ike policy srv-pol proposals srv-prop
set security group-vpn server ike policy srv-pol pre-shared-key ascii-text "$9$c
  1grK8-VYZUHX7UHqmF3Sre"
set security group-vpn server ike gateway gw1 ike-policy srv-pol
set security group-vpn server ike gateway gw1 address 10.1.0.1
set security group-vpn server ike gateway gw2 ike-policy srv-pol
set security group-vpn server ike gateway gw2 address 10.2.0.1
set security group-vpn server ipsec proposal group-prop authentication-algorithm
  hmac-sha1-96
set security group-vpn server ipsec proposal group-prop encryption-algorithm 3des-cbc
set security group-vpn server ipsec proposal group-prop lifetime-seconds 3600
set security group-vpn server group grp1 group-id 1
set security group-vpn server group grp1 ike-gateway gw1
set security group-vpn server group grp1 ike-gateway gw2
set security group-vpn server group grp1 anti-replay-time-window 120
```

```
set security group-vpn server group grp1 server-address 20.0.0.1
set security group-vpn server group grp1 server-member-communication
  communication-type unicast
set security group-vpn server group grp1 server-member-communication
  encryption-algorithm aes-128-cbc
set security group-vpn server group grp1 server-member-communication
  sig-hash-algorithm md5
set security group-vpn server group grp1 server-member-communication certificate
  srv-cert
set security group-vpn server group grp1 ipsec-sa group-sa proposal group-prop
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 source
  10.1.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 destination
  10.2.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 source-port
  0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 destination-port
  0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p1 protocol 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 source
  10.2.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 destination
  10.1.0.0/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 source-port
  0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2
  destination-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p2 protocol 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 source
  10.1.1.1/16
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 destination
  239.1.1.1/32
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 source-port
  0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3
  destination-port 0
set security group-vpn server group grp1 ipsec-sa group-sa match-policy p3 protocol 0
set security group-vpn co-location
set security group-vpn member ipsec vpn v1 ike-gateway g1
set security group-vpn member ipsec vpn v1 group-vpn-external-interface ge-0/1/0
set security zones security-zone trust address-book address 10_subnet 10.0.0.0/8
set security zones security-zone untrust address-book address 10_subnet 10.0.0.0 /8
set security policies from-zone trust to-zone untrust policy scope1 match source-address
  10_subnet
set security policies from-zone trust to-zone untrust policy scope1 match
  destination-address 10_subnet
set security policies from-zone trust to-zone untrust policy scope1 match application any
set security policies from-zone trust to-zone untrust policy scope1 then permit tunnel
  ipsec-group-vpn v1
set security policies from-zone untrust to-zone trust policy scope1 match source-address
  10_subnet
set security policies from-zone untrust to-zone trust policy scope1 match
  destination-address 10_subnet
set security policies from-zone untrust to-zone trust policy scope1 match application any
set security policies from-zone untrust to-zone trust policy scope1 then permit tunnel
  ipsec-group-vpn v1
```



**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure group VPN with server-member colocation:

1. Configure the loopback address on the device.

```
[edit interfaces]
user@host# set lo0 unit 0 family inet address 20.0.0.1/32
```

2. Configure the interface that member1 uses to connect to the MPLS network.

```
[edit interfaces]
user@host# set ge-0/1/0 unit 0 family inet address 10.1.0.1/32
```

3. Configure group VPN colocation on the device.

```
[edit security group-vpn]
user@host# set co-location
```

4. Configure IKE Phase 1 SA for the server (this configuration must match the Phase 1 SA configured on group members).

```
[edit security group-vpn server ike proposal srv-prop]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm 3des-cbc
```

5. Define the IKE policy and set the remote gateways.

```
[edit security group-vpn server ike]
user@host# set policy srv-pol proposals srv-prop mode main pre-shared-key
  ascii-text "$9$c1grK8-VYZUHX7UHqmF3Sre"
user@host# set gateway gw1 ike-policy srv-pol address 10.1.0.1
user@host# set gateway gw2 ike-policy srv-pol address 10.2.0.1
```

6. Configure the Phase 2 SA exchange for the server.

```
[edit security group-vpn server ipsec proposal group-prop]
user@host# set authentication-algorithm hmac-sha1-96
user@host# set encryption-algorithm 3des-cbc
user@host# set lifetime-seconds 3600
```

7. Configure the group identifier, IKE gateway, anti-replay time, and server address on the server.

```
[edit security group-vpn server group grp1]
user@host# set group-id 1 anti-replay-time-window 120 server-address 20.0.0.1
user@host# set ike-gateway gw1
user@host# set ike-gateway gw2
```

8. Configure server to member communications.

```
[edit security group-vpn server group grp1]
user@host# set server-member-communication communication-type unicast
  encryption-algorithm aes-128-cbc sig-hash-algorithm md5 certificate "srv-cert"
```

9. Configure the group policies to be downloaded to group members.

```
[edit security group-vpn server group grp1 ipsec-sa group-sa ]
```

```

user@host# set proposal group-prop match-policy p1 source 10.1.0.0/16 destination
10.2.0.0/16 source-port 0 destination-port 0 protocol 0
user@host# set proposal group-prop match-policy p2 source 10.2.0.0/16 destination
10.1.0.0/16 source-port 0 destination-port 0 protocol 0
user@host# set proposal group-prop match-policy p3 source 10.1.1.1/16 destination
239.1.1.1/32 source-port 0 destination-port 0 protocol 0

```

10. Configure Phase 1 SA for member1 (this configuration must match the Phase 1 SA configured for the group server).

```

[edit security group-vpn member ike proposal prop1]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group2
user@host# set authentication-algorithm sha1
user@host# set encryption-algorithm 3des-cbc

```

11. Define the policy and set the remote gateway for member1.

```

[edit security group-vpn member ike]
user@host# set policy pol1 mode main proposals prop1 pre-shared-key ascii-text
"$9$c1grK8-VYZUHX7UHqmF3Sre"
user@host# set gateway g1 ike-policy pol1 address 20.0.0.1 local-address 10.1.0.1

```

12. Configure the group identifier, IKE gateway, and interface for member1.

```

[edit security group-vpn member ipsec]
user@host# set vpn v1 group 1 ike-gateway g1 group-vpn-external-interface ge-0/1/0

```

13. Configure address book entries for the 10.0.0.0/8 subnet.

```

[edit security zones]
user@member1# set security-zone trust address-book address 10_subnet 10.0.0.0/8
user@member1# set security-zone untrust address-book address 10_subnet
10.0.0.0/8

```

14. Configure a scope policy from the trust zone to the untrust zone that allows unicast traffic to and from the 10.0.0.0/8 subnetwork.

```

[edit security policies from-zone trust to-zone untrust]
user@member1# set policy scope1 match source-address 10_subnet
destination-address 10_subnet application any
user@member1# set policy scope1 then permit tunnel ipsec-group-vpn v1

```

15. Configure a scope policy from the untrust zone to the trust zone that allows unicast traffic to and from the 10.0.0.0/8 subnetwork.

```

[edit security policies from-zone untrust to-zone trust]
user@member1# set policy scope1 match source-address 10_subnet
destination-address 10_subnet application any
user@member1# set policy scope1 then permit tunnel ipsec-group-vpn v1

```

**Results** From configuration mode, confirm your configuration by entering the **show security group-vpn** and **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.



**NOTE:** In the list of configured security policies, make sure that the scope policies are listed before the default policies.

```
[edit]
user@host# show security group-vpn
member {
  ike {
    proposal prop1 {
      authentication-method pre-shared-keys;
      dh-group group2;
      authentication-algorithm sha1;
      encryption-algorithm 3des-cbc;
    }
    policy pol1 {
      mode main;
      proposals prop1;
      pre-shared-key ascii-text "$9$c1grK8-VYZUHX7UHqmF3Sre"; ## SECRET-DATA
    }
    gateway g1 {
      ike-policy pol1;
      address 20.0.0.1;
      local-address 10.1.0.1;
    }
  }
}
ipsec {
  vpn v1 {
    ike-gateway g1;
    group-vpn-external-interface ge-0/1/0;
    group 1;
  }
}
server {
  ike {
    proposal srv-prop {
      authentication-method pre-shared-keys;
      dh-group group2;
      authentication-algorithm sha1;
      encryption-algorithm 3des-cbc;
    }
    policy srv-pol {
      mode main;
      proposals srv-prop;
      pre-shared-key ascii-text "$9$c1grK8-VYZUHX7UHqmF3Sre"; ## SECRET-DATA
    }
    gateway gw1 {
      ike-policy srv-pol;
      address 10.1.0.1;
    }
    gateway gw2 {
      ike-policy srv-pol;
      address 10.2.0.1;
    }
  }
}
```

```
    }
  }
  ipsec {
    proposal group-prop {
      authentication-algorithm hmac-sha1-96;
      encryption-algorithm 3des-cbc;
      lifetime-seconds 3600;
    }
  }
  group grp1 {
    group-id 1;
    ike-gateway gw1;
    ike-gateway gw2;
    anti-replay-time-window 120;
    server-address 20.0.0.1;
    server-member-communication {
      communication-type unicast;
      encryption-algorithm aes-128-cbc;
      sig-hash-algorithm md5;
      certificate srv-cert;
    }
    ipsec-sa group-sa {
      proposal group-prop;
      match-policy p1 {
        source 10.1.0.0/16;
        destination 10.2.0.0/16;
        source-port 0;
        destination-port 0;
        protocol 0;
      }
      match-policy p2 {
        source 10.2.0.0/16;
        destination 10.1.0.0/16;
        source-port 0;
        destination-port 0;
        protocol 0;
      }
      match-policy p3 {
        source 10.1.1.1/16;
        destination 239.1.1.1/32;
        source-port 0;
        destination-port 0;
        protocol 0;
      }
    }
  }
}
co-location;

[edit]
user@host# show security policies
from-zone trust to-zone trust {
  policy default-permit {
    match {
      source-address any;
      destination-address any;
    }
  }
}
```

```
        application any;
    }
    then {
        permit;
    }
}
}
from-zone trust to-zone untrust {
    policy scope1 {
        match {
            source-address 10_subnet;
            destination-address 10_subnet;
            application any;
        }
        then {
            permit {
                tunnel {
                    ipsec-group-vpn v1;
                }
            }
        }
    }
}
policy default-permit {
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
}
from-zone untrust to-zone trust {
    policy default-deny {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            deny;
        }
    }
}
policy scope1 {
    match {
        source-address 10_subnet;
        destination-address 10_subnet;
        application any;
    }
    then {
        permit {
            tunnel {
                ipsec-group-vpn v1;
            }
        }
    }
}
```

```
    }  
  }  
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

---

To confirm that the configuration is working properly, perform these tasks:

- Verifying Group VPN Member Registration on page 688
- Verifying Group VPN Server Security Associations for IKE on page 688
- Verifying Group VPN Server Security Associations for IPsec on page 688
- Verifying Group VPN Member Security Associations for IKE on page 688
- Verifying Group VPN Member Security Associations for IPsec on page 688

#### *Verifying Group VPN Member Registration*

**Purpose** Verify that the group VPN members are registered correctly.

**Action** From operational mode, enter the **show security group-vpn registered-members** command.

#### *Verifying Group VPN Server Security Associations for IKE*

**Purpose** Verify the SAs for the group VPN server for IKE.

**Action** From operational mode, enter the **show security group-vpn server ike security-associations** command.

#### *Verifying Group VPN Server Security Associations for IPsec*

**Purpose** Verify the SAs for the group VPN server for IPsec.

**Action** From operational mode, enter the **show security group-vpn server ipsec security-associations** command.

#### *Verifying Group VPN Member Security Associations for IKE*

**Purpose** Verify the SAs for the group VPN members for IKE.

**Action** From operational mode, enter the **show security group-vpn member ike security-associations** command.

#### *Verifying Group VPN Member Security Associations for IPsec*

**Purpose** Verify the SAs for the group VPN members for IPsec.

**Action** From operational mode, enter the **show security group-vpn member ipsec security-associations** command.

- Related Documentation**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Group VPN Overview on page 655
  - Group VPN Configuration Overview on page 663
  - Example: Configuring Group VPNs on page 663

## Server-Group Communications

- Understanding Server-Member Communication on page 689
- Understanding Group Key Operations on page 690
- Understanding Heartbeat Messages on page 693
- Example: Configuring Server-Member Communication for Unicast Rekey Messages on page 694
- Example: Configuring Server-Member Communication for Multicast Rekey Messages on page 695

### Understanding Server-Member Communication

Server-member communication allows the server to send GDOI **groupkey-push** messages to members. If server-member communication is not configured for the group, members can send GDOI **groupkey-pull** messages to register and reregister with the server, but the server is not able to send rekey messages to members.

Server-member communication is configured for the group by using the **server-member-communication** configuration statement at the [edit security group-vpn server] hierarchy. The following options can be defined:

- Encryption algorithm used for communications between the server and member. You can specify 3des-cbc, aes-128-cbc, aes-192-cbc, aes-256-cbc, or des-cbc. There is no default algorithm.
- Authentication algorithm (md5 or sha1) used to authenticate the member to the server. There is no default algorithm.
- Whether the server sends unicast or multicast rekey messages to group members and parameters related to the communication type. See “Understanding Group Key Operations” on page 690.
- Interval at which the server sends heartbeat messages to the group member. This allows the member to determine whether the server has rebooted, which would require the member to reregister with the server. The default is 300 seconds. See “Understanding Heartbeat Messages” on page 693.
- Lifetime for the key encryption key (KEK). The default is 3600 seconds.



**NOTE:** Configuring server-member communication is necessary for the group server to send rekey messages to members, but there might be situations in which this behavior is not desired. For example, if group members are dynamic peers (such as in a home office), the devices are not always up and the IP address of a device might be different each time it is powered up. Configuring server-member communication for a group of dynamic peers can result in unnecessary transmissions by the server. If you want IKE Phase 1 SA negotiation to always be performed to protect GDOI negotiation, do not configure server-member communication.

If server-member communication for a group is not configured, the membership list displayed by the **show security group-vpn server registered-members** command shows group members who have registered with the server; members can be active or not. When server-member communication for a group is configured, the group membership list is cleared. If the communication type is configured as unicast, the **show security group-vpn server registered-members** command shows only active members. If the communication type is configured as multicast, the **show security group-vpn server registered-members** command shows members who have registered with the server after the configuration; the membership list does not necessarily represent active members because members might drop out after registration.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Group Key Operations on page 690](#)
- [Understanding VPN Group Configuration on page 662](#)
- [Example: Configuring Server-Member Communication for Unicast Rekey Messages on page 694](#)
- [Example: Configuring Server-Member Communication for Multicast Rekey Messages on page 695](#)

## Understanding Group Key Operations

This topic contains the following sections:

- [Group Keys on page 690](#)
- [Rekey Messages on page 691](#)
- [Member Registration on page 692](#)
- [Key Activation on page 692](#)

### Group Keys

The group server maintains a database to track the relationship among VPN groups, group members, and group keys. There are two kinds of group keys that the server downloads to members:

- **Key Encryption Key (KEK)**—Used to encrypt rekey messages. One KEK is supported per group.



- Traffic Encryption Key (TEK)—Used to encrypt and decrypt IPsec data traffic between group members.

The key associated with an SA is accepted by a group member only if there is a matching scope policy configured on the member. An accepted key is installed for the group VPN, whereas a rejected key is discarded.

### Rekey Messages

If the group is configured for server-member communications (see “Understanding Server-Member Communication” on page 689), the server periodically sends SA and key refreshes to group members with rekey (GDOI **groupkey-push**) messages. Rekey messages are sent before SAs expire; this ensures that valid keys are available for encrypting traffic between group members.

The server also sends rekey messages to provide new keys to members when there is a change in group membership or the group SA has changed (for example, a group policy is added or deleted).

Server-member communications options must be configured on the server to allow the server to send rekey messages to group members. These options specify the type of message and the intervals at which the messages are sent, as explained in the following sections:

- Types of Rekey Messages on page 691
- Rekey Intervals on page 692

#### **Types of Rekey Messages**

There are two types of rekey messages:

- Unicast rekey messages—The group server sends one copy of the rekey message to each group member. Upon receipt of the rekey message, members must send an acknowledgment (ACK) to the server. If the server does not receive an ACK from a member (including retransmission of rekey messages), the server considers the member to be inactive and removes it from the membership list. The server stops sending rekey messages to the member.

The **number-of-retransmission** and **retransmission-period** configuration statements for server-member communications control the resending of rekey messages by the server when no ACK is received from a member.

- Multicast rekey messages—The group server sends one copy of the rekey message from the specified outgoing interface to the configured multicast group address. Members do not send acknowledgment of receipt of multicast rekey messages. The registered membership list does not necessarily represent active members because members might drop out after initial registration. All members of the group must be configured to support multicast messages.



**NOTE:** IP multicast protocols must be configured to allow delivery of multicast traffic in the network. For detailed information about configuring multicast protocols on Juniper Networks devices, see the *Junos OS Multicast Protocols Configuration Guide*.

### Rekey Intervals

The interval at which the server sends rekey messages is calculated based on the values of the **lifetime-seconds** and **activation-time-delay** configuration statements at the [edit security group-vpn server group] hierarchy. The interval is calculated as **lifetime-seconds** minus  $4 * (\text{activation-time-delay})$ .

The **lifetime-seconds** for the KEK is configured as part of the server-member communications; the default is 3600 seconds. The **lifetime-seconds** for the TEK is configured for the IPsec proposal; the default is 3600 seconds. The **activation-time-delay** is configured for the group on the server; the default is 15 seconds. Using the default values for **lifetime-seconds** and **activation-time-delay**, the interval at which the server sends rekey messages is 3600 minus  $4 * 15$ , or 3540 seconds.

### Member Registration

If a group member does not receive a new SA key from the server before the current key expires, the member must reregister with the server and obtain updated keys with a GDOI **groupkey-pull** exchange. In this case, the interval at which the server sends rekey messages is calculated as follows: **lifetime-seconds** minus  $3 * (\text{activation-time-delay})$ . Using the default values for **lifetime-seconds** and **activation-time-delay**, the interval at which the server sends rekey messages is 3600 minus  $3 * 15$ , or 3555 seconds.

Member reregistration can occur for the following reasons:

- The member detects a server reboot by the absence of heartbeats received from the server.
- The rekey message from the group server is lost or delayed, and the TEK lifetime has expired.

### Key Activation

When a member receives a new key from the server, it waits a period of time before using the key for encryption. This period of time is determined by the **activation-time-delay** configuration statement and whether the key is received through a rekey message sent from the server or as a result of the member reregistering with the server.

If the key is received through a rekey message sent from the server, the member waits  $2 * (\text{activation-time-delay})$  seconds before using the key. If the key is received through member reregistration, the member waits the number of seconds specified by the **activation-time-delay** value.

A member retains the two most recent keys sent from the server for each group SA installed on the member. Both keys can be used for decryption, while the most recent

key is used for encryption. The previous key is removed the number of seconds specified by the **activation-time-delay** value after the new key is activated.

The default for the **activation-time-delay** configuration statement is 15 seconds. Setting this time period too small can result in a packet being dropped at a remote group member before the new key is installed. Consider the network topology and system transport delays when you change the **activation-time-delay** value. For unicast transmissions, the system transport delay is proportional to the number of group members.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Group VPN Overview on page 655](#)
- [Understanding the GDOI Protocol on page 657](#)
- [Understanding Group Servers and Members on page 658](#)
- [Understanding Dynamic Policies on page 660](#)
- [Group VPN Configuration Overview on page 663](#)

## Understanding Heartbeat Messages

When server-member communication is configured, the server sends heartbeat messages to members at specified intervals (the default interval is 300 seconds). The heartbeat mechanism allows members to reregister with the server if the specified number of heartbeats is not received. For example, members will not receive heartbeat messages during a server reboot. When the server has rebooted, members reregister with the server.

Heartbeats are transmitted through **groupkey-push** messages. The sequence number is incremented on each heartbeat message, which protects members from reply attacks. Unlike rekey messages, heartbeat messages are not acknowledged by recipients and are not retransmitted by the server.

Heartbeat messages contain the following information:

- Current state and configuration of the keys on the server
- Relative time, if antireplay is enabled

By comparing the information in the heartbeats, a member can detect whether it has missed server information or rekey messages. The member reregisters to synchronize itself with the server.



**NOTE:** Heartbeat messages can increase network congestion and cause unnecessary member reregistrations. Thus, heartbeat detection can be disabled on the member if necessary.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Group VPN Configuration Overview on page 663](#)
- [Understanding Server-Member Communication on page 689](#)

- Understanding the GDOI Protocol on page 657
- Understanding Group Servers and Members on page 658

## Example: Configuring Server-Member Communication for Unicast Rekey Messages

This example shows how to enable the server to send unicast rekey messages to group members to ensure that valid keys are available for encrypting traffic between group members.

- Requirements on page 694
- Overview on page 694
- Configuration on page 694
- Verification on page 695

### Requirements

---

Before you begin:

- Configure the group server and members for IKE Phase 1 negotiation.
- Configure the group server and members for Phase 2 IPsec SA.
- Configure the group **g1** on the group server.

### Overview

---

In this example, you specify the following server-member communication parameters for group **g1**:

- The server sends unicast rekey messages to group members.
- 3des-cbc is used to encrypt traffic between the server and members.
- sha1 is used for member authentication.

Default values are used for server heartbeats, KEK lifetime, and retransmissions.

### Configuration

---

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure server-member communication:

1. Set the communications type.  

```
[edit security group-vpn server group g1 server-member-communication]  
user@host# set communications-type unicast
```
2. Set the encryption algorithm.  

```
[edit security group-vpn server group g1 server-member-communication]  
user@host# set encryption-algorithm 3des-cbc
```
3. Set the member authentication.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set sig-hash-algorithm sha1
```

### Verification

To verify the configuration is working properly, enter the **show security group-vpn server group g1 server-member-communication** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Group VPN Configuration Overview on page 663
- Understanding Server-Member Communication on page 689
- Understanding Group Key Operations on page 690
- Understanding VPN Group Configuration on page 662

### Example: Configuring Server-Member Communication for Multicast Rekey Messages

This example shows how to enable the server to send multicast rekey messages to group members to ensure that valid keys are available for encrypting traffic between group members.

- Requirements on page 695
- Overview on page 696
- Configuration on page 696
- Verification on page 697

### Requirements

Before you begin:

- Configure the group server and members for IKE Phase 1 negotiation and Phase 2 IPsec SA. See “Example: Configuring Group VPNs” on page 663 or “Example: Configuring Group VPN with Server-Member Colocation” on page 680.
- Configure ge-0/0/1.0, which is the interface the server will use for sending multicast messages. See *Junos OS Routing Protocols and Policies Configuration Guide for Security Devices*.
- Configure the multicast group address 226.1.1.1. See *Junos OS Routing Protocols and Policies Configuration Guide for Security Devices*.



**NOTE:** IP multicast protocols must be configured to allow delivery of multicast traffic in the network. This example does not show multicast configuration. For information about configuring multicast protocols on Juniper Networks security devices, see the *Junos OS Routing Protocols and Policies Configuration Guide for Security Devices*.

## Overview

In this example, you specify the following server-member communication for group **g1**:

- The server sends multicast rekey messages to group members by means of multicast address 226.1.1.1 and interface ge-0/0/1.0.
- 3des-cbc is used to encrypt traffic between the server and members.
- sha1 is used for member authentication.

Default values are used for server heartbeats, KEK lifetime, and retransmissions.

## Configuration

### CLI Quick Configuration

To quickly configure the server to send multicast rekey messages to group members, copy the following commands and paste them into the CLI:

```
[edit]
set security group-vpn server group g1 server-member-communication
  communication-type multicast
set security group-vpn server group g1 server-member-communication multicast-group
  226.1.1.1
set security group-vpn server group g1 server-member-communication
  multicast-outgoing-interface ge-0/0/1.0
set security group-vpn server group g1 server-member-communication
  encryption-algorithm 3des-cbc
set security group-vpn server group g1 server-member-communication sig-hash-algorithm
  sha1
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure server-member communication for multicast rekey messages:

1. Set the communications type.
 

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set communication-type multicast
```
2. Set the multicast group.
 

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set multicast-group 226.1.1.1
```
3. Set the interface for outgoing multicast messages.
 

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set multicast-outgoing-interface ge-0/0/1.0
```
4. Set the encryption algorithm.
 

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set encryption-algorithm 3des-cbc
```
5. Set the member authentication.
 

```
[edit security group-vpn server group g1 server-member-communication]
```

```
user@host# set sig-hash-algorithm sha1
```

**Results** From configuration mode, confirm your configuration by entering the **show security group-vpn server group g1 server-member-communication** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security group-vpn server group g1 server-member-communication
communication-type multicast;
multicast-group 226.1.1.1;
multicast-outgoing-interface ge-0/0/1.0;
encryption-algorithm 3des-cbc;
sig-hash-algorithm sha1;
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Server-Member Communication for Multicast Rekey Messages on page 697

#### *Verifying Server-Member Communication for Multicast Rekey Messages*

**Purpose** Verify that server-member communication parameters for multicast rekey message are configured properly to ensure that valid keys are available for encrypting traffic between group members.

**Action** From operational mode, enter the **show security group-vpn server group g1 server-member-communication** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Group VPN Configuration Overview on page 663
- Understanding Server-Member Communication on page 689
- Understanding Group Key Operations on page 690
- Understanding VPN Group Configuration on page 662

## Understanding Group VPN Limitations

The following are not supported in this release for group VPNs:

- Non-default routing instances
- Chassis cluster
- Server clusters
- Route-based group VPN
- Public Internet-based deployment

- SNMP
- Deny policy from Cisco GET VPN server
- J-Web interface for configuration and monitoring

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Group VPN Overview on page 655
- Understanding the GDOI Protocol on page 657
- Understanding Group Servers and Members on page 658

## Understanding Interoperability with Cisco GET VPN

---

Cisco's implementation of GDOI is called Group Encryption Transport (GET) VPN. While group VPN in Junos OS and Cisco's GET VPN are both based on RFC 3547, *The Group Domain of Interpretation*, there are some implementation differences that you need to be aware of when deploying GDOI in a networking environment that includes both Juniper Networks security devices and Cisco routers. For more information, see the current Junos OS Release notes.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- VPN Overview on page 451
- Understanding IKE and IPsec Packet Processing on page 458
- Understanding the GDOI Protocol on page 657
- Understanding Group Servers and Members on page 658



## PART 7

# Intrusion Detection and Prevention

- IDP Policies on page 701
- Application-Level Distributed Denial of Service on page 763
- IDP Signature Database on page 777
- IDP Application Identification on page 795
- IDP SSL Inspection on page 809
- IDP Class of Service Action on page 817
- IDP Performance and Capacity Tuning on page 825
- IDP Logging on page 827



# IDP Policies

- IDP Policies Overview on page 701
- Example: Enabling IDP in a Security Policy on page 702
- IDP Inline Tap Mode on page 705
- IDP Rules and Rulebases on page 707
- IDP Applications and Application Sets on page 730
- IDP Attacks and Attack Objects on page 735

## IDP Policies Overview

---

The Junos OS Intrusion Detection and Prevention (IDP) policy enables you to selectively enforce various attack detection and prevention techniques on network traffic passing through an IDP-enabled device. It allows you to define policy rules to match a section of traffic based on a zone, network, and application, and then take active or passive preventive actions on that traffic.

An IDP policy defines how your device handles the network traffic. It allows you to enforce various attack detection and prevention techniques on traffic traversing your network.

A policy is made up of *rulebases* and each rulebase contains a set of *rules*. You define rule parameters, such as traffic match conditions, action, and logging requirements, then add the rules to rule bases. After you create an IDP Policy by adding rules in one or more rulebases, you can select that policy to be the active policy on your device.

Junos OS allows you to configure multiple IDP policies, but a device can have only one active IDP policy at a time. You can install the same IDP policy on multiple devices, or you can install a unique IDP policy on each device in your network. A single policy can contain only one instance of any type of rulebase.

You can perform the following tasks to manage IDP policies:

- Create new IDP policies starting from scratch. See “Example: Defining Rules for an IDP IPS Rulebase” on page 717.
- Create an IDP policy starting with one of the predefined templates provided by Juniper Networks (see “Understanding Predefined IDP Policy Templates” on page 778).
- Add or delete rules within a rulebase. You can use any of the following IDP objects to create rules:

- Zone and network objects available in the base system
- Predefined service objects provided by Juniper Networks
- Custom application objects
- Predefined attack objects provided by Juniper Networks
- Create custom attack objects (see “Example: Configuring IDP Signature-Based Attacks” on page 755).
- Update the signature database provided by Juniper Networks. This database contains all predefined objects.
- Maintain multiple IDP policies. Any one of the policies can be applied to the device.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding IDP Policy Rules on page 707](#)
- [Understanding IDP Terminal Rules on page 724](#)
- [Understanding IDP Application Sets on page 730](#)
- [Understanding Custom Attack Objects on page 736](#)
- [Example: Enabling IDP in a Security Policy on page 702](#)

## Example: Enabling IDP in a Security Policy

---

This example shows how to configure two security policies to enable IDP services on all traffic flowing in both directions on the device.

- [Requirements on page 702](#)
- [Overview on page 702](#)
- [Configuration on page 703](#)
- [Verification on page 705](#)

### Requirements

Before you begin:

- Configure network interfaces. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).
- Create security zones. See “Example: Creating Security Zones” on page 114.
- Configure applications. See “Example: Configuring IDP Applications and Services” on page 731.

### Overview

For transit traffic to pass through IDP inspection, you configure a security policy and enable IDP application services on all traffic that you want to inspect. Security policies

contain rules defining the types of traffic permitted on the network and the way that the traffic is treated inside the network. Enabling IDP in a security policy directs traffic that matches the specified criteria to be checked against the IDP rulebases.

To allow transit traffic to pass through without IDP inspection, specify a *permit* action for the rule without enabling the IDP application services. Traffic matching the conditions in this rule passes through the device without IDP inspection.

This example shows how to configure two policies, `idp-app-policy-1` and `idp-app-policy-2`, to enable IDP services on all traffic flowing in both directions on the device. The `idp-app-policy-1` policy directs all traffic flowing from previously configured Zone1 to Zone2 to be checked against IDP rulebases. The `idp-app-policy-2` policy directs all traffic flowing from Zone2 to Zone1 to be checked against IDP rulebases.



**NOTE:** The action set in the security policy action must be *permit*. You cannot enable IDP for traffic that the device denies or rejects.

## Configuration

### CLI Quick Configuration

To quickly enable IDP services on all traffic flowing in both directions on the device, copy the following commands and paste them into the CLI:

[edit]

```
set security policies from-zone Zone1 to-zone Zone2 policy idp-app-policy-1 match
  source-address any destination-address any application any
```

```
set security policies from-zone Zone1 to-zone Zone2 policy idp-app-policy-1 then permit
  application-services idp
```

```
set security policies from-zone Zone2 to-zone Zone1 policy idp-app-policy-2 match
  source-address any destination-address any application any
```

```
set security policies from-zone Zone2 to-zone Zone1 policy idp-app-policy-2 then permit
  application-services idp
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To enable IDP services on all traffic flowing in both directions on the device:

1. Create a security policy for the traffic flowing in one direction.

```
[edit security policies from-zone Zone1 to-zone Zone2 policy idp-app-policy-1]
user@host# set match source-address any destination-address any application
any
```

2. Specify the action to be taken on traffic that matches conditions specified in the policy.

```
[edit security policies from-zone Zone1 to-zone Zone2 policy idp-app-policy-1]
```

```
user@host# set then permit application-services idp
```

3. Create another security policy for the traffic flowing in the other direction.

```
[edit security policies from-zone Zone2 to-zone Zone1 policy idp-app-policy-2]
user@host# set match source-address any destination-address any application
any
```

4. Specify the action to be taken on traffic that matches the conditions specified in the policy.

```
[edit security policies from-zone Zone2 to-zone Zone1 policy idp-app-policy-2]
user@host# set then permit application-services idp
```

**Results** From configuration mode, confirm your configuration by entering the `show security policies` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security policies
from-zone Zone1 to-zone Zone2 {
  policy idp-app-policy-1 {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit {
        application-services {
          idp;
        }
      }
    }
  }
}
from-zone Zone2 to-zone Zone1 {
  policy idp-app-policy-2 {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit {
        application-services {
          idp;
        }
      }
    }
  }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

## Verification

To confirm that the configuration is working properly, perform this task:

- [Verifying the Configuration on page 705](#)

### Verifying the Configuration

---

**Purpose** Verify that the security policy configuration is correct.

**Action** From operational mode, enter the **show security policies** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [IDP Policies Overview on page 701](#)
- [Understanding IDP Policy Rules on page 707](#)
- [Understanding IDP Policy Rulebases on page 713](#)

## IDP Inline Tap Mode

---

- [Understanding IDP Inline Tap Mode on page 705](#)
- [Example: Configuring IDP Inline Tap Mode on page 706](#)

## Understanding IDP Inline Tap Mode

The main purpose of inline tap mode is to provide best case deep inspection analysis of traffic while maintaining over all performance and stability of the device. The inline tap feature provides passive, inline detection of application layer threats for traffic matching security policies which have the IDP application service enabled. When a device is in inline tap mode, packets pass through firewall inspection and are also copied to the independent IDP module. This allows the packets to get to the next service module without waiting for IDP processing results. By doing this, when the traffic input is beyond the IDP throughput limit, the device can still sustain processing as long as it does not go beyond the modules limits, such as with the firewall. If the IDP process fails, all other features of the device will continue to function normally. Once the IDP process recovers, it will resume processing packets for inspection. Since inline tap mode puts IDP in a passive mode for monitoring, preventative actions such as session close, drop, and mark diffserv are deferred. The action drop packet is ignored.

Inline tap mode can only be configured if the forwarding process mode is set to maximize IDP sessions, which ensures stability and resiliency for firewall services. You also do not need a separate tap or span port to use inline tap mode.



**NOTE:** You must restart the device when switching to inline tap mode or back to regular mode.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Example: Configuring IDP Inline Tap Mode on page 706](#)
  - [IDP Policies Overview on page 701](#)
  - [Understanding IDP Policy Rules on page 707](#)
  - [Understanding IDP Policy Rulebases on page 713](#)

## Example: Configuring IDP Inline Tap Mode

This example shows how to configure a device for inline tap mode.

### Requirements

Before you begin, review the inline tap mode feature. See “Understanding IDP Inline Tap Mode” on page 705.

### Overview

The inline tap mode feature provides passive, inline detection of Application Layer threats for traffic matching security policies that have the IDP application service enabled.



**NOTE:** IDP inline tap mode does not require a separate tap or span port.

### Configuration

#### Step-by-Step Procedure

To configure a device for inline tap mode:

1. Set inline tap mode.

```
[edit]
user@host# set security forwarding-process application-services
maximize-idp-sessions inline-tap
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

3. Restart the system from operational mode.

```
user@host> request system reboot
```



**NOTE:** When switching to inline tap mode or back to regular mode, you must restart the device .

4. If you want to switch the device back to regular mode, delete inline tap mode configuration.

```
[edit security]
user@host# delete forwarding-process application-services maximize-idp-sessions
inline-tap
```



---

### Verification

To verify that inline tap mode is enabled, enter the **show security idp status** command. The line item for the forwarding process mode shows “Forwarding process mode : maximizing sessions (Inline-tap)”.

#### Related Documentation

- IDP Policies Overview on page 701
- Understanding IDP Policy Rules on page 707
- Understanding IDP Policy Rulebases on page 713

## IDP Rules and Rulebases

---

- Understanding IDP Policy Rules on page 707
- IDP Rulebases on page 713
- Understanding IDP Application-Level DDoS Rulebases on page 715
- IDP IPS Rulebase on page 716
- IDP Exempt Rulebase on page 721
- IDP Terminal Rules on page 724
- IDP DSCP Rules on page 727

### Understanding IDP Policy Rules

Each instruction in an Intrusion Detection and Prevention (IDP) policy is called a rule. Rules are created in rulebases.

Rulebases are a set of rules that combine to define an IDP policy. Rules provide context to detection mechanisms by specifying which part of the network traffic the IDP system should look in to find attacks. When a rule is matched, it means that an attack has been detected in the network traffic, triggering the action for that rule. The IDP system performs the specified action and protects your network from that attack.

IDP policy rules are made up of the following components:

- Understanding IDP Rule Match Conditions on page 707
- Understanding IDP Rule Objects on page 708
- Understanding IDP Rule Actions on page 710
- Understanding IDP Rule IP Actions on page 711
- Understanding IDP Rule Notifications on page 712

### Understanding IDP Rule Match Conditions

---

Match conditions specify the type of network traffic you want IDP to monitor for attacks.

Match conditions use the following characteristics to specify the type of network traffic to be monitored:

- **From-zone and to-zone**—All traffic flows from a source to a destination zone. You can select any zone for the source or destination. You can also use zone exceptions to specify unique to and from zones for each device. Specify **any** to monitor network traffic originating from and to any zone. The default value is **any**.
- **Source IP Address**—Specify the source IP address from which the network traffic originates. You can specify **any** to monitor network traffic originating from any IP address. You can also specify **source-except** to specify all sources except the specified addresses. The default value is **any**.
- **Destination IP address**—Specify the destination IP address to which the network traffic is sent. You can set this to **any** to monitor network traffic sent to any IP address. You can also specify **destination-except** to specify all destinations except the specified addresses. The default value is **any**.
- **Application**—Specify the Application Layer protocols supported by the destination IP address. You can specify **any** for all applications and **default** for the application configured in the attack object for the rule.

### Understanding IDP Rule Objects

---

Objects are reusable logical entities that you can apply to rules. Each object that you create is added to a database for the object type.

You can configure the following types of objects for IDP rules.

#### **Zone Objects**

A zone or security zone is a collection of one or more network interfaces. IDP uses zone objects configured in the base system.

#### **Address or Network Objects**

Address objects represent components of your network, such as host machines, servers, and subnets. You use address objects in IDP policy rules to specify the network components that you want to protect.

#### **Application or Service Objects**

Service objects represent network services that use Transport Layer protocols such as TCP, UDP, RPC, and ICMP. You use service objects in rules to specify the service an attack uses to access your network. Juniper Networks provides predefined service objects, a database of service objects that are based on industry-standard services. If you need to add service objects that are not included in the predefined service objects, you can create custom service objects. IDP supports the following types of service objects:

- **Any**—Allows IDP to match all Transport Layer protocols.
- **TCP**—Specifies a TCP port or a port range to match network services for specified TCP ports. You can specify **junos-tcp-any** to match services for all TCP ports.
- **UDP**—Specifies a UDP port or a port range to match network services for specified UDP ports. You can specify **junos-udp-any** to match services for all UDP ports.

- **RPC**—Specifies a remote procedure call (RPC from Sun Microsystems) program number or a program number range. IDP uses this information to identify RPC sessions.
- **ICMP**—Specifies a type and code that is a part of an ICMP packet. You can specify `junos-icmp-all` to match all ICMP services.

### Attack Objects

IDP attack objects represent known and unknown attacks. IDP includes a predefined attack object database that is periodically updated by Juniper Networks. Attack objects are specified in rules to identify malicious activity. Each attack is defined as an attack object, which represents a known pattern of attack. Whenever this known pattern of attack is encountered in the monitored network traffic, the attack object is matched. The three main types of attack objects are described in Table 77 on page 709:

**Table 77: IDP Attack Objects Description**

Attack Objects	Description
Signature Attack Objects	Signature attack objects detect known attacks using stateful attack signatures. An attack signature is a pattern that always exists within an attack; if the attack is present, so is the attack signature. With stateful signatures, IDP can look for the specific protocol or service used to perpetrate the attack, the direction and flow of the attack, and the context in which the attack occurs. Stateful signatures produce few false positives because the context of the attack is defined, eliminating huge sections of network traffic in which the attack would not occur.
Protocol Anomaly Attack Objects	Protocol anomaly attack objects identify unusual activity on the network. They detect abnormal or ambiguous messages within a connection according to the set of rules for the particular protocol being used. Protocol anomaly detection works by finding deviations from protocol standards, most often defined by RFCs and common RFC extensions. Most legitimate traffic adheres to established protocols. Traffic that does not, produces an anomaly, which may be created by attackers for specific purposes, such as evading an intrusion prevention system (IPS).
Compound Attack Objects	A compound attack object combines multiple signatures and/or protocol anomalies into a single object. Traffic must match all of the combined signatures and/or protocol anomalies to match the compound attack object; you can specify the order in which signatures or anomalies must match. Use compound attack objects to refine your IDP policy rules, reduce false positives, and increase detection accuracy. A compound attack object enables you to be very specific about the events that need to occur before IDP identifies traffic as an attack. You can use <b>And</b> , <b>Or</b> , and <b>Ordered and</b> operations to define the relationship among different attack objects within a compound attack and the order in which events occur.

### Attack Object Groups

IDP contains a large number of predefined attack objects. To help keep IDP policies organized and manageable, attack objects can be grouped. An attack object group can contain one or more attack objects of different types. Junos OS supports the following two types of attack groups:

- Static groups—Contain a fixed set of attack objects.
- Dynamic groups—Contain attack objects based on a certain matching criteria. For example, a dynamic group can contain all attacks related to an application. During signature update, the dynamic group membership is automatically updated based on the matching criteria for that group.

### Understanding IDP Rule Actions

*Actions* specify the actions you want IDP to take when the monitored traffic matches the attack objects specified in the rules.

Table 78 on page 710 shows the actions you can specify for IDP rules:

**Table 78: IDP Rule Actions**

Term	Definition
<b>No Action</b>	No action is taken. Use this action when you only want to generate logs for some traffic.
<b>Ignore Connection</b>	Stops scanning traffic for the rest of the connection if an attack match is found. IDP disables the rulebase for the specific connection.  <b>NOTE:</b> This action does not mean ignore an attack.
<b>Diffserv Marking</b>	Assigns the indicated Differentiated Services code point (DSCP) value to the packet in an attack, then passes the packet on normally.  Note that DSCP value is not applied to the first packet that is detected as an attack, but is applied to subsequent packets.
<b>Drop Packet</b>	Drops a matching packet before it can reach its destination but does not close the connection. Use this action to drop packets for attacks in traffic that is prone to spoofing, such as UDP traffic. Dropping a connection for such traffic could result in a denial of service that prevents you from receiving traffic from a legitimate source-IP address.  <b>NOTE:</b> When an IDP policy is configured using a non-packet context defined in a custom signature for any application and has the action drop_packet, when IDP identifies an attack the decoder will promote drop_packet to drop_connection. With a DNS protocol attack, this is not the case. The DNS decoder will not promote drop_packet to drop_connection when an attack is identified. This will ensure that only DNS attack traffic will be dropped and valid DNS requests will continue to be processed. This will also avoid TCP retransmission for the valid TCP DNS requests..

Table 78: IDP Rule Actions (*continued*)

Term	Definition
<b>Drop Connection</b>	Drops all packets associated with the connection, preventing traffic for the connection from reaching its destination. Use this action to drop connections for traffic that is not prone to spoofing.
<b>Close Client</b>	Closes the connection and sends an RST packet to the client but not to the server.
<b>Close Server</b>	Closes the connection and sends an RST packet to the server but not to the client.
<b>Close Client and Server</b>	Closes the connection and sends an RST packet to both the client and the server.
<b>Recommended</b>	<p>All predefined attack objects have a default action associated with them. This is the action that Juniper Networks recommends when that attack is detected.</p> <p><b>NOTE:</b> This action is supported only for IPS rulebases.</p> <p>Recommended —A list of all attack objects that Juniper Networks considers to be serious threats, organized into categories.</p> <ul style="list-style-type: none"> <li>• Attack type groups attack objects by type (anomaly or signature). Within each type, attack objects are grouped by severity.</li> <li>• Category groups attack objects by predefined categories. Within each category, attack objects are grouped by severity.</li> <li>• Operating system groups attack objects by the operating system to which they apply: BSD, Linux, Solaris, or Windows. Within each operating system, attack objects are grouped by services and severity.</li> <li>• Severity groups attack objects by the severity assigned to the attack. IDP has five severity levels: Critical, Major, Minor, Warning, and Info. Within each severity, attack objects are grouped by category.</li> </ul>

### Understanding IDP Rule IP Actions

IP actions are actions that apply on future connections that use the same IP action attributes. For example, you can configure an IP action in the rule to block all future HTTP sessions between two hosts if an attack is detected on a session between the hosts. Or you can specify a timeout value that defines that the action should be applied only if new sessions are initiated within that specified timeout value. The default timeout value for IP actions is 0, which means that IP actions are never timed out.

IP actions are similar to other actions; they direct IDP to drop or close the connection. However, because you now also have the attacker's IP address, you can choose to block the attacker for a specified time. If attackers cannot immediately regain a connection to your network, they might try to attack easier targets. Use IP actions in conjunction with actions and logging to secure your network.

IP action attributes are a combination of the following fields:

- Source IP address
- Destination IP address
- Destination port
- From-zone
- Protocol

Table 79 on page 712 summarizes the types IP actions supported by IDP rules:

**Table 79: IDP Rule IP Actions**

Term	Definition
<b>Notify</b>	Does not take any action against future traffic, but logs the event. This is the default.
<b>Drop/Block Session</b>	All packets of any session matching the IP action rule are dropped silently.
<b>Close Session</b>	Any new sessions matching this IP action rule are closed by sending RST packets to the client and server.

When traffic matches multiple rules, the most severe IP action of all matched rules is applied. The most severe IP action is the Close Session action, the next in severity is the Drop/Block Session action, and then the Notify action.

### Understanding IDP Rule Notifications

Notification defines how information is to be logged when an action is performed. When attacks are detected, you can choose to log an attack and create log records with attack information and send that information to the log server.

By using notifications, you can also configure the following options that instruct the log server to perform specific actions on logs generated for each rule:

- **Set Alerts**—Specify an alert option for a rule in the IDP policy. When the rule is matched, the corresponding log record displays an alert in the alert column of the Log Viewer. Security administrators use alerts to become aware of and react to important security events.
- **Set Severity Level**—Set severity levels in logging to support better organization and presentation of log records on the log server. You can use the default severity settings of the selected attack objects or choose a specific severity for your rule. The severity you configure in the rules overrides the inherited attack severity. You can set the severity level to the following levels:
  - Info—2
  - Warning—3
  - Minor—4

- Major—5
- Critical—7

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding IDP Policy Rulebases on page 713
- Understanding IDP Application-Level DDoS Rulebases on page 715
- Understanding IDP IPS Rulebases on page 716
- Understanding IDP Exempt Rulebases on page 721
- Understanding IDP Terminal Rules on page 724
- Understanding DSCP Rules in IDP Policies on page 727
- Understanding Predefined IDP Policy Templates on page 778

## IDP Rulebases

- Understanding IDP Policy Rulebases on page 713
- Example: Inserting a Rule in the IDP Rulebase on page 714
- Example: Deactivating and Activating Rules in an IDP Rulebase on page 714

### Understanding IDP Policy Rulebases

Intrusion Detection and Prevention (IDP) policies are collections of rules and rulebases. A rulebase is an ordered set of rules that use a specific detection method to identify and prevent attacks.

Rules are instructions that provide context to detection mechanisms by specifying which part of the network traffic the IDP system should look in to find attacks. When a rule is matched, it means that an attack has been detected in the network traffic, triggering the action for that rule. The IDP system performs the specified action and protects your network from that attack.

Each rulebase can have multiple rules—you determine the sequence in which rules are applied to network traffic by placing them in the desired order. Each rulebase in the IDP system uses specific detection methods to identify and prevent attacks. Junos OS supports two types of rulebases—intrusion prevention system (IPS) rulebase and exempt rulebase.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding IDP Policy Rules on page 707
- Understanding IDP Application-Level DDoS Rulebases on page 715
- Understanding IDP IPS Rulebases on page 716
- Understanding IDP Exempt Rulebases on page 721
- Example: Inserting a Rule in the IDP Rulebase on page 714
- Example: Deactivating and Activating Rules in an IDP Rulebase on page 714

### Example: Inserting a Rule in the IDP Rulebase

---

This example shows how to insert a rule in the IDP rulebase.

#### Requirements

Before you begin:

- Configure network interfaces. See [Junos OS Interfaces Configuration Guide for Security Devices](#).
- Define rules in a rulebase. See “Example: Defining Rules for an IDP IPS Rulebase” on page 717.

#### Overview

The IDP rule-matching algorithm starts from the top of the rulebase and checks traffic against all rules in the rulebase that match the specified match conditions. You determine the sequence in which rules are applied to network traffic by placing them in the desired order. When you add a rule to the rulebase, it is placed at the end of the existing list of rules. To place a rule in any other location than at the end of the rulebase, you *insert* the rule at the desired location in the rulebase. This example places rule R2 before rule R1 in the IDP IPS rulebase in a policy called base-policy.

#### Configuration

#### Step-by-Step Procedure

To insert a rule in the rulebase:

1. Define the position of the rule in the rulebase based on the order in which you want the rule to be evaluated.

```
[edit]
user@host# insert security idp idp-policy base-policy rulebase-ips rule R2 before
rule R1
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

#### Verification

To verify the configuration is working properly, enter the **show security idp status** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding IDP Policy Rules on page 707](#)
- [Understanding IDP Policy Rulebases on page 713](#)
- [Example: Defining Rules for an IDP Exempt Rulebase on page 722](#)
- [Example: Enabling IDP Protection Against Application-Level DDoS Attacks on page 768](#)
- [Example: Enabling IDP in a Security Policy on page 702](#)

### Example: Deactivating and Activating Rules in an IDP Rulebase

---

This example shows how to deactivate and activate a rule in a rulebase.



**Requirements**

Before you begin:

- Configure network interfaces. See [Junos OS Interfaces Configuration Guide for Security Devices](#).
- Define rules in a rulebase. See “Example: Defining Rules for an IDP IPS Rulebase” on page 717.

**Overview**

In a rulebase, you can disable and enable rules by using the **deactivate** and **activate** commands. The **deactivate** command comments out the specified statement from the configuration. Rules that have been deactivated do not take effect when you issue the **commit** command. The **activate** command adds the specified statement back to the configuration. Rules that have been activated take effect when you next issue the **commit** command. This example shows how to deactivate and reactivate rule R2 in an IDP IPS rulebase that is associated with a policy called base-policy.

**Configuration****Step-by-Step Procedure**

To deactivate and activate a rule in a rulebase:

1. Specify the rule that you want to deactivate.
 

```
[edit]
user@host# deactivate security idp idp-policy base-policy rulebase-ips rule R2
```
2. Activate the rule.
 

```
[edit]
user@host# activate security idp idp-policy base-policy rulebase-ips rule R2
```
3. If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

**Verification**

To verify the configuration is working properly, enter the **show security idp status** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding IDP Policy Rules on page 707
- Understanding IDP Policy Rulebases on page 713
- Example: Defining Rules for an IDP Exempt Rulebase on page 722
- Example: Enabling IDP Protection Against Application-Level DDoS Attacks on page 768
- Example: Enabling IDP in a Security Policy on page 702

**Understanding IDP Application-Level DDoS Rulebases**

The application-level DDoS rulebase defines parameters used to protect servers, such as DNS or HTTP, from application-level distributed denial-of-service (DDoS) attacks.

You can set up custom application metrics based on normal server activity requests to determine when clients should be considered an attack client. The application-level DDoS rulebase is then used to define the source match condition for traffic that should be monitored, then takes the defined action: close server, drop connection, drop packet, or no action. It can also perform an IP action: ip-block, ip-close, ip-notify, or timeout. Table 80 on page 716 summarizes the options that you can configure in the application-level DDoS rulebase rules.

**Table 80: Application-Level DDoS Rulebase Components**

Term	Definition
<b>Match condition</b>	Specify the network traffic you want the device to monitor for attacks.
<b>Action</b>	Specify the actions you want Intrusion Detection and Prevention (IDP) to take when the monitored traffic matches the application-ddos objects specified in the application-level DDoS rule.
<b>IP Action</b>	Enables you to implicitly block a source address to protect the network from future intrusions while permitting legitimate traffic. You can configure one of the following IP action options in application-level DDoS: ip-block, ip-close, and ip-notify.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- IDP Policies Overview on page 701
- Understanding IDP Policy Rulebases on page 713
- Understanding IDP Policy Rules on page 707
- IDP Application-Level DDoS Attack Overview on page 763
- IDP Application-Level DDoS Protection Overview on page 763
- Example: Enabling IDP Protection Against Application-Level DDoS Attacks on page 768

## IDP IPS Rulebase

- Understanding IDP IPS Rulebases on page 716
- Example: Defining Rules for an IDP IPS Rulebase on page 717

### Understanding IDP IPS Rulebases

The intrusion prevention system (IPS) rulebase protects your network from attacks by using attack objects to detect known and unknown attacks. It detects attacks based on stateful signature and protocol anomalies. Table 81 on page 717 summarizes the options that you can configure in the IPS-rulebase rules.

Table 81: IPS Rulebase Components

Term	Definition
<b>Match condition</b>	Specify the type of network traffic you want the device to monitor for attacks. For more information about match conditions, see “Understanding IDP Policy Rules” on page 707.
<b>Attack objects/groups</b>	Specify the attacks you want the device to match in the monitored network traffic. Each attack is defined as an attack object, which represents a known pattern of attack. For more information about attack objects, see “Understanding IDP Policy Rules” on page 707.
<b>Terminal flag</b>	Specify a terminal rule. The device stops matching rules for a session when a terminal rule is matched. For more information about terminal rules, see “Understanding IDP Terminal Rules” on page 724.
<b>Action</b>	Specify the action you want the system to take when the monitored traffic matches the attack objects specified in the rules. If an attack triggers multiple rule actions, then the most severe action among those rules is executed. For more information about actions, see “Understanding IDP Policy Rules” on page 707.
<b>IP Action</b>	Enables you to protect the network from future intrusions while permitting legitimate traffic. You can configure one of the following IP action options in the IPS rulebase—notify, drop, or close. For more information about IP actions, see “Understanding IDP Policy Rules” on page 707.
<b>Notification</b>	Defines how information is to be logged when action is performed. You can choose to log an attack, create log records with the attack information, and send information to the log server. For more information, see “Understanding IDP Policy Rules” on page 707.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding IDP Policy Rules on page 707](#)
- [Understanding IDP Policy Rulebases on page 713](#)
- [Understanding IDP Exempt Rulebases on page 721](#)
- [Understanding IDP Terminal Rules on page 724](#)
- [Understanding Predefined IDP Policy Templates on page 778](#)
- [Example: Defining Rules for an IDP IPS Rulebase on page 717](#)

#### [Example: Defining Rules for an IDP IPS Rulebase](#)

This example shows how to define rules for an IDP IPS rulebase.

- [Requirements on page 718](#)
- [Overview on page 718](#)

- Configuration on page 718
- Verification on page 720

### Requirements

Before you begin:

- Configure network interfaces. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).
- Create security zones. See “Example: Creating Security Zones” on page 114.
- Enable IDP in security policies. See “Example: Enabling IDP in a Security Policy” on page 702.

### Overview

Each rule is composed of match conditions, objects, actions, and notifications. When you define an IDP rule, you must specify the type of network traffic you want IDP to monitor for attacks by using the following characteristics—source zone, destination zone, source IP address, destination IP address, and the Application Layer protocol supported by the destination IP address. The rules are defined in rulebases, and rulebases are associated with policies.

This example describes how to create a policy called `base-policy`, specify a rulebase for this policy, and then add rule `R1` to this rulebase. In this example, rule `R1`:

- Specifies the match condition to include any traffic from a previously configured zone called `trust` to another previously configured zone called `untrust`. The match condition also includes a predefined attack group `Critical - TELNET`. The application setting in the match condition is `default` and matches any application configured in the attack object.
- Specifies an action to drop connection for any traffic that matches the criteria for rule `R1`.
- Enables attack logging and specifies that an alert flag is added to the attack log.
- Specifies a severity level as `critical`.

After defining the rule, you specify `base-policy` as the active policy on the device.

### Configuration

#### CLI Quick Configuration

To quickly define rules for an IDP IPS rulebase, copy the following commands and paste them into the CLI.

```
[edit]
set security idp idp-policy base-policy
set security idp idp-policy base-policy rulebase-ips rule R1 match from-zone trust to-zone
  untrust source-address any destination-address any application default
set security idp idp-policy base-policy rulebase-ips rule R1 match attacks
  predefined-attack-groups "TELNET-Critical"
set security idp idp-policy base-policy rulebase-ips rule R1 then action drop-connection
set security idp idp-policy base-policy rulebase-ips rule R1 then notification log-attacks
  alert
set security idp idp-policy base-policy rulebase-ips rule R1 then severity critical
```

```
set security idp active-policy base-policy
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To define rules for an IDP IPS rulebase:

1. Create a policy by assigning a meaningful name to it.

```
[edit]
user@host# edit security idp idp-policy base-policy
```

2. Associate a rulebase with the policy.

```
[edit security idp idp-policy base-policy]
user@host# edit rulebase-ips
```

3. Add rules to the rulebase.

```
[edit security idp idp-policy base-policy rulebase-ips]
user@host# edit rule R1
```

4. Define the match criteria for the rule.

```
[edit security idp idp-policy base-policy rulebase-ips rule R1]
user@host# set match from-zone trust to-zone untrust source-address any
destination-address any application default
```

5. Define an attack as match criteria.

```
[edit security idp idp-policy base-policy rulebase-ips rule R1]
user@host# set match attacks predefined-attack-groups "TELNET-Critical"
```

6. Specify an action for the rule.

```
[edit security idp idp-policy base-policy rulebase-ips rule R1]
user@host# set then action drop-connection
```

7. Specify notification and logging options for the rule.

```
[edit security idp idp-policy base-policy rulebase-ips rule R1]
user@host# set then notification log-attacks alert
```

8. Set the severity level for the rule.

```
[edit security idp idp-policy base-policy rulebase-ips rule R1]
user@host# set then severity critical
```

9. Activate the policy.

```
[edit]
user@host# set security idp active-policy base-policy
```

**Results** From configuration mode, confirm your configuration by entering the **show security idp** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security idp
idp-policy base-policy {
```

```

rulebase-ips {
  rule R1 {
    match {
      from-zone trust;
      source-address any;
      to-zone untrust;
      destination-address any;
      application default;
      attacks {
        predefined-attack-groups Critical-TELNET;
      }
    }
    then {
      action {
        drop-connection;
      }
      notification {
        log-attacks {
          alert;
        }
      }
      severity critical;
    }
  }
}
}
active-policy base-policy;

```

If you are done configuring the device, enter **commit** from configuration mode.

### **Verification**

To confirm that the configuration is working properly, perform this task:

- [Verifying the Configuration on page 720](#)

### **Verifying the Configuration**

**Purpose** Verify that the rules for the IDP IPS rulebase configuration are correct.

**Action** From operational mode, enter the **show security idp status** command.

- Related Documentation**
- [Junos OS CLI Reference](#)
  - [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding IDP IPS Rulebases on page 716](#)
  - [Example: Enabling IDP in a Security Policy on page 702](#)
  - [Example: Inserting a Rule in the IDP Rulebase on page 714](#)
  - [Example: Deactivating and Activating Rules in an IDP Rulebase on page 714](#)

## IDP Exempt Rulebase

- Understanding IDP Exempt Rulebases on page 721
- Example: Defining Rules for an IDP Exempt Rulebase on page 722

### Understanding IDP Exempt Rulebases

The exempt rulebase works in conjunction with the intrusion prevention system (IPS) rulebase to prevent unnecessary alarms from being generated. You configure rules in this rulebase to exclude known false positives or to exclude a specific source, destination, or source/destination pair from matching an IPS rule. If traffic matches a rule in the IPS rulebase, the system attempts to match the traffic against the exempt rulebase before performing the action specified. Carefully written rules in an exempt rulebase can significantly reduce the number of false positives generated by an IPS rulebase.

Configure an exempt rulebase in the following conditions:

- When an IDP rule uses an attack object group that contains one or more attack objects that produce false positives or irrelevant log records.
- When you want to exclude a specific source, destination, or source/destination pair from matching an IDP rule. This prevents IDP from generating unnecessary alarms.



**NOTE:** Make sure to configure the IPS rulebase before configuring the exempt rulebase.

Table 82 on page 721 summarizes the options that you can configure in the exempt-rulebase rules.

**Table 82: Exempt Rulebase Options**

Term	Definition
<b>Match condition</b>	Specify the type of network traffic you want the device to monitor for attacks in the same way as in the IPS rulebase. However, in the exempt rulebase, you cannot configure an application; it is always set to <b>any</b> .
<b>Attack objects/groups</b>	Specify the attack objects that you do <i>not</i> want the device to match in the monitored network traffic.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- IDP Policies Overview on page 701
- Understanding IDP Policy Rules on page 707
- Understanding IDP Policy Rulebases on page 713
- Understanding IDP IPS Rulebases on page 716
- Understanding Predefined IDP Policy Templates on page 778

- Example: Defining Rules for an IDP Exempt Rulebase on page 722

### Example: Defining Rules for an IDP Exempt Rulebase

This example shows how to define rules for an exempt IDP rulebase.

- Requirements on page 722
- Overview on page 722
- Configuration on page 722
- Verification on page 723

#### Requirements

Before you begin, create rules in the IDP IPS rulebase. See “Example: Defining Rules for an IDP IPS Rulebase” on page 717.

#### Overview

When you create an exempt rule, you must specify the following:

- Source and destination for traffic you want to exempt. You can set the source or destination to **Any** to exempt network traffic originating from any source or sent to any destination. You can also set **source-except** or **destination-except** to specify all the sources or destinations except the specified source or destination addresses.
- The attacks you want IDP to exempt for the specified source/destination addresses. You must include at least one attack object in an exempt rule.

This example shows that the IDP policy generates false positives for the attack FTP:USER:ROOT on an internal network. You configure the rule to exempt attack detection for this attack when the source IP is from your internal network.

#### Configuration

##### CLI Quick Configuration

To quickly define rules for an exempt IDP rulebase, copy the following commands and paste them into the CLI.

```
[edit]
set security idp idp-policy base-policy
set security idp idp-policy base-policy rulebase-exempt rule R1 match from-zone trust
to-zone any
set security idp idp-policy base-policy rulebase-exempt rule R1 match source-address
internal-devices destination-address any
set security idp idp-policy base-policy rulebase-exempt rule R1 match attacks
predefined-attacks "FTP:USER:ROOT"
set security idp active-policy base-policy
```

##### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To define rules for an exempt IDP rulebase:

1. Specify the IDP IPS rulebase for which you want to define and exempt the rulebase.



```
[edit]
user@host# edit security idp idp-policy base-policy
```

- Associate the exempt rulebase with the policy and zones, and add a rule to the rulebase.

```
[edit security idp idp-policy base-policy]
user@host# set rulebase-exempt rule R1 match from-zone trust to-zone any
```

- Specify the source and destination addresses for the rulebase.

```
[edit security idp idp-policy base-policy]
user@host# set rulebase-exempt rule R1 match source-address internal-devices
destination-address any
```

- Specify the attacks that you want to exempt from attack detection.

```
[edit security idp idp-policy base-policy]
user@host# set rulebase-exempt rule R1 match attacks predefined-attacks
"FTP:USER:ROOT"
```

- Activate the policy.

```
[edit]
user@host# set security idp active-policy base-policy
```

**Results** From configuration mode, confirm your configuration by entering the **show security idp** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security idp
idp-policy base-policy {
  rulebase-exempt {
    rule R1 {
      match {
        from-zone trust;
        source-address internal-devices;
        to-zone any;
        destination-address any;
        attacks {
          predefined-attacks FTP:USER:ROOT;
        }
      }
    }
  }
}
active-policy base-policy;
```

If you are done configuring the device, enter **commit** from configuration mode.

### **Verification**

To confirm that the configuration is working properly, perform this task:

- Verifying the Configuration on page 723

### **Verifying the Configuration**

**Purpose** Verify that the defined rules were exempt from the IDP rulebase configuration.

**Action** From operational mode, enter the **show security idp status** command.

- Related Documentation**
- [Junos OS CLI Reference](#)
  - [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding IDP Exempt Rulebases on page 721
  - Example: Inserting a Rule in the IDP Rulebase on page 714
  - Example: Deactivating and Activating Rules in an IDP Rulebase on page 714
  - Example: Enabling IDP in a Security Policy on page 702

## IDP Terminal Rules

- Understanding IDP Terminal Rules on page 724
- Example: Setting Terminal Rules in Rulebases on page 725

### Understanding IDP Terminal Rules

---

The Intrusion Detection and Prevention (IDP) rule-matching algorithm starts from the top of the rulebase and checks traffic against all rules in the rulebase that match the source, destination, and service. However, you can configure a rule to be *terminal*. A *terminal* rule is an exception to this algorithm. When a match is discovered in a terminal rule for the source, destination, zones, and application, IDP does not continue to check subsequent rules for the same source, destination, and application. It does not matter whether or not the traffic matches the attack objects in the matching rule.

You can use a terminal rule for the following purposes:

- To set different actions for different attacks for the same Source and Destination.
- To disregard traffic that originates from a known trusted source. Typically, the action is **None** for this type of terminal rule.
- To disregard traffic sent to a server that is vulnerable only to a specific set of attacks. Typically, the action is **Drop Connection** for this type of terminal rule.

Use caution when defining terminal rules. An inappropriate terminal rule can leave your network open to attacks. Remember that traffic matching the source, destination, and application of a terminal rule is not compared to subsequent rules, even if the traffic does not match an attack object in the terminal rule. Use a terminal rule only when you want to examine a certain type of traffic for one specific set of attack objects. Be particularly careful about terminal rules that use **any** for both the source and destination. Terminal rules should appear near the top of the rulebase before other rules that would match the same traffic.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - IDP Policies Overview on page 701
  - Understanding IDP Policy Rules on page 707
  - Understanding IDP Policy Rulebases on page 713

- Understanding IDP IPS Rulebases on page 716
- Understanding IDP Exempt Rulebases on page 721
- Example: Setting Terminal Rules in Rulebases on page 725

### Example: Setting Terminal Rules in Rulebases

This example shows how to configure terminal rules.

- Requirements on page 725
- Overview on page 725
- Configuration on page 725
- Verification on page 726

#### Requirements

Before you begin:

- Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Enable IDP application services in a security policy. See “Example: Enabling IDP in a Security Policy” on page 702.
- Create security zones. See “Example: Creating Security Zones” on page 114.
- Define rules. See “Example: Inserting a Rule in the IDP Rulebase” on page 714.

#### Overview

By default, rules in the IDP rulebase are not terminal, which means IDP examines all rules in the rulebase and executes all matches. You can specify that a rule is terminal; that is, if IDP encounters a match for the source, destination, and service specified in a terminal rule, it does not examine any subsequent rules for that connection.

This example shows how to configure terminal rules. You define rule R2 to terminate the match algorithm if the source IP of the traffic originates from a known trusted network in your company. If this rule is matched, IDP disregards traffic from the trusted network and does not monitor the session for malicious data.

#### Configuration

#### CLI Quick Configuration

To quickly configure terminal rules, copy the following commands and paste them into the CLI:

```
[edit]
set security idp idp-policy base-policy rulebase-ips rule R2
set security idp idp-policy base-policy rulebase-ips rule R2 match source-address internal
destination-address any
set security idp idp-policy base-policy rulebase-ips rule R2 terminal
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To configure terminal rules:

1. Create an IDP policy.

```
[edit]
user@host# edit set security idp idp-policy base-policy
```

2. Define a rule and set its match criteria.

```
[edit security idp idp-policy base-policy]
user@host# set rulebase-ips rule R2 match source-address internal
destination-address any
```

3. Set the terminal flag for the rule.

```
[edit security idp idp-policy base-policy]
user@host# set rulebase-ips rule R2 terminal
```

**Results** From configuration mode, confirm your configuration by entering the **show security idp** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security idp
idp-policy base-policy {
  rulebase-ips {
    rule R2 {
      match {
        source-address internal;
        destination-address any;
      }
      terminal;
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying the Configuration on page 726

### Verifying the Configuration

**Purpose** Verify that the terminal rules were configured correctly.

**Action** From operational mode, enter the **show security idp status** command.

**Related Documentation**

- [Junos OS CLI Reference](#)
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

- Understanding IDP Terminal Rules on page 724
- Example: Defining Rules for an IDP IPS Rulebase on page 717
- Example: Enabling IDP in a Security Policy on page 702

## IDP DSCP Rules

- Understanding DSCP Rules in IDP Policies on page 727
- Example: Configuring DSCP Rules in an IDP Policy on page 727

### Understanding DSCP Rules in IDP Policies

Differentiated Services code point (DSCP) is an integer value encoded in the 6-bit field defined in IP packet headers. It is used to enforce class-of-service (CoS) distinctions. CoS allows you to override the default packet forwarding behavior and assign service levels to specific traffic flows.

You can configure DSCP value as an action in an IDP policy rule. You first define the traffic by defining match conditions in the IDP policy and then associate a DiffServ marking action with it. Based on the DSCP value, behavior aggregate classifiers set the forwarding class and loss priority for the traffic deciding the forwarding treatment the traffic receives.

All packets that match the IDP policy rule have the CoS field in their IP header rewritten with the DSCP value specified in the matching policy. If the traffic matches multiple rules with differing DSCP values, the first IDP rule that matches takes effect and this IDP rule then applies to all traffic for that session.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- IDP Policies Overview on page 701
- Understanding IDP Policy Rules on page 707
- Understanding IDP Policy Rulebases on page 713
- Understanding IDP IPS Rulebases on page 716
- Understanding IDP Exempt Rulebases on page 721
- Example: Configuring DSCP Rules in an IDP Policy on page 727

### Example: Configuring DSCP Rules in an IDP Policy

This example shows how to configure DSCP values in an IDP policy.

- Requirements on page 727
- Overview on page 728
- Configuration on page 728
- Verification on page 730

#### Requirements

Before you begin:

- Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Enable IDP application services in a security policy. See “Example: Enabling IDP in a Security Policy” on page 702.
- Create security zones. See “Example: Creating Security Zones” on page 114.
- Define rules. See “Example: Inserting a Rule in the IDP Rulebase” on page 714.

### Overview

Configuring DSCP values in IDP policies provides a method of associating CoS values—thus different levels of reliability—for different types of traffic on the network.

This example shows how to create a policy called `policy1`, specify a rulebase for this policy, and then add rule `R1` to this rulebase. In this example, rule `R1`:

- Specifies the match condition to include any traffic from a previously configured zone called `trust` to another previously configured zone called `untrust`. The match condition also includes a predefined attack group called `HTTP - Critical`. The application setting in the match condition is specified as the default and matches any application configured in the attack object.
- Specifies an action to rewrite the CoS field in the IP header with the DSCP value 50 for any traffic that matches the criteria for rule `R1`.

### Configuration

**CLI Quick Configuration** To quickly configure DSCP values in an IDP policy, copy the following commands and paste them into the CLI:

```
[edit]
set security idp idp-policy base-policy
set security idp idp-policy base-policy rulebase-ips rule R1 match from-zone Zone-1 to-zone
  Zone-2 source-address any destination-address any application default
set security idp idp-policy base-policy rulebase-ips rule R1 match attacks
  predefined-attack-groups "HTTP - Critical"
set security idp idp-policy base-policy rulebase-ips rule R1 then action mark-diffserv 50
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To configure DSCP values in an IDP policy:

1. Create a policy by assigning a meaningful name to it.

```
[edit]
user@host# edit security idp idp-policy base-policy
```

2. Associate a rulebase with the policy.

```
[edit security idp idp-policy base-policy]
user@host# edit rulebase-ips
```

3. Add rules to the rulebase

```
[edit security idp idp-policy base-policy rulebase-ips]
user@host# edit rule R1
```

4. Define the match criteria for the rule.

```
[edit security idp idp-policy base-policy rulebase-ips R1]
user@host# set match from-zone trust to-zone untrust source-address any
destination-address any application default
```

```
user@host# set match attacks predefined-attack-group "HTTP - Critical"
```

5. Specify an action for the rule.

```
[edit security idp idp-policy base-policy rulebase-ips R1]
user@host# set then action mark-diffserv 50
```

6. Continue to specify any notification or logging options for the rule, if required.
7. Activate the policy.

```
[edit]
user@host# set security idp active-policy base-policy
```

**Results** From configuration mode, confirm your configuration by entering the **show security idp** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security idp
idp-policy base-policy{
  rulebase-ips {
    rule R1 {
      match {
        from-zone trust;
        source-address any;
        to-zone untrust;
        destination-address any;
        application default;
        attacks {
          predefined-attack-groups HTTP-Critical;
        }
      }
      then {
        action {
          mark-diffserv {
            50;
          }
        }
      }
    }
  }
}
active-policy base-policy;
```

If you are done configuring the device, enter **commit** from configuration mode.

**Verification**

To confirm that the configuration is working properly, perform this task:

- [Verifying the Configuration on page 730](#)

**Verifying the Configuration**

**Purpose** Verify that the DSCP values were configured in an IDP policy.

**Action** From operational mode, enter the **show security idp status** command.

- Related Documentation**
- [Junos OS CLI Reference](#)
  - [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding DSCP Rules in IDP Policies on page 727](#)
  - [Example: Enabling IDP in a Security Policy on page 702](#)
  - [Example: Defining Rules for an IDP IPS Rulebase on page 717](#)

## IDP Applications and Application Sets

---

- [Understanding IDP Application Sets on page 730](#)
- [Example: Configuring IDP Applications and Services on page 731](#)
- [Example: Configuring IDP Applications Sets on page 733](#)

### Understanding IDP Application Sets

Applications or services represent Application Layer protocols that define how data is structured as it travels across the network. Because the services you support on your network are the same services that attackers must use to attack your network, you can specify which services are supported by the destination IP to make your rules more efficient. Juniper Networks provides predefined applications and application sets that are based on industry-standard applications. If you need to add applications that are not included in the predefined applications, you can create custom applications or modify predefined applications to suit your needs.

You specify an application, or service, to indicate that a policy applies to traffic of that type. Sometimes the same applications or a subset of them can be present in multiple policies, making them difficult to manage. Junos OS allows you to create groups of applications called *application sets*.

Application sets simplify the process by allowing you to manage a small number of application sets, rather than a large number of individual application entries.

The application (or application set) is configured as a match criterion for packets. Packets must be of the application type specified in the policy for the policy to apply to the packet. If the packet matches the application type specified by the policy and all other criteria match, then the policy action is applied to the packet. You can use predefined or custom applications and refer to them in a policy.



- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - IDP Policies Overview on page 701
  - Understanding IDP Policy Rules on page 707
  - Understanding IDP Policy Rulebases on page 713
  - Example: Configuring IDP Applications and Services on page 731

## Example: Configuring IDP Applications and Services

This example shows how to create an application and associate it with an IDP policy.

- Requirements on page 731
- Overview on page 731
- Configuration on page 731
- Verification on page 733

### Requirements

Before you begin:

- Configure network interfaces. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).
- Enable IDP application services in a security policy. See “Example: Enabling IDP in a Security Policy” on page 702.

### Overview

To create custom applications, specify a meaningful name for an application and associate parameters with it—for example, inactivity timeout, or application protocol type. In this example, you create a special FTP application called `cust-app`, specify it as a match condition in the IDP policy ABC running on port 78, and specify the inactivity timeout value as 6000 seconds.

### Configuration

- CLI Quick Configuration** To quickly create an application and associate it with an IDP policy, copy the following commands and paste them into the CLI:

```
[edit]
set applications application cust-app application-protocol ftp protocol tcp
  destination-port 78 inactivity-timeout 6000
set security idp idp-policy ABC rulebase-ips rule ABC match application cust-app
set security idp idp-policy ABC rulebase-ips rule ABC then action no-action
set security idp active-policy ABC
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To create an application and associate it with an IDP policy:

1. Create an application and specify its properties.

```
[edit applications application cust-app]
user@host# set application-protocol ftp protocol tcp destination-port 78
inactivity-timeout 6000
```

2. Specify the application as a match condition in a policy.

```
[edit security idp idp-policy ABC rulebase-ips rule ABC]
user@host# set match application cust-app
```

3. Specify the no action condition.

```
[edit security idp idp-policy ABC rulebase-ips rule ABC]
user@host# set then action no-action
```

4. Activate the policy.

```
[edit]
user@host# set security idp active-policy ABC
```

**Results** From configuration mode, confirm your configuration by entering the **show security idp** and **show applications** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security idp
idp-policy ABC {
  rulebase-ips {
    rule R1 {
      match {
        application cust-app;
      }
    }
  }
}
active-policy ABC;
```

```
[edit]
user@host# show applications
application cust-app {
  application-protocol ftp;
  protocol tcp;
  destination-port 78;
  inactivity-timeout 6000;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

---

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying the Configuration on page 733

### **Verifying the Configuration**

**Purpose** Verify that the application was associated with the IDP policy.

**Action** From operational mode, enter the **show security idp status** command.

- Related Documentation**
- *Junos OS CLI Reference*
  - *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding IDP Application Sets on page 730
  - Example: Configuring IDP Applications Sets on page 733
  - Example: Enabling IDP in a Security Policy on page 702

## Example: Configuring IDP Applications Sets

This example shows how to create an application set and associate it with an IDP policy.

- Requirements on page 733
- Overview on page 733
- Configuration on page 734
- Verification on page 735

### Requirements

---

Before you begin:

- Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Enable IDP application services in a security policy. See “Example: Enabling IDP in a Security Policy” on page 702.
- Define applications. See “Example: Configuring Applications and Application Sets” on page 189.

### Overview

---

To configure an application set, you add predefined or custom applications separately to an application set and assign a meaningful name to the application set. Once you name the application set you specify the name as part of the policy. For this policy to apply on a packet, the packet must match any one of the applications included in this set.

This example describes how to create an application set called SrvAccessAppSet and associate it with IDP policy ABC. The application set SrvAccessAppSet combines three

applications. Instead of specifying three applications in the policy rule, you specify one application set. If all of the other criteria match, any one of the applications in the application set serves as valid matching criteria.

### Configuration

**CLI Quick Configuration** To quickly create an application set and associate it with an IDP policy, copy the following commands and paste them into the CLI:

```
[edit]
set applications application-set SrvAccessAppSet application junos-ssh
set applications application-set SrvAccessAppSet application junos-telnet
set applications application-set SrvAccessAppSet application cust-app
set security idp idp-policy ABC rulebase-ips rule ABC match application SrvAccessAppSet
set security idp idp-policy ABC rulebase-ips rule ABC then action no-action
set security idp active-policy ABC
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To create an application set and associate it with an IDP policy:

1. Create an application set and include three applications in the set.

```
[edit applications application-set SrvAccessAppSet]
user@host# set application junos-ssh
user@host# set application junos-telnet
user@host# set application cust-app
```

2. Create an IDP policy.

```
[edit]
user@host# edit security idp idp-policy ABC
```

3. Associate the application set with an IDP policy.

```
[edit security idp idp-policy ABC]
user@host# set rulebase-ips rule ABC match application SrvAccessAppSet
```

4. Specify an action for the policy.

```
[edit security idp idp-policy ABC]
user@host# set rulebase-ips rule ABC then action no-action
```

5. Activate the policy.

```
[edit]
user@host# set security idp active-policy ABC
```

**Results** From configuration mode, confirm your configuration by entering the **show security idp** and **show applications** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security idp
idp-policy ABC {
  rulebase-ips {
```

```

rule R1 {
  match {
    application SrvAccessAppSet;
  }
  then {
    action {
      no-action;
    }
  }
}
}
}
active-policy ABC;

[edit]
user@host# show applications
application-set SrvAccessAppSet {
  application ssh;
  application telnet;
  application custApp;
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying the Configuration on page 735

#### **Verifying the Configuration**

**Purpose** Verify that the application set was associated with the IDP policy.

**Action** From operational mode, enter the **show security idp status** command.

**Related Documentation**

- *Junos OS CLI Reference*
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding IDP Application Sets on page 730
- Example: Configuring IDP Applications and Services on page 731
- Example: Enabling IDP in a Security Policy on page 702

## IDP Attacks and Attack Objects

- Understanding Custom Attack Objects on page 736
- IDP Protocol Decoders on page 752
- IDP Signature-Based Attacks on page 754
- IDP Protocol Anomaly-Based Attacks on page 758
- Listing IDP Test Conditions for a Specific Protocol on page 761

## Understanding Custom Attack Objects

You can create custom attack objects to detect new attacks or customize predefined attack objects to meet the unique needs of your network.

To configure a custom attack object, you specify a unique name for it and then specify additional information, such as a general description and keywords, which can make it easier for you to locate and maintain the attack object.

Certain properties in the attack object definitions are common to all types of attacks, such as attack name, description, severity level, service or application binding, time binding, recommended action, and protocol or port binding. Some fields are specific to an attack type and are available only for that specific attack definition.

This topic includes the following sections:

- Attack Name on page 736
- Severity on page 736
- Service and Application Bindings on page 736
- Protocol and Port Bindings on page 740
- Time Bindings on page 742
- Attack Properties (Signature Attacks) on page 743
- Attack Properties (Protocol Anomaly Attacks) on page 748
- Attack Properties (Compound or Chain Attacks) on page 749

---

### Attack Name

Specify an alphanumeric name for the object. You might want to include the protocol the attack uses in the attack name.

---

### Severity

Specifies the brutality of the attack on your network. Severity categories, in order of increasing brutality, are info, warning, minor, major, critical (see “Understanding IDP Rule Notifications” on page 712). Critical attacks are the most dangerous—typically these attacks attempt to crash your server or gain control of your network. Informational attacks are the least dangerous, and typically are used by network administrators to discover holes in their own security systems.

---

### Service and Application Bindings

The service or application binding field specifies the service that the attack uses to enter your network.



**NOTE:** Specify either the service or the protocol binding in a custom attack. In case you specify both, the service binding takes precedence.

---

- Any—Specify **any** if you are unsure of the correct service and want to match the signature in all services. Because some attacks use multiple services to attack your network, you

might want to select the **Any** service binding to detect the attack regardless of which service the attack chooses for a connection.

- **Service**—Most attacks use a specific service to attack your network. You can select the specific service used to perpetrate the attack as the service binding. Table 83 on page 737 displays supported services and default ports associated with the services.

**Table 83: Supported Services for Service Bindings**

Service	Description	Default Port
<b>AIM</b>	AOL Instant Messenger. America Online Internet service provider (ISP) provides Internet, chat, and instant messaging applications.	TCP/5190
<b>BGP</b>	Border Gateway Protocol	TCP/179
<b>Chargen</b>	Character Generator Protocol is a UDP- or TCP-based debugging and measurement tool.	TCP/19, UDP/19
<b>DHCP</b>	Dynamic Host Configuration Protocol allocates network addresses and delivers configuration parameters from server to hosts.	UDP/67, UDP/68
<b>Discard</b>	Discard protocol is an Application Layer protocol that describes a process for discarding TCP or UDP data sent to port 9.	TCP/9, UDP/9
<b>DNS</b>	Domain Name System translates domain names into IP addresses.	TCP/53, UDP/53
<b>Echo</b>	Echo	TCP/7, UDP/7
<b>Finger</b>	Finger is a UNIX program that provides information about users.	TCP/79, UDP/79
<b>FTP</b>	File Transfer Protocol (FTP) allows the sending and receiving of files between machines.	TCP/21, UDP/21
<b>Gnutella</b>	Gnutella is a public domain file sharing protocol that operates over a distributed network.	TCP/6346
<b>Gopher</b>	Gopher organizes and displays Internet servers' contents as a hierarchically structured list of files.	TCP/70
<b>H225RAS</b>	H.225.0/RAS (Registration, Admission, and Status)	UDP/1718, UDP/1719

Table 83: Supported Services for Service Bindings (*continued*)

Service	Description	Default Port
<b>HTTP</b>	HyperText Transfer Protocol is the underlying protocol used by the World Wide Web (WWW).	TCP/80, TCP/81, TCP/88, TCP/3128, TCP/7001 (Weblogic), TCP/8000, TCP/8001, TCP/8100 (JRun), TCP/8200 (JRun), TCP/8080, TCP/8888 (Oracle-9i), TCP/9080 (Websphere), UDP/80
<b>ICMP</b>	Internet Control Message Protocol	
<b>IDENT</b>	Identification protocol is a TCP/IP Application Layer protocol used for TCP client authentication.	TCP/113
<b>IKE</b>	Internet Key Exchange protocol (IKE) is a protocol to obtain authenticated keying material for use with ISAKMP.	UDP/500
<b>IMAP</b>	Internet Message Access Protocol is used for retrieving messages.	TCP/143, UDP/143
<b>IRC</b>	Internet Relay Chat (IRC) allows people connected to the Internet to join live discussions.	TCP/6667
<b>LDAP</b>	Lightweight Directory Access Protocol is a set of protocols used to access information directories.	TCP/389
<b>lpr</b>	Line Printer Daemon protocol is a TCP-based protocol used for printing applications.	TCP/515
<b>MSN</b>	Microsoft Network Messenger is a utility that allows you to send instant messages and talk online.	TCP/1863
<b>MSRPC</b>	Microsoft Remote Procedure Call	TCP/135, UDP/135
<b>MSSQL</b>	Microsoft SQL is a proprietary database server tool that allows for the creation, access, modification, and protection of data.	TCP/1433, TCP/3306
<b>MYSQL</b>	MySQL is a database management system available for both Linux and Windows.	TCP/3306
<b>NBDS</b>	NetBIOS Datagram Service application, published by IBM, provides connectionless (datagram) applications to PCs connected with a broadcast medium to locate resources, initiate sessions, and terminate sessions. It is unreliable and the packets are not sequenced.	UDP/137 (NBName), UDP/138 (NBDS)



Table 83: Supported Services for Service Bindings (*continued*)

Service	Description	Default Port
<b>NFS</b>	Network File System uses UDP to allow network users to access shared files stored on computers of different types. SUN RPC is a building block of NFS.	TCP/2049, UDP/2049
<b>nntp</b>	Network News Transport Protocol is a protocol used to post, distribute, and retrieve USENET messages.	TCP/119
<b>NTP</b>	Network Time Protocol provides a way for computers to synchronize to a time reference.	UDP/123
<b>POP3</b>	Post Office Protocol is used for retrieving e-mail.	UDP/110, TCP/110
<b>Portmapper</b>	Service that runs on nodes on the Internet to map an ONC RPC program number to the network address of the server that listens for the program number.	TCP/111, UDP/111
<b>RADIUS</b>	Remote Authentication Dial-In User Service application is a server program used for authentication and accounting purposes.	UDP/1812, UDP/1813
<b>rexec</b>	Rexec	TCP/512
<b>rlogin</b>	RLOGIN starts a terminal session on a remote host.	TCP/513
<b>rsh</b>	RSH executes a shell command on a remote host.	TCP/514
<b>rtsp</b>	Real-Time Streaming Protocol (RTSP) is for streaming media applications	TCP/554
<b>SIP</b>	Session Initiation Protocol (SIP) is an Application-Layer control protocol for creating, modifying, and terminating sessions.	TCP/5060, UDP/5060
<b>SMB</b>	Server Message Block (SMB) over IP is a protocol that allows you to read and write files to a server on a network.	TCP/139, TCP/445
<b>SMTP</b>	Simple Mail Transfer Protocol is used to send messages between servers.	TCP/25, UDP/25
<b>SNMP</b>	Simple Network Management Protocol is a set of protocols for managing complex networks.	TCP/161, UDP/161
<b>SNMPTRAP</b>	SNMP trap	TCP/162, UDP/162
<b>SQLMON</b>	SQL monitor (Microsoft)	UDP/1434
<b>SSH</b>	SSH is a program to log into another computer over a network through strong authentication and secure communications on a channel that is not secure.	TCP/22, UDP/22

Table 83: Supported Services for Service Bindings (*continued*)

Service	Description	Default Port
SSL	Secure Sockets Layer	TCP/443, TCP/80
syslog	Syslog is a UNIX program that sends messages to the system logger.	UDP/514
Telnet	Telnet is a UNIX program that provides a standard method of interfacing terminal routers and terminal-oriented processes to each other.	TCP/23, UDP/23
TNS	Transparent Network Substrate	TCP/1521, TCP/1522, TCP/1523, TCP/1524, TCP/1525, TCP/1526, TCP/1527, TCP/1528, TCP/1529, TCP/1530, TCP/2481, TCP/1810, TCP/7778
TFTP	Trivial File Transfer Protocol	UDP/69
VNC	Virtual Network Computing facilitates viewing and interacting with another computer or mobile router connected to the Internet.	TCP/5800, TCP/5900
Whois	Network Directory Application Protocol is a way to look up domain names.	TCP/43
YMSG	Yahoo! Messenger is a utility that allows you to check when others are online, send instant messages, and talk online.	TCP/5050

### Protocol and Port Bindings

Protocol or port bindings allow you to specify the protocol that an attack uses to enter your network. You can specify the name of the network protocol, or the protocol number.



**NOTE:** Specify either the service or the protocol binding in a custom attack. In case you specify both, the service binding takes precedence.

- IP—You can specify any of the supported network layer protocols using protocol numbers. Table 84 on page 740 lists protocol numbers for different protocols.

Table 84: Supported Protocols and Protocol Numbers

Protocol Name	Protocol Number
IGMP	2
IPIP	4

Table 84: Supported Protocols and Protocol Numbers (*continued*)

Protocol Name	Protocol Number
EGP	8
PUP	12
TP	29
IPV6	41
ROUTING	43
FRAGMENT	44
RSVP	46
GRE	47
ESP	50
AH	51
ICMPV6	58
NONE	59
DSTOPTS	60
MTP	92
ENCAP	98
PIM	103
COMP	108
RAW	255

- ICMP, TCP, and UDP—Attacks that do not use a specific service might use specific ports to attack your network. Some TCP and UDP attacks use standard ports to enter your network and establish a connection.
- RPC—The remote procedure call (RPC) protocol is used by distributed processing applications to handle interaction between processes remotely. When a client makes a remote procedure call to an RPC server, the server replies with a remote program; each remote program uses a different program number. To detect attacks that use RPC, configure the service binding as RPC and specify the RPC program ID.

Table 85 on page 742 displays sample formats for key protocols.

Table 85: Sample Formats for Protocols

Protocol Name	Protocol Number	Description
ICMP	<Port>ICMP</Port>	Specify the protocol name.
IP	<Port>IP/protocol-number</Port>	Specify the Network Layer protocol number.
RPC	<Port>RPC/program-number</Port>	Specify the RPC program number.
TCP or UDP	<ul style="list-style-type: none"> <li>• &lt;Port&gt;TCP &lt;/Port&gt;</li> <li>• &lt;Port&gt;TCP/port &lt;/Port&gt;</li> <li>• &lt;Port&gt;TCP/minport-maxport &lt;/Port&gt;</li> </ul>	Specifying the port is optional for TCP and UDP protocols. For example, you can specify either of the following: <ul style="list-style-type: none"> <li>• &lt;Port&gt;UDP&lt;/Port&gt;</li> <li>• &lt;Port&gt;UDP/10&lt;/Port&gt;</li> <li>• &lt;Port&gt;UDP/10-100&lt;/Port&gt;</li> </ul>

### Time Bindings

Use time bindings to configure the time attributes for the custom attack object. Time attributes control how the attack object identifies attacks that repeat for a certain number of times. By configuring the scope and count of an attack, you can detect a sequence of the same attacks over a period of time (one minute) across sessions.

#### Scope

Specify the scope within which the count of an attack occurs:

- **Source**—Specify this option to detect attacks from the source address for the specified number of times, regardless of the destination address. This means that for a given attack, a threshold value is maintained for each attack from the source address. The destination address is ignored. For example, anomalies are detected from two different pairs (**ip-a**, **ip-b**) and (**ip-a**, **ip-c**) that have the same source address **ip-a** but different destination addresses **ip-b** and **ip-c**. Then the number of matches for **ip-a** increments to **2**. Suppose the threshold value or *count* is also set to **2**, then the signature triggers the attack event.
- **Destination**—Specify this option to detect attacks sent to the destination address for the specified number of times, regardless of the source address. This means that for a given attack, a threshold value is maintained for each attack from the destination address. The source address is ignored. For example, if anomalies are detected from two different pairs (**ip-a**, **ip-b**) and (**ip-c**, **ip-b**) that have the same destination address **ip-b** but different source addresses **ip-a** and **ip-c**. Then the number of matches for **ip-b** increments to **2**. Suppose the threshold value or *count* is also set to **2**, then the signature triggers the attack event.
- **Peer**—Specify this option to detect attacks between source and destination IP addresses of the sessions for the specified number of times. This means that the threshold value is applicable for a pair of source and destination addresses. Suppose anomalies are detected from two different source and destination pairs (**ip-a**, **ip-b**) and (**ip-a**, **ip-c**).

Then the number of matches for each pair is set to 1, even though both pairs have a common source address.

### **Count**

Count or threshold value specifies the number of times that the attack object must detect an attack within the specified scope before the device considers the attack object to match the attack. If you bind the attack object to multiple ports and the attack object detects that attack on different ports, each attack on each port is counted as a separate occurrence. For example, when the attack object detects an attack on **TCP/80** and then on **TCP/8080**, the count is two.

Once the **count** match is reached, each attack that matches the criteria causes the attack count to increase by one. This count cycle lasts for a duration of 60 seconds, after which the cycle repeats.

### **Attack Properties (Signature Attacks)**

Signature attack objects use a stateful attack signature (a pattern that always exists within a specific section of the attack) to detect known attacks. They also include the protocol or service used to perpetrate the attack and the context in which the attack occurs. The following properties are specific to signature attacks, and you can configure them when configuring signature attack:



**NOTE:** Attack context, flow type, and direction are mandatory fields for the signature attack definition.

### **Attack Context**

An attack context defines the location of the signature. If you know the service and the specific service context, specify that service and then specify the appropriate service contexts. If you know the service, but are unsure of the specific service context, specify one of the following general contexts:

- **first-data-packet**—Specify this context to detect the attack in only the first data packet.
- **first-packet**—Specify this context to detect the attack in only the first packet of a stream. When the flow direction for the attack object is set to **any**, the device checks the first packet of both the server-to-client and the client-to-server flows. If you know that the attack signature appears in the first packet of a session, choosing **first packet** instead of **packet** reduces the amount of traffic the device needs to monitor, which improves performance.
- **packet**—Specify this context to match the attack pattern within a packet. When you select this option, you must also specify the service binding to define the service header options. Although not required, specifying these additional parameters improves the accuracy of the attack object and thereby improves performance.
- **line**—Specify this context to detect a pattern match within a specific line within your network traffic.
- **normalized-stream**—Specify this context to detect the attack in an entire normalized stream. The normalized stream is one of the multiple ways of sending information. In

this stream the information in the packet is normalized before a match is performed. Suppose `www.yahoo.com/sports` is the same as `www.yahoo.com/s%70orts`. The normalized form to represent both of these URLs might be `www.yahoo.com/sports`. Choose **normalized stream** instead of **stream**, unless you want to detect some pattern in its exact form. For example, if you want to detect the exact pattern `www.yahoo.com/s%70orts`, then select **stream**.

- **normalized-stream256**—Specify this context to detect the attack in only the first 256 bytes of a normalized stream.
- **normalized-stream1k**—Specify this context to detect the attack in only the first 1024 bytes of a normalized stream.
- **normalized-stream-8k**—Specify this context to detect the attack in only the first 8192 bytes of a normalized stream.
- **stream**—Specify this context to reassemble packets and extract the data to search for a pattern match. However, the device cannot recognize packet boundaries for stream contexts, so data for multiple packets is combined. Specify this option only when no other context option contains the attack.
- **stream256**—Specify this context to reassemble packets and search for a pattern match within the first 256 bytes of a traffic stream. When the flow direction is set to **any**, the device checks the first 256 bytes of both the server-to-client and client-to-server flows. If you know that the attack signature will appear in the first 256 bytes of a session, choosing **stream256** instead of **stream** reduces the amount of traffic that the device must monitor and cache, thereby improving performance.
- **stream1k**—Specify this context to reassemble packets and search for a pattern match within the first 1024 bytes of a traffic stream. When the flow direction is set to **any**, the device checks the first 1024 bytes of both the server-to-client and client-to-server flows. If you know that the attack signature will appear in the first 1024 bytes of a session, choosing **stream1024** instead of **stream** reduces the amount of traffic that the device must monitor and cache, thereby improving performance.
- **stream8k**—Specify this context to reassemble packets and search for a pattern match within the first 8192 bytes of a traffic stream. When the flow direction is set to **any**, the device checks the first 8192 bytes of both the server-to-client and client-to-server flows. If you know that the attack signature will appear in the first 8192 bytes of a session, choosing **stream8192** instead of **stream** reduces the amount of traffic that the device must monitor and cache, thereby improving performance.

#### ***Attack Direction***

You can specify the connection direction of the attack. Using a single direction (instead of **Any**) improves performance, reduces false positives, and increases detection accuracy.

- Client to server (detects the attack only in client-to-server traffic)
- Server to client (detects the attack only in server-to-client traffic)
- Any (detects the attack in either direction)

### Attack Pattern

Attack patterns are signatures of the attacks you want to detect. A signature is a pattern that always exists within an attack; if the attack is present, so is the signature. To create the attack pattern, you must first analyze the attack to detect a pattern (such as a segment of code, a URL, or a value in a packet header), then create a syntactical expression that represents that pattern. You can also negate a pattern. Negating a pattern means that the attack is considered matched if the pattern defined in the attack does *not* match the specified pattern.



**NOTE:** Pattern negation is supported for packet, line, and application based contexts only and not for stream and normalized stream contexts.

### Protocol-Specific Parameters

Specifies certain values and options existing within packet headers. These parameters are different for different protocols. In a custom attack definition, you can specify fields for only one of the following protocols—TCP, UDP, or ICMP. Although, you can define IP protocol fields with TCP or UDP in a custom attack definition.



**NOTE:** Header parameters can be defined only for attack objects that use a packet or first packet context. If you specified a line, stream, stream 256, or a service context you cannot specify header parameters.

If you are unsure of the options or flag settings for the malicious packet, leave all fields blank and Intrusion Detection and Prevention (IDP) attempts to match the signature for all header contents.

Table 86 on page 745 displays fields and flags that you can set for attacks that use the IP protocol.

**Table 86: IP Protocol Fields and Flags**

Field	Description
<b>Type of Service</b>	Specify a value for the service type. Common service types are: <ul style="list-style-type: none"> <li>• 0000 Default</li> <li>• 0001 Minimize Cost</li> <li>• 0002 Maximize Reliability</li> <li>• 0003 Maximize Throughput</li> <li>• 0004 Minimize Delay</li> <li>• 0005 Maximize Security</li> </ul>
<b>Total Length</b>	Specify a value for the number of bytes in the packet, including all header fields and the data payload.
<b>ID</b>	Specify a value for the unique value used by the destination system to reassemble a fragmented packet.

Table 86: IP Protocol Fields and Flags (*continued*)

Field	Description
<b>Time to Live</b>	Specify an integer value in the range of 0–255 for the time-to-live (TTL) value of the packet. This value represents the number of devices the packet can traverse. Each router that processes the packet decrements the TTL by 1; when the TTL reaches 0, the packet is discarded.
<b>Protocol</b>	Specify a value for the protocol used.
<b>Source</b>	Enter the source address of the attacking device.
<b>Destination</b>	Enter the destination address of the attack target.
<b>Reserved Bit</b>	This bit is not used.
<b>More Fragments</b>	When set (1), this option indicates that the packet contains more fragments. When unset (0), it indicates that no more fragments remain.
<b>Don't Fragment</b>	When set (1), this option indicates that the packet cannot be fragmented for transmission.

Table 87 on page 746 displays packet header fields and flags that you can set for attacks that use the TCP protocol.

Table 87: TCP Header Fields and Flags

Field	Description
<b>Source Port</b>	Specify a value for the port number on the attacking device.
<b>Destination Port</b>	Specify a value for the port number of the attack target.
<b>Sequence Number</b>	Specify a value for the sequence number of the packet. This number identifies the location of the data in relation to the entire data sequence.
<b>ACK Number</b>	Specify a value for the ACK number of the packet. This number identifies the next sequence number; the ACK flag must be set to activate this field.
<b>Header Length</b>	Specify a value for the number of bytes in the TCP header.
<b>Data Length</b>	Specify a value for the number of bytes in the data payload. For SYN, ACK, and FIN packets, this field should be empty.
<b>Window Size</b>	Specify a value for the number of bytes in the TCP window size.



Table 87: TCP Header Fields and Flags (*continued*)

Field	Description
<b>Urgent Pointer</b>	Specify a value for the urgent pointer. The value indicates that the data in the packet is urgent; the URG flag must be set to activate this field.
<b>URG</b>	When set, the urgent flag indicates that the packet data is urgent.
<b>ACK</b>	When set, the acknowledgment flag acknowledges receipt of a packet.
<b>PSH</b>	When set, the push flag indicates that the receiver should push all data in the current sequence to the destination application (identified by the port number) without waiting for the remaining packets in the sequence.
<b>RST</b>	When set, the reset flag resets the TCP connection, discarding all packets in an existing sequence.
<b>SYN</b>	When set, the SYN flag indicates a request for a new session.
<b>FIN</b>	When set, the final flag indicates that the packet transfer is complete and the connection can be closed.
<b>R1</b>	This reserved bit (1 of 2) is not used.
<b>R2</b>	This reserved bit (2 of 2) is not used.

Table 88 on page 747 displays packet header fields and flags that you can set for attacks that use the UDP protocol.

Table 88: UDP Header Fields and Flags

Field	Description
<b>Source Port</b>	Specify a value for the port number on the attacking device.
<b>Destination Port</b>	Specify a value for the port number of the attack target.
<b>Data Length</b>	Specify a value for the number of bytes in the data payload.

Table 89 on page 747 displays packet header fields and flags that you can set for attacks that use the ICMP protocol.

Table 89: ICMP Header Fields and Flags

Field	Description
<b>ICMP Type</b>	Specify a value for the primary code that identifies the function of the request or reply packet.

Table 89: ICMP Header Fields and Flags (*continued*)

Field	Description
ICMP Code	Specify a value for the secondary code that identifies the function of the request or reply packet within a given type.
Sequence Number	Specify a value for the sequence number of the packet. This number identifies the location of the request or reply packet in relation to the entire sequence.
ICMP ID	Specify a value for the identification number. The identification number is a unique value used by the destination system to associate request and reply packets.
Data Length	Specify a value for the number of bytes in the data payload.

**Sample Signature Attack Definition**

The following is a sample signature attack definition:

```

<Entry>
<Name>sample-sig</Name>
<Severity>Major</Severity>
<Attacks><Attack>
<TimeBinding><Count>2</Count>
<Scope>dst</Scope></TimeBinding>
<Application>FTP</Application>
<Type>signature</Type>
<Context>packet</Context>
<Negate>>true</Negate>
<Flow>Control</Flow>
<Direction>any</Direction>
<Headers><Protocol><Name>ip</Name>
<Field><Name>ttl</Name>
<Match>==</Match><Value>128</Value></Field>
</Protocol><Name>tcp</Name>
<Field><Name><Match>&lt;</Match>
<value>1500</Value>
</Field></Protocol></Headers>
</Attack></Attacks>
</Entry>

```

**Attack Properties (Protocol Anomaly Attacks)**

A protocol anomaly attack object detects unknown or sophisticated attacks that violate protocol specifications (RFCs and common RFC extensions). You cannot create new protocol anomalies, but you can configure a new attack object that controls how your device handles a predefined protocol anomaly when detected.



**NOTE:** The service or application binding is a mandatory field for protocol anomaly attacks.

The following properties are specific to protocol anomaly attacks. Both attack direction and test condition are mandatory fields for configuring anomaly attack definitions.

### **Attack Direction**

Attack direction allows you to specify the connection direction of an attack. Using a single direction (instead of **Any**) improves performance, reduces false positives, and increases detection accuracy:

- Client to server (detects the attack only in client-to-server traffic)
- Server to client (detects the attack only in server-to-client traffic)
- Any (detects the attack in either direction)

### **Test Condition**

Test condition is a condition to be matched for an anomaly attack. Juniper Networks supports certain predefined test conditions. In the following example, the condition is a message that is too long. If the size of the message is longer than the preconfigured value for this test condition, the attack is matched.

```
<Attacks>
<Attack>
<Type>anomaly</Type>
...
<Test>MESSAGE_TOO_LONG</Test>
<Value>yes</Value>
...
</Attack>
</Attacks>
```

### **Sample Protocol Anomaly Attack Definition**

The following is a sample protocol anomaly attack definition:

```
<Entry>
<Name>sample-anomaly</Name>
<Severity>Info</Severity>
<Attacks><Attack>
<TimeBinding><Count>2</Count>
<Scope>peer</Scope></TimeBinding>
<Application>TCP</Application>
<Type>anomaly</Type>
<Test>OPTIONS_UNSUPPORTED</Test>
<Direction>any</Direction>
</Attack></Attacks>
</Entry>
```

### **Attack Properties (Compound or Chain Attacks)**

A compound or chain attack object detects attacks that use multiple methods to exploit a vulnerability. This object combines multiple signatures and/or protocol anomalies into a single attack object, forcing traffic to match a pattern of combined signatures and anomalies within the compound attack object before traffic is identified as an attack. By combining and even specifying the order in which signatures or anomalies must match, you can be very specific about the events that need to take place before the device identifies traffic as an attack.

You must specify a minimum of 2 members (attacks) in a compound attack. You can specify up to 32 members in compound attack. Members can be either signature or anomaly attacks.

The following properties are specific to compound attacks:

### **Scope**

Scope allows you to specify if the attack is matched within a session or across transactions in a session. If the specified service supports multiple transactions within a single session, you can also specify whether the match should occur over a single session or can be made across multiple transactions within a session:

- Specify *session* to allow multiple matches for the object within the same session.
- Specify *transaction* to match the object across multiple transactions that occur within the same session.

### **Order**

Use ordered match to create a compound attack object that must match each member signature or protocol anomaly in the order you specify. If you do not specify an ordered match, the compound attack object still must match all members, but the attack pattern or protocol anomalies can appear in the attack in random order.

### **Reset**

Specifies that a new log is generated each time an attack is detected within the same session. If this field is set to **no** then the attack is logged only once for a session.

### **Expression (Boolean expression)**

Using the boolean expression field disables the ordered match function. The boolean expression field makes use of the member name or member index properties. The following three boolean operators are supported along with parenthesis, which helps determine precedence:

- **or**—If either of the member name patterns match, the expression matches.
- **and**—If both of the member name patterns match, the expression matches. It does not matter which order the members appear in.
- **oand (ordered and)**—If both of the member name patterns match, and if they appear in the same order as specified in the boolean expression, the expression matches.

Suppose you have created five signature members, labelled **s1-s5**. Suppose you know that the attack always contains the pattern **s1**, followed by either **s2** or **s3**. You also know that the attack always contains **s4** and **s5**, but their positions in the attack can vary. In this case, you might create the following boolean expression: **((s1 oand s2) or (s1 oand s3)) and (s4 and s5)**



**NOTE:** You can either define an ordered match or an expression (not both) in a custom attack definition.

### Member Index

Member Index is specified in chain attacks to identify a member (attack) uniquely. In the following example, member index is used to identify the members **m01** and **m02** in the defined expression:

```
<Expression>m02 AND m01</Expression>
<Order>no</Order>
<Reset>no</Reset>
<ScopeOption/>
<Members>
<Attack>
<Member>m01</Member>
<Type>Signature</Type>
...
<Pattern><!CDATA[.*/latestversion]]></Pattern>
<Regex/>
</Attack>
<Attack><Member>m02</Member>
<Type>Signature</Type>
...
<Pattern><!CDATA[\\Skype\\!.*]]></Pattern>
<Regex/>
</Attack>
<Attack>
```



**NOTE:** When defining the expression, you must specify the member index for all members.

### Sample Compound Attack Definition

The following is a sample compound attack definition:

```
<Entry>
<Name>sample-chain</Name>
<Severity>Critical</Severity>
<Attacks><Attack>
<Application>HTTP</Application>
<Type>Chain</Type>
<Order>yes</Order>
<Reset>yes</Reset>
<Members><Attack>
<Type>Signature</Type>
<Context>packet</Context>
<Pattern><!CDATA[Unknown[]]></Pattern>
<Flow>Control</Flow>
<Direction>cts</Direction>
</Attack><Attack>
<Type>anomaly</Type>
<Test>CHUNK_LENGTH_OVERFLOW</Test>
<Direction>any</Direction>
</Attack></Members>
</Attack></Attacks>
</Entry>
```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- IDP Policies Overview on page 701
- Understanding IDP Policy Rulebases on page 713
- Understanding Predefined IDP Attack Objects and Object Groups on page 781
- Understanding IDP Protocol Decoders on page 752
- Understanding IDP Signature-Based Attacks on page 754
- Understanding IDP Protocol Anomaly-Based Attacks on page 758

## IDP Protocol Decoders

- Understanding IDP Protocol Decoders on page 752
- Example: Configuring IDP Protocol Decoders on page 753
- Understanding Multiple IDP Detector Support on page 754

### Understanding IDP Protocol Decoders

Protocol decoders are used by Intrusion Detection and Prevention (IDP) to check protocol integrity and protocol contextual information by looking for anomalies and ensuring that RFC standards are met. An anomaly can be any part of a protocol, such as the header, message body, or other individual fields that deviate from RFC standards for that protocol. For example, in the case of SMTP, if SMTP MAIL TO precedes SMTP HELO, that is an anomaly in the SMTP protocol.

When protocol contextual information is available, protocol decoders check for attacks within those contexts. For example, for SMTP, if an e-mail is sent to user@company.com, user@company.com is the contextual information and SMTP MAIL TO is the context. By using protocol contextual data, rather than the entire packet, for attack detection, protocol decoders improve overall performance and accuracy.

If there is a policy configured with a rule that matches the protocol decoder check for SMTP, the rule triggers and the appropriate action is taken.

The IDP module ships with a preconfigured set of protocol decoders. These protocol decoders have default settings for various protocol-specific contextual checks they perform. You can use these defaults or you can tune them to meet your site's specific needs. To display the list of available protocol decoders, enter the following command:

```
user@host # show security idp sensor-configuration detector protocol-name ?
```

For a more detailed view of the current set of protocol decoders and their default context values, you can view the *detector-capabilities.xml* file located in the `/var/db/idpd/sec-download` folder on the device. When you download a new security package, you also receive this file which lists current protocols and default decoder context values.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- IDP Policies Overview on page 701

- Understanding Custom Attack Objects on page 736
- Understanding IDP Protocol Anomaly-Based Attacks on page 758
- Understanding Multiple IDP Detector Support on page 754
- Understanding IDP Signature-Based Attacks on page 754
- Example: Configuring IDP Protocol Decoders on page 753

### Example: Configuring IDP Protocol Decoders

This example shows how to configure IDP protocol decoder tunables.

#### Requirements

Before you begin, review the IDP protocol decoders feature. See “Understanding IDP Protocol Decoders” on page 752.

#### Overview

The Junos IDP module ships with a set of preconfigured protocol decoders. These protocol decoders have default settings for various protocol-specific contextual checks that they perform. You can use the default settings or tune them to meet your site's specific needs. This example shows you how to tune the protocol decoder for FTP.

#### Configuration

#### Step-by-Step Procedure

To configure IDP protocol decoder tunables:

1. View the list of protocols that have tunable parameters.
 

```
[edit]
user@host# edit security idp sensor-configuration detector protocol-name FTP
```
2. Configure tunable parameters for the FTP protocol.
 

```
[edit security idp sensor-configuration-detector protocol-name FTP]
user@host# set tunable-name sc_ftp_failed_logins tunable-value 4
user@host# set tunable-name sc_ftp_failed_flags tunable value 1
user@host# set tunable-name sc_ftp_line_length tunable-value 1024
user@host# set tunable-name sc_ftp_password_length tunable-value 64
user@host# set tunable-name sc_ftp_sitestring_length tunable-value 512
user@host# set tunable-name sc_ftp_username_length tunable-value 32
```
3. If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

#### Verification

To verify the configuration is working properly, enter the `show security idp status` command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Multiple IDP Detector Support on page 754
- Understanding IDP Signature-Based Attacks on page 754

## Understanding Multiple IDP Detector Support

---

When a new security package is received, it contains attack definitions and a detector. In any given version of a security package, the attack definitions correspond to the capabilities of the included detector. When policy aging is disabled on the device (see the *reset-on-policy* command in the *Junos OS CLI Reference* for policy aging commands), only one policy is in effect at any given time. But if policy aging is enabled and there is a policy update, the existing policy is not unloaded when the new policy is loaded. Therefore, both policies can be in effect on the device. In this case, all existing sessions will continue to be inspected by existing policies and new sessions are inspected with new policies. Once all the existing sessions using the older policy have terminated or expired, the older policy is then unloaded.

When a policy is loaded, it is also associated with a detector. If the new policy being loaded has an associated detector that matches the detector already in use by the existing policy, the new detector is not loaded and both policies use a single associated detector. But if the new detector does not match the current detector, the new detector is loaded along with the new policy. In this case, each loaded policy will then use its own associated detector for attack detection.

Note that a maximum of two detectors can be loaded at any given time. If two detectors are already loaded (by two or more policies), and loading a new policy requires also loading a new detector, then at least one of the loaded detectors must be unloaded before the new detector can be loaded. Before a detector is unloaded, all policies that use the corresponding detector are unloaded as well.

You can view the current policy and corresponding detector version by entering the following command:

```
user@host> show security idp status
```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding IDP Protocol Decoders on page 752](#)
- [Example: Configuring IDP Protocol Decoders on page 753](#)
- [Understanding IDP Signature-Based Attacks on page 754](#)

## IDP Signature-Based Attacks

- [Understanding IDP Signature-Based Attacks on page 754](#)
- [Example: Configuring IDP Signature-Based Attacks on page 755](#)

### Understanding IDP Signature-Based Attacks

---

To configure a custom attack object, you specify a unique name for it and then specify additional information, which can make it easier for you to locate and maintain the attack object.

Certain properties in the attack object definitions are common to all types of attacks, such as attack name, severity level, service or application binding, time binding, and



protocol or port binding. Some fields are specific to an attack type and are available only for that specific attack definition.

Signature attack objects use a stateful attack signature (a pattern that always exists within a specific section of the attack) to detect known attacks. They also include the protocol or service used to perpetrate the attack and the context in which the attack occurs. The following properties are specific to signature attacks, and you can configure them when configuring signature attack—attack context, attack direction, attack pattern, and protocol-specific parameters (TCP, UDP, ICMP, or IP header fields).

When configuring signature-based attacks, keep the following in mind:

- Attack context and direction are mandatory fields for the signature attack definition.
- Pattern negation is supported for packet, line, and application-based contexts only and not for stream and normalized stream contexts.
- When configuring the protocol-specific parameters, you can specify fields for only one of the following protocols—IP, TCP, UDP, or ICMP.
- When configuring a protocol binding, you can specify only one of the following—IP, ICMP, TCP, UDP, RPC or applications.
  - IP—Protocol number is a mandatory field.
  - TCP and UDP—You can specify either a single port (minimum-port) or a port range (minimum-port and maximum-port). If you do not specify a port, the default value is taken (0-65535).
  - RPC—Program number is a mandatory field.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [IDP Policies Overview on page 701](#)
- [Understanding Custom Attack Objects on page 736](#)
- [Understanding Predefined IDP Attack Objects and Object Groups on page 781](#)
- [Understanding IDP Protocol Decoders on page 752](#)
- [Example: Configuring IDP Signature-Based Attacks on page 755](#)
- [Example: Configuring IDP Protocol Anomaly-Based Attacks on page 759](#)

### [Example: Configuring IDP Signature-Based Attacks](#)

This example shows how to create a signature-based attack object.

- [Requirements on page 756](#)
- [Overview on page 756](#)
- [Configuration on page 756](#)
- [Verification on page 758](#)

**Requirements**

Before you begin, configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

**Overview**

In this example, you create a signature attack called sig1 and assign it the following properties:

- Recommended action (drop packet)—Drops a matching packet before it can reach its destination but does not close the connection.
- Time binding—Specifies the scope as **source** and the count as **10**. When scope is **source**, all attacks from the same source are counted, and when the number of attacks reaches the specified count (**10**), the attack is logged. In this example, every tenth attack from the same source is logged.
- Attack context (packet)—Matches the attack pattern within a packet.
- Attack direction (any)—Detects the attack in both directions—client-to-server and server-to-client traffic.
- Protocol (TCP)—Specifies the TTL value of 128.
- Shellcode (Intel)—Sets the flag to detect shellcode for Intel platforms.
- Protocol binding—Specifies the TCP protocol and ports 50 through 100.

Once you have configured a signature-based attack object, you specify the attack as match criteria in an IDP policy rule. See “Example: Defining Rules for an IDP IPS Rulebase” on page 717.

**Configuration****CLI Quick Configuration**

To quickly create a signature-based attack object, copy the following commands and paste them into the CLI:

```
[edit]
set security idp custom-attack sig1 severity major
set security idp custom-attack sig1 recommended-action drop-packet
set security idp custom-attack sig1 time-binding scope source count 10
set security idp custom-attack sig1 attack-type signature context packet
set security idp custom-attack sig1 attack-type signature shellcode intel
set security idp custom-attack sig1 attack-type signature protocol ip ttl value 128 match
equal
set security idp custom-attack sig1 attack-type signature protocol-binding tcp
minimum-port 50 maximum-port 100
set security idp custom-attack sig1 attack-type signature direction any
```

**Step-by-Step Procedure**

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To create a signature-based attack object:

1. Specify a name for the attack.

- ```
[edit]
user@host# edit security idp custom-attack sig1
```
2. Specify common properties for the attack.
 

```
[edit security idp custom-attack sig1]
user@host# set severity major
user@host# set recommended-action drop-packet
user@host# set time-binding scope source count 10
```
  3. Specify the attack type and context.
 

```
[edit security idp custom-attack sig1]
user@host# set attack-type signature context packet
```
  4. Specify the attack direction and the shellcode flag.
 

```
[edit security idp custom-attack sig1]
user@host# set attack-type signature shellcode intel
```
  5. Set the protocol and its fields.
 

```
[edit security idp custom-attack sig1]
user@host# set attack-type signature protocol ip ttl value 128 match equal
```
  6. Specify the protocol binding and ports.
 

```
[edit security idp custom-attack sig1]
user@host# set attack-type signature protocol-binding tcp minimum-port 50
maximum-port 100
```
  7. Specify the direction.
 

```
[edit security idp custom-attack sig1]
user@host# set attack-type signature direction any
```

**Results** From configuration mode, confirm your configuration by entering the **show security idp** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security idp
custom-attack sig1 {
  recommended-action drop-packet;
  severity major;
  time-binding {
    count 10;
    scope source;
  }
  attack-type {
    signature {
      protocol-binding {
        tcp {
          minimum-port 50 maximum-port 100;
        }
      }
    }
    context packet;
    direction any;
    shellcode intel;
    protocol {
```

```
ip {
  ttl {
    match equal;
    value 128;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

#### **Verification**

To confirm that the configuration is working properly, perform this task:

- [Verifying the Configuration on page 758](#)

#### **Verifying the Configuration**

**Purpose** Verify that the signature-based attack object was created.

**Action** From operational mode, enter the **show security idp status** command.

#### **Related Documentation**

- [Junos OS CLI Reference](#)
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding IDP Signature-Based Attacks on page 754](#)
- [Understanding Custom Attack Objects on page 736](#)
- [Understanding Predefined IDP Attack Objects and Object Groups on page 781](#)
- [Understanding IDP Protocol Decoders on page 752](#)
- [Example: Configuring IDP Protocol Anomaly-Based Attacks on page 759](#)

## **IDP Protocol Anomaly-Based Attacks**

- [Understanding IDP Protocol Anomaly-Based Attacks on page 758](#)
- [Example: Configuring IDP Protocol Anomaly-Based Attacks on page 759](#)

### **Understanding IDP Protocol Anomaly-Based Attacks**

A protocol anomaly attack object detects unknown or sophisticated attacks that violate protocol specifications (RFCs and common RFC extensions). You cannot create new protocol anomalies, but you can configure a new attack object that controls how your device handles a predefined protocol anomaly when detected.

The following properties are specific to protocol anomaly attacks:

- Attack direction
- Test condition

When configuring protocol anomaly-based attacks, keep the following in mind:

- The service or application binding is a mandatory field for protocol anomaly attacks. Besides the supported applications, services also include IP, TCP, UDP, ICMP, and RPC.
- The attack direction and test condition properties are mandatory fields for configuring anomaly attack definitions.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- IDP Policies Overview on page 701
- Understanding IDP Protocol Decoders on page 752
- Understanding Custom Attack Objects on page 736
- Understanding Predefined IDP Attack Objects and Object Groups on page 781
- Example: Configuring IDP Protocol Anomaly-Based Attacks on page 759

#### Example: Configuring IDP Protocol Anomaly-Based Attacks

This example shows how to create a protocol anomaly-based attack object.

- Requirements on page 759
- Overview on page 759
- Configuration on page 760
- Verification on page 761

##### **Requirements**

Before you begin, configure network interfaces. See the [Junos OS Interfaces Configuration Guide for Security Devices](#)

##### **Overview**

In this example, you create a protocol anomaly attack called anomaly1 and assign it the following properties:

- Time binding—Specifies the scope as **peer** and count as **2** to detect anomalies between source and destination IP addresses of the sessions for the specified number of times.
- Severity (info)—Provides information about any attack that matches the conditions.
- Attack direction (any)—Detects the attack in both directions—client-to-server and server-to-client traffic.
- Service (TCP)—Matches attacks using the TCP service.
- Test condition (OPTIONS\_UNSUPPORTED)—Matches certain predefined test conditions. In this example, the condition is to match if the attack includes unsupported options.
- Shellcode (sparc)—Sets the flag to detect shellcode for Sparc platforms.

Once you have configured the protocol anomaly-based attack object, you specify the attack as match criteria in an IDP policy rule. See “Example: Defining Rules for an IDP IPS Rulebase” on page 717.

### Configuration

**CLI Quick Configuration** To quickly create a protocol anomaly-based attack object, copy the following commands and paste them into the CLI:

```
[edit]
set security idp custom-attack anomaly1 severity info
set security idp custom-attack anomaly1 time-binding scope peer count 2
set security idp custom-attack anomaly1 attack-type anomaly test
  OPTIONS_UNSUPPORTED
set security idp custom-attack sa
set security idp custom-attack sa attack-type anomaly service TCP
set security idp custom-attack sa attack-type anomaly direction any
set security idp custom-attack sa attack-type anomaly shellcode sparc
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To create a protocol anomaly-based attack object:

1. Specify a name for the attack.

```
[edit]
user@host# edit security idp custom-attack anomaly1
```

2. Specify common properties for the attack.

```
[edit security idp custom-attack anomaly1]
user@host# set severity info
user@host# set time-binding scope peer count 2
```

3. Specify the attack type and test condition.

```
[edit security idp custom-attack anomaly1]
user@host# set attack-type anomaly test OPTIONS_UNSUPPORTED
```

4. Specify other properties for the anomaly attack.

```
[edit security idp custom-attack anomaly1]
user@host# set attack-type anomaly service TCP
user@host# set attack-type anomaly direction any
user@host# attack-type anomaly shellcode sparc
```

**Results** From configuration mode, confirm your configuration by entering the **show security idp** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security idp
custom-attack anomaly1 {
  severity info;
  time-binding {
```

```

        count 2;
        scope peer;
    }
    attack-type {
        anomaly {
            test OPTIONS_UNSUPPORTED;
            service TCP;
            direction any;
            shellcode sparc;
        }
    }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying the Configuration on page 761

### Verifying the Configuration

**Purpose** Verify that the protocol anomaly-based attack object was created.

**Action** From operational mode, enter the **show security idp status** command.

### Related Documentation

- [Junos OS CLI Reference](#)
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding IDP Protocol Anomaly-Based Attacks on page 758
- Example: Updating the IDP Signature Database Manually on page 785
- Example: Updating the Signature Database Automatically on page 788

## Listing IDP Test Conditions for a Specific Protocol

When configuring IDP custom attacks, you can specify list test conditions for a specific protocol. To list test conditions for ICMP:

1. List supported test conditions for ICMP and choose the one you want to configure. The supported test conditions are available in the CLI at the **[edit security idp custom-attack test] attack-type anomaly** hierarchy level.

```
user@host#set test icmp?
```

```
Possible completions:
```

```
<test> Protocol anomaly condition to be checked
```

```

ADDRESSMASK_REQUEST
DIFF_CHECKSUM_IN_RESEND
DIFF_CHECKSUM_IN_RESPONSE
DIFF_LENGTH_IN_RESEND

```

2. Configure the service for which you want to configure the test condition.  
`user@host# set service ICMP`
3. Configure the test condition (specifying the protocol name is not required).  
`user@host# set test ADDRESSMASK_REQUEST`
4. If you are done configuring the device, enter **commit** from configuration mode.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding IDP Protocol Anomaly-Based Attacks on page 758](#)
- [Example: Configuring IDP Protocol Anomaly-Based Attacks on page 759](#)



# Application-Level Distributed Denial of Service

- IDP Application-Level DDoS Attack Overview on page 763
- IDP Application-Level DDoS Protection Overview on page 763
- Example: Enabling IDP Protection Against Application-Level DDoS Attacks on page 768
- Understanding Application-level DDoS Statistics Reporting on page 772
- Example: Configuring Application-Level DDoS Statistics Reporting on page 775

## IDP Application-Level DDoS Attack Overview

---

The intent of an application-level DDoS attack is to overwhelm the targeted server, such as a DNS or HTTP servers, so it can not perform it's intended services. This is done by making a tremendous amount of application requests from malicious bot clients that often use spoofed IP addresses.

Application-level DDoS attacks are different than traditional Layer 3 and Layer 4 DDoS attacks, such as a SYN flood. From a Layer 3 and Layer 4 perspective, the attack can appear as legitimate transactions. Traditional Layer 3 and Layer 4 DDoS solutions can only rate limit these attacks and begin the application transactions, instead of denying the attacks.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- IDP Policies Overview on page 701
- Understanding IDP Application-Level DDoS Rulebases on page 715
- IDP Application-Level DDoS Protection Overview on page 763
- Example: Enabling IDP Protection Against Application-Level DDoS Attacks on page 768

## IDP Application-Level DDoS Protection Overview

---

- Understanding the Application-Level DDoS Module on page 764
- Understanding the Application-Level DDoS Definition on page 765
- Understanding the Application-Level DDoS Rule on page 766

- Understanding Application-Level DDoS IP-Action on page 767
- Understanding Application-Level DDoS Session Action on page 768

## Understanding the Application-Level DDoS Module

The application-level distributed denial-of-service (application-level DDoS) IDP module uses application-level metrics to differentiate between normal and malicious application requests. It then identifies the offending source addresses and can drop or deny these requests. Based on user-configured application thresholds, when the client application transactions exceed the defined thresholds, session and ip-actions are applied on traffic from the offending client address. This feature protects servers against DNS and HTTP application-level DDoS attacks.

To identify malicious bot clients, you create a policy with `rulebase-ddos` to monitor specific traffic and define application-level DDoS application metrics and thresholds to monitor that traffic. When the threshold are exceeded, your defined action is taken on the client to protect the application server.

IDP performs multistage analysis from connection monitoring, to protocol analysis, to bot client classification, and maintains the state for each protected server. You can configure `connection-rate-threshold` in the application-level DDoS application definition to monitor connection rate. When connection thresholds rate are exceeded, IDP transitions to protocol analysis for deep content inspection and maintains statistical data on application transactions. The application-level DDoS attacks can be classified into either heavy hitters or random hitters. Heavy hitters perform identical application transactions in a fast and repeated fashion, for example, querying a nonexisting domain name repeatedly. Random hitters perform random application transactions, for example, querying a random domain name, one per each request. You can configure `context value-hit-rate-threshold` to detect heavy hitters and `context hit-rate-threshold` to detect random hitters. If either of the context thresholds are exceeded, IDP transitions to the bot client classification stage, where it tracks the application transactions on a per-client basis based on user-configured `time-binding` thresholds. A benign client will not perform identical and repeated transactions, whereas malicious bot clients will. Once `time-binding` thresholds are exceeded, identified bot clients will be blocked with the configured ip-action and session actions.

You can also configure a list of regular expressions under `exclude-context-values` to exempt certain context values from being considered for application-level DDoS processing. This is helpful for requests for well-known resources that can often hit context thresholds, for example, a DNS query for domain name `google.com`.

Protocol analysis stage uses a default interval of 60 seconds for `context hit-rate-threshold` and `value-hit-rate-threshold`. For example, if you configure 10,000 as the `value-hit-rate-threshold`, the context value would be monitored against a 10,000 hits limit in a 60-second interval.

IDP also uses hysteresis for state transitions to avoid thrashing between the states. A default of 20 percent lower limit will be used from the configured connection and context thresholds for falling behind in state. For example, if you configure a `context value-hit-rate-threshold` of 10,000, IDP transitions from protocol analysis to bot client

classification after 10,000 hits in 60 seconds for identical context values, and falls behind in state only when such hits are smaller than 8000 in 60 seconds.

We recommend configuring time-binding thresholds in the application-ddos definition, because it is critical to differentiate between benign clients and malicious bot clients. However, if you choose not to define time-binding thresholds, IDP will not do bot client classification. In this case, if application transactions exceed context thresholds, the configured ip-action and session actions will be performed. Note that without bot client classification, benign clients might get denied when making a request to the protected server.

IDP maintains application-level DDoS state for the current policy only. For more information, see the policy aging reference in “Understanding Multiple IDP Detector Support” on page 754. Traffic from sessions using older policy will not be inspected for application-level DDoS. If a new policy is loaded, application-level DDoS state for each protected server will be relearned.

## Understanding the Application-Level DDoS Definition

You can only configure one application-ddos definition for each protected server. However, you can use the same application-ddos definition in two or more rules with specific destination-address, to-zone, or both to protect two or more servers with the same desired application-ddos thresholds.

Table 90 on page 765 shows the parameters that can be set for application-ddos. For more details, see the *Junos OS CLI Reference* guide.



**NOTE:** Application-level denial-of-service (application-level DDoS) detection will not work if two rules with different application-level DDoS applications process traffic going to a single destination application server. When setting up application-level DDoS rules, make sure you do not configure rulebase-ddos rules that have two different application-ddos objects while the traffic destined to one application server can process more than one rule. Essentially, for each protected application server, you have to configure application-level DDoS rules so traffic destined for one protected server only hits one application-level DDoS rule.

application-level DDoS rules are terminal, which means that once traffic is processed by one rule, it will not be processed by other rules.

**Table 90: Application DDoS Parameters**

| Parameter                   | Description                                                         |
|-----------------------------|---------------------------------------------------------------------|
| <b>service service-name</b> | The Application Layer service to be monitored, such as DNS or HTTP. |

Table 90: Application DDoS Parameters (*continued*)

| Parameter                          | Description                                                                                                                                                                                                                                                                                             |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>exclude-context-value</b>       | Configure a list of common context value patterns that should be excluded from application-level DDoS detection. For example, if you have a webserver that receives a high number of HTTP requests on the home/landing page, you can exclude it from application-level DDoS detection.                  |
| <b>connection-rate threshold</b>   | The connections-per-second threshold at which to start monitoring the application context values.                                                                                                                                                                                                       |
| <b>context <i>context-name</i></b> | Name of the application context that the IDP protocol decoder generates while parsing the application protocol from traffic data.                                                                                                                                                                       |
| <b>hit-rate-threshold</b>          | Number of context hits in tick interval (60 seconds by default) to start bot client classification, if timebinding parameters are configured. If timebinding parameters are not configured, the configured policy actions are taken.                                                                    |
| <b>value-hit-rate-threshold</b>    | Number of context value hits in tick interval to start bot client classification, if timebinding parameters are configured. If time binding parameters are not configured, the configured policy actions are taken.                                                                                     |
| <b>max-context-values</b>          | The top <i>n</i> context values that should be monitored, reported, or both.                                                                                                                                                                                                                            |
| <b>time-binding-period</b>         | The time-binding period to determine if a client should be classified as a malicious bot client or not. This setting is used in conjunction with time-binding count to detect an attack if a client request for the same context value exceeds time-binding-count times in time-binding-period seconds. |
| <b>time-binding-count</b>          | The number of context or context value hits from each client over the time binding period to determine if it should be considered a malicious bot client.                                                                                                                                               |

## Understanding the Application-Level DDoS Rule

You configure one or more application-Level DDoS rules to define the traffic that should be monitored to protect your servers.

Table 91 on page 766 shows the parameters that can be set for application-ddos.

Table 91: application-level DDoS Rule Parameters

| Parameter             | Description             |
|-----------------------|-------------------------|
| <b>from-zone</b>      | Match source zone.      |
| <b>source-address</b> | Match source address.   |
| <b>to-zone</b>        | Match destination zone. |

Table 91: application-level DDoS Rule Parameters (*continued*)

| Parameter                  | Description                                                                            |
|----------------------------|----------------------------------------------------------------------------------------|
| <b>destination-address</b> | Match destination address.                                                             |
| <b>application</b>         | Choose default to select the application service from the application-ddos definition. |
| <b>application-ddos</b>    | Specify the DDoS application.                                                          |

## Understanding Application-Level DDoS IP-Action

You configure ip-action either to drop future sessions from identified bot client addresses for a specified time or to rate-limit future connections.

Table 92 on page 767 shows the available parameters for configuring an application-level DDoS IP action.

Table 92: Application-Level DDoS IP-Action Parameters

| Parameter                       | Description                                                                                                                                                                       |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ip-block</b>                 | Blocks future connections of any session that matches the IP action.                                                                                                              |
| <b>ip-close</b>                 | Closes future connections of any client address that matches the IP action by sending an RST packet to the client.<br><br>If TCP is not used, the connection is dropped silently. |
| <b>ip-connection-rate-limit</b> | Rate-limits future connections based on a connections per second limit that you set. This parameter can be used to reduce the number of attacks from a client.                    |
| <b>ip-notify</b>                | Takes no action against matching future connections, but logs the event.                                                                                                          |
| <b>destination-address</b>      | Matches traffic based on the destination address of the attack traffic.                                                                                                           |
| <b>service</b>                  | Matches traffic based on the source address, destination address, destination port, and protocol of the attack traffic. This is the default.                                      |
| <b>source-address</b>           | Matches traffic based on the source address of the attack traffic.                                                                                                                |
| <b>source-zone</b>              | Matches traffic based on the source zone of the attack traffic.                                                                                                                   |
| <b>zone-service</b>             | Matches traffic based on the source zone, destination address, destination port, and protocol of the attack traffic.                                                              |
| <b>log</b>                      | Logs the information about the IP action against the traffic that matches a rule.                                                                                                 |
| <b>timeout</b>                  | Specifies the number of seconds that you want the IP action to remain in effect after a traffic match.                                                                            |

## Understanding Application-Level DDoS Session Action

Session action determines what action should be performed on the identified bot client.

Table 93 on page 768 shows the parameters that can be set for action.

**Table 93: application-level DDoS Action Parameters**

| Parameter              | Description                                                                                                                                                                                                                                                                                                                                                    |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>close-server</b>    | Closes the connection and sends an RST packet to the server but not to the client.                                                                                                                                                                                                                                                                             |
| <b>drop-connection</b> | Drops all packets associated with the connection, preventing traffic for the connection from reaching its destination. Use this action to drop connections for traffic that is not prone to spoofing.                                                                                                                                                          |
| <b>drop-packet</b>     | Drops a matching packet before it can reach its destination but does not close the connection. Use this action to drop packets for attacks in traffic that is prone to spoofing, such as UDP traffic. Dropping a connection for such traffic could result in a denial of service that prevents you from receiving traffic from a legitimate source-IP address. |
| <b>no-action</b>       | No action is taken. Use this action when you want to generate logs for only some traffic.                                                                                                                                                                                                                                                                      |

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding IDP Application-Level DDoS Rulebases on page 715](#)
- [IDP Application-Level DDoS Attack Overview on page 763](#)
- [Example: Enabling IDP Protection Against Application-Level DDoS Attacks on page 768](#)

## Example: Enabling IDP Protection Against Application-Level DDoS Attacks

This example shows how to use the application-level DDoS module to protect a DNS server from an application-level DDoS attack.

- [Requirements on page 768](#)
- [Overview on page 768](#)
- [Configuration on page 769](#)
- [Verification on page 772](#)

### Requirements

No special configuration beyond device initialization is required before configuring this feature.

### Overview

Before configuring application-level DDoS protection for a DNS server, observe the average load of DNS requests on the server you want to protect so you can decide the

application thresholds to configure. Next, define the application thresholds and when the client application for transactions exceeds those thresholds, session and IP actions are applied on traffic from the offending client address.

For example, if the DNS server is expected to handle a normal load of 1000 requests per second, choose 20 percent in excess of the normal load (1200 requests per second) as the connection-rate-threshold value. This value is essentially 60,000 transactions in 60 seconds, so choose 20 percent in excess of this load (72,000) as the context hit-rate-threshold value. You can choose a context value-hit-rate-threshold based on the maximum load of requests for the same domain name being queried. For example, if it is impractical for DNS to receive queries for domain xyz.com in excess of 2000 times in 60 seconds, set the context value-hit-rate-threshold to 20 percent more than that value, which would be 2400 times in 60 seconds.

For monitoring and reporting, you can optionally set the max-context-values to 100, so at the maximum, the most active 100 DNS query requests will be monitored and reported. If a client is in this range, it is most likely a malicious bot client. Once bot clients are identified, you can configure ip-action as ip-block with a timeout of 600 seconds (the bot client gets access denied for 10 minutes) and session action is set as drop-packet.

In this example, IDP starts deep protocol analysis when the number of connections per second exceeds 1200. IDP also starts bot client classification if either the total number of queries for context dns-type-name exceeds 72,000 or if requests for the same query value exceeds 2400.



**NOTE:** When an application-level DDoS attack occurs on the application server, it will have much higher transaction rates than it does under normal or even peak load. Therefore, best practice is to set higher thresholds than the normal peak of the application server so it does not trigger unnecessary client classification processing. This setting improves the overall performance of the Juniper Networks device because the application-level DDoS module will not start client classification until the server has actually reached abnormal transaction rates.



**NOTE:** You can only define one DDoS application per application-level DDoS rule. Create additional rules to monitor multiple DDoS applications.

Each application-level DDoS rule is a terminal rule, meaning that only one matching rule is considered for incoming traffic matching.

## Configuration

**CLI Quick Configuration** To quickly configure IDP protection against application-level DDoS attacks, copy the following commands and paste them into the CLI:

```
[edit]
set security idp idp-policy AppDDos-policy-1 rulebase-ddos rule AppDDos-rule1 match
source-address any
```

```

set security idp idp-policy AppDDos-policy-1 rulebase-ddos rule AppDDos-rule1 match
  to-zone any
set security idp idp-policy AppDDos-policy-1 rulebase-ddos rule AppDDos-rule1 match
  destination-address any
set security idp idp-policy AppDDos-policy-1 rulebase-ddos rule AppDDos-rule1 match
  application default
set security idp idp-policy AppDDos-policy-1 rulebase-ddos rule AppDDos-rule1 match
  application-ddos dns-server-1
set security idp idp-policy AppDDos-policy-1 rulebase-ddos rule AppDDos-rule1 then
  action drop-packet
set security idp idp-policy AppDDos-policy-1 rulebase-ddos rule AppDDos-rule1 then
  ip-action ip-block
set security idp idp-policy AppDDos-policy-1 rulebase-ddos rule AppDDos-rule1 then
  ip-action timeout 600
set security idp application-ddos dns-server-1 connection-rate-threshold 1200
set security idp application-ddos dns-server-1 context dns-type-name hit-rate-threshold
  72000
set security idp application-ddos dns-server-1 context dns-type-name
  value-hit-rate-threshold 2400
set security idp application-ddos dns-server-1 context dns-type-name max-context-values
  100
set security idp application-ddos dns-server-1 context dns-type-name time-binding-count
  10
set security idp application-ddos dns-server-1 context dns-type-name time-binding-period
  30
set security idp application-ddos dns-server-1 context dns-type-name
  exclude-context-values .*google.com
set security idp application-ddos dns-server-1 context dns-type-name
  exclude-context-values .*yahoo.com
set security idp application-ddos dns-server-1 service dns

```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure IDP protection against application-level DDoS attacks:

1. Define the type of traffic, the protocol context to monitor, and thresholds to use to trigger an action.

```

[edit security idp]
user@host# set application-ddos dns-server-1
user@host# set application-ddos dns-server-1 service dns
user@host# set application-ddos dns-server-1 connection-rate-threshold 1200
user@host# set application-ddos dns-server-1 context dns-type-name
  hit-rate-threshold 72000
user@host# set application-ddos dns-server-1 context dns-type-name
  value-hit-rate-threshold 2400
user@host# set application-ddos dns-server-1 context dns-type-name
  max-context-values 100
user@host# set application-ddos dns-server-1 context dns-type-name
  time-binding-count 10
user@host# set application-ddos dns-server-1 context dns-type-name
  time-binding-period 30

```

2. Set context values that will be exempt from monitoring.



```
[edit security idp]
user@host# set application-ddos dns-server-1 context dns-type-name
exclude-context-values .*google.com
[edit security idp]
user@host# set application-ddos dns-server-1 context dns-type-name
exclude-context-values .*yahoo.com
```

3. Set the IDP policy rule for rulebase-ddos to define the source and destination of monitored traffic.

```
[edit security idp]
user@host# set idp-policy AppDDoS-policy-1 rulebase-ddos rule AppDDoS-rule1
match source-address any
[edit security idp]
user@host# set idp-policy AppDDoS-policy-1 rulebase-ddos rule AppDDoS-rule1
match to-zone any
[edit security idp]
user@host# set idp-policy AppDDoS-policy-1 rulebase-ddos rule AppDDoS-rule1
match destination-address any
[edit security idp]
user@host# set idp-policy AppDDoS-policy-1 rulebase-ddos rule AppDDoS-rule1
match application default
[edit security idp]
user@host# set idp-policy AppDDoS-policy-1 rulebase-ddos rule AppDDoS-rule1
match application-ddos dns-server-1
[edit security idp]
```

4. Define the action to be taken when application-level DDoS attack traffic is detected.

```
[edit security idp]
user@host# set idp-policy AppDDoS-policy-1 rulebase-ddos rule AppDDoS-rule1
then action drop-packet
[edit security idp]
user@host# set idp-policy AppDDoS-policy-1 rulebase-ddos rule AppDDoS-rule1
then ip-action ip-block
[edit security idp]
user@host# set idp-policy AppDDoS-policy-1 rulebase-ddos rule AppDDoS-rule1
then ip-action timeout 600
```

**Results** From configuration mode, confirm your configuration by entering the **show security idp** and **show security idp application-ddos dns-server-1** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security idp
idp-policy AppDDoS-policy-1 {
  rulebase-ddos {
    rule AppDDoS-rule1 {
      match {
        source-address any;
        to-zone any;
        destination-address any;
        application default;
        application-ddos {
          dns-server-1;
        }
      }
    }
  }
}
```

```

    }
  }
  then {
    action {
      drop-packet;
    }
    ip-action {
      ip-block;
      timeout 600;
    }
  }
}
}
}
[edit]
user@host#show security idp application-ddos dns-server-1
context dns-type-name {
  hit-rate-threshold 72000;
  value-hit-rate-threshold 2400;
  max-context-values 100;
  time-binding-count 10;
  time-binding-period 30;
  exclude-context-values [ .*google.com .*yahoo.com ];
}

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform this task:

- Verifying IDP Protection Against Application-Level DDoS Attacks on page 772

### [Verifying IDP Protection Against Application-Level DDoS Attacks](#)

**Purpose** Verify basic statistics for the servers being protected by the IDP application-level DDoS feature.

**Action** From operational mode, enter the **show security idp application-ddos** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding IDP Application-Level DDoS Rulebases on page 715
- IDP Application-Level DDoS Attack Overview on page 763
- IDP Application-Level DDoS Protection Overview on page 763

## [Understanding Application-level DDoS Statistics Reporting](#)

To successfully mitigate application-level distributed denial-of-service (DDoS) attacks on your network environment, you need to set the appropriate rule thresholds. To identify the appropriate thresholds, you need to analyze network statistical data. With application-level DDoS statistic reporting, you can collect application information on

connection, context and rates, and data records from application requests destined for your protected servers. With this information, you can determine trends to help you create more efficient rules for your environment.

Following are the main features of statistic reporting:

- Application-level DDoS reporting on application request data.
- Statistics collection of application connection rates and context rates on a periodic basis that you define. The default snapshot interval is once every 1 minute and the range is 1 through 60 minutes.
- Report files are written to the Routing Engine (RE) data storage device for extensive storage space.
- Automatic file compression of statistical report files when file size reaches 10 MB.



**NOTE:** Statistic reports are saved on the Routing Engine (RE) data storage device in the `/var/log/addos` directory. There must be at least 2 GB of free space to allow report logging.

The IDP module polls for application-level DDoS records and takes a snapshot of current activity at intervals that you define. Each statistical record collected represents an application request data entry (context value) up to 4 KB. Information collected includes the server IP address, zone, connection, and context rates, protocol, and Layer 7 service and context values. The `max-context-values` setting determines how many records should be collected per application context.

The filenaming convention for reports stored in `/var/log/addos` comprises the prefix `addos-stats` along with the record creation timestamp in the format `YYYYMMDDHHMMSS` (year/month/day/hour/minute/seconds). For example: `addos-stats-20100501091500`, is May 1, 2010 at 9:15 AM.

The report files are in comma-separated value (`.csv`) format and should be copied off the device to be analyzed in a program that can read `.csv` files, such as Excel. See Table 94 on page 773 for descriptions of each field in the application-level DDoS statistic record.

**Table 94: Application-Level DDoS Statistic Record Fields**

| Field                       | Description                                                                                      |
|-----------------------------|--------------------------------------------------------------------------------------------------|
| <code>time</code>           | Time the event occurred.                                                                         |
| <code>record-type</code>    | Type of record that is created. Type <code>app-record</code> is supported.                       |
| <code>record-data</code>    | Identifies the type of data collected ( <code>addos-http-url</code> or <code>addos-dns</code> ). |
| <code>destination-ip</code> | Destination IP of the application request.                                                       |
| <code>ddos-app-name</code>  | Name of the configured application object defined in the application-level DDoS rule.            |

Table 94: Application-Level DDoS Statistic Record Fields (*continued*)

| Field                   | Description                                                                                                              |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------|
| conn/sec                | Connection attempts per second by the application.                                                                       |
| context-name            | Context name in the application header.                                                                                  |
| context-hits/tick       | Number of context hits per tick interval. The default tick interval is 60 seconds.                                       |
| context-value-hits/tick | Number of context value hits per tick interval. The default tick interval is 60 seconds.                                 |
| context-value           | Application context name. The context-value is reported both in hexadecimal and ASCII formats and is no larger than 4 K. |

The following output shows an application-level DDoS statistic record.

```
2010:01:16:04:23:53,app-record,my-http,5.0.0.1,trust,6,http-url-parsed,1234/60sec,1234/60sec,ascii:/abc.html
hex:2F6162632e68746d6c
2010:01:16:04:23:53,app-record,my-http,5.0.0.1,trust,6,http-url-parsed,932791/60sec,932791/60sec,ascii:/index.html
hex:2F696e6465782e68746d6c
```

The following screen shot shows a formatted application-level DDoS statistic report.

| time                 | record-type | record-data    | destination-ip | ddos-app-name | conn/sec | context-name           | context-hits/tick | context-value-hits/tick | context-value                            |
|----------------------|-------------|----------------|----------------|---------------|----------|------------------------|-------------------|-------------------------|------------------------------------------|
| "2010:01:22:20:56:34 | app-record  | addos-http-url | 61.0.3.43      | bps-servers   | 24       | http-header-user-agent | 199/60sec         | 199/60sec               | ascii:Mozilla/4.0 (compatible; MSIE 6.0; |
| "2010:01:22:20:56:34 | app-record  | addos-http-url | 61.0.3.7       | bps-servers   | 31       | http-header-user-agent | 196/60sec         | 196/60sec               | ascii:Mozilla/4.0 (compatible; MSIE 6.0; |
| "2010:01:22:20:56:34 | app-record  | addos-dns      | 71.2.0.72      | test-servers  | 16       | dns-type-name          | 330/60sec         | 11/60sec                | ascii:.host28 hex:0001686f73743238"      |
| "2010:01:22:20:56:34 | app-record  | addos-dns      | 71.2.0.94      | test-servers  | 23       | dns-type-name          | 529/60sec         | 11/60sec                | ascii:.host28 hex:0001686f73743238"      |



**NOTE:** To clear out statistics files that are no longer needed, you run the operational command `request system storage cleanup`.

You can use the statistical data you collect to analyze application-level DDoS activity and identify the types and rates of application activity hitting your server. Typically, you will initially set your rules to have low thresholds with no action; then, once you profile your environment by analyzing the collected statistics, you can protect your servers by setting appropriate limits and configuring effective actions for attacks.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Configuring Application-Level DDoS Statistics Reporting on page 775](#)
- [IDP Application-Level DDoS Attack Overview on page 763](#)
- [Understanding IDP Policy Rules on page 707](#)
- [Understanding IDP Application-Level DDoS Rulebases on page 715](#)

## Example: Configuring Application-Level DDoS Statistics Reporting

This example shows how to configure application-level DDoS statistics reporting so you can analyze network statistical data. This data can be used to help you set the appropriate rule thresholds to enable protection against application-level DDoS attacks on your network.

### Requirements

Statistics reports are saved on the Routing Engine data storage device in the `/var/log/addos` directory. There must be at least 2 GB of free space to allow report logging.

### Overview

Application-level DDoS statistics reporting allows you to collect application information on connection, context and rates, and data records from application requests destined for your protected servers. With this information, you can determine trends to help you create more efficient rules for your environment.

The default interval to take a snapshot of the statistics is once every 1 minute and the range is from 1 through 60 minutes. In this example, you enable statistics reporting with an interval of 5 minutes. At 5-minute intervals, the application-level DDoS module takes a snapshot of current activity and places a report file in the `/var/log/addos` directory in comma-separated format.

### Configuration

#### Step-by-Step Procedure

To configure application-level DDoS statistics reporting:

1. Enable application-level DDoS statistics reporting.

```
[edit]
```

```
user@host# set security idp sensor-configuration application-ddos statistics interval 5
```

2. Disable application-level DDoS statistic reporting.

```
[edit]
```

```
user@host# delete security idp sensor-configuration application-ddos statistics
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
```

```
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security idp application-statistics** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Application-level DDoS Statistics Reporting on page 772

- [IDP Application-Level DDoS Attack Overview on page 763](#)
- [Understanding IDP Policy Rules on page 707](#)

# IDP Signature Database

- Understanding the IDP Signature Database on page 777
- Predefined IDP Policy Templates on page 778
- IDP Signature Databases on page 781
- Example: Adding a Detector Sensor Configuration (J-Web) on page 789
- Verifying the Signature Database on page 790

## Understanding the IDP Signature Database

---

The signature database is one of the major components of Intrusion Detection and Prevention (IDP). It contains definitions of different objects—such as attack objects, application signatures objects, and service objects—that are used in defining IDP policy rules. As a response to new vulnerabilities, Juniper Networks periodically provides a file containing attack database updates on the Juniper website. You can download this file to protect your network from new threats.

The IDP signature database is stored on the IDP enabled device and contains definitions of predefined attack objects and groups. These attack objects and groups are designed to detect known attack patterns and protocol anomalies within the network traffic. You can configure attack objects and groups as match conditions in IDP policy rules.



**NOTE:** You must install the IDP signature-database-update license key on your device for downloading and installing daily signature database updates provided by Juniper Networks. The IDP signature license key does not provide grace period support. For license details, see the *Junos OS Administration Guide for Security Devices*.

You can perform the following tasks to manage the IDP signature database:

- Update the signature database—Download the attack database updates available on the Juniper Networks website. New attacks are discovered daily, so it is important to keep your signature database up to date.
- Verify the signature database version—Each signature database has a different version number with the latest database having the highest number. You can use the CLI to display the signature database version number.

- Update the protocol detector engine—You can download the protocol detector engine updates along with downloading the signature database. The IDP protocol detector contains Application Layer protocol decoders. The detector is coupled with the IDP policy and is updated together. It is always needed at policy update time, even if there is no change in the detector.
- Schedule signature database updates—You can configure the IDP-enabled device to automatically update the signature database after a set interval.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- IDP Policies Overview on page 701
- Understanding IDP Policy Rulebases on page 713
- Understanding IDP Policy Rules on page 707
- Understanding Predefined IDP Policy Templates on page 778
- Understanding the IDP Signature Database Version on page 783
- Example: Defining Rules for an IDP IPS Rulebase on page 717
- Example: Adding a Detector Sensor Configuration (J-Web) on page 789

## Predefined IDP Policy Templates

- Understanding Predefined IDP Policy Templates on page 778
- Downloading and Using Predefined IDP Policy Templates (CLI Procedure) on page 780

### Understanding Predefined IDP Policy Templates

Juniper Networks provides predefined policy templates that you can use as a starting point for creating your own policies. Each template is set of rules of a specific rulebase type that you can copy and then update according to your requirements. These templates are available in the **templates.xml** file on a secured Juniper Networks website. To start using a template, you run a command from the CLI to download and copy this file to a **/var/db/scripts/commit** directory.

Each policy template contains rules that use the default actions associated with the attack objects. You should customize these templates to work on your network by selecting your own source and destination addresses and choosing IDP actions that reflect your security needs.

Table 95 on page 778 summarizes the predefined IDP policy templates provided by Juniper Networks.

**Table 95: Predefined IDP Policy Templates**

| Template Name | Description                                              |
|---------------|----------------------------------------------------------|
| DMZ Services  | Protects a typical demilitarized zone (DMZ) environment. |
| DNS Server    | Protects Domain Name System (DNS) services.              |



Table 95: Predefined IDP Policy Templates (*continued*)

| Template Name          | Description                                                                                                                                                                       |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>File Server</b>     | Protects file sharing services, such as Network File System (NFS), FTP, and others.                                                                                               |
| <b>Getting Started</b> | Contains very open rules. Useful in controlled lab environments, but should not be deployed on heavy traffic live networks.                                                       |
| <b>IDP Default</b>     | Contains a good blend of security and performance.                                                                                                                                |
| <b>Recommended</b>     | Contains only the attack objects tagged as <i>recommended</i> by Juniper Networks. All rules have their Actions column set to take the recommended action for each attack object. |
| <b>Web Server</b>      | Protects HTTP servers from remote attacks.                                                                                                                                        |

To use predefined policy templates:

1. Download the policy templates from the Juniper Networks website.
2. Install the policy templates.
3. Enable the **templates.xml** script file. Commit scripts in the `/var/db/scripts/commit` directory are ignored if they are not enabled.
4. Choose a policy template that is appropriate for you and customize it if you need to.
5. Activate the policy that you want to run on the system. Activating the policy might take a few minutes. Even after a commit complete message is displayed in the CLI, the system might continue to compile and push the policy to the dataplane.



**NOTE:** Occasionally, the compilation process might fail for a policy. In this case, the active policy showing in your configuration might not match the actual policy running on your device. Run the `show security idp status` command to verify the running policy. Additionally, you can view the IDP log files to verify the policy load and compilation status (see “Verifying the Signature Database” on page 790).

6. Delete or deactivate the commit script file. By deleting the commit script file, you avoid the risk of overwriting modifications to the template when you commit the configuration. Deactivating the statement adds an inactive tag to the statement, effectively commenting out the statement from the configuration. Statements marked inactive do not take effect when you issue the **commit** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding the IDP Signature Database on page 777](#)
- [Downloading and Using Predefined IDP Policy Templates \(CLI Procedure\) on page 780](#)

## Downloading and Using Predefined IDP Policy Templates (CLI Procedure)

Before you begin, configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

To download and use a predefined policy template:

1. Download the script file **templates.xml** to the **/var/db/idpd/sec-download/sub-download** directory. This script file contains predefined IDP policy templates.

```
user@host> request security idp security-package download policy-templates
```

2. Copy the **templates.xml** file to the **/var/db/scripts/commit** directory and rename it to **templates.xsl**.

```
user@host> request security idp security-package install policy-templates
```

3. Enable the **templates.xsl** scripts file. At commit time, the Junos OS management process (mgd) looks in the **/var/db/scripts/commit** directory for scripts and runs the script against the candidate configuration database to ensure the configuration conforms to the rules dictated by the scripts.

```
user@host# set system scripts commit file templates.xsl
```

4. Commit the configuration. Committing the configuration saves the downloaded templates to the Junos OS configuration database and makes them available in the CLI at the **[edit security idp idp-policy]** hierarchy level.

5. Display the list of downloaded templates.

```
user@host#set security idp active-policy ?
```

```
Possible completions:
<active policy> Set active policy
  DMZ_Services
  DNS_Service
  File_Server
  Getting_Started
  IDP_Default
  Recommended
  Web_Server
```

6. Activate the predefined policy. The following statement specifies the *Recommended* predefined IDP policy as the active policy:

```
user@host# set security idp active-policy Recommended
```

7. Delete or deactivate the commit script file. By deleting the commit script file, you avoid the risk of overwriting modifications to the template when you commit the configuration. Run one of the following commands:

```
user@host# delete system scripts commit file templates.xsl
user@host# deactivate system scripts commit file templates.xsl
```

8. If you are finished configuring the device, commit the configuration.
9. You can verify the configuration by using the **show security idp status** command. For more information, see the *Junos OS CLI Reference*.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Predefined IDP Policy Templates on page 778](#)
- [Example: Defining Rules for an IDP IPS Rulebase on page 717](#)
- [Example: Defining Rules for an IDP Exempt Rulebase on page 722](#)

## IDP Signature Databases

---

- [Understanding Predefined IDP Attack Objects and Object Groups on page 781](#)
- [Understanding the IDP Signature Database Version on page 783](#)
- [Updating the IDP Signature Database Overview on page 783](#)
- [Updating the IDP Signature Database Manually Overview on page 784](#)
- [Example: Updating the IDP Signature Database Manually on page 785](#)
- [Example: Updating the Signature Database Automatically on page 788](#)

### Understanding Predefined IDP Attack Objects and Object Groups

The security package for Intrusion Detection and Prevention (IDP) contains a database of predefined IDP attack objects and IDP attack object groups that you can use in IDP policies to match traffic against known and unknown attacks. Juniper Networks updates the predefined attack objects and groups on a regular basis with newly discovered attack patterns.

Updates to the attack object database can include:

- New descriptions or severities for existing attack objects
- New attack objects
- Deletion of obsolete attack objects

This topic includes the following sections:

- [Predefined Attack Objects on page 781](#)
- [Predefined Attack Object Groups on page 782](#)

#### Predefined Attack Objects

---

Predefined attack objects are listed in an alphabetical order. These attack objects have unique names that help you identify the attack. The first part of the name indicates the group to which the attack object belongs. For example:

- **FTP:USER:ROOT**—Belongs to the **FTP:USER** group. It detects attempts to log in to an FTP server using the **root** account.

- **HTTP:HOTMAIL:FILE-UPLOAD**—Belongs to the **HTTP:HOTMAIL** group. It detects files attached to e-mails sent via the Web-based e-mail service **Hotmail**.

### Predefined Attack Object Groups

The predefined attack groups list displays the attack objects in the categories described below. A set of recommended attack objects that Juniper Networks considers to be serious threats are also available in this list. The recommended attack objects are organized into the following categories:

**Table 96: Predefined Attack Object Groups**

| Attack Object Group | Description                                                                                                                                                                                      |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Attack Type         | Groups attack objects by type (anomaly or signature). Within each type, attack objects are grouped by severity.                                                                                  |
| Category            | Groups attack objects by predefined categories. Within each category, attack objects are grouped by severity.                                                                                    |
| Operating System    | Groups attack objects by the operating system to which they apply: BSD, Linux, Solaris, or Windows. Within each operating system, attack objects are grouped by services and severity.           |
| Severity            | Groups attack objects by the severity assigned to the attack. IDP has five severity levels: Critical, Major, Minor, Warning, Info. Within each severity, attack objects are grouped by category. |
| Web Services        | Groups attack objects by common Web services. These services are grouped by severity levels—Warning, Critical, Major, Minor, Info.                                                               |
| Miscellaneous       | Groups attack objects by performance level. Attack objects affecting IDP performance over a certain level are grouped under this category.                                                       |
| Response            | Groups attack objects in traffic flowing in the server to client direction.                                                                                                                      |

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding the IDP Signature Database on page 777](#)
- [Updating the IDP Signature Database Overview on page 783](#)
- [Updating the IDP Signature Database Manually Overview on page 784](#)
- [Example: Updating the IDP Signature Database Manually on page 785](#)
- [Example: Updating the Signature Database Automatically on page 788](#)
- [Example: Defining Rules for an IDP IPS Rulebase on page 717](#)
- [Example: Defining Rules for an IDP Exempt Rulebase on page 722](#)

## Understanding the IDP Signature Database Version

New attack objects are added to the signature database server frequently; downloading these updates and installing them on your managed devices regularly ensures that your network is effectively protected against the latest threats. As new attack objects are added to the signature database server, the version number of the database is updated with the latest database version number. Each signature database has a different version number with the latest database having the highest number.

When updating the signature database, the signature database update client connects to the Juniper Networks website and obtains the update using an HTTPS connection. This update—difference between the existing signature database and latest signature database—is calculated based on the version number that is assigned to each signature database. After you download the updates, the updated information is merged with the existing signature database and the version number is set to that of the latest signature database.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Predefined IDP Attack Objects and Object Groups on page 781](#)
- [Updating the IDP Signature Database Overview on page 783](#)
- [Updating the IDP Signature Database Manually Overview on page 784](#)
- [Example: Updating the IDP Signature Database Manually on page 785](#)
- [Example: Updating the Signature Database Automatically on page 788](#)

## Updating the IDP Signature Database Overview

Juniper Networks regularly updates the predefined attack database and makes it available on the Juniper Networks website. This database includes attack object groups that you can use in Intrusion Detection and Prevention (IDP) policies to match traffic against known attacks. Although you cannot create, edit, or delete predefined attack objects, you can use the CLI to update the list of attack objects that you can use in IDP policies.

To update the signature database, you download a security package from the Juniper Networks website. The security package consists of the following IDP components:

- Attack objects
- Attack object groups
- Application objects
- Updates to the IDP Detector Engine
- IDP Policy templates (Policy templates are downloaded independently. See “Understanding Predefined IDP Policy Templates” on page 778.)

By default, when you download the security package, you download the following components into a Staging folder in your device: the latest version of the complete attack object groups table, application objects table, and the updates to the IDP Detector Engine.

Because the attack objects table is typically of a large size, by default the system downloads only updates to the attack objects table. However, you can download the complete attack objects table by using the **full-update** configuration option.

After downloading the security package, you must install the package to update the security database with the newly downloaded updates from the Staging folder in your device.

After installing a security package, when you commit the configuration, all policies are checked for their syntax (not only the active policy). This checking is the same as a commit check. If an attack configured in any of the existing policies is removed from the new signature database that you download, the commit check fails. When you update the IDP signature database, attacks configured in policies are not updated automatically. For example, suppose you configure a policy to include an attack **FTP:USER:ROOT** that is available in the signature database version **1200** on your system. Then, you download signature database version **1201**, which no longer includes the attack **FTP:USER:ROOT**. Because an attack configured in your policy is missing from the newly downloaded database, the commit check in the CLI fails. To successfully commit your configuration, you must remove the attack (**FTP:USER:ROOT**) from your policy configuration.



**CAUTION:** IDP signature updates might fail if a new IDP policy load fails for any reason. When a new IDP policy load fails, the last known good IDP policy is loaded. Once the issue with the new policy load is resolved, and the new valid policy is active, signature updates will work properly.

---

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding the IDP Signature Database on page 777](#)
- [Understanding Predefined IDP Attack Objects and Object Groups on page 781](#)
- [Understanding the IDP Signature Database Version on page 783](#)
- [Updating the IDP Signature Database Manually Overview on page 784](#)
- [Example: Updating the IDP Signature Database Manually on page 785](#)
- [Example: Updating the Signature Database Automatically on page 788](#)

## Updating the IDP Signature Database Manually Overview

Juniper Networks regularly updates the predefined attack database and makes it available on the Juniper Networks website. This database includes attack object groups that you can use in Intrusion Detection and Prevention (IDP) policies to match traffic against known attacks. Although you cannot create, edit, or delete predefined attack objects, you can use the CLI to update the list of attack objects that you can use in IDP policies. After downloading the security package, you must install the package to update the security database with the newly downloaded updates from the Staging folder in your device.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding the IDP Signature Database on page 777](#)
  - [Understanding Predefined IDP Attack Objects and Object Groups on page 781](#)
  - [Understanding the IDP Signature Database Version on page 783](#)
  - [Updating the IDP Signature Database Overview on page 783](#)
  - [Example: Updating the IDP Signature Database Manually on page 785](#)
  - [Example: Updating the Signature Database Automatically on page 788](#)

## Example: Updating the IDP Signature Database Manually

This example shows how to update the IDP signature database manually.

- [Requirements on page 785](#)
- [Overview on page 785](#)
- [Configuration on page 785](#)
- [Verification on page 788](#)

### Requirements

---

Before you begin, configure network interfaces. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).

### Overview

---

Juniper Networks regularly updates the predefined attack database and makes it available as a security package on the Juniper Networks website. This database includes attack object and attack object groups that you can use in IDP policies to match traffic against known attacks.

In this example, you download the security package with the complete table of attack objects and attack object groups. Once the installation is completed, the attack objects and attack object groups are available in the CLI under the predefined-attack-groups and predefined-attacks configuration statements at the [edit security idp idp-policy] hierarchy level. You create a policy and specify the new policy as the active policy. You also download only the updates that Juniper Networks has recently uploaded and then update the attack database, the running policy, and the detector with these new updates.

### Configuration

---

- CLI Quick Configuration**
- CLI quick configuration is not available for this example because manual intervention is required during the configuration.

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To manually download and update the signature database:

1. Specify the URL for the security package.

```
[edit]
user@host# set security idp security-package url
https://services.netscreen.com/cgi-bin/index.cgi
```



**NOTE:** By default it will take URL as `https://services.netscreen.com/cgi-bin/index.cgi`.

2. Commit the configuration.

```
[edit]
user@host# commit
```

3. Switch to operational mode.

```
[edit]
user@host# exit
```

4. Download the security package.

```
user@host>request security idp security-package download full-update
```

5. Check the security package download status.

```
user@host>request security idp security-package download status
```

6. Update the attack database using install command.

```
user@host>request security idp security-package install
```

7. Check the attack database update status with the following command. The command output displays information about the downloaded and installed versions of attack database versions.

```
user@host>request security idp security-package install status
```

8. Switch to configuration mode.

```
user@host>configure
```

9. Create an IDP policy.

```
[edit ]
user@host#edit security idp idp-policy policy1
```

10. Associate attack objects or attack object groups with the policy.

```
[edit security idp idp-policy policy1]
user@host#set rulebase-ips rule rule1 match attacks predefined-attack-groups
"Response_Critical"
```

11. Set action.



```
[edit security idp idp-policy policy1]
user@host#set rulebase-ips rule rule1 then action no-action
```

12. Activate the policy.

```
[edit]
user@host#set security idp active-policy policy1
```

13. Commit the configuration.

```
[edit]
user@host# commit
```

14. After a week, download only the updates that Juniper Networks has recently uploaded.

```
user@host>request security idp security-package download
```

15. Check the security package download status.

```
user@host>request security idp security-package download status
```

16. Update the attack database, the active policy, and the detector with the new changes.

```
user@host>request security idp security-package install
```

17. Check the attack database, the active policy and the detector using install status.

```
user@host>request security idp security-package install status
```

**Results** From configuration mode, confirm your configuration by entering the **show security idp** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security idp
idp-policy policy1 {
  rulebase-ips {
    rule rule1 {
      match {
        attacks {
          predefined-attack-groups Response_Critical;
        }
      }
      then {
        action {
          no-action;
        }
      }
    }
  }
}
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

---

To confirm that the configuration is working properly, perform this task:

- Verifying the IDP Signature Database Manually on page 788

#### *Verifying the IDP Signature Database Manually*

**Purpose** Display the IDP signature database manually.

**Action** From operational mode, enter the **show security idp** command.

#### **Related Documentation**

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Updating the IDP Signature Database Manually Overview on page 784
- Example: Updating the Signature Database Automatically on page 788
- Understanding the IDP Signature Database on page 777

## Example: Updating the Signature Database Automatically

This example shows how to download signature database updates automatically.

- Requirements on page 788
- Overview on page 788
- Configuration on page 788
- Verification on page 789

### Requirements

---

Before you begin, configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

### Overview

---

Juniper Networks regularly updates the predefined attack database and makes it available as a security package on the Juniper Networks website. This database includes attack object and attack object groups that you can use in IDP policies to match traffic against known attacks. You can configure your device to download the signature database updates automatically at a specified interval.

In this example, you download the security package with the complete table of attack objects and attack object groups every 48 hours, starting at 11:59 p.m. on December 10. You also enable an automatic download and update of the security package.

### Configuration

---

#### **Step-by-Step Procedure**

To download and update the predefined attack objects:

1. Specify the URL for the security package.

[edit]

```
user@host# set security idp security-package url  
https://services.netscreen.com/cgi-bin/index.cgi
```



**NOTE:** By default it will take URL as `https://services.netscreen.com/cgi-bin/index.cgi`.

- Specify the time, interval and download timeout value for the download.

```
[edit]
user@host# set security idp security-package automatic interval 48
download-timeout 3 start-time 2009-12-10.23:59:00
```

- Enable the automatic download and update of the security package.

```
[edit]
user@host# set security idp security-package automatic enable
```

- If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security idp** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Updating the IDP Signature Database Manually Overview on page 784](#)
- [Understanding the IDP Signature Database on page 777](#)

## Example: Adding a Detector Sensor Configuration (J-Web)

In this example, you add a detector sensor configuration for File Transfer Protocol (FTP) with Tunable Value 1000, which can be further tuned using the Basic and Advanced configuration tabs on the IDP Sensor configuration page.

To add a detector sensor configuration:

- Select **Configure>Security>IDP>Sensor**.
- Select the **Detector** tab.
- Click **Add**.
- In the Protocol list, select **FTP**.
- In the Tunable Name list, select **sc\_ftp\_flags**.
- In the Tunable Value box, type **1000**.
- Click **OK** to save the configuration.



**NOTE:** For more information about how to configure this feature using the J-Web Configure menu, navigate to the **Configure>Security>IDP>Sensor** page in the J-Web user interface and click **Help**.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [IDP Policies Overview on page 701](#)
- [Understanding the IDP Signature Database on page 777](#)
- [Understanding IDP Application Identification on page 795](#)
- [Example: Configuring IDP Policies for Application Identification on page 798](#)

## Verifying the Signature Database

- [Verifying the IDP Policy Compilation and Load Status on page 790](#)
- [Verifying the IDP Signature Database Version on page 792](#)

### Verifying the IDP Policy Compilation and Load Status

**Purpose** Display the IDP log files to verify the IDP policy load and compilation status. When activating an IDP policy, you can view the IDP logs and verify if the policy is loaded and compiled successfully.

**Action** To track the load and compilation progress of an IDP policy, configure either one or both of the following in the CLI:

- You can configure a log file, which will be located in `/var/log/`, and set trace option flags to record these operations:

```
user@host# set security idp traceoptions file idpd
user@host# set security idp traceoptions flag all
```

- You can configure your device to log system log messages to a file in the `/var/log` directory:

```
user@host# set system syslog file messages any any
```

After committing the configuration in the CLI, enter either of the following commands from the shell prompt in the UNIX-level shell:

### Sample Output

```
user@host> start shell
user@host% tail -f /var/log/idpd
Aug 3 15:46:42 chiron clear-log[2655]: logfile cleared
Aug 3 15:47:12 idpd_config_read: called: check: 0
Aug 3 15:47:12 idpd commit in progres ...
Aug 3 15:47:13 Entering enable processing.
Aug 3 15:47:13 Enable value (default)
Aug 3 15:47:13 IDP processing default.
Aug 3 15:47:13 idp config knob set to (2)
Aug 3 15:47:13 Warning: active policy configured but no application package
```

```

installed, attack may not be detected!
Aug 3 15:47:13 idpd_need_policy_compile:480 Active policy path
/var/db/idpd/sets/idpengine.set
Aug 3 15:47:13 Active Policy (idpengine) rule base configuration is changed so
need to recompile active policy
Aug 3 15:47:13 Compiling policy idpengine...
Aug 3 15:47:13 Apply policy configuration, policy ops bitmask = 41
Aug 3 15:47:13 Starting policy(idpengine) compile with compress dfa...
Aug 3 15:47:35 policy compilation memory estimate: 82040
Aug 3 15:47:35 ...Passed
Aug 3 15:47:35 Starting policy package...
Aug 3 15:47:36 ...Policy Packaging Passed
Aug 3 15:47:36 [get_secupdate_cb_status] state = 0x1
Aug 3 15:47:36 idpd_policy_apply_config idpd_policy_set_config()
Aug 3 15:47:36 Reading sensor config...
Aug 3 15:47:36 sensor/idp node does not exist, apply defaults
Aug 3 15:47:36 sensor conf saved
Aug 3 15:47:36 idpd_dev_add_ipc_connection called...
Aug 3 15:47:36 idpd_dev_add_ipc_connection: done.
Aug 3 15:47:36 idpd_policy_apply_config: IDP state (2) being set
Aug 3 15:47:36 idpd_comm_server_get_event:545: evGetNext got event.
Aug 3 15:47:36 idpd_comm_server_get_event:553: evDispatch OK
Aug 3 15:47:36 Apply policy configuration, policy ops bitmask = 4
Aug 3 15:47:36 Starting policy load...
Aug 3 15:47:36 Loading policy(/var/db/idpd/bins/idpengine.bin.gz.v +
/var/db/idpd/sec-repository/installed-detector/libidp-detector.so.tgz.v +
/var/db/idpd/bins/compressed_ai.bin)...
Aug 3 15:47:36 idpd_dev_add_ipc_connection called...
Aug 3 15:47:36 idpd_dev_add_ipc_connection: done.
Aug 3 15:47:37 idpd_policy_load: creating temp tar directory
'/var/db/idpd//bins/52b58e5'
Aug 3 15:47:37 sc_policy_unpack_tgz: running addver cmd '/usr/bin/addver -r
/var/db/idpd/sec-repository/installed-detector/libidp-detector.so.tgz.v
/var/db/idpd//bins/52b58e5/___temp.tgz > /var/log/idpd.addver'
Aug 3 15:47:38 sc_policy_unpack_tgz: running tar cmd '/usr/bin/tar -C
/var/db/idpd//bins/52b58e5 -xzf /var/db/idpd//bins/52b58e5/___temp.tgz'
Aug 3 15:47:40 idpd_policy_load: running cp cmd 'cp
/var/db/idpd//bins/52b58e5/detector4.so /var/db/idpd//bins/detector.so'
Aug 3 15:47:43 idpd_policy_load: running chmod cmd 'chmod 755
/var/db/idpd//bins/detector.so'
Aug 3 15:47:44 idpd_policy_load: running rm cmd 'rm -fr
/var/db/idpd//bins/52b58e5'
Aug 3 15:47:45 idpd_policy_load: detector version: 10.3.160100209
Aug 3 15:47:45 idpd_comm_server_get_event:545: evGetNext got event.
Aug 3 15:47:45 idpd_comm_server_get_event:553: evDispatch OK
Aug 3 15:47:45 idp_policy_loader_command: sc_klibs_subs_policy_pre_compile()
returned 0 (EOK)
Aug 3 15:47:45 idpd_policy_load: IDP_LOADER_POLICY_PRE_COMPILE returned EAGAIN,
retrying... after (5) secs
Aug 3 15:47:50 idpd_comm_server_get_event:545: evGetNext got event.
Aug 3 15:47:50 idpd_comm_server_get_event:553: evDispatch OK
Aug 3 15:47:50 idp_policy_loader_command: sc_klibs_subs_policy_pre_compile()
returned 0 (EOK)
Aug 3 15:47:50 idpd_policy_load: idp policy parser pre compile succeeded, after
(1) retries
Aug 3 15:47:50 idpd_policy_load: policy parser compile subs s0 name
/var/db/idpd/bins/idpengine.bin.gz.v.1 buf 0x0 size 0zones 0xee34c7 z_size 136
detector /var/db/idpd//bins/detector.so ai_buf 0x0 ai_size 0 ai
/var/db/idpd/bins/compressed_ai.bin
Aug 3 15:47:50 idpd_comm_server_get_event:545: evGetNext got event.
Aug 3 15:47:50 idpd_comm_server_get_event:553: evDispatch OK

```

```

Aug 3 15:47:50 idpd_comm_server_get_event:545: evGetNext got event.
Aug 3 15:47:50 idpd_comm_server_get_event:553: evDispatch OK
Aug 3 15:47:50 idpd_policy_load: idp policy parser compile succeeded
Aug 3 15:47:50 idpd_comm_server_get_event:545: evGetNext got event.
Aug 3 15:47:50 idpd_comm_server_get_event:553: evDispatch OK
Aug 3 15:47:50 idpd_policy_load: idp policy pre-install succeeded
Aug 3 15:47:50 idpd_comm_server_get_event:545: evGetNext got event.
Aug 3 15:47:50 idpd_comm_server_get_event:553: evDispatch OK
Aug 3 15:47:50 idpd_comm_server_get_event:545: evGetNext got event.
Aug 3 15:47:50 idpd_comm_server_get_event:553: evDispatch OK
Aug 3 15:47:50 idpd_policy_load: idp policy install succeeded
Aug 3 15:47:50 idpd_comm_server_get_event:545: evGetNext got event.
Aug 3 15:47:50 idpd_comm_server_get_event:553: evDispatch OK
Aug 3 15:47:50 idpd_policy_load: idp policy post-install succeeded
Aug 3 15:47:51 IDP policy[/var/db/idpd/bins/idpengine.bin.gz.v] and
detector[/var/db/idpd/sec-repository/installed-detector/libidp-detector.so.tgz.v]
loaded successfully.
Aug 3 15:47:51 Applying sensor configuration
Aug 3 15:47:51 idpd_dev_add_ipc_connection called...
Aug 3 15:47:51 idpd_dev_add_ipc_connection: done.
Aug 3 15:47:51 idpd_comm_server_get_event:545: evGetNext got event.
Aug 3 15:47:51 idpd_comm_server_get_event:553: evDispatch OK
Aug 3 15:47:51 idpd_comm_server_get_event:545: evGetNext got event.
Aug 3 15:47:51 idpd_comm_server_get_event:553: evDispatch OK
Aug 3 15:47:51
...idpd commit end
Aug 3 15:47:51 Returning from commit mode, status = 0.
Aug 3 15:47:51 [get_secupdate_cb_status] state = 0x1
Aug 3 15:47:51 Got signal SIGCHLD....

```

## Sample Output

```

user@host> start shell
user@host% tail -f /var/log/messages
Aug 3 15:46:56 chiron mgd[2444]: UI_COMMIT_PROGRESS: Commit operation in
progress: no commit script changes
Aug 3 15:46:56 chiron mgd[2444]: UI_COMMIT_PROGRESS: Commit operation in
progress: no transient commit script changes
Aug 3 15:46:56 chiron mgd[2444]: UI_COMMIT_PROGRESS: Commit operation in
progress: finished loading commit script changes
Aug 3 15:46:56 chiron mgd[2444]: UI_COMMIT_PROGRESS: Commit operation in
progress: exporting juniper.conf
.....
Aug 3 15:47:51 chiron idpd[2678]: IDP_POLICY_LOAD_SUCCEEDED: IDP
policy[/var/db/idpd/bins/idpengine.bin.gz.v] and
detector[/var/db/idpd/sec-repository/installed-detector/libidp-detector.so.tgz.v]
loaded successfully(Regular load).
Aug 3 15:47:51 chiron idpd[2678]: IDP_COMMIT_COMPLETED: IDP policy commit is
complete.
.....
Aug 3 15:47:51 chiron chiron sc_set_flow_max_sessions: max sessions set 16384

```

**Meaning** Displays log messages showing the procedures that run in the background after you commit the **set security idp active-policy** command. This sample output shows that the policy compilation, sensor configuration, and policy load are successful.

## Verifying the IDP Signature Database Version

**Purpose** Display the signature database version.

**Action** From the operational mode in the CLI, enter **show security idp security-package-version**.

### Sample Output

```
user@host> show security idp security-package-version
Attack database version:31(Wed Apr 16 15:53:46 2008)
  Detector version :9.1.140080400
  Policy template version :N/A
```

**Meaning** The output displays the version numbers for the signature database, protocol detector, and the policy template on the IDP-enabled device. Verify the following information:

- **Attack database version**—On April 16, 2008, the version of the signature database active on the device is **31**.
- **Detector version**—Displays the version number of the IDP protocol detector currently running on the device.
- **Policy template version**—Displays the version of the policy template that is installed in the `/var/db/scripts/commit` directory when you run the **request security idp security-package install policy-templates** configuration statement in the CLI.

For a complete description of **show security idp security-package-version** output, see the *Junos OS CLI Reference*.





# IDP Application Identification

- IDP Application Identification on page 795
- IDP Application Identification for Nested Applications on page 800
- IDP Application System Cache on page 801
- IDP Memory and Session Limits on page 804
- Verifying IDP Counters for Application Identification Processes on page 806

## IDP Application Identification

---

- Understanding IDP Application Identification on page 795
- Understanding IDP Service and Application Bindings by Attack Objects on page 796
- Example: Configuring IDP Policies for Application Identification on page 798
- Disabling Application Identification for an IDP Policy (CLI Procedure) on page 799

## Understanding IDP Application Identification

Juniper Networks provides predefined application signatures that detect Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) applications running on nonstandard ports. Identifying these applications allows Intrusion Detection and Prevention (IDP) to apply appropriate attack objects to applications running on nonstandard ports. It also improves performance by narrowing the scope of attack signatures for applications without decoders.

The IDP sensor monitors the network and detects suspicious and anomalous network traffic based on specific rules defined in IDP rulebases. It applies attack objects to traffic based on protocols or applications. Application signatures enable the sensor to identify known and unknown applications running on nonstandard ports and to apply the correct attack objects.

Application signatures are available as part of the security package provided by Juniper Networks. You download predefined application signatures along with the security package updates. You cannot create application signatures. For information on downloading the security package, see “Updating the IDP Signature Database Manually Overview” on page 784.

The application signatures identify an application by matching patterns in the first packet of a session. The IDP sensor matches patterns for both client-to-server and server-to-client sessions.

Application identification is enabled by default and is automatically turned on when you configure the default application in the IDP policy. However, when you specify an application in the policy rule, application identification is disabled and attack objects are applied based on the specified application. This specific application configuration overwrites the automatic identification process. For instructions on specifying applications in policy rules, see “Example: Configuring IDP Applications and Services” on page 731.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding IDP Service and Application Bindings by Attack Objects on page 796](#)
- [Understanding IDP Application Identification for Nested Applications on page 800](#)
- [Understanding the IDP Application System Cache on page 801](#)
- [Understanding Memory and Session Limit Settings for IDP Application Identification on page 804](#)
- [Example: Configuring IDP Policies for Application Identification on page 798](#)
- [Disabling Application Identification for an IDP Policy \(CLI Procedure\) on page 799](#)

## Understanding IDP Service and Application Bindings by Attack Objects

Attack objects can bind to applications and services in different ways:

- Attack objects can bind to an application implicitly and not have a service definition. They bind to an application based on the name of a context or anomaly.
- Attack objects can bind to a service using a service name.
- Attack objects can bind to a service using TCP or UDP ports, ICMP types or codes or RPC program numbers.

Whether the specified application or service binding applies or not depends on the complete attack object definition as well as the IDP policy configuration:

- If you specify an application in an attack object definition, the service field is ignored. The attack object binds to the application instead of the specified service. However, if you specify a service and no application in the attack object definition, the attack object binds to the service. Table 97 on page 796 summarizes the behavior of application and service bindings with application identification.

**Table 97: Applications and Services with Application Identification**

| Attack Object Fields       | Binding Behavior                                                                   | Application Identification |
|----------------------------|------------------------------------------------------------------------------------|----------------------------|
| <b>:application (http)</b> | <ul style="list-style-type: none"> <li>• Binds to the application HTTP.</li> </ul> | Enabled                    |
| <b>:service (smtp)</b>     | <ul style="list-style-type: none"> <li>• The service field is ignored.</li> </ul>  | Enabled                    |
| <b>:service (http)</b>     | Binds to the application HTTP.                                                     | Enabled                    |

Table 97: Applications and Services with Application Identification (*continued*)

| Attack Object Fields | Binding Behavior      | Application Identification |
|----------------------|-----------------------|----------------------------|
| :service (tcp/80)    | Binds to TCP port 80. | Disabled                   |

For example, in the following attack object definition, the attack object binds to the application **HTTP**, the application identification is enabled, and the service field **SMTP** is ignored.

```

: ("http-test"
 :application ("http")
 :service ("smtp")
 :rectype (signature)
 :signature (
 :pattern (".*TERM=xterm; export TERM=xterm; exec bash - i\x0a\x.*")
 :type (stream)
 )
 :type (attack-ip)
 )

```

- If an attack object is based on service specific contexts (for example, **http-url**) and anomalies (for example, **tftp\_file\_name\_too\_long**), both application and service fields are ignored. Service contexts and anomalies imply application; thus when you specify these in the attack object, application identification is applied.
- If you configure a specific application in a policy, you overwrite the application binding specified in an attack object. Table 98 on page 797 summarizes the binding with the application configuration in the IDP policy.

Table 98: Application Configuration in an IDP Policy

| Application Type in the Policy | Binding Behavior                                                                | Application Identification                                                                                                                            |
|--------------------------------|---------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Default</b>                 | Binds to the application or service configured in the attack object definition. | <ul style="list-style-type: none"> <li>• Enabled for application-based attack objects</li> <li>• Disabled for service-based attack objects</li> </ul> |
| <b>Specific application</b>    | Binds to the application specified in the attack object definition.             | Disabled                                                                                                                                              |
| <b>Any</b>                     | Binds to all applications.                                                      | Disabled                                                                                                                                              |

- If you specify an application in an IDP policy, the application type configured in the attack object definition and in the IDP policy must match. The policy rule cannot specify two different applications (one in the attack object and the other in the policy).

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding IDP Application Identification on page 795
- IDP Policies Overview on page 701

- Understanding the IDP Signature Database on page 777
- Example: Configuring IDP Policies for Application Identification on page 798
- Disabling Application Identification for an IDP Policy (CLI Procedure) on page 799

## Example: Configuring IDP Policies for Application Identification

This example shows how to configure the IDP policies for application identification.

- Requirements on page 798
- Overview on page 798
- Configuration on page 798
- Verification on page 799

### Requirements

---

Before you begin:

- Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Download the application package.

### Overview

---

In this example, you create an IDP policy ABC and define rule 123 in the IPS rulebase. You specify default as the application type in an IDP policy rule. If you specify an application instead of default the application identification feature will be disabled for this rule and IDP will match the traffic with the specified application type. The applications defined under application-identification cannot be referenced directly at this time.

### Configuration

---

#### Step-by-Step Procedure

To configure IDP policies for application identification:

1. Create an IDP policy.  

```
[edit]  
user@host# set security idp idp-policy ABC
```
2. Specify the application type.  

```
[edit]  
user@host# set security idp idp-policy ABC rulebase-ips rule 123 match application  
default
```
3. Specify an action to take when the match condition is met.  

```
[edit]  
user@host# set security idp idp-policy ABC rulebase-ips rule 123 then action  
no-action
```
4. If you are done configuring the device, commit the configuration.  

```
[edit]  
user@host# commit
```

## Verification

To verify the configuration is working properly, enter the **show security idp** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding the Junos OS Application Identification Application Package on page 1104](#)
- [Understanding IDP Application Identification on page 795](#)
- [Disabling Application Identification for an IDP Policy \(CLI Procedure\) on page 799](#)
- [Example: Setting Memory and Session Limits for IDP Application Identification Services on page 805](#)
- [Verifying Application System Cache Statistics on page 1124](#)

## Disabling Application Identification for an IDP Policy (CLI Procedure)

Application identification is enabled by default. You can disable application identification with the CLI.

To disable and application identification:

1. Specify the **disable** configuration option.

```
user@host# set security idp sensor-configuration application-identification disable
```

2. If you want to reenable application identification, delete the configuration statement that specifies disabling of application identification.

```
user@host# delete security idp sensor-configuration application-identification disable
```

3. If you are finished configuring the device, commit the configuration.
4. From configuration mode in the CLI, enter the **show security idp** command to verify the configuration. For more information, see the *Junos OS CLI Reference*.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding the IDP Signature Database on page 777](#)
- [Understanding IDP Application Identification on page 795](#)
- [Example: Configuring IDP Policies for Application Identification on page 798](#)
- [Example: Setting Memory and Session Limits for IDP Application Identification Services on page 805](#)
- [Verifying Application System Cache Statistics on page 1124](#)

## IDP Application Identification for Nested Applications

- Understanding IDP Application Identification for Nested Applications on page 800
- Activating IDP Application Identification for Nested Applications (CLI Procedure) on page 800
- Example: Adding IDP Application Information to Attack Logging for Nested Applications (CLI) on page 800

### Understanding IDP Application Identification for Nested Applications

With the greater use of application protocol encapsulation, the need arises to support the identification of multiple different applications running on the same Layer 7 protocols. For example, applications such as Facebook and Yahoo Messenger can both run over HTTP, but there is a need to identify them as two different applications running on the same Layer 7 protocol. In order to do this, the current application identification layer is split into two layers: Layer 7 applications and Layer 7 protocols.

Included predefined application signatures have been created to detect the Layer 7 applications whereas the existing Layer 7 protocol signatures still function in the same manner. These predefined application signatures can be used in attack objects.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding IDP Application Identification on page 795
- Activating IDP Application Identification for Nested Applications (CLI Procedure) on page 800
- Example: Adding IDP Application Information to Attack Logging for Nested Applications (CLI) on page 800

### Activating IDP Application Identification for Nested Applications (CLI Procedure)

Application identification for nested applications is turned on by default. You can manually turn this identification off by using the CLI.

```
user@host# edit security idp sensor-configuration application-identification
no-nested-application-identification
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding IDP Application Identification for Nested Applications on page 800
- Example: Adding IDP Application Information to Attack Logging for Nested Applications (CLI) on page 800

### Example: Adding IDP Application Information to Attack Logging for Nested Applications (CLI)

Nested application information added to IDP attack logging after “service” and before “rule” provides information on detected Layer 7 applications. In the following example, “Facebook” appears in the log file as nested application information.

```
Aug 29 20:46:32 4.0.0.254 RT_IDP: IDP_ATTACK_LOG_EVENT: IDP: at 1251603992, SIG
Attack log <4.0.0.1:33000->5.0.0.1:210> for TCP protocol and service SERVICE_IDP
application FACEBOOK by rule 1 of rulebase IPS in policy idpengine. attack: repeat=0,
action=NONE, severity=MEDIUM, name=http-url-attack-test, NAT
<8.11.163.220:0->0.0.0.0:0>, time-elapsed=0, inbytes=0, outbytes=0, inpackets=0,
outpackets=0, intf:untrust:ge-0/0/1.0->trust:ge-0/0/0.0, and misc-message -
```



**NOTE:** For further information on IDP logging, refer to “Understanding IDP Logging” on page 827.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding IDP Application Identification for Nested Applications on page 800
- Activating IDP Application Identification for Nested Applications (CLI Procedure) on page 800

## IDP Application System Cache

- Understanding the IDP Application System Cache on page 801
- Understanding IDP Application System Cache Information for Nested Application Identification on page 802
- Deactivating IDP Application System Cache Information for Nested Application Identification (CLI Procedure) on page 802
- Verifying IDP Application System Cache Statistics on page 803

### Understanding the IDP Application System Cache

Application system cache (ASC) saves the mapping between an application type and the corresponding destination IP address, destination port, protocol type, and service.

Once an application is identified, its information is saved in the cache so that only one pattern matching is required for an application running on a particular system, thereby expediting the identification process.

A mapping is saved in the cache only if the matched signature contains both client-to-server and server-to-client patterns. This process protects the system from hackers who might send malicious packets through a legitimate server port so that it is interpreted as a different application.

By default, the application system cache saves the mapping information for 3600 seconds. However, you can configure the cache timeout value by using the CLI.

To minimize the impact on performance, application system cache is refreshed only when Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) traffic triggers a cache lookup. Without a cache lookup, the entries in the application system cache remain unchanged even after cache timeout.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

- Understanding IDP Application Identification on page 795
- Understanding IDP Application Identification for Nested Applications on page 800
- Understanding IDP Application System Cache Information for Nested Application Identification on page 802
- Deactivating IDP Application System Cache Information for Nested Application Identification (CLI Procedure) on page 802
- Verifying IDP Application System Cache Statistics on page 803

## Understanding IDP Application System Cache Information for Nested Application Identification

Nested application identification information is saved in the application system cache to improve performance. This cache is updated when a different application is identified. The only circumstances in which nested application information is not cached are the following:

- The application system cache is turned off for nested application identification.
- The matched application signatures have only client-to-server members.
- There is no valid server-to-client response seen for a transaction. This is done to prevent an attacker from sending invalid client-to-server requests to poison the application system cache.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding IDP Application Identification on page 795
- Understanding IDP Application Identification for Nested Applications on page 800
- Understanding the IDP Application System Cache on page 801
- Deactivating IDP Application System Cache Information for Nested Application Identification (CLI Procedure) on page 802
- Verifying IDP Application System Cache Statistics on page 803

## Deactivating IDP Application System Cache Information for Nested Application Identification (CLI Procedure)

Caching for nested applications is turned on by default. You can manually turn this caching off using the CLI.

```
user@host# edit security idp sensor-configuration application-identification
no-nested-application-system-cache
```

When you use the show command for the application system cache, nested application information is displayed as follows:

```
user@host# show security idp application-identification application-system-cache

Vsys-ID IP address Port Protocol Service Application
0 5.0.0.1 80 TCP HTTP FACEBOOK
0 5.0.0.2 80 TCP HTTP NONE
```



- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding the IDP Application System Cache on page 801
  - Understanding IDP Application System Cache Information for Nested Application Identification on page 802
  - Verifying IDP Application System Cache Statistics on page 803

## Verifying IDP Application System Cache Statistics

**Purpose** Verify the IDP application system cache (ASC) statistics.

**Action** From the CLI, enter the `show security idp application-identification application-system-cache` command.

## Sample Output

```
user@host> show security idp application-identification application-system-cache
IDP Application System Cache statistics:
```

| Vsys-ID | IP address | Port | Protocol | Service |
|---------|------------|------|----------|---------|
| 0       | 20.0.0.4   | 23   | tcp      | TELNET  |
| 0       | 20.0.0.6   | 23   | tcp      | TELNET  |
| 0       | 20.0.0.2   | 23   | tcp      | TELNET  |
| 0       | 20.0.0.2   | 25   | tcp      | SMTP    |
| 0       | 20.0.0.6   | 25   | tcp      | SMTP    |
| 0       | 20.0.0.4   | 25   | tcp      | SMTP    |
| 0       | 20.0.0.3   | 135  | tcp      | MSRPC   |
| 0       | 20.0.0.5   | 139  | tcp      | SMB     |
| 0       | 20.0.0.7   | 139  | tcp      | SMB     |
| 0       | 20.0.0.3   | 143  | tcp      | IMAP    |
| 0       | 20.0.0.5   | 143  | tcp      | IMAP    |
| 0       | 20.0.0.3   | 139  | tcp      | SMB     |
| 0       | 20.0.0.7   | 143  | tcp      | IMAP    |
| 0       | 20.0.0.3   | 80   | tcp      | HTTP    |
| 0       | 20.0.0.5   | 80   | tcp      | HTTP    |
| 0       | 20.0.0.7   | 80   | tcp      | HTTP    |

**Meaning** The output shows a summary of the ASC statistics information. Verify the following information:

- Vsys-ID—Displays the virtual system identification number.
- IP address—Displays the destination address.
- Port—Displays the destination port on the server.
- Service—Displays the name of the service or application identified on the destination port.

For a complete description of `show security idp application-identification application-system-cache` output, see the *Junos OS CLI Reference*.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding IDP Service and Application Bindings by Attack Objects on page 796](#)
  - [IDP Policies Overview on page 701](#)
  - [Deactivating IDP Application System Cache Information for Nested Application Identification \(CLI Procedure\) on page 802](#)
  - [Example: Configuring IDP Policies for Application Identification on page 798](#)
  - [Disabling Application Identification for an IDP Policy \(CLI Procedure\) on page 799](#)
  - [Verifying Application System Cache Statistics on page 1124](#)

## IDP Memory and Session Limits

- [Understanding Memory and Session Limit Settings for IDP Application Identification on page 804](#)
- [Example: Setting Memory and Session Limits for IDP Application Identification Services on page 805](#)

### Understanding Memory and Session Limit Settings for IDP Application Identification

Although you cannot create application signatures with the IDP signature database, you can configure sensor settings to limit the number of sessions running application identification and also limit memory usage for application identification.

- **Memory limit for a session**—You can configure the maximum amount of memory bytes that can be used to save packets for application identification for one TCP or UDP session. You can also configure a limit for global memory usage for application identification. Application identification is disabled for a session after the system reaches the specified memory limit for the session. However, IDP continues to match patterns. The matched application is saved to cache so that the next session can use it. This protects the system from attackers trying to bypass application identification by purposefully sending large client-to-server packets.
- **Number of sessions**—You can configure the maximum number of sessions that can run application identification at the same time. Application identification is disabled after the system reaches the specified number of sessions. You limit the number of sessions so that you can prevent a denial-of-service (DOS) attack, which occurs when too many connection requests overwhelm and exhaust all the allocated resources on the system.

Table 99 on page 804 provides the capacity of a central point (CP) session numbers for SRX3400, SRX3600, SRX5600, and SRX5800 devices.

**Table 99: Maximum CP Session Numbers**

| SRX Series Devices | Maximum Sessions | Central Point (CP) |
|--------------------|------------------|--------------------|
| SRX3400            | 2.25 million     | Combo-mode CP      |

Table 99: Maximum CP Session Numbers (*continued*)

| SRX Series Devices | Maximum Sessions | Central Point (CP) |
|--------------------|------------------|--------------------|
| SRX3600            | 2.25 million     | Combo-mode CP      |
| SRX5600            | 9 million        | Full CP            |
|                    | 2.25 million     | Combo-mode CP      |
| SRX5800            | 10 million       | Full CP            |
|                    | 2.25 million     | Combo-mode CP      |

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding IDP Application Identification on page 795](#)
- [IDP Policies Overview on page 701](#)
- [Understanding the IDP Signature Database on page 777](#)
- [Example: Updating the IDP Signature Database Manually on page 785](#)
- [Example: Setting Memory and Session Limits for IDP Application Identification Services on page 805](#)

### Example: Setting Memory and Session Limits for IDP Application Identification Services

This example shows how to configure memory and session limits for IDP application identification services.

- [Requirements on page 805](#)
- [Overview on page 805](#)
- [Configuration on page 806](#)
- [Verification on page 806](#)

#### Requirements

Before you begin:

- Configure network interfaces. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).
- Download the signature database. See “Example: Updating the IDP Signature Database Manually” on page 785.

#### Overview

In this example, you configure the limit so that only 600 sessions can run application identification at the same time. You also configure 5000 memory bytes as the maximum amount of memory that can be used for saving packets for application identification for one TCP session.

## Configuration

**Step-by-Step Procedure** To configure memory and session limits for IDP application identification services:

- Specify the session limit for application identification.
 

```
[edit]
user@host# set security idp sensor-configuration application-identification
max-sessions 600
```
- Specify the memory limits for application identification.
 

```
[edit]
user@host# set security idp sensor-configuration application-identification
max-tcp-session-packet-memory 5000
```
- If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

## Verification

To verify the configuration is working properly, enter the **show security idp** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding Memory and Session Limit Settings for IDP Application Identification on page 804

## Verifying IDP Counters for Application Identification Processes

**Purpose** Verify the IDP counters for the application identification processes.

**Action** From the CLI, enter the **show security idp counters application-identification** command.

### Sample Output

```
user@host> show security idp counters application-identification
IDP counters:

IDP counter type           Value
AI cache hits              2682
AI cache misses            3804
AI matches                  74
AI no-matches              27
AI-enabled sessions        3804
AI-disabled sessions       2834
AI-disabled sessions due to cache hit 2682
AI-disabled sessions due to configuration 0
AI-disabled sessions due to protocol remapping 0
AI-disabled sessions due to non-TCP/UDP flows 118
AI-disabled sessions due to no AI signatures 0
AI-disabled sessions due to session limit 0
AI-disabled sessions due to session packet memory limit 34
AI-disabled sessions due to global packet memory limit 0
```

**Meaning** The output shows a summary of the application identification counters. Verify the following information:

- AI cache hits—Displays the number of hits on the application identification cache
- AI cache misses—Displays the number of times the application matches but the application identification cache entry is not added.
- AI matches—Displays the number of times the application matches, and an application identification cache entry is added.
- AI no-matches—Displays the number of times when application does not match.
- AI-enabled sessions—Displays the number of sessions on which application identification is enabled.
- AI-disabled sessions—Displays the number of sessions on which application identification is disabled.
- AI-disabled sessions due to cache hit—Displays the number of sessions on which application identification is disabled after a cache entry is matched. Application identification process is discontinued for this session.
- AI-disabled sessions due to configuration—Displays the number of sessions on which application identification is disabled because of the sensor configuration.
- AI-disabled sessions due to protocol remapping—Displays the number of sessions for which application identification is disabled because you have configured a specific service in the IDP policy rule definition.
- AI-disabled sessions due to non-TCP/UDP flows—Displays the number of sessions for which application identification is disabled because the session is not a TCP or UDP session.
- AI-disabled sessions due to no AI signatures—Displays the number of sessions for which application identification is disabled because no match is found on the application identification signatures.
- AI-disabled due to session limit—Displays the number of sessions for which application identification is disabled because sessions have reached the maximum limit configured. Application identification is disabled for future sessions too.
- AI-disabled due to session packet memory limit—Displays the sessions for which application identification is disabled because sessions have reached the maximum memory limit on TCP or UDP flows. Application identification is disabled for future sessions too.
- AI-disabled due to global packet memory limit—Displays the sessions for which application identification is disabled because the maximum memory limit is reached. Application identification is disabled for future sessions too.

For a complete description of **show security idp counters** output, see the *Junos OS CLI Reference*.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

- [Understanding IDP Application Identification on page 795](#)
- [Example: Setting Memory and Session Limits for IDP Application Identification Services on page 805](#)
- [Understanding IDP Service and Application Bindings by Attack Objects on page 796](#)
- [Verifying Application System Cache Statistics on page 1124](#)

# IDP SSL Inspection

- IDP SSL Overview on page 809
- Supported IDP SSL Ciphers on page 810
- Understanding IDP Internet Key Exchange on page 811
- Understanding IDP SSL Server Key Management and Policy Configuration on page 812
- Displaying IDP SSL Keys and Associated Servers on page 812
- Adding IDP SSL Keys and Associated Servers on page 813
- Deleting IDP SSL Keys and Associated Servers on page 813
- Configuring an IDP SSL Inspection (CLI Procedure) on page 814

## IDP SSL Overview

---

Secure Sockets Layer (SSL), also called Transport Layer Security (TLS), is a protocol suite for Web security that provides authentication, confidentiality and message integrity. Authentication guards against fraudulent transmissions by enabling a Web browser to validate the identity of a webserver. Confidentiality mechanisms ensure that communications are private. SSL enforces confidentiality by encrypting data to prevent unauthorized users from eavesdropping on electronic communications. Finally, message integrity ensures that the contents of a communication have not been tampered with.

Each SSL session begins with a handshake during which the client and server agree on the specific security key and the encryption algorithms to use for that session. At this time, the client also authenticates the server. Optionally, the server can authenticate the client. Once the handshake is complete, transfer of encrypted data can begin.

Juniper Networks provides Intrusion Detection and Prevention (IDP) SSL inspection that uses the SSL protocol suite consisting of different SSL versions, ciphers, and key exchange methods. Combined with the Application Identification feature, the SSL Inspection feature enables SRX Series devices to inspect HTTP traffic encrypted in SSL on any port. The following SSL protocols are supported:

- SSLv2
- SSLv3
- TLS

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- IDP Policies Overview on page 701
- Supported IDP SSL Ciphers on page 810
- Understanding IDP Internet Key Exchange on page 811
- Understanding IDP SSL Server Key Management and Policy Configuration on page 812
- Configuring an IDP SSL Inspection (CLI Procedure) on page 814

## Supported IDP SSL Ciphers

An SSL cipher comprises encryption cipher, authentication method, and compression. Junos OS supports all OPENSSL supported ciphers that do not involve the use of temporary private keys. For authentication, NULL, MD5, and SHA-1 authentication methods are supported.



**NOTE:** Compression and SSLv2 ciphers are not supported. Currently, most SSL servers automatically upgrade to a TLS cipher when an SSLv2 cipher is received in a client “hello” message. Check your browser to see how strong the ciphers can be and which ones your browser supports. (If the cipher is not in the list of supported ciphers, the session is ignored for deep packet inspection.)

Table 100 on page 810 shows the encryption algorithms supported by the SRX Series devices.

**Table 100: Supported Encryption Algorithms**

| Cipher       | Exportable | Type   | Key Material | Expanded Key Material | Effective Key Bits | IV Size |
|--------------|------------|--------|--------------|-----------------------|--------------------|---------|
| NULL         | No         | Stream | 0            | 0                     | 0                  | N/A     |
| DES-CBC-SHA  | No         | Block  | 8            | 8                     | 56                 | 8       |
| DES-CBC3-SHA | No         | Block  | 24           | 24                    | 168                | 8       |
| AES128-SHA   | No         | Block  | 16           | 16                    | 128                | 16      |
| AES256-SHA   | No         | Block  | 32           | 32                    | 256                | 16      |

For more information on encryption algorithms, see “VPN Overview” on page 451. Table 101 on page 810 shows the supported SSL ciphers.

**Table 101: Supported SSL Ciphers**

| Cipher Suites | Value |
|---------------|-------|
|---------------|-------|



Table 101: Supported SSL Ciphers (*continued*)

|                               |        |
|-------------------------------|--------|
| TLS_RSA_WITH_NULL_MD5         | 0x0001 |
| TLS_RSA_WITH_NULL_SHA         | 0x0002 |
| TLS_RSA_WITH_DES_CBC_SHA      | 0x0009 |
| TLS_RSA_WITH_3DES_EDE_CBC_SHA | 0x000A |
| TLS_RSA_WITH_AES_128_CBC_SHA  | 0x002F |
| TLS_RSA_WITH_AES_256_CBC_SHA  | 0x0035 |



**NOTE:** RC4 and IDEA ciphers are not supported because of license and OPENSSL library availability.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [IDP SSL Overview on page 809](#)
- [Understanding IDP Internet Key Exchange on page 811](#)
- [Understanding IDP SSL Server Key Management and Policy Configuration on page 812](#)

## Understanding IDP Internet Key Exchange

Internet Key Exchange (IKE) establishes a premaster secret that is used to generate symmetric keys for bulk data encryption and authentication. Section F.1.1 of RFC 2246 defines Transport Layer Security (TLS) authentication and key exchange methods. The two key exchange methods are:

- **RSA—Rivest-Shamir-Adleman (RSA)** is a key exchange algorithm that governs the way participants create symmetric keys or a secret that is used during an SSL session. The RSA key exchange algorithm is the most commonly used method.
- **DSA—Digital Signature Algorithm (DSA)** adds an additional authentication option to the IKE Phase 1 proposals. The DSA can be configured and behaves analogously to the RSA, requiring the user to import or create DSA certificates and configure an IKE proposal to use the DSA. Digital certificates are used for RSA signatures, DSA signatures, and the RSA public key encryption based method of authentication in the IKE protocol.
- **Diffie-Hellman—Diffie-Hellman (DH)** is a key exchange method that allows participants to produce a shared secret value. The strength of the technique is that it allows participants to create the secret value over an unsecured medium without passing the secret value through the wire.

The key exchange methods can use either a fixed or a temporary server key. IDP can successfully retrieve the premaster secret only if a fixed server key is used. For more information on Internet Key Exchange, see “Understanding Certificates and PKI” on page 569.



**NOTE:** Juniper IDP does not decrypt SSL sessions that use Diffie-Hellman key exchange.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- IDP SSL Overview on page 809
- Supported IDP SSL Ciphers on page 810
- Understanding IDP SSL Server Key Management and Policy Configuration on page 812
- Configuring an IDP SSL Inspection (CLI Procedure) on page 814

## Understanding IDP SSL Server Key Management and Policy Configuration

The device can support up to 1000 server private keys. Each key can have up to 100 servers that use it. This capacity is the same regardless of the number of SPUs available on the device because essentially each SPU needs to be able to access all the keys.

Multiple servers can share the same private key; however, one server can have only one private key. SSL decryption is disabled by default. Both plain and encrypted keys are supported.



**NOTE:** Junos OS does not encrypt SSL keys file.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- IDP SSL Overview on page 809
- Displaying IDP SSL Keys and Associated Servers on page 812
- Adding IDP SSL Keys and Associated Servers on page 813
- Deleting IDP SSL Keys and Associated Servers on page 813
- Configuring an IDP SSL Inspection (CLI Procedure) on page 814

## Displaying IDP SSL Keys and Associated Servers

- To display all installed server keys and associated server, use the following CLI command:

```
user@host> show security idp ssl-inspection key
```

Displays all server keys and IP addresses bound to those keys. The following example shows CLI output when the **show security idp ssl-inspection key** command is used:

```
Total SSL keys : 2
SSL server key and ip address :
  Key : key1, server : 1.1.1.1
  Key : key2, server : 2.2.2.2
```

Key : key2, server : 2.2.2.3

- To display IP addresses bound to a specific key, use the following CLI command:

```
user@host> show security idp ssl-inspection key <key-name>
```

The following is an example of the CLI output received when the **show security idp ssl-inspection key <key-name>** command is used:

```
Key : key1, server : 1.1.1.1
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding IDP SSL Server Key Management and Policy Configuration on page 812
- Adding IDP SSL Keys and Associated Servers on page 813
- Deleting IDP SSL Keys and Associated Servers on page 813

## Adding IDP SSL Keys and Associated Servers

When you are installing a key, you can password protect the key and also associate it to a server.

To install a Privacy-Enhanced Mail (PEM) key, use the following CLI command:

```
user@host> request security idp ssl-inspection key add <key-name> [file <file-path>]
server <server-ip> [password <password-string>]
```



**NOTE:** In a two-node SRX cluster, the key has to be manually copied over to both Node 0 and Node 1 at the same location for the request command to be successful.

You can also associate the key with a server at a later time by using the **add server** CLI command. A server can be associated with only one key. To associate a server to the installed key, use the following CLI command:

```
user@host> request security idp ssl-inspection key add <key-name> server <server-ip>
```



**NOTE:** The maximum key name length is 32 bytes, including the ending “\0”.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding IDP SSL Server Key Management and Policy Configuration on page 812
- Displaying IDP SSL Keys and Associated Servers on page 812
- Deleting IDP SSL Keys and Associated Servers on page 813

## Deleting IDP SSL Keys and Associated Servers

- To delete all keys and servers, use the following CLI command:

```
user@host> request security idp ssl-inspection key delete
```

All installed keys are deleted along with any associated servers.

- To delete a specific key and all associated servers with that key, use the following CLI command:

```
user@host> request security idp ssl-inspection key delete <key-name>
```

Deletes the specified key and all servers associated with that key.

- To delete a single server, use the following CLI command:

```
user@host> request security idp ssl-inspection key delete <key-name> server
<server-ip>
```

Deletes the specified server that is bound to the specified key.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding IDP SSL Server Key Management and Policy Configuration on page 812
- Displaying IDP SSL Keys and Associated Servers on page 812
- Adding IDP SSL Keys and Associated Servers on page 813

## Configuring an IDP SSL Inspection (CLI Procedure)

SSL decoder is enabled by default. If you need to manually enable it via CLI, use the following CLI command.

```
set security idp sensor-configuration detector protocol-name SSL tunable-name sc_ssl_flags
tunable-value 1
```

To configure an IDP SSL inspection, use the following CLI procedure:

```
[edit security]
idp {
  sensor-configuration {
    ssl-inspection {
      sessions <number>;
    }
  }
}
```

The sensor now inspects traffic for which it has a key/server pair.



**NOTE:** Maximum supported sessions per SPU: default value is 10,000 and range is 1 to 100,000. The session limit is per SPU, and it is the same regardless of the number of SPUs on the device.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- IDP SSL Overview on page 809
- Understanding IDP Internet Key Exchange on page 811

- [Understanding IDP SSL Server Key Management and Policy Configuration on page 812](#)



# IDP Class of Service Action

- IDP Class of Service Action Overview on page 817
- Example: Applying the CoS Action in an IDP Policy on page 818

## IDP Class of Service Action Overview

---

Differentiated Services (DS) is a system for tagging (or “marking”) traffic at a position within a hierarchy of priority. Differentiated Services codepoint (DSCP) marking maps the Junos OS Class of Service (CoS) level to the DSCP field in the IP packet header. On SRX1400, SRX3400, SRX3600, SRX5600, and SRX5800 devices, DSCP values of IP packets can be rewritten by the following two software modules:

- Differentiated Services code point (DSCP) rewriter at an egress interface.
- IDP module according to IDP policies.

In the data plane, before a packet reaches an egress interface, the IDP module can notify the security flow module to rewrite the packet's DSCP value. The IDP module and the interface-based rewriter rewrite DSCP values based on different and independent rules. The IDP module rewrites a packet's DSCP value based on IDP policies; whereas the interface-based writer rewrites a packet's DSCP value based on packet classification results. Therefore the rewriting decisions of the IDP module and the interface-based rewriter can be different.

An interface-based rewriter rewrites DSCP values by comparing a packet's forwarding class against a set of forwarding classes configured as rewrite rules. A forwarding class that does not belong to this set of forwarding classes is used to notify an interface-based rewriter to not rewrite a packet's DSCP value when it has been set by the IDP module.



**NOTE:** In addition to influencing the rewriting of a packet's DSCP value, forwarding classes are also used to prioritize the traffic in the device. By assigning a forwarding class to a queue number, you affect the scheduling and marking of a packet as it transits an SRX Series device. For information on forwarding classes, see [Forwarding Classes Overview and Defining Forwarding Classes \(J-Web Procedure\)](#) in the *Junos OS Class of Service Configuration Guide for Security Devices*.

---

When the IDP module rewrites a packet's DSCP value, IDP can set the forwarding class associated with the packet such that the forwarding class is out of the set of forwarding classes defined as the rule for an egress interface-based rewriter. For information on rewrite rules, see Rewrite Rules Overview and Example: Configuring and Applying Rewrite Rules in the *Junos OS Class of Service Configuration Guide for Security Devices*.

When the interface-based rewriter processes the packet, it notices that the packet's forwarding class does not match any of the classes defined in the rewrite rule, therefore it does not change the DSCP value of the packet. Consequently, the packet's DSCP value is marked by the IDP module and the interface-based rewriter is bypassed. Separate forwarding classes for the IDP module and the interface-based rewriter can be defined using the **set forwarding-class** statement at the [edit class-of-service] hierarchy level. For example, forwarding classes fc0, fc1, fc2, and fc3 can be defined for the IDP module, while forwarding classes fc4, fc5, fc6, and fc7 can be defined for the interface-based rewriters. In Junos OS, multiple forwarding classes can be mapped to one priority queue. Therefore the number of forwarding classes can be more than the number of queues.



**NOTE:** When both the interface-based rewriter and the IDP modules try to rewrite DSCP values, the IDP module is given precedence over the interface-based rewriter because IDP marks DSCP values with more information about the packets and has stricter security criteria than the interface-based rewriter module.

---

For a configuration example that shows how you can rewrite DSCP values with the IDP module and bypass the interface-based rewriter, see "Example: Applying the CoS Action in an IDP Policy" on page 818.

### Example: Applying the CoS Action in an IDP Policy

---

As packets enter or exit a network, devices might be required to alter the CoS settings of the packet. Rewrite rules set the value of the CoS bits within the packet's header. In addition, you often need to rewrite a given marker (for example, DSCP) at the inbound interfaces of a device to accommodate BA classification by core devices.

On SRX1400, SRX3400, SRX3600, SRX5600, and SRX5800 devices, DSCP values of IP packets can be rewritten by the following two software modules:

- DSCP rewriter at an egress interface
- IDP module according to IDP policies

This example describes how to create an IDP policy that defines a forwarding class as an action item to rewrite the DSCP value of a packet.

- Requirements on page 819
- Overview on page 819
- Configuration on page 819
- Verification on page 824



## Requirements

Before you begin, review the CoS components. See the *Junos OS Class of Service Configuration Guide for Security Devices*.

## Overview

This example shows how you can rewrite DSCP values with the IDP module and bypass the interface-based rewriter. When you create an IDP policy to rewrite DSCP values, you must specify the following:

- Configure separate forwarding classes for the IDP module and the interface-based rewriters. In this example, eight forwarding classes, fc1 through fc8, are configured. Out of these eight forwarding classes, four classes, fc1 through fc4, are assigned to interface-based rewriters; the other four, fc5 through fc8, are assigned to the IDP module. These eight forwarding classes are mapped to four priority queues, queue 0 through queue 3.
- Configure the DSCP rewriter (rw\_dscp) with forwarding classes, fc1 through fc4.
- Configure a DSCP classifier (c1) with the same forwarding classes as the DSCP rewriter. Essentially the classifier provides inputs, forwarding classes, and loss priorities to the rewriter.
- Apply the DSCP rewriter, rw\_dscp, to a logical interface, ge-0/0/5.
- Apply the classifier, c1, to an ingress logical interface, ge-0/0/6.
- Create a new IDP policy (cos-policy) and assign class-of-service forwarding-class fc5 as the action.



**NOTE:** To ensure DSCP rewriting by IDP, it is important that you do not configure an IDP policy and interface-based DSCP rewrite rules with the same forwarding class.

## Configuration

### CLI Quick Configuration

To quickly rewrite DSCP values with the IDP module and bypass the interface-based rewriter, copy the following commands and paste them into the CLI:

```
[edit ]
set class-of-service forwarding-classes queue 0 fc1
set class-of-service forwarding-classes queue 1 fc2
set class-of-service forwarding-classes queue 2 fc3
set class-of-service forwarding-classes queue 3 fc4
set class-of-service forwarding-classes queue 0 fc5
set class-of-service forwarding-classes queue 1 fc6
set class-of-service forwarding-classes queue 2 fc7
set class-of-service forwarding-classes queue 3 fc8
set class-of-service rewrite-rules dscp rw_dscp
set class-of-service rewrite-rules dscp rw_dscp forwarding-class fc1 loss-priority low
code-point 000000
```

```

set class-of-service rewrite-rules dscp rw_dscp forwarding-class fc2 loss-priority low
  code-point 001000
set class-of-service rewrite-rules dscp rw_dscp forwarding-class fc3 loss-priority low
  code-point 010000
set class-of-service rewrite-rules dscp rw_dscp forwarding-class fc4 loss-priority low
  code-point 011000
set class-of-service classifiers dscp c1 forwarding-class fc1 loss-priority low code-points
  11111
set class-of-service classifiers dscp c1 forwarding-class fc2 loss-priority low code-points
  110000
set class-of-service classifiers dscp c1 forwarding-class fc3 loss-priority low code-points
  100000
set class-of-service classifiers dscp c1 forwarding-class fc4 loss-priority low code-points
  000000
set class-of-service interfaces ge-0/0/5 unit 0 rewrite-rules dscp rw_dscp
set class-of-service interfaces ge-0/0/6 unit 0 classifiers dscp c1
set security idp idp-policy cos-policy
set security idp idp-policy cos-policy rulebase-ips
set security idp idp-policy cos-policy rulebase-ips rule r1
set security idp idp-policy cos-policy rulebase-ips rule r1 match from-zone any to-zone
  any application default
set security idp idp-policy cos-policy rulebase-ips rule r1 match attacks
  predefined-attack-groups 'P2P - All'
set security idp idp-policy cos-policy rulebase-ips rule r1 then action class-of-service
  forwarding-class fc5
set security idp idp-policy cos-policy rulebase-ips rule r1 then action class-of-service
  dscp-code-point 62
set security idp idp-policy cos-policy rulebase-ips rule r1 then notification log-attacks
set security idp idp-policy cos-policy rulebase-ips rule r1 then severity critical

```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure an IDP policy that uses a forwarding class as a notification action for DSCP rewriting, perform the following tasks:

1. Configure forwarding classes.

To configure a one-to-one mapping between the eight forwarding classes and the four priority queues, include the following statements at the `[edit class-of-service]` hierarchy level:

```

[edit class-of-service]
user@host# set forwarding-class fc1 queue-num 0
user@host# set forwarding-class fc2 queue-num 1
user@host# set forwarding-class fc3 queue-num 2
user@host# set forwarding-class fc4 queue-num 3
user@host# set forwarding-class fc5 queue-num 0
user@host# set forwarding-class fc6 queue-num 1
user@host# set forwarding-class fc7 queue-num 2
user@host# set forwarding-class fc8 queue-num 3

```

2. Configure a DSCP rewriter with forwarding classes.

```

[edit class-of-service]

```

```

user@host# set rewrite-rules dscp rw_dscp forwarding-class fc1 loss-priority low
code-point 000000
user@host# set rewrite-rules dscp rw_dscp forwarding-class fc2 loss-priority low
code-point 001000
user@host# set rewrite-rules dscp rw_dscp forwarding-class fc3 loss-priority low
code-point 010000
user@host# set rewrite-rules dscp rw_dscp forwarding-class fc4 loss-priority low
code-point 011000

```

3. Configure a BA classifier with the same forwarding classes as the DSCP rewriter.

```

[edit class-of-service]
user@host# set classifiers dscp c1 forwarding-class fc1 loss-priority low code-points
111111
user@host# set classifiers dscp c1 forwarding-class fc2 loss-priority low code-points
110000
user@host# set classifiers dscp c1 forwarding-class fc3 loss-priority low code-points
100000
user@host# set classifiers dscp c1 forwarding-class fc4 loss-priority low code-points
000000

```

4. Apply the rewriter to a logical interface.

```

[edit class-of-service]
user@host# set interfaces ge-0/0/5 unit 0 rewrite-rules dscp rw_dscp

```

5. Apply the classifier to a logical interface.

```

[edit class-of-service]
user@host# set interfaces ge-0/0/6 unit 0 classifiers dscp c1

```

6. Configure the IDP policy with the action of forwarding class.

The following steps show how an IDP policy includes a class-of-service forwarding class as one of the actions. In policy *cos-policy*, forwarding class fc5 is defined as an action in conjunction with the action of dscp-code-point 62, which requires the IDP module to rewrite DSCP values to 62. Taking actions of R1, the IDP module conducts the security flow module to rewrite the packets' DSCP values as 62 and set their forwarding classes as fc5.

To set a forwarding class as one of the actions in an IDP policy, perform the following tasks:

- a. Create a policy by assigning a meaningful name to it.

```

[edit ]
user@host# edit security idp idp-policy cos-policy

```

- b. Associate a rulebase with the policy.

```

[edit security idp idp-policy cos-policy ]
user@host# edit rulebase-ips

```

- c. Add rules to the rulebase.

```

[edit security idp idp-policy cos-policy rulebase-ips]
user@host# edit rule R1

```

- d. Define the match criteria for the rule.

```
[edit security idp idp-policy cos-policy rulebase-ips rule R1]
user@host# set match from-zone any to-zone any application default
```

- e. Define an attack as match criteria.

```
[edit security idp idp-policy cos-policy rulebase-ips rule R1]
user@host# set match attacks predefined-attack-groups 'P2P - All'
```

- f. Specify forwarding class as an action for the rule.

```
[edit security idp idp-policy cos-policy rulebase-ips rule R1]
user@host# set then action class-of-service forwarding-class fc5
```

- g. Specify dscp-code-point as an action for the rule.

```
[edit security idp idp-policy cos-policy rulebase-ips rule R1]
user@host# set then action class-of-service dscp-code-point 62
```

- h. Specify notification and logging options for the rule.

```
[edit security idp idp-policy cos-policy rulebase-ips rule R1]
user@host# set then notification log-attacks alert
```

- i. Set the severity level for the rule.

```
[edit security idp idp-policy cos-policy rulebase-ips rule R1]
user@host# set then severity critical
```

- j. Activate the policy.

```
[edit]
user@host# set security idp active-policy cos-policy
```

**Results** From configuration mode, confirm your configuration by entering the **show security idp** and **show class-of-service** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security idp
idp-policy cos-policy {
  rulebase-ips {
    rule R1 {
      match {
        from-zone any;
        to-zone any;
        application default;
        attacks {
          predefined-attack-groups P2P - All;
        }
      }
    }
  }
  then {
    action {
      class-of-service {
        forwarding-class fc5;
        dscp-code-point 62;
      }
    }
  }
  notification {
```



```
    forwarding-class fc2 {  
        loss-priority low code-point 001000;  
    }  
    forwarding-class fc3 {  
        loss-priority low code-point 010000;  
    }  
    forwarding-class fc4 {  
        loss-priority low code-point 011000;  
    }  
    }  
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying IDP Policy Configuration on page 824
- Verifying CoS Configuration on page 824

### Verifying IDP Policy Configuration

**Purpose** Verify that the forwarding class fc5 is configured as an action in the IDP policy.

**Action** From operational mode, enter the **show security idp idp-policy cos-policy** command.

### Verifying CoS Configuration

**Purpose** Verify if the one-to-one mapping between the eight forwarding classes and the four priority queues, application of the BA classifier to the interfaces, and the rewrite rule are working.

**Action** From operational mode, enter the **show class-of-service** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Junos OS CLI Reference](#)
- [Junos OS Class of Service Configuration Guide for Security Devices](#)
- Understanding IDP Policy Rules on page 707
- Example: Enabling IDP in a Security Policy on page 702

# IDP Performance and Capacity Tuning

- Performance and Capacity Tuning for IDP Overview on page 825
- Configuring Session Capacity for IDP (CLI Procedure) on page 826

## Performance and Capacity Tuning for IDP Overview

---

This topic provides an overview on performance and capacity tuning for an Intrusion Detection and Prevention (IDP) session.

If you are deploying IDP policies, you can configure the device to increase IDP session capacity. By using the provided commands to change the way the system allocates resources, you can achieve higher IDP session capacity.

By using the **maximize-idp-sessions** command, you can increase the IDP session capacity. In this mode, by default, the device assigns a greater weight value to firewall functions. Based on your IDP policy, you can shift the weight to IDP functions to maximize IDP performance. By shifting weight, you are increasing capacity and allocating more processing power for the given service.



**NOTE:** You should not configure the device to increase IDP session capacity if you are not using an IDP policy.

The device ships with an implicit default session capacity setting. This default value adds weight to firewall sessions. You can manually override the default by adding the **maximize-idp-sessions** setting to your configuration. When you do this, in addition to IDP session scaling, you can choose to assign weight values of equal, firewall, or IDP to firewall and IDP functions. Typically, when you only include IDP-recommended attacks or client-to-server attacks in your IDP policy, IDP functions consume less CPU resources, for this reason, you would select weight firewall to maximize device performance. Alternatively, if you add server-to-client attacks to your IDP policy, IDP functions consume higher CPU resources. For this reason, you would select weight IDP to maximize performance. Essentially, you will need to configure the weight based on the desired IDP policy and performance. You do this by examining the CPU resource utilization on the packet forwarding engine by using the command **show security monitoring fpc number**. For details on this command, see the [Junos OS CLI Reference](#).

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - IDP Policies Overview on page 701
  - Configuring Session Capacity for IDP (CLI Procedure) on page 826

## Configuring Session Capacity for IDP (CLI Procedure)

The configuration instructions in this topic describe how modify session capacity for IDP policies.

You do this by adding the **maximize-idp-sessions** command and then adding the weight option to specify IDP sessions.



**NOTE:** The weight option depends on the **maximize-idp-sessions** command being set.

1. If you have an active IDP policy, you can configure the device to increase IDP session capacity by entering following command:

```
user@host# set security forwarding-process application-services maximize-idp-sessions
```

2. You can further adjust the weight of the firewall and IDP processing functions, such as in the case of heavier IDP policies with the following command:

```
user@host# set security forwarding-process application-services maximize-idp-sessions weight idp
```

3. Commit your changes. You must reboot the device for any session capacity setting changes to take effect.



**NOTE:** If the device has **maximize-idp-sessions** weight enabled for IDP, and you do not have an IDP policy configured, a warning message appears when you commit your configuration. If you see this warning, you should remove your configured settings.

To turn **maximize-idp-sessions** settings off, remove the **maximize-idp-sessions** configuration.



**NOTE:** You must reboot the device for any **maximize-idp-sessions** setting changes to take effect.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - IDP Policies Overview on page 701
  - Performance and Capacity Tuning for IDP Overview on page 825



# IDP Logging

- Understanding IDP Logging on page 827
- IDP Application-Level DDoS Logging on page 828
- IDP Log Suppression Attributes on page 830
- Security Packet Capture on page 832
- Understanding IDP Log Information Usage on the Infranet Controller on page 838

## Understanding IDP Logging

---

The basic Junos OS system logging continues to function after Intrusion Detection and Prevention (IDP) is enabled. An IDP-enabled device continues to record events that occur because of routine operations, such as a user login into the configuration database. It records failure and error conditions, such as failure to access a configuration file. You can configure files to log system messages and also assign attributes, such as severity levels, to messages. In addition to the regular system log messages, IDP generates event logs for attacks.

IDP generates event logs when an event matches an IDP policy rule in which logging is enabled. When you configure a rule for logging, the device creates a log entry for each event that matches that rule. You can use the CLI or J-Web to configure the policy rules to generate event logs.

Because IDP event logs are generated during an attack, log generation happens in bursts, generating a much larger volume of messages during an attack. In comparison to other event messages, the message size is also much larger for attack generated messages. The log volume and message size are important concerns for log management. To better manage the volume of log messages, IDP supports log suppression.

By configuring log suppression you can suppress multiple instances of the same log occurring from the same or similar sessions over the same period of time. Enabling log suppression ensures that minimal numbers of logs are generated for the same event or attack that occurs multiple times.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- For information about monitoring events, managing system log files, and configuring packet capture see the *Junos OS Administration Guide for Security Devices*.
- IDP Policies Overview on page 701

- Understanding Application-Level DDoS Logging on page 828
- Understanding IDP Log Suppression Attributes on page 830
- Understanding Security Packet Capture on page 832
- Understanding IDP Log Information Usage on the IC Series UAC Appliance on page 838

## IDP Application-Level DDoS Logging

---

- Understanding Application-Level DDoS Logging on page 828
- Enabling Attack and IP-Action Logging (CLI Procedure) on page 830

### Understanding Application-Level DDoS Logging

Intrusion Detection and Prevention (IDP) generates three types of application-level distributed denial-of-service (application-level DDoS) event logs: attack, state transition, and ip-action. These event logs provide visibility into the application-level DDoS state and provide notifications on occurrences of application-level DDoS attacks for each protected application server.

IDP generates application-level DDoS attack event logs when logging is enabled and an event matches an application-level DDoS policy rule. When you configure a rule with logging enabled, the device creates a log entry for each attack event that matches the rule. For more information about the application-level DDoS rulebase, see “Understanding IDP Application-Level DDoS Rulebases” on page 715.

The attack event log contains the following information:

- Time generated (the date/time in which the log is generated)
- Ingress and egress zone and interface information
- Sources and destination IP address and port numbers
- Connection, context, and context value rates
- Time-binding information
- Policy name
- Rulebase name and rule name
- application-level DDoS application name
- Layer 4 protocol
- Application service (such as DNS and HTTP)
- Context and value rates
- Context value (presented in ASCII and hexadecimal formats)
- Action taken on the event

To reduce the volume of application-level DDoS attack event logs, when you configure an application-level DDoS application with time-binding-count in a rule that has logging

enabled, IDP generates an application-level DDoS attack event log only when an attack is detected for time-binding-count times for each time-binding-period seconds. Without time-binding-count configured for an application-level DDoS application, IDP generates an application-level DDoS attack event log for each detected attack, and these logs are subjected to log suppression. The repeat-count field in the log represents how many times this log event would have been sent if log suppression was applied.

IDP generates application-level DDoS state transition event logs when the number of application transactions exceeds or falls behind the configured connection or context hit rate thresholds. State transition event logs are enabled by default, and IDP generates state transition event logs based on user-configured connection, context, or context value thresholds. IDP exhibits hysteresis for state transitions, due to this fact, the state transition log event is generated after incoming traffic connection or context rates have fallen behind by 20 percent (by default) of the configured threshold.



**NOTE:** State transition logging is enabled by default and cannot be enabled or disabled, it is part of the standard system logging.

The state event log contains the following information:

- Time generated (the date/time in which the log is generated)
- IP address of the protected server
- Port
- Interface and zone
- Policy name
- Rulebase name and rule name
- application-level DDoS application name
- Layer 4 protocol
- Application service (such as DNS and HTTP)
- Description of the transition event
- Description of the context value (presented in ASCII and hexadecimal formats)



**NOTE:** The `rulebase-ddos` command is only available for high-end SRX Series devices (SRX3000 and SRX5000 series).

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding IDP Log Suppression Attributes on page 830](#)
- [Understanding IDP Logging on page 827](#)
- [Understanding IDP Log Information Usage on the IC Series UAC Appliance on page 838](#)

- IDP Application-Level DDoS Attack Overview on page 763
- Enabling Attack and IP-Action Logging (CLI Procedure) on page 830

## Enabling Attack and IP-Action Logging (CLI Procedure)

To enable attack and ip-action logging, perform the following steps:

- Enable attack logs

```
[edit security idp]
user@host# set idp-policy AppDDoS-policy-name rulebase-ddos rule
AppDDoS-rule-name then notification log-attacks
```
- Enable ip-action logs

```
[edit security idp]
user@host# set idp-policy AppDDoS-policy-name rulebase-ddos rule
AppDDoS-rule-name then ip-action log
```

Once enabled, the application-level DDoS logs will appear in the regular system logs.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- For information about monitoring events and managing system log files, see the *Junos OS Administration Guide for Security Devices*.
- Understanding Application-Level DDoS Logging on page 828
- Understanding IDP Log Suppression Attributes on page 830
- Understanding IDP Logging on page 827
- Understanding IDP Log Information Usage on the IC Series UAC Appliance on page 838
- IDP Application-Level DDoS Attack Overview on page 763

---

## IDP Log Suppression Attributes

- Understanding IDP Log Suppression Attributes on page 830
- Example: Configuring IDP Log Suppression Attributes on page 831

## Understanding IDP Log Suppression Attributes

Log suppression ensures that minimal numbers of logs are generated for the same event or attack that occurs multiple times. Log suppression is enabled by default. You can configure certain log suppression attributes to suppress logs according to your needs. When configuring log suppression, keep in mind that log suppression can negatively impact sensor performance if you set the reporting interval too high.

You can configure the following log suppression attributes:

- Include destination addresses while performing log suppression—You can choose to combine log records for events with a matching source address. By default, the IDP sensor does not consider destination when matching events for log suppression.
- Number of log occurrences after which log suppression begins—You can specify the number of instances that a specific event must occur before log suppression begins. By default, log suppression begins after the first occurrence.
- Maximum number of logs that log suppression can operate on—When log suppression is enabled, Intrusion Detection and Prevention (IDP) must cache log records so that it can identify when multiple occurrences of the same event occur. You can specify how many log records are tracked simultaneously by IDP. By default, the maximum number of log records that IDP can operate on is 16,384.
- Time after which suppressed logs are reported—When log suppression is enabled, IDP maintains a count of occurrences of the same event. After the specified number of seconds have passed, IDP writes a single log entry containing the count of occurrences. By default, IDP reports suppressed logs after 5 seconds.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding IDP Logging on page 827](#)
- [IDP Policies Overview on page 701](#)
- [Understanding IDP Policy Rules on page 707](#)
- [Example: Configuring IDP Log Suppression Attributes on page 831](#)

### Example: Configuring IDP Log Suppression Attributes

This example shows how to configure log suppression attributes.

#### Requirements

Before you begin:

- Configure network interfaces. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).
- Download the signature database. See “Updating the IDP Signature Database Manually Overview” on page 784.

#### Overview

Log suppression ensures that minimal numbers of logs are generated for the same event or attack that occurs multiple times. Log suppression is enabled by default. You can configure certain log suppression attributes to suppress logs according to your needs.

In this example, you configure log suppression to begin after the second occurrence of an event and specify that logs are reported after 20 seconds.

---

## Configuration

---

### Step-by-Step Procedure

To configure log suppression attributes:

1. Specify the log number after which you want to start log suppression.  
[edit]  
user@host# **set security idp sensor-configuration log suppression start-log 2**
2. Specify the maximum time after which suppressed logs are reported.  
[edit]  
user@host# **set security idp sensor-configuration log suppression max-time-report 20**
3. If you are done configuring the device, commit the configuration.  
[edit]  
user@host# **commit**

---

## Verification

---

To verify log statistics, enter the **show security idp counters log** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Updating the IDP Signature Database Manually Overview on page 784](#)
- [Example: Defining Rules for an IDP IPS Rulebase on page 717](#)
- [Understanding IDP Log Suppression Attributes on page 830](#)

---

## Security Packet Capture

---

- [Understanding Security Packet Capture on page 832](#)
- [Example: Configuring Security Packet Capture on page 833](#)
- [Example: Configuring Packet Capture for Datapath Debugging on page 835](#)
- [Verifying Security Packet Capture on page 838](#)

## Understanding Security Packet Capture

Viewing packets that precede and follow an attack helps you determine the purpose and extent of an attempted attack, whether an attack was successful, and if any network damage was caused by an attack. Packet analysis also aids in defining attack signatures to minimize false positives.

If packet capture is enabled when an attack is logged, a specified number of packets before and after the attack can be captured for the session. When all packets have been collected, they are transmitted in Device Management Interface (DMI) to a host device for offline analysis.

A notification option in the IDP policy rule enables packet capture when a rule match occurs. The option further defines the number of packets to be captured and the duration of packet capture for the associated session.

An IDP sensor configuration defines the device specifications for the packet capture. Options for this command determine the memory to be allocated for packet capture, and the source and host devices between which the packet capture object will be transmitted.

A **show** command displays packet capture counters that provide details about the progress, success, and failure of packet capture activity on the device.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding IDP Logging on page 827](#)
- [Example: Configuring Security Packet Capture on page 833](#)
- [Example: Configuring Packet Capture for Datapath Debugging on page 835](#)
- [Verifying Security Packet Capture on page 838](#)

### Example: Configuring Security Packet Capture

This example shows how to configure the security packet capture.

- [Requirements on page 833](#)
- [Overview on page 833](#)
- [Configuration on page 833](#)
- [Verification on page 835](#)

#### Requirements

Before you begin, configure network interfaces. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).

#### Overview

In this example, you configure a packet capture for rule 1 of policy pol0. The rule specifies that, if an attack occurs, 10 packets before the attack and 3 packets after the attack will be captured, and that the post-attack capture should time out after 60 seconds. The sensor configuration is modified to allocate 5 percent of available memory and 15 percent of the IDP sessions to packet capture. When the packet capture object is prepared, it is transmitted from device 10.56.97.3 to port 5 on device 10.24.45.7.

#### Configuration

#### CLI Quick Configuration

To quickly configure the security packet capture, copy the following commands and paste them into the CLI:

```
[edit]
set security idp idp-policy pol0 rulebase-ips rule 1 then notification packet-log pre-attack
  10 post-attack 3 post-attack-timeout 60
set security idp sensor-configuration packet-log total-memory 5 max-sessions 15
  source-address 10.56.97.3 host 10.24.45.7 port 5
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the security packet capture:

1. Navigate to the notification level for rule 1, policy pol0 in the configuration hierarchy.

```
[edit]
user@host# edit security idp idp-policy pol0 rulebase-ips rule 1 then notification
```

2. Define the size and timing constraints for each packet capture.

```
[edit security idp idp-policy pol0 rulebase-ips rule 1 then notification]
user@host# set packet-log pre-attack 10 post-attack 3 post-attack-timeout 60
```

3. Enable the security idp sensor-configuration.

```
[edit]
user@host# edit security idp sensor-configuration
```

4. Allocate the device resources to be used for packet capture.

```
[edit security idp sensor-configuration]
user@host# set packet-log total-memory 5 max-sessions 15
```

5. Identify the source and host devices for transmitting the packet-capture object.

```
[edit security idp sensor-configuration]
user@host# set packet-log source-address 10.56.97.3 host 10.24.45.7 port 5
```

**Results** From configuration mode, confirm your configuration by entering the **show security idp** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security idp
idp-policy pol0 {
  rulebase-ips {
    rule 1 {
      then {
        notification {
          packet-log {
            pre-attack 10;
            post-attack 3;
            post-attack-timeout 60;
          }
        }
      }
    }
  }
}
sensor-configuration {
  packet-log {
    total-memory 5;
    max-sessions 15;
    source-address 10.56.97.3;
    host {
```



```

10.24.45.7;
port 5;
}
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Security Packet Capture on page 835

#### Verifying Security Packet Capture

**Purpose** Verify security packet capture.

**Action** From operational mode, enter the **show security idp counters packet-log** command.

```
user@host> show security idp counters packet-log
```

| IDP counters:                                             | Value |
|-----------------------------------------------------------|-------|
| Total packets captured since packet capture was activated | 0     |
| Total sessions enabled since packet capture was activated | 0     |
| Sessions currently enabled for packet capture             | 0     |
| Packets currently captured for enabled sessions           | 0     |
| Packet clone failures                                     | 0     |
| Session log object failures                               | 0     |
| Session packet log object failures                        | 0     |
| Sessions skipped because session limit exceeded           | 0     |
| Packets skipped because total memory limit exceeded       | 0     |

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding Security Packet Capture on page 832
  - Example: Configuring Packet Capture for Datapath Debugging on page 835
  - Verifying Security Packet Capture on page 838

### Example: Configuring Packet Capture for Datapath Debugging

This example shows how to configure packet capture to monitor traffic that passes through the device. Packet Capture then dumps the packets into a PCAP file format that can be later examined by the tcpdump utility.

- Requirements on page 835
- Overview on page 836
- Configuration on page 836
- Verification on page 837

### Requirements

Before you begin, see “Debugging the Data Path (CLI Procedure)” on page 22.

## Overview

A filter is defined to filter traffic; then an action profile is applied to the filtered traffic. The action profile specifies a variety of actions on the processing unit. One of the supported actions is packet dump, which sends the packet to the Routing Engine and stores it in proprietary form to be read using the **show security datapath-debug capture** command.

## Configuration

### CLI Quick Configuration

To quickly configure packet capture, copy the following commands and paste them into the CLI.

```
[edit]
set security datapath-debug capture-file my-capture
set security datapath-debug capture-file format pcap
set security datapath-debug capture-file size 1m
set security datapath-debug capture-file files 5
set security datapath-debug maximum-capture-size 400
set security datapath-debug action-profile do-capture event np-ingress packet-dump
set security datapath-debug packet-filter my-filter action-profile do-capture
set security datapath-debug packet-filter my-filter source-prefix 1.2.3.4/32
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure packet capture:

1. Edit the security datapath-debug option for the multiple processing units along the packet-processing path:

```
[edit]
user@host# edit security datapath-debug
```

2. Enable the capture file, the file format, the file size, and the number of files. Size number limits the size of the capture file. After the limit size is reached, if the file number is specified, then the capture file will be rotated to filename *x*, where *x* is auto-incremented until it reaches the specified index and then returns to zero. If no files index is specified, the packets will be discarded after the size limit is reached. The default size is 512 kilobytes.

```
[edit security datapath-debug]
user@host# set capture-file my-capture format pcap size 1m files 5
[edit security datapath-debug]
user@host# set maximum-capture-size 400
```

3. Enable action profile and set the event. Set the action profile as do-capture and the event type as np-ingress:

```
[edit security datapath-debug]
user@host# edit action-profile do-capture
[edit security datapath-debug action-profile do-capture]
user@host# edit event np-ingress
```

4. Enable packet dump for the action profile:

```
[edit security datapath-debug action-profile do-capture event np-ingress]
user@host# set packet-dump
```

5. Enable packet filter, action, and filter options. The packet filter is set to my-filter, the action profile is set to do-capture, and filter option is set to source-prefix 1.2.3.4/32.

```
[edit security datapath-debug]
user@host# set security datapath-debug packet-filter my-filter action-profile
do-capture
```

```
[edit security datapath-debug]
user@host# set security datapath-debug packet-filter my-filter source-prefix
1.2.3.4/32
```

**Results** From configuration mode, confirm your configuration by entering the **show security datapath-debug** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it. The following is output from the **show security datapath-debug** command:

```
security datapath-debug {
  capture-file my-capture format pcap size 1m files 5;
  maximum-capture-size 100
  action-profile do-capture {
    event np-ingress {
      packet-dump
    }
  }
  packet-filter my-filter {
    source-prefix 1.2.3.4/32
    action-profile do-capture }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Packet Capture on page 837

#### **Verifying Packet Capture**

**Purpose** Verify if the packet capture is working.

**Action** From operational mode, enter the **request security datapath-debug capture start** command to start packet capture and enter the **request security datapath-debug capture stop** command to stop packet capture.

To view the results, from CLI operational mode, access the local UNIX shell and navigate to the directory `/var/log/my-capture`. The result can be read by using the `tcpdump` utility.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Debugging the Data Path \(CLI Procedure\) on page 22](#)

- Understanding Data Path Debugging for SRX Series Devices on page 21
- Understanding Security Packet Capture on page 832
- Example: Configuring Security Packet Capture on page 833
- Verifying Security Packet Capture on page 838

## Verifying Security Packet Capture

**Purpose** Monitor packet capture statistics issuing the following **show** command from the CLI prompt.

**Action** user@host> **show security idp counters packet-log**

| IDP counters:                                             | Value |
|-----------------------------------------------------------|-------|
| Total packets captured since packet capture was activated | 0     |
| Total sessions enabled since packet capture was activated | 0     |
| Sessions currently enabled for packet capture             | 0     |
| Packets currently captured for enabled sessions           | 0     |
| Packet clone failures                                     | 0     |
| Session log object failures                               | 0     |
| Session packet log object failures                        | 0     |
| Sessions skipped because session limit exceeded           | 0     |
| Packets skipped because total memory limit exceeded       | 0     |

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding Security Packet Capture on page 832
  - Example: Configuring Security Packet Capture on page 833
  - Example: Configuring Packet Capture for Datapath Debugging on page 835

## Understanding IDP Log Information Usage on the Infranet Controller

The infranet controller for the Unified Access Control (UAC) appliance can use Intrusion Detection and Prevention (IDP) attack log information sent from the Juniper Networks device to apply access policies for traffic in which IDP logs indicate an attack has been detected. Using a secure channel of communication, these IDP logs are sent to the infranet controller directly and securely. IDP attack logs are sent to the infranet controller through the JUEP communication channel.

This topic contains the following sections:

- Message Filtering to the Infranet Controller on page 838
- Configuring Infranet Controller Logging on page 839

### Message Filtering to the Infranet Controller

When you configure the infranet controller to receive IDP log messages, you set certain filtering parameters on the infranet controller. Without this filtering, the infranet controller could potentially receive too many log messages. The filtering parameters could include the following:

- The infranet controller should only receive communications from IDP for sessions it has authenticated. See the *Unified Access Control Administration Guide* for details.
- You can create infranet controller filters for receiving IDP logs files based on their severity. For example, if on the infranet controller the severity is set to high, then IDP only sends logs which have a severity greater than or equal to high. See the *Unified Access Control Administration Guide* for details.
- From the infranet controller, you can disable the receiving of all IDP logs. See the *Unified Access Control Administration Guide* for details.

## Configuring Infranet Controller Logging

All the configuration for receiving and filtering IDP logs is done on the infranet controller. You should refer to the *Unified Access Control Administration Guide* for configuration information for receiving IDP logs and details on the JUEP communication channel.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding IDP Log Suppression Attributes on page 830](#)
- [Understanding IDP Logging on page 827](#)
- [Understanding Application-Level DDoS Logging on page 828](#)



## PART 8

# Unified Threat Management

- Unified Threat Management Overview on page 843
- Antispam Filtering on page 851
- Full Antivirus Protection on page 869
- Express Antivirus Protection on page 929
- Sophos Antivirus Protection on page 951
- Content Filtering on page 969
- Web Filtering on page 985





# Unified Threat Management Overview

- Unified Threat Management Overview on page 843
- Understanding UTM Custom Objects on page 844
- UTM Licensing on page 845
- WELF Logging for UTM Features on page 846

## Unified Threat Management Overview

---

Unified Threat Management (UTM) is a term used to describe the consolidation of several security features into one device, protecting against multiple threat types. The advantage of UTM is streamlined installation and management of these multiple security capabilities.

The security features provided as part of the UTM solution are:

- **Antispam**—E-mail spam consists of unwanted e-mail messages, usually sent by commercial, malicious, or fraudulent entities. The antispam feature examines transmitted e-mail messages to identify e-mail spam. When the device detects an e-mail message deemed to be spam, it either drops the message or tags the message header or subject field with a preprogrammed string. The antispam feature uses a constantly updated spam block list (SBL). Sophos updates and maintains the IP-based SBL. The antispam feature is a separately licensed subscription service.
- **Full File-Based Antivirus**—A virus is executable code that infects or attaches itself to other executable code to reproduce itself. Some malicious viruses erase files or lock up systems. Other viruses merely infect files and overwhelm the target host or network with bogus data. The full file-based antivirus feature provides file-based scanning on specific Application Layer traffic checking for viruses against a virus signature database. It collects the received data packets until it has reconstructed the original application content, such as an e-mail file attachment, and then scans this content. Kaspersky Lab provides the internal scan engine. The full file-based antivirus scanning feature is a separately licensed subscription service.
- **Express Antivirus**—Express antivirus scanning is offered as a less CPU intensive alternative to the full file-based antivirus feature. The express antivirus feature, like the full antivirus feature, scans specific Application Layer traffic for viruses against a virus signature database. However, unlike full antivirus, express antivirus does not reconstruct the original application content. Rather, it just sends (streams) the received data packets, as is, to the scan engine. With express antivirus, the virus scanning is

executed by a hardware pattern matching engine. This improves performance while scanning is occurring, but the level of security provided is lessened. Juniper Networks provides the scan engine. The express antivirus scanning feature is a separately licensed subscription service.

- **Content Filtering**—Content filtering blocks or permits certain types of traffic based on the MIME type, file extension, protocol command, and embedded object type. Content filtering does not require a separate license.
- **Web Filtering**—Web filtering lets you manage Internet usage by preventing access to inappropriate Web content. There are three types of Web filtering solutions. In the case of the integrated Web filtering solution, the decision-making for blocking or permitting Web access is done on the device after it identifies the category for a URL either from user-defined categories or from a category server (Websense provides the CPA Server). The integrated Web filtering feature is a separately licensed subscription service. The redirect Web filtering solution intercepts HTTP requests and forwards the server URL to an external URL filtering server provided by Websense to determine whether to block or permit the requested Web access. Redirect Web filtering does not require a separate license. With Juniper Local Web Filtering, the decision-making for blocking or permitting Web access is done on the device after it identifies the category for a URL from user-defined categories stored on the device. With Local filtering, there is no additional Juniper license or remote category server required.

#### **Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding UTM Custom Objects on page 844](#)
- [Understanding UTM Licensing on page 845](#)
- [Updating UTM Licenses \(CLI Procedure\) on page 846](#)
- [Understanding WELF Logging for UTM Features on page 846](#)
- [Example: Configuring WELF Logging for UTM Features on page 847](#)

---

## Understanding UTM Custom Objects

Before you can configure most UTM features, you must first configure the custom objects for the feature in question. Custom objects are global parameters for UTM features. This means that configured custom objects can be applied to all UTM policies where applicable, rather than only to individual policies.

The following UTM features make use of certain custom objects:

- Anti-Virus (see “Full Antivirus Pattern Update Configuration Overview” on page 871)
- Web Filtering (see “Example: Configuring Integrated Web Filtering” on page 989)
- Anti-Spam (see “Server-Based Antispam Filtering Configuration Overview” on page 853)
- Content Filtering (see “Content Filtering Configuration Overview” on page 972)

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Unified Threat Management Overview on page 843
  - Understanding UTM Licensing on page 845
  - Updating UTM Licenses (CLI Procedure) on page 846
  - Understanding WELF Logging for UTM Features on page 846
  - Example: Configuring WELF Logging for UTM Features on page 847

## UTM Licensing

- Understanding UTM Licensing on page 845
- Updating UTM Licenses (CLI Procedure) on page 846

### Understanding UTM Licensing

The majority of UTM features function as a subscription service requiring a license. You can redeem this license once you have purchased your subscription license SKUs. You redeem your license by entering your authorization code and chassis serial number into the Customer Service LMS interface. Once your entitlement is generated, you can use the CLI from your device to send a license update request to the LMS server. The LMS server then sends your subscription license directly to the device.



**NOTE:** UTM requires 1 GB of memory. If your J2320, J2350, or J4350 device has only 512 MB of memory, you must upgrade the memory to 1 GB to run UTM.

**Table 102: UTM Feature Subscription Service License Requirements**

| UTM Feature               | Requires License |
|---------------------------|------------------|
| Antispam                  | Yes              |
| Antivirus: full           | Yes              |
| Antivirus: express        | Yes              |
| Content Filtering         | No               |
| Web Filtering: integrated | Yes              |
| Web Filtering: redirect   | No               |
| Web Filtering: local      | No               |

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Unified Threat Management Overview on page 843

- Understanding UTM Custom Objects on page 844
- Updating UTM Licenses (CLI Procedure) on page 846
- Understanding WELF Logging for UTM Features on page 846
- Example: Configuring WELF Logging for UTM Features on page 847

## Updating UTM Licenses (CLI Procedure)

To apply your UTM subscription license to the device, use the following CLI command:

```
user@host> request system license update
```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Unified Threat Management Overview on page 843
- Understanding UTM Custom Objects on page 844
- Understanding UTM Licensing on page 845
- Understanding WELF Logging for UTM Features on page 846
- Example: Configuring WELF Logging for UTM Features on page 847

## WELF Logging for UTM Features

---

- Understanding WELF Logging for UTM Features on page 846
- Example: Configuring WELF Logging for UTM Features on page 847

## Understanding WELF Logging for UTM Features

UTM features support the WELF standard. The WELF Reference defines the WebTrends industry standard log file exchange format. Any system logging to this format is compatible with Firewall Suite 2.0 and later, Firewall Reporting Center 1.0 and later, and Security Reporting Center 2.0 and later.

A WELF log file is composed of records. Each record is a single line in the file. Records are always in chronological order. The earliest record is the first record in the file; the most recent record is the last record in the file. WELF places no restrictions on log filenames or log file rotation policies.



**NOTE:** Each WELF record is composed of fields. The record identifier field (**id=**) must be the first field in a record. All other fields can appear in any order.

---

The following is a sample WELF record:

```
id=firewall time="2000-2-4 12:01:01" fw=192.168.0.238 pri=6 rule=3 proto=http
src=192.168.0.23 dst=6.1.0.36 rg=www.webtrends.com/index.html op=GET result=0
rcvd=1426
```

The fields from the example WELF record include the following required elements (all other fields are optional):

- **id** (Record identifier)
- **time** (Date/time)
- **fw** (Firewall IP address or name)
- **pri** (Priority of the record)

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Unified Threat Management Overview on page 843](#)
- [Understanding UTM Custom Objects on page 844](#)
- [Understanding UTM Licensing on page 845](#)
- [Updating UTM Licenses \(CLI Procedure\) on page 846](#)
- [Example: Configuring WELF Logging for UTM Features on page 847](#)

### Example: Configuring WELF Logging for UTM Features

This example shows how to configure WELF logging for UTM features.

- [Requirements on page 847](#)
- [Overview on page 847](#)
- [Configuration on page 847](#)
- [Verification on page 849](#)

#### Requirements

Before you begin, review the fields used to create a WELF log file and record. See “Understanding WELF Logging for UTM Features” on page 846.

#### Overview

A WELF log file is composed of records. Each record is a single line in the file. Records are always in chronological order. The earliest record is the first record in the file; the most recent record is the last record in the file. WELF places no restrictions on log filenames or log file rotation policies. In this example, the severity level is emergency and the name of the security log stream is **stream-utm-welf**.

#### Configuration

#### CLI Quick Configuration

To quickly configure WELF logging for UTM features, copy the following commands and paste them into the CLI.

```
[edit]
set security log source-address 1.2.3.4 stream utm-welf
```

```

set security log source-address 1.2.3.4 stream utm-welf format welf
set security log source-address 1.2.3.4 stream utm-welf format welf category
content-security
set security log source-address 1.2.3.4 stream utm-welf format welf category
content-security severity emergency
set security log source-address 1.2.3.4 stream utm-welf format welf category
content-security severity emergency host 5.6.7.8

```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure WELF logging for UTM features:

1. Set the security log source IP address.

```

[edit security log]
user@host# set source-address 1.2.3.4

```



**NOTE:** You must save the WELF logging messages to a dedicated WebTrends server.

2. Name the security log stream.

```

[edit security log]
user@host# set source-address 1.2.3.4 stream utm-welf

```

3. Set the format for the log messages.

```

[edit security log]
user@host# set source-address 1.2.3.4 stream utm-welf format welf

```

4. Set the category of log messages that are sent.

```

[edit security log]
user@host# set source-address 1.2.3.4 stream utm-welf format welf category
content-security

```

5. Set the severity level of log messages that are sent.

```

[edit security log]
user@host# set source-address 1.2.3.4 stream utm-welf format welf category
content-security severity emergency

```

6. Enter the host address of the dedicated WebTrends server to which the log messages are to be sent.

```

[edit security log]
user@host# set source-address 1.2.3.4 stream utm-welf format welf category
content-security severity emergency host 5.6.7.8

```

**Results** From configuration mode, confirm your configuration by entering the **show security log** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]

```

```
user@host# show security log
stream utm-welf {
  severity emergency;
  format welf;
  category content-security;
  host {
    5.6.7.8;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

---

To confirm that the configuration is working properly, perform this task:

- Verifying the Security Log on page 849

#### **Verifying the Security Log**

**Purpose** Verify that the WELF log for UTM features is complete.

**Action** From operational mode, enter the **show security utm status** command to verify if the UTM service is running or not.

#### **Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Unified Threat Management Overview](#) on page 843
- [Understanding UTM Custom Objects](#) on page 844
- [Understanding UTM Licensing](#) on page 845
- [Updating UTM Licenses \(CLI Procedure\)](#) on page 846





# Antispam Filtering

- [Antispam Filtering Overview on page 851](#)
- [Server-Based Spam Filtering on page 851](#)
- [Local List Spam Filtering on page 859](#)
- [Understanding Spam Message Handling on page 867](#)

## Antispam Filtering Overview

---

Spam consists of unwanted e-mail messages, usually sent by commercial, malicious, or fraudulent entities. The antispam feature examines transmitted e-mail messages to identify spam. When the device detects a message deemed to be spam, it blocks the e-mail message or tags the e-mail message header or subject with a preprogrammed string.

Antispam filtering allows you to use both a third-party server-based spam block list (SBL) and to optionally create your own local whitelists (benign) and blacklists (malicious) for filtering against e-mail messages. The antispam feature is not meant to replace your antispam server, but to complement it.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Server-Based Antispam Filtering on page 852](#)
- [Server-Based Antispam Filtering Configuration Overview on page 853](#)
- [Understanding Local List Antispam Filtering on page 859](#)
- [Local List Antispam Filtering Configuration Overview on page 860](#)
- [Understanding Spam Message Handling on page 867](#)

## Server-Based Spam Filtering

---

- [Understanding Server-Based Antispam Filtering on page 852](#)
- [Server-Based Antispam Filtering Configuration Overview on page 853](#)
- [Example: Configuring Server-Based Antispam Filtering on page 853](#)

## Understanding Server-Based Antispam Filtering



**NOTE:** Server-based spam filtering supports only IP-based spam block list blacklist lookup. Sophos updates and maintains the IP-based spam block list. Server-based antispam filtering is a separately licensed subscription service. When your antispam license key expires, you can continue to use locally defined black and whitelists.

Server-based antispam filtering requires Internet connectivity with the spam block list (SBL) server. Domain Name Service (DNS) is required to access the SBL server. The firewall performs SBL lookups through the DNS protocol. The lookups are against the IP address of the sender (or relaying agent) of the e-mail, adding the name of the SBL server as the authoritative domain. The DNS server then forwards each request to the SBL server, which returns a DNS response to the device. The device then interprets the DNS response to determine if the e-mail sender is a spammer.

IP addresses that are included in the block lists are generally considered to be invalid addresses for mail servers or easily compromised addresses. Criteria for listing an IP address as a spammer on the SBL can include:

- Running an SMTP open relay service
- Running open proxy servers (of various kinds)
- Being a zombie host possibly compromised by a virus, worm, Trojan, or spyware
- Using a dynamic IP range
- Being a confirmed spam source with a known IP address

By default, the device first checks incoming e-mail against local white and blacklists. If there are no local lists, or if the sender is not found on local lists, the device proceeds to query the SBL server over the Internet. When both server-based spam filtering and local list spam filtering are enabled, checks are done in the following order:

1. The local whitelist is checked. If there is a match, no further checking is done. If there is no match...
2. The local blacklist is checked. If there is a match, no further checking is done. If there is no match...
3. The SBL server list is checked.



**NOTE:** SBL server matching stops when the antispam license key is expired.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Antispam Filtering Overview on page 851](#)
- [Server-Based Antispam Filtering Configuration Overview on page 853](#)

- Example: Configuring Server-Based Antispam Filtering on page 853
- Local List Antispam Filtering Configuration Overview on page 860
- Understanding Local List Antispam Filtering on page 859
- Understanding Spam Message Handling on page 867

## Server-Based Antispam Filtering Configuration Overview

For each UTM feature, configure feature parameters in the following order:

1. Configure UTM custom objects for the feature:

```
user@host# set security utm custom-objects
```

2. Configure the main feature parameters, using feature profiles.

```
user@host# set security utm feature-profile anti-spam
```

3. Configure a UTM policy for each protocol, and attach this policy to a profile.

```
user@host# set security utm utm-policy utmp1 anti-spam smtp-profile smtp1
```



**NOTE:** Antispam filtering is only supported for the SMTP protocol.

4. Attach the UTM policy to a security policy.

```
user@host# set security policies from-zone trust to-zone untrust policy p1 then permit
application-services utm-policy utmp1
```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Antispam Filtering Overview on page 851
- Understanding Server-Based Antispam Filtering on page 852
- Example: Configuring Server-Based Antispam Filtering on page 853
- Understanding Local List Antispam Filtering on page 859
- Local List Antispam Filtering Configuration Overview on page 860
- Understanding Spam Message Handling on page 867

## Example: Configuring Server-Based Antispam Filtering

This example shows how to configure server-based antispam filtering.

- Requirements on page 854
- Overview on page 854
- Configuration on page 854
- Verification on page 858

## Requirements

Before you begin, review how to configure the feature parameters for each UTM feature. See “Server-Based Antispam Filtering Configuration Overview” on page 853.

## Overview

Server-based antispam filtering requires Internet connectivity with the spam block list (SBL) server. Domain Name Service (DNS) is required to access the SBL server.

## Configuration

### CLI Quick Configuration

To quickly configure server-based antispam filtering, copy the following commands and paste them into the CLI.

```
[edit]
set security utm feature-profile anti-spam sbl profile sblprofile1 sbl-default-server
set security utm feature-profile anti-spam sbl profile sblprofile1 sbl-default-server
  spam-action block
set security utm feature-profile anti-spam sbl profile sblprofile1 sbl-default-server
  custom-tag-string ***spam***
set security utm utm-policy spampolicy1 anti-spam smtp-profile sblprofile1
set security policies from-zone trust to-zone untrust policy utmsecuritypolicy1 match
  source-address any
set security policies from-zone trust to-zone untrust policy utmsecuritypolicy1 match
  destination-address any
set security policies from-zone trust to-zone untrust policy utmsecuritypolicy1 match
  application junos-smtp
set security policies from-zone trust to-zone untrust policy utmsecuritypolicy1 then permit
  application-services utm-policy spampolicy1
```

### J-Web Quick Configuration

To configure server-based antispam filtering:

1. Configure a profile and enable/disable the SBL server lookup. Select **Configure>Security>UTM>Anti-Spam**.
  - a. In the Anti-Spam profiles configuration window, click **Add** to configure a profile for the SBL server, or click **Edit** to modify an existing item.
  - b. In the Profile name box, enter a unique name for the antispam profile that you are creating.
  - c. If you are using the default server, select **Yes** next to Default SBL server. If you are not using the default server, select **No**.



**NOTE:** The SBL server is predefined on the device. The device comes preconfigured with the name and address of the SBL server. If you do not select Yes, you are disabling server-based spam filtering. You should disable it only if you are using only local lists or if you do not have a license for server-based spam filtering.

- d. In the Custom tag string box, enter a custom string for identifying a message as spam. By default, the device uses **\*\*\*SPAM\*\*\***.
  - e. From the antispam action list, select the action that the device should take when it detects spam. Options include Tag subject, Block email, and Tag header.
2. Configure a UTM policy for SMTP to which you attach the antispam profile.
    - a. Select **Configure>Security>Policy>UTM Policies**.
    - b. In the UTM policy configuration window, click **Add**.
    - c. In the policy configuration window, select the **Main** tab.
    - d. In the Policy name box, type a unique name for the UTM policy.
    - e. In the Session per client limit box, type a session per client limit. Valid values range from 0 to 2000.
    - f. From the Session per client over limit list, select the action that the device should take when the session per client limit for this UTM policy is exceeded. Options include Log and permit and Block.
    - g. Select the **Anti-Spam profiles** tab in the pop-up window.
    - h. From the SMTP profile list, select an antispam profile to attach to this UTM policy.
  3. Attach the UTM policy to a security policy.
    - a. Select **Configure>Security>Policy>FW Policies**.
    - b. In the Security Policy window, click **Add** to configure a security policy with UTM or click **Edit** to modify an existing policy.
    - c. In the Policy tab, type a name in the **Policy Name** box.
    - d. Next to From Zone, select a zone from the list.
    - e. Next to To Zone, select a zone from the list.
    - f. Choose a source address.
    - g. Choose a destination address.
    - h. Choose an application by selecting **junos-smtp** (for antispam) in the Application Sets box and move it to the Matched box.
    - i. Next to Policy Action, select one of the following: **Permit**, **Deny**, or **Reject**.



**NOTE:** When you select Permit for Policy Action, several additional fields become available in the Applications Services tab, including UTM Policy.

- j. Select the **Application Services** tab.
- k. Next to UTM Policy, select the appropriate policy from the list. This attaches your UTM policy to the security policy.
- l. Click **OK** to check your configuration and save it as a candidate configuration.
- m. If the policy is saved successfully, you receive a confirmation, and you must click **OK** again. If the profile is not saved successfully, click **Details** in the pop-up window to discover why.



NOTE:

- You must activate your new policy to apply it.
- In SRX Series devices the confirmation window that notifies you that the policy is saved successfully, disappears automatically.

- n. If you are done configuring the device, click **Commit Options>Commit**.

**Step-by-Step  
Procedure**

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure server-based antispam filtering:

1. Create a profile.

```
[edit security]
user@host# set utm feature-profile anti-spam sbl profile sblprofile1
```

2. Enable or disable the default SBL server lookup.

```
[edit security]
user@host# set utm feature-profile anti-spam sbl profile sblprofile1
sbl-default-server
```



NOTE: If you are using server-based antispam filtering, you should type `sbl-default-server` to enable the default SBL server. (The SBL server is predefined on the device. The device comes preconfigured with the name and address of the SBL server.) You should disable server-based antispam filtering using the `no-sbl-default-server` option only if you are using only local lists or if you do not have a license for server-based spam filtering.

3. Configure the action to be taken by the device when spam is detected (block, tag-header, or tag-subject).

```
[edit security]
```

```
user@host# set utm feature-profile anti-spam sbl profile sblprofile1 sbl-default-server
spam-action block
```

4. Configure a custom string for identifying a message as spam.

```
[edit security]
user@host# set utm feature-profile anti-spam sbl profile sblprofile1
sbl-default-server custom-tag-string ***spam***
```

5. Attach the spam feature profile to the UTM policy.

```
[edit security]
user@host# set utm utm-policy spampolicy1 anti-spam smtp-profile sblprofile1
```

6. Configure a security policy for UTM to which to attach the UTM policy.

```
[edit]
user@host# set security policies from-zone trust to-zone untrust policy
utmsecuritypolicy1 match source-address any
user@host# set security policies from-zone trust to-zone untrust policy
utmsecuritypolicy1 match destination-address any
user@host# set security policies from-zone trust to-zone untrust policy
utmsecuritypolicy1 match application junos-smtp
user@host# set security policies from-zone trust to-zone untrust policy
utmsecuritypolicy1 then permit application-services utm-policy spampolicy1
```



**NOTE:** The device comes preconfigured with a default antispam policy. The policy is called `junos-as-defaults`. It contains the following configuration parameters:

```
anti-spam {
  sbl {
    profile junos-as-defaults {
      sbl-default-server;
      spam-action block;
      custom-tag-string "***SPAM***";
    }
  }
}
```

**Results** From configuration mode, confirm your configuration by entering the `show security utm` and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security utm
feature-profile {
  anti-spam {
    sbl {
      profile sblprofile1 {
        sbl-default-server;
        spam-action block;
        custom-tag-string ***spam***;
      }
    }
  }
}
```

```

    }
  }
  utm-policy spampolicy1 {
    anti-spam {
      smtp-profile sblprofile1;
    }
  }
}

[edit]
user@host# show security policies
from-zone trust to-zone untrust {
  policy utmsecuritypolicy1 {
    match {
      source-address any;
      destination-address any;
      application junos-smtp;
    }
    then {
      permit {
        application-services {
          utm-policy spampolicy1;
        }
      }
    }
  }
}
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Antispam Statistics on page 858

### *Verifying Antispam Statistics*

**Purpose** Verify the antispam statistics.

**Action** From operational mode, enter the **show security utm anti-spam status** and **show security utm anti-spam statistics** commands.

The following information appears:

```

SBL Whitelist Server:
SBL Blacklist Server:
msgsecurity.juniper.net
DNS Server:
Primary : 1.2.3.4, Src Interface: ge-0/0/0
Secondary: 2.3.4.5, Src Interface: ge-0/0/1
Ternary : 0.0.0.0, Src Interface: fe-0/0/2

Total connections: #
Denied connections: #
Total greetings: #
Denied greetings: #
Total e-mail scanned: #

```



White list hit: #  
 Black list hit: #  
 Spam total: #  
 Spam tagged: #  
 Spam dropped: #  
 DNS errors: #  
 Timeout errors: #  
 Return errors: #  
 Invalid parameter errors: #  
 Statistics start time:  
 Statistics for the last 10 days.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Antispam Filtering Overview on page 851](#)
- [Understanding Server-Based Antispam Filtering on page 852](#)
- [Server-Based Antispam Filtering Configuration Overview on page 853](#)
- [Understanding Local List Antispam Filtering on page 859](#)
- [Local List Antispam Filtering Configuration Overview on page 860](#)
- [Understanding Spam Message Handling on page 867](#)

## Local List Spam Filtering

- [Understanding Local List Antispam Filtering on page 859](#)
- [Local List Antispam Filtering Configuration Overview on page 860](#)
- [Example: Configuring Local List Antispam Filtering on page 861](#)

### Understanding Local List Antispam Filtering

When creating your own local whitelist and blacklist for antispam filtering, you can filter against domain names, e-mail addresses, and/or IP addresses. Pattern matching works a bit differently depending upon the type of matching in question. For example, pattern matching for domain names uses a longest suffix match algorithm. If the sender e-mail address has a domain name of aaa.bbb.ccc, the device tries to match "aaa.bbb.ccc" in the list. If no match is found, it tries to match "bbb.ccc", and then "ccc". IP address matching, however, does not allow for partial matches.

Antispam filtering uses local lists for matching in the following manner:

1. **Sender IP:** The sender IP is checked against the local whitelist, then the local blacklist, and then the SBL IP-based server (if enabled).
2. **Sender Domain:** The domain name is checked against the local whitelist and then against the local blacklist.
3. **Sender E-mail Address:** The sender e-mail address is checked against the local whitelist and then against the local blacklist.

By default, the device first checks incoming e-mail against the local whitelist and blacklist. If the sender is not found on either list, the device proceeds to query the SBL server over the Internet. When both server-based antispam filtering and local list antispam filtering are enabled, checks are done in the following order:

1. The local whitelist is checked. If there is a match, no further checking is done. If there is no match...
2. The local blacklist is checked. If there is a match, no further checking is done. If there is no match...
3. The SBL server list is checked.



**NOTE:** Local black and whitelist matching continues after the antispam license key is expired.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Antispam Filtering Overview on page 851](#)
- [Local List Antispam Filtering Configuration Overview on page 860](#)
- [Example: Configuring Local List Antispam Filtering on page 861](#)
- [Server-Based Antispam Filtering Configuration Overview on page 853](#)
- [Understanding Spam Message Handling on page 867](#)

## Local List Antispam Filtering Configuration Overview

For each UTM feature, configure feature parameters in the following order:

1. Configure UTM custom objects for the feature:

```
user@host# set security utm custom-objects url-pattern url-pattern-name
```

2. Configure the main feature parameters, using feature profiles.

```
user@host# set security utm feature-profile anti-spam as-profile-name
```

3. Configure a UTM policy for each protocol, and attach this policy to a profile.

```
user@host# set security utm utm-policy utmp1 anti-spam smtp-profile smtp1
```

4. Attach the UTM policy to a security policy.

```
user@host# set security policies from-zone trust to-zone untrust policy p1 then permit
application-services utm-policy utmp1
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Antispam Filtering Overview on page 851](#)
- [Understanding Local List Antispam Filtering on page 859](#)
- [Example: Configuring Local List Antispam Filtering on page 861](#)

- Understanding Server-Based Antispam Filtering on page 852
- Understanding Spam Message Handling on page 867

## Example: Configuring Local List Antispam Filtering

This example shows how to configure local list antispam filtering.

- Requirements on page 861
- Overview on page 861
- Configuration on page 861
- Verification on page 866

### Requirements

Before you begin, review how to configure the feature parameters for each UTM feature. See “Local List Antispam Filtering Configuration Overview” on page 860.

### Overview

Antispam filtering uses local lists for matching. When creating your own local whitelist and blacklist for antispam filtering, you can filter against domain names, e-mail addresses, and/or IP addresses.

### Configuration

#### CLI Quick Configuration

To quickly configure local list antispam filtering, copy the following commands and paste them into the CLI:

```
[edit]
set security utm custom-objects url-pattern as-black value [150.61.8.134]
set security utm custom-objects url-pattern as-white value [150.1.2.3]
set security utm feature-profile anti-spam address-whitelist as-white
set security utm feature-profile anti-spam sbl profile localprofile1
set security utm feature-profile anti-spam sbl profile localprofile1 spam-action block
set security utm feature-profile anti-spam sbl profile localprofile1 custom-tag-string
***spam***
set security utm utm-policy spampolicy2 anti-spam smtp-profile localprofile1
set security policies from-zone trust to-zone untrust policy utmsecuritypolicy2 match
source-address any
set security policies from-zone trust to-zone untrust policy utmsecuritypolicy2 match
destination-address any
set security policies from-zone trust to-zone untrust policy utmsecuritypolicy2 match
application junos-smtp
set security policies from-zone trust to-zone untrust policy utmsecuritypolicy2 then permit
application-services utm-policy spampolicy2
```

#### J-Web Quick Configuration

To configure local list antispam filtering:

1. Create local whitelist and blacklist custom objects by configuring a URL pattern list.
  - a. Select **Configure>Security>UTM>Custom Objects**.
  - b. In the UTM custom objects configuration window, select the **URL Pattern List** tab.

- c. Click **Add** to create URL pattern lists.
- d. Next to URL Pattern Name, type a unique name.



.....

**NOTE:** If you are creating a whitelist, it is helpful to indicate this in the list name. The same applies to a blacklist. The name you enter here becomes available in the Address Whitelist and Address Blacklist fields when you are configuring your antispam profiles.

.....

- e. Next to URL Pattern Value, type the URL pattern for whitelist or blacklist antispam filtering.
2. Configure antispam filtering to use the whitelist and blacklist custom objects.
    - a. Select **Configure>Security>UTM>Global options**.
    - b. In the right pane, select the **Anti-Spam** tab.
    - c. Under Anti-Spam, select an Address Whitelist and/or an Address Blacklist from the list for local lists for spam filtering. (These lists are configured as custom objects.)
    - d. Click **OK**.
    - e. If the configuration item is saved successfully, you receive a confirmation, and you must click **OK** again. If it is not saved successfully, click **Details** in the pop-up window to discover why.
    - f. In the left pane under Security, select the **Anti-Spam** tab.
    - g. Click **Add** to configure an anti-spam profile. The profile configuration pop-up window appears.
    - h. In the Profile name box, enter a unique name.
    - i. If you are using the default server, select **Yes** beside Default SBL server. If you are not using the default server, select **No**.



.....

**NOTE:** If you select No, you are disabling server-based spam filtering. You disable it only if you are using local lists or if you do not have a license for server-based spam filtering.

.....

- j. In the Custom tag string box, type a custom string for identifying a message as spam. By default, the device uses **\*\*\*SPAM\*\*\***.
  - k. In the Actions list, select the action that the device should take when it detects spam. Options include Tag subject, Block email, and Tag header.
3. Configure a UTM policy for SMTP to which you attach the antispam profile.
    - a. Select **Configure>Security>Policy>UTM Policies**.
    - b. In the UTM policy configuration window, click **Add** to configure a UTM policy. The policy configuration pop-up window appears.
    - c. Select the **Main** tab.
    - d. In the Policy name box, type a unique name.
    - e. In the Session per client limit box, type a session per client limit. Valid values range from 0 through 2000.
    - f. From the Session per client over limit list, select the action that the device should take when the session per client limit for this UTM policy is exceeded. Options include Log and permit and Block.
    - g. Select the **Anti-Spam profiles** tab.
    - h. From the SMTP profile list, select the antispam profile that you are attaching to this UTM policy.
  4. Attach the UTM policy to a security policy.
    - a. Select **Configure>Security>Policy>FW Policies**.
    - b. In the Security Policy window, click **Add** to configure a security policy with UTM. The policy configuration pop-up window appears.
    - c. In the Policy tab, type a name in the **Policy Name** box.
    - d. Next to From Zone, select a zone from the list.
    - e. Next to To Zone, select a zone from the list.
    - f. Choose a source address.
    - g. Choose a destination address.
    - h. Choose an application by selecting **junos-smtp** (for antispam) in the Application Sets box and move it to the Matched box.
    - i. Next to Policy Action, select one of the following: **Permit, Deny, or Reject**.



NOTE: When you select Permit for policy action, several additional fields become available in the Applications Services tab, including UTM Policy.

- j. Select the **Application Services** tab.
- k. Next to UTM Policy, select the appropriate policy from the list. This attaches your UTM policy to the security policy.
- l. Click **OK** to check your configuration and save it as a candidate configuration.
- m. If the policy is saved successfully, you receive a confirmation, and you must click **OK** again. If the profile is not saved successfully, click **Details** in the pop-up window to discover why.



**NOTE:** You must activate your new policy to apply it.

- n. If you are done configuring the device, click **Commit Options>Commit**.

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure local list antispam filtering:

1. Configure the local list spam blocking by first creating your global local spam lists.
 

```
[edit security]
user@host# set utm custom-objects url-pattern as-black value [150.61.8.134]
user@host# set utm custom-objects url-pattern as-white value [150.1.2.3]
```

2. Configure the local list antispam feature profile by first attaching your custom-object blacklist or whitelist or both.

```
[edit security]
user@host# set utm feature-profile anti-spam address-whitelist as-white
```



**NOTE:** When both the whitelist and the blacklist are in use, the whitelist is checked first. If there is no match, then the blacklist is checked.

3. Configure a profile for your local list spam blocking.

```
[edit security]
user@host# set utm feature-profile anti-spam sbl profile localprofile1
```



**NOTE:** Although you are not using the sbl for local list spam blocking, you configure your profile from within that command similar to the server-based spam blocking procedure.

4. Configure the action to be taken by the device when spam is detected (block, tag-header, tag-subject).

```
[edit security]
user@host# set utm feature-profile anti-spam sbl profile localprofile1 spam-action
block
```

5. Configure a custom string for identifying a message as spam.

```
[edit security]
user@host# set utm feature-profile anti-spam sbl profile localprofile1
custom-tag-string ***spam***
```

6. Attach the spam feature profile to the UTM policy.

```
[edit security]
user@host# set utm utm-policy spampolicy2 anti-spam smtp-profile localprofile1
```

7. Configure a security policy for UTM, and attach the UTM policy to the security policy.

```
[edit]
user@host# set security policies from-zone trust to-zone untrust policy
utmsecuritypolicy2 match source-address any
user@host# set security policies from-zone trust to-zone untrust policy
utmsecuritypolicy2 match destination-address any
user@host# set security policies from-zone trust to-zone untrust policy
utmsecuritypolicy2 match application junos-smtp
user@host# set security policies from-zone trust to-zone untrust policy
utmsecuritypolicy2 then permit application-services utm-policy spampolicy2
```

**Results** From configuration mode, confirm your configuration by entering the **show security utm** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security utm
custom-objects {
  anti-spam {
    url-pattern patternwhite;
    address-whitelist as-white;
    sbl {
      profile localprofile1 {
        spam-action block;
        custom-tag-string ***spam***;
      }
    }
  }
}
utm-policy spampolicy2 {
  anti-spam {
    smtp-profile localprofile1;
  }
}
```

```
[edit]
user@host# show security policies
from-zone trust to-zone untrust {
  policy utmsecuritypolicy2 {
    match {
      source-address any;
      destination-address any;
      application junos-smtp;
    }
  }
}
```

```

    }
    then {
      permit {
        application-services {
          utm-policy spampolicy2;
        }
      }
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Antispam Statistics on page 866

#### *Verifying Antispam Statistics*

**Purpose** Verify the antispam statistics.

**Action** From operational mode, enter the **show security utm anti-spam status** and **show security utm anti-spam statistics** commands.

The following information appears:

```

SBL Whitelist Server:
SBL Blacklist Server:
msgsecurity.juniper.net
DNS Server:
Primary : 1.2.3.4, Src Interface: ge-0/0/0
Secondary: 2.3.4.5, Src Interface: ge-0/0/1
Ternary : 0.0.0.0, Src Interface: fe-0/0/2

Total connections: #
Denied connections: #
Total greetings: #
Denied greetings: #
Total e-mail scanned: #
White list hit: #
Black list hit: #
Spam total: #
Spam tagged: #
Spam dropped: #
DNS errors: #
Timeout errors: #
Return errors: #
Invalid parameter errors: #
Statistics start time:
Statistics for the last 10 days.

```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Antispam Filtering Overview on page 851



- [Understanding Local List Antispam Filtering on page 859](#)
- [Local List Antispam Filtering Configuration Overview on page 860](#)
- [Understanding Spam Message Handling on page 867](#)

## Understanding Spam Message Handling

There are two possible actions the device can take when spam is detected. It can perform a drop action or a tag action.

- [Blocking Detected Spam on page 867](#)
- [Tagging Detected Spam on page 867](#)

### Blocking Detected Spam

The device can block and drop detected spam at either the connection level or the e-mail level:

- [Blocking spam at the connection level](#)

When the SMTP sender is identified as a spam sender based on its IP address, the SMTP connection is rejected and dropped. An error message with a proper error code from the firewall is sent out on behalf of the SMTP server. An example of such an error message is:

```
554 Transaction failed due to anti spam setting
```

- [Blocking spam at the e-mail level](#)

When a particular e-mail sender is identified as spam sender based on its sender address, the e-mail is rejected and dropped. An error message with a proper error code from the firewall is sent back to the sender on behalf of the server. An example of such an error message is:

```
550 Requested action not taken: mailbox unavailable
```

### Tagging Detected Spam

The device can allow and tag the e-mail if the message sender is detected as a spammer. This tagging can occur at the connection level so that all the e-mails for the connection in question are tagged. Otherwise, you can tag only an individual e-mail. Two tagging methods are supported:

- [Tag the subject:](#) A user-defined string is added at the beginning of the subject of the e-mail.
- [Tag the header:](#) A user-defined string is added to the e-mail header.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Antispam Filtering Overview on page 851](#)
- [Server-Based Antispam Filtering Configuration Overview on page 853](#)

- [Local List Antispam Filtering Configuration Overview on page 860](#)

## CHAPTER 32

# Full Antivirus Protection

- Full Antivirus Protection Overview on page 869
- Full Antivirus Scanner Pattern Database on page 870
- Full Antivirus File Scanning on page 875
- Full Antivirus Application Protocol Scanning on page 886
- Full Antivirus Scan Results and Notification Options on page 899
- Full Antivirus Configuration Overview on page 906
- Configuring Full Antivirus (J-Web Procedure) on page 907
- Example: Configuring Full Antivirus (CLI) on page 914
- Monitoring Antivirus Sessions and Scan Results on page 924

## Full Antivirus Protection Overview

---

A virus is executable code that infects or attaches itself to other executable code in order to reproduce itself. Some malicious viruses erase files or lock up systems, while other viruses merely infect files and can overwhelm the target host or network with bogus data. The full file-based antivirus feature provides file-based scanning on specific Application Layer traffic checking for viruses against a virus signature database. It collects the received data packets until it has reconstructed the original application content, such as an e-mail file attachment, and then scans this content.

The full file-based antivirus scanning feature is a separately licensed subscription service. Kaspersky Lab provides the scan engine for full file-based antivirus. When your antivirus license key expires, you can continue to use locally stored antivirus signatures without any updates. But in that case, if the local database is deleted, antivirus scanning is disabled.



**NOTE:** The express antivirus feature provides better performance but lower security. Note that if you switch from full file-based antivirus protection to express antivirus protection, you must reboot the device in order for express antivirus to begin working.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

- Understanding Full Antivirus Pattern Updates on page 870
- Full Antivirus Pattern Update Configuration Overview on page 871
- Understanding Full Antivirus Scan Level Settings on page 876
- Understanding the Full Antivirus Internal Scan Engine on page 875
- Full Antivirus Configuration Overview on page 906

## Full Antivirus Scanner Pattern Database

---

- Understanding Full Antivirus Pattern Updates on page 870
- Full Antivirus Pattern Update Configuration Overview on page 871
- Example: Configuring the Full Antivirus Pattern Update Server on page 872
- Example: Automatically Updating Full Antivirus Patterns (J-Web) on page 873
- Example: Automatically Updating Full Antivirus Patterns on page 874
- Manually Updating, Reloading, and Deleting Full Antivirus Patterns (CLI Procedure) on page 875

## Understanding Full Antivirus Pattern Updates

The full file-based antivirus protection signature database is called the Juniper Full antivirus database (downloaded by the **pattern-update** command). This database is different from the database used by express antivirus. It detects all destructive malicious code, including viruses (polymorphic and other advanced virus types), worms, Trojans, and malware.

Updates to the pattern file are added as new viruses are discovered. When Kaspersky Lab updates the signatures in its pattern database, the security device downloads these updates so that the antivirus scanner is using the latest, most up-to-date signatures when scanning traffic. The security device can perform these updates automatically (the default), or you can perform pattern update downloads manually.

The database pattern server is accessible through HTTP or HTTPS. By default, the antivirus module checks for database updates automatically every 60 minutes. You can change this interval and you can trigger updates manually, as well. The number of files that are downloaded during an update and the duration of the download process can vary.

A local copy of the pattern database is saved in persistent data storage (that is, the flash disk). If the device is rebooted, the local copy remains available for the antivirus scan engine to use during the antivirus scan engine initialization time, without the need for network access to the pattern database server.



**NOTE:** If the auto-update fails, the updater automatically retries to update three more times. If the database download continues to fail, the updater stops trying and waits for the next periodic update before trying again.

---



**NOTE:** Once your subscription expires, you have a 30 day grace period during which you can continue to update the antivirus pattern file. Once that grace period expires, the update server no longer permits antivirus pattern file updates.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Full Antivirus Pattern Update Configuration Overview on page 871
- Example: Configuring the Full Antivirus Pattern Update Server on page 872
- Example: Automatically Updating Full Antivirus Patterns (J-Web) on page 873
- Example: Automatically Updating Full Antivirus Patterns on page 874
- Manually Updating, Reloading, and Deleting Full Antivirus Patterns (CLI Procedure) on page 875

## Full Antivirus Pattern Update Configuration Overview

Before you begin, there are several prerequisites that must be met in order to perform a successful pattern database update:

- You must have a valid antivirus scanner license.
- You must have network connectivity and access to the pattern database server.
- Your DNS settings and port settings (port 80) must be correct.

To update the patterns for the antivirus signature database:

1. On the security device, specify the URL address of the pattern-update server.
2. (Optional) Specify how often the device should automatically check for pattern-server updates.

After the security device downloads the server-initialization file, the device checks that the pattern file is valid. The device then parses the file to obtain information about it, including the file version, size, and location of the pattern file server.

If the pattern file on the security device is out-of-date (or nonexistent because this is the first time you are loading it), and, if the antivirus pattern-update service subscription is still valid, the device automatically retrieves an updated pattern file from the pattern file server.

The following is an example of the CLI for configuring the database update feature:

```
utm {
  feature-profile {
    anti-virus {
      type
        kaspersky-lab-engine {
```

```

    pattern-update
      url url
      interval minutes
    }
  }
}

```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Understanding Full Antivirus Pattern Updates on page 870
- Example: Configuring the Full Antivirus Pattern Update Server on page 872
- Example: Automatically Updating Full Antivirus Patterns (J-Web) on page 873
- Example: Automatically Updating Full Antivirus Patterns on page 874
- Manually Updating, Reloading, and Deleting Full Antivirus Patterns (CLI Procedure) on page 875

### Example: Configuring the Full Antivirus Pattern Update Server

This example shows how to configure the pattern-update server on the security device.

- Requirements on page 872
- Overview on page 872
- Configuration on page 873
- Verification on page 873

#### Requirements

Before you begin:

- Obtain a valid antivirus scanner license. See “Full Antivirus Protection Overview” on page 869.
- Get network connectivity and access to the pattern database server. See “Understanding Full Antivirus Pattern Updates” on page 870.
- Configure your DNS settings and port settings (port 80) correctly. See “DNS Overview” on page 127.

#### Overview

To configure the pattern-update server on the security device, enter the URL address of the pattern-update server. In this example, you update the URL for an SRX210 Services Gateway.

By default, the Juniper-Kaspersky URL for full antivirus protection is `http://update.juniper-updates.net/AV/device-name`, where *device-name* is the name of your device, such as, SRX210.

## Configuration

### Step-by-Step Procedure

To configure the pattern-update server on a security device:

1. Specify the URL of the pattern-update server.

[edit]

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine
pattern-update url http://update.juniper-updates.net/AV/SRX210
```



**NOTE:** Other than the platform name, you should not change this URL unless you are experiencing problems with it and have called for support.

2. If you are done configuring the device, commit the configuration.

[edit]

```
user@host# commit
```

## Verification

To verify the configuration is working properly, enter the **show security utm** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Pattern Update Configuration Overview on page 871
- Example: Automatically Updating Full Antivirus Patterns (J-Web) on page 873
- Example: Automatically Updating Full Antivirus Patterns on page 874
- Manually Updating, Reloading, and Deleting Full Antivirus Patterns (CLI Procedure) on page 875

## Example: Automatically Updating Full Antivirus Patterns (J-Web)

In this example, you configure the security device to update the pattern file automatically every 120 minutes. (The default antivirus pattern-update interval is 60 minutes.)

To automatically update antivirus patterns:

1. Select **Configure>UTM>Anti-Virus**.
2. Next to Interval, in the Kaspersky Lab Engine section, enter **120** in the box.
3. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Understanding Full Antivirus Pattern Updates on page 870
- Full Antivirus Pattern Update Configuration Overview on page 871

- Example: Configuring the Full Antivirus Pattern Update Server on page 872
- Example: Automatically Updating Full Antivirus Patterns on page 874
- Manually Updating, Reloading, and Deleting Full Antivirus Patterns (CLI Procedure) on page 875

## Example: Automatically Updating Full Antivirus Patterns

This example shows how to update the pattern file automatically on a security device.

- Requirements on page 874
- Overview on page 874
- Configuration on page 874
- Verification on page 874

### Requirements

---

Before you begin:

- Obtain a valid antivirus scanner license. See “Full Antivirus Protection Overview” on page 869.
- Get network connectivity and access to the pattern database server. See “Understanding Full Antivirus Pattern Updates” on page 870.
- Configure your DNS settings and port settings (port 80) correctly. See “DNS Overview” on page 127.

### Overview

---

In this example, you configure the security device to update the pattern file automatically every 120 minutes. (The default antivirus pattern-update interval is 60 minutes.)

### Configuration

---

#### Step-by-Step Procedure

To configure the security device to update the pattern file automatically:

1. Set the interval.  

```
[edit]  
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine  
pattern-update interval 120
```
2. If you are done configuring the device, commit the configuration.  

```
[edit]  
user@host# commit
```

### Verification

---

To verify the configuration is working properly, enter the **show security utm** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Pattern Update Configuration Overview on page 871



- Example: Configuring the Full Antivirus Pattern Update Server on page 872
- Example: Automatically Updating Full Antivirus Patterns (J-Web) on page 873
- Manually Updating, Reloading, and Deleting Full Antivirus Patterns (CLI Procedure) on page 875

### Manually Updating, Reloading, and Deleting Full Antivirus Patterns (CLI Procedure)

To manually update antivirus patterns, enter the following CLI statement:

```
user@host> request security utm anti-virus kaspersky-lab-engine pattern-update
```

To manually reload antivirus patterns, enter the following CLI statement:

```
user@host> request security utm anti-virus kaspersky-lab-engine pattern-reload
```

To manually delete antivirus patterns, enter the following CLI statement:

```
user@host> request security utm anti-virus kaspersky-lab-engine pattern-delete
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Understanding Full Antivirus Pattern Updates on page 870
- Full Antivirus Pattern Update Configuration Overview on page 871
- Example: Configuring the Full Antivirus Pattern Update Server on page 872
- Example: Automatically Updating Full Antivirus Patterns (J-Web) on page 873
- Example: Automatically Updating Full Antivirus Patterns on page 874

## Full Antivirus File Scanning

- Understanding the Full Antivirus Internal Scan Engine on page 875
- Global, Profile-Based, and Policy-Based Full Antivirus Scan Settings on page 876
- Full Antivirus Scan Modes on page 879
- Full Antivirus Intelligent Prescreening on page 881
- Full Antivirus Content Size Limits on page 883
- Full Antivirus Decompression Layer Limit on page 884
- Full Antivirus Scanning Timeout on page 885
- Full Antivirus Scan Session Throttling on page 885

### Understanding the Full Antivirus Internal Scan Engine

The full file-based antivirus module is the software subsystem on the gateway device that scans specific Application Layer traffic to protect users from virus attacks and to prevent viruses from spreading. The antivirus software subsystem consists of a virus signature database, an application proxy, the scan manager, and the scan engine.

Kaspersky Lab provides the scan engine and it works in the following manner:

1. A client establishes a TCP connection with a server and then starts a transaction.
2. If the application protocol in question is marked for antivirus scanning, the traffic is forwarded to an application proxy for parsing.
3. When the scan request is sent, the scan engine scans the data by querying a virus pattern database.
4. The scan manager monitors antivirus scanning sessions, checking the properties of the data content against the existing antivirus settings.
5. After scanning has occurred, the result is then handled by the scan manager.

The Kaspersky Lab scan engine supports regular file scanning and script file scanning. With regular file scanning, the input object is a regular file. The engine matches the input content with all possible signatures. With script file scanning, the input object is a script file. It can be JavaScript, VBScript, mIRC script, bat scripts (DOS bat files), and other text scripts. The engine matches the input content only with signatures for script files. Script scanning is only applicable for HTML content over the HTTP protocol. There are two criteria for this scan type. First, the content-type field of this HTML document must be text or HTML. Second, there is no content encoding in the HTTP header. If those two criteria are met, an HTML parser is used to parse the HTML document for scripts.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Full Antivirus Protection Overview on page 869](#)
- [Understanding Full Antivirus Scan Level Settings on page 876](#)
- [Example: Configuring Full Antivirus Scan Settings at Different Levels on page 877](#)
- [Understanding Full Antivirus Scan Mode Support on page 879](#)

### Global, Profile-Based, and Policy-Based Full Antivirus Scan Settings

- [Understanding Full Antivirus Scan Level Settings on page 876](#)
- [Example: Configuring Full Antivirus Scan Settings at Different Levels on page 877](#)

#### Understanding Full Antivirus Scan Level Settings

The antivirus module allows you to configure scanning options on a global level, on a UTM profile level, or on a firewall policy level. Each configuration level has the following implications:

- **Global antivirus settings**—Settings are applied to all antivirus sessions. Global settings are general overall configurations for the antivirus module or settings that are not specific for profiles.
- **Profile-based settings**—Antivirus settings are different for different protocols within the same policy.
- **Policy-based settings**—Antivirus settings are different for different policies. Policy-based antivirus settings are applied to all scan-specified traffic defined in a firewall policy.

The majority of antivirus settings are configured within an antivirus profile, bound to specified protocols, and used by designated policies. These UTM policies are then applied to the traffic according to firewall policies. If a firewall policy with an antivirus setting matches the properties of a traffic flow, the antivirus setting is applied to the traffic session. Therefore, you can apply different antivirus settings for different protocols and for different traffic sessions.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Full Antivirus Protection Overview on page 869](#)
- [Understanding the Full Antivirus Internal Scan Engine on page 875](#)
- [Example: Configuring Full Antivirus Scan Settings at Different Levels on page 877](#)
- [Understanding Full Antivirus Scan Mode Support on page 879](#)

#### [Example: Configuring Full Antivirus Scan Settings at Different Levels](#)

This example shows how to configure full antivirus scan settings at different levels.

- [Requirements on page 877](#)
- [Overview on page 877](#)
- [Configuration on page 877](#)
- [Verification on page 878](#)

#### **Requirements**

Before you begin, decide the type of scanning option you require. See “Understanding Full Antivirus Scan Level Settings” on page 876.

#### **Overview**

In this example, you define antivirus scanning options on any of the following levels:

- Global level
- UTM profile level using the kasprof1 UTM profile
- Firewall policy level using the p1 UTM policy

#### **Configuration**

#### CLI Quick Configuration

To quickly configure antivirus scanning options at different levels, copy the following commands and paste them into the CLI:

```
[edit]
set security utm feature-profile anti-virus kaspersky-lab-engine pattern-update interval
  20
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options default block
set utm-policy p1 anti-virus http-profile av-profile
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure antivirus scanning options at different levels:

1. Configure scanning options at the global level.

```
[edit security utm]
user@host# set feature-profile anti-virus kaspersky-lab-engine pattern-update
interval 20
```

2. Configure scanning options at the UTM profile level.

```
[edit security utm]
user@host# set feature-profile anti-virus kaspersky-lab-engine profile kasprof1
fallback-options default block
```

3. Configure scanning options at the UTM policy level.

```
[edit security utm]
user@host# set utm-policy p1 anti-virus http-profile av-profile
```

**Results** From configuration mode, confirm your configuration by entering the **show security utm** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
user@host# show security utm
...
  utm-policy p1 {
    anti-virus {
      http-profile av-profile
      ftp {
        upload-profile av-profile
        download-profile av-profile
      }
    }
  }
  ...
```

If you are done configuring the device, enter **commit** from configuration mode.

### **Verification**

To confirm that the configuration is working properly, perform this task:

- Verifying Scan Settings at Different Levels on page 878

### **Verifying Scan Settings at Different Levels**

**Purpose** Verify the scan settings at different levels.

**Action** From operational mode, enter the **show configuration security utm** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Full Antivirus Protection Overview on page 869
  - Understanding the Full Antivirus Internal Scan Engine on page 875
  - Understanding Full Antivirus Scan Mode Support on page 879

## Full Antivirus Scan Modes

- Understanding Full Antivirus Scan Mode Support on page 879
- Example: Configuring Full Antivirus File Extension Scanning on page 880
- Configuring Full Antivirus File Extension Scanning (CLI Procedure) on page 881

### Understanding Full Antivirus Scan Mode Support

---

The Kaspersky Lab scan engine supports two modes of scanning:

- **scan-all**—This option tells the scan engine to scan all the data it receives.
- **scan-by-extension**—This option bases all scanning decisions on the file extensions found in the traffic in question.

When scanning content, you can use a file extension list to define a set of file extensions that are used in file extension scan mode (**scan-by-extension**). The antivirus module can then scan files with extensions on the scan-extension list. If an extension is not defined in an extension list, the file with that extension is not scanned in scan-by-extension mode. If there is no extension present, the file in question is scanned.

When using a file extension list to scan content, please note the following requirements:

- File extension entries are case-insensitive.
- The maximum length of the file extension list name is 29 bytes.
- The maximum length of each file extension entry is 15 bytes.
- The maximum entry number in a file extension list is 255.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Full Antivirus Protection Overview on page 869
  - Understanding the Full Antivirus Internal Scan Engine on page 875
  - Understanding Full Antivirus Scan Level Settings on page 876
  - Example: Configuring Full Antivirus Scan Settings at Different Levels on page 877

### Example: Configuring Full Antivirus File Extension Scanning

This example shows how to configure full antivirus file extension scanning.

- Requirements on page 880
- Overview on page 880
- Configuration on page 880
- Verification on page 880

#### Requirements

Before you begin, decide the mode of scanning you require. See “Understanding Full Antivirus Scan Mode Support” on page 879.

#### Overview

In this example, you perform the following tasks:

1. Create a file called `extlist1` for the `kasprof1` profile, and add extensions such as `.zip`, `.js`, and `.vbs` to the `extlist1`.
2. Configure the scan mode setting. You can choose to scan all files or to scan only the files that have the extensions that you specify. This example uses the scan by-extension option to configure the device to use the `extlist1` file.

#### Configuration

##### Step-by-Step Procedure

To configure full antivirus file extension scanning:

1. Create a extension for the list and add extensions to the filename extension list.
 

```
[edit]
user@host# set security utm custom-objects filename-extension extlist1 value [zip
js vbs]
```
2. Configure scan extension settings.
 

```
[edit]
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine profile
kasprof1 scan-options scan-extension extlist1
```
3. Configure the scan mode setting.
 

```
[edit]
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine profile
kasprof1 scan-options scan-mode by-extension
```
4. If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

#### Verification

To verify the configuration is working properly, enter the `show security utm` command.

##### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869

- [Configuring Full Antivirus File Extension Scanning \(CLI Procedure\) on page 881](#)

### Configuring Full Antivirus File Extension Scanning (CLI Procedure)

To configure file-extension scanning, use the following CLI configuration statements:

```
security utm {
  custom-objects {
    filename-extension { ; set of list
      name extension-list-name; #mandatory
      value windows-extension-string;
    }
  }
}

security utm feature-profile anti-virus kaspersky-lab-engine profile name {
  scan-options {
    scan-extension ext-list
  }
}
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Full Antivirus Protection Overview on page 869](#)
- [Example: Configuring Full Antivirus File Extension Scanning on page 880](#)
- [Understanding Full Antivirus Scan Mode Support on page 879](#)

## Full Antivirus Intelligent Prescreening

- [Understanding Full Antivirus Intelligent Prescreening on page 881](#)
- [Example: Configuring Full Antivirus Intelligent Prescreening on page 882](#)

### Understanding Full Antivirus Intelligent Prescreening

By default, intelligent prescreening is enabled to improve antivirus scanning performance. The antivirus module generally begins to scan data after the gateway device has received all the packets of a file. Intelligent prescreening tells the antivirus module to begin scanning a file much earlier. In this case, the scan engine uses the first packet or the first several packets to determine if a file could possibly contain malicious code. The scan engine does a quick check on these first packets and if it finds that it is unlikely that the file is infected, it then decides that it is safe to bypass the normal scanning procedure.



**NOTE:** Intelligent prescreening is only intended for use with non-encoded traffic. It is not applicable for MIME encoded traffic, mail protocols (SMTP, POP3, IMAP) and HTTP POST.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Full Antivirus Protection Overview on page 869](#)
- [Example: Configuring Full Antivirus Intelligent Prescreening on page 882](#)

- Understanding Full Antivirus Scan Mode Support on page 879

### Example: Configuring Full Antivirus Intelligent Prescreening

This example shows how to configure full antivirus intelligent prescreening. By default, intelligent prescreening is enabled to improve antivirus scanning performance.

- Requirements on page 882
- Overview on page 882
- Configuration on page 882
- Verification on page 882

#### Requirements

Before you begin, understand how intelligent prescreening enables the improvement of antivirus scanning performance. See “Understanding Full Antivirus Intelligent Prescreening” on page 881.

#### Overview

In this example, you perform the following tasks:

- Enable intelligent prescreening for the `kasprof1` profile.
- Disable intelligent prescreening for the `kasprof1` profile.

#### Configuration

#### Step-by-Step Procedure

To enable or disable full antivirus intelligent prescreening:

1. Enable intelligent prescreening for the `kasprof1` profile.

[edit]

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine profile
kasprof1 scan-options intelligent-prescreening
```

2. Disable intelligent prescreening for the `kasprof1` profile.

[edit]

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine profile
kasprof1 scan-options no-intelligent-prescreening
```



**NOTE:** Intelligent prescreening is intended only for use with non-encoded traffic. It is not applicable to mail protocols (SMTP, POP3, IMAP) or HTTP POST.

3. If you are done configuring the device, commit the configuration.

[edit]

```
user@host# commit
```

#### Verification

To verify the configuration is working properly, enter the `show security utm` command.



- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Full Antivirus Protection Overview on page 869
  - Understanding Full Antivirus Scan Level Settings on page 876

## Full Antivirus Content Size Limits

- Understanding Full Antivirus Content Size Limits on page 883
- Configuring Full Antivirus Content Size Limits (CLI Procedure) on page 883

### Understanding Full Antivirus Content Size Limits

Due to resource constraints, there is a default, device-dependent limit on maximum content size for the database. The content size value is configurable. There is also a lower and upper limit for maximum content size. (This range is device dependent and is not configurable.)

The content size check occurs before the scan request is sent. The exact timing of this is protocol dependent. If the protocol header contains an accurate content length field, the content size check takes place when the content length field is extracted during header parsing. The content size usually refers to file size. If there is no content length field, the size is checked while the antivirus module is receiving packets. The content size, in this case, refers to accumulated TCP payload size.



**NOTE:** This setting can be used in all protocols.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Full Antivirus Protection Overview on page 869
  - Configuring Full Antivirus Content Size Limits (CLI Procedure) on page 883

### Configuring Full Antivirus Content Size Limits (CLI Procedure)

To configure content size limits, use the following CLI configuration statements:

```
security utm feature-profile anti-virus kaspersky-lab-engine profile name {
  scan-options {
    content-size-limit KB;
  }
}
```

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Full Antivirus Protection Overview on page 869
  - Understanding Full Antivirus Content Size Limits on page 883

## Full Antivirus Decompression Layer Limit

- Understanding Full Antivirus Decompression Layer Limits on page 884
- Configuring Full Antivirus Decompression Layer Limits (CLI Procedure) on page 884

### Understanding Full Antivirus Decompression Layer Limits

The decompression layer limit specifies how many layers of nested compressed files and files with internal extractable objects, such as archive files (tar), MS Word and PowerPoint files, the internal antivirus scanner can decompress before it executes the virus scan. For example, if a message contains a compressed .zip file that contains another compressed .zip file, there are two compression layers. Decompressing both files requires a decompress layer setting of 2.

It is worth noting that during the transfer of data, some protocols use content encoding. The antivirus scan engine must decode this layer, which is considered a decompression level, before it scans for viruses.

There are three kinds of compressed data:

- compressed file (zip, rar, gzip)
- encoded data (MIME)
- packaged data (OLE, .CAP, .MSI, .TAR, .EML)

A decompression Layer could be a layer of a zipped file or an embedded object in packaged data. The antivirus engine scans each layer before unpacking the next layer, until it either reaches the user-configured decompress limit, reaches the device decompress layer limit, finds a virus or other malware, or decompresses the data completely, whichever comes first.

As the virus signature database becomes larger and the scan algorithms become more sophisticated, the scan engine has the ability to look deeper into the data for embedded malware. As a result, it can uncover more layers of compressed data. The Juniper device's level of security is limited by decompress limit, which is based on the memory allocated to the security service. If a virus is not found within the decompress limit, the user has an option to either pass or drop the data.



**NOTE:** This setting can be used in all protocols.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Configuring Full Antivirus Decompression Layer Limits (CLI Procedure) on page 884

### Configuring Full Antivirus Decompression Layer Limits (CLI Procedure)

To configure decompression layer limits, use the following CLI configuration statements:

```
security utm feature-profile anti-virus kaspersky-lab-engine profile name {
```

```

scan-options {
  decompress-layer-limit number
}

```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Understanding Full Antivirus Decompression Layer Limits on page 884

## Full Antivirus Scanning Timeout

- Understanding Full Antivirus Scanning Timeouts on page 885
- Configuring Full Antivirus Scanning Timeouts (CLI Procedure) on page 885

### Understanding Full Antivirus Scanning Timeouts

The scanning timeout value includes the time frame from when the scan request is generated to when the scan result is returned by the scan engine. The time range can be 1 to 1800 seconds. By default, it is 180 seconds.



**NOTE:** This timeout parameter is used by all supported protocols. Each protocol can have a different timeout value.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Configuring Full Antivirus Scanning Timeouts (CLI Procedure) on page 885

### Configuring Full Antivirus Scanning Timeouts (CLI Procedure)

To configure scanning timeouts, use the following CLI configuration statements:

```

security utm feature-profile anti-virus kaspersky-lab-engine profile name {
  scan-options {
    timeout-value seconds {
    }
  }
}

```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Understanding Full Antivirus Scanning Timeouts on page 885

## Full Antivirus Scan Session Throttling

- Understanding Full Antivirus Scan Session Throttling on page 886
- Configuring Full Antivirus Scan Session Throttling (CLI Procedure) on page 886

### [Understanding Full Antivirus Scan Session Throttling](#)

In an attempt to consume all available resources and hinder the ability of the scan engine to scan other traffic, a malicious user might generate a large amount of traffic all at once. To prevent such activity from succeeding, a session throttle is imposed for antivirus resources, thereby restricting the amount of traffic a single source can consume at one time. The limit is an integer with 100 as the default setting. This integer refers to the maximum allowed sessions from a single source. You may change this default limit, but understand that if this limit is set high, that is comparable to no limit.

**Over-limit** is a fallback setting for the connection-per-client limit. The default behavior of over-limit is to block sessions. This is a per-policy setting. You can specify different settings for different UTM policies.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Configuring Full Antivirus Scan Session Throttling (CLI Procedure) on page 886

### [Configuring Full Antivirus Scan Session Throttling \(CLI Procedure\)](#)

To configure scan session throttling, use the following CLI configuration statements:

```
security utm utm-policy name
  traffic-options {
    sessions-per-client {
      limit number;
      over-limit { log-and-permit | block }
    }
  }
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Understanding Full Antivirus Scan Session Throttling on page 886

## [Full Antivirus Application Protocol Scanning](#)

---

- Understanding Full Antivirus Application Protocol Scanning on page 886
- HTTP Full Antivirus Scanning on page 887
- FTP Full Antivirus Scanning on page 892
- SMTP Full Antivirus Scanning on page 893
- POP3 Full Antivirus Scanning on page 895
- IMAP Full Antivirus Scanning on page 897

### [Understanding Full Antivirus Application Protocol Scanning](#)

You can turn antivirus scanning on and off on a per protocol basis. If scanning for a protocol is disabled in an antivirus profile, there is no application intelligence for this protocol. Therefore, in most cases, traffic using this protocol is not scanned. But if the

protocol in question is based on another protocol for which scanning is enabled in an antivirus profile, then the traffic is scanned as that enabled protocol.

The internal antivirus scan engine supports scanning for specific Application Layer transactions allowing you to select the content (HTTP, FTP, SMTP, POP3, or IMAP traffic) to scan. For each content type that you are scanning, you have different configuration options.

Profile-based settings, including enable/disable, scan-mode, and scan result handling settings, may not be applicable to all supported protocols. The following table lists profile-based settings and their protocol support.

**Table 103: Supported Profile-based Settings By Protocol**

| Profile Setting                                                            | Protocol Support                   |
|----------------------------------------------------------------------------|------------------------------------|
| Enable or disable scanning on per protocol basis                           | All protocols support this feature |
| "Full Antivirus Scan Modes" on page 879, including file extension scanning | All protocols support this feature |
| "Full Antivirus Content Size Limits" on page 883                           | All protocols support this feature |
| "Full Antivirus Decompression Layer Limit" on page 884                     | All protocols support this feature |
| "Full Antivirus Scanning Timeout" on page 885                              | All protocols support this feature |
| "Understanding HTTP Tricking" on page 889                                  | HTTP only                          |
| "Understanding Antivirus Scanning Fallback Options" on page 902            | All protocols support this feature |
| Protocol specific messages                                                 | All protocols support this feature |
| "E-Mail Virus-Detected Notifications" on page 900                          | SMTP, POP3, and IMAP only          |
| "Custom Message Virus-Detected Notifications" on page 901                  | All protocols support this feature |

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Understanding HTTP Scanning on page 888
- Enabling HTTP Scanning (CLI Procedure) on page 889
- Understanding Protocol-Only Virus-Detected Notifications on page 900

### HTTP Full Antivirus Scanning

- Understanding HTTP Scanning on page 888
- Enabling HTTP Scanning (CLI Procedure) on page 889
- Understanding HTTP Tricking on page 889

- [Configuring HTTP Tricking to Prevent Timeouts During Antivirus Scanning \(CLI Procedure\)](#) on page 889
- [Understanding MIME Whitelists](#) on page 890
- [Example: Configuring MIME Whitelists to Bypass Antivirus Scanning](#) on page 890
- [Understanding URL Whitelists](#) on page 891
- [Configuring URL Whitelists to Bypass Antivirus Scanning \(CLI Procedure\)](#) on page 892

### [Understanding HTTP Scanning](#)

If antivirus scanning is enabled for Hypertext Transfer Protocol (HTTP) traffic in a content security profile, TCP traffic to defined HTTP service ports (generally port 80) is monitored. For HTTP traffic, the security device scans both HTTP responses and requests (get, post, and put commands).



**NOTE:** For HTTP antivirus scanning, both HTTP 1.0 and 1.1 are supported. If the protocol version is HTTP 0.x, the antivirus scanner attempts to scan the traffic. Unknown protocols are bypassed. For example, some application protocols use HTTP as the transport but do not comply with HTTP 1.0 or 1.1. These are considered unknown protocols and are not scanned.

This is a general description of how HTTP traffic is intercepted, scanned, and acted upon by the antivirus scanner:

1. An HTTP client sends an HTTP request to a webserver or a webserver responds to an HTTP request.
2. The security device intercepts the request and passes the data to the antivirus scanner, which scans it for viruses.
3. After completing the scan, the device follows one of two courses:
  - If there is no virus, the device forwards the request to the webserver.
  - If there is a virus, the device drops the request and sends an HTTP message reporting the infection to the client.

With script-only scanning, the input object is a script file. It can be JavaScript, VBScript, mIRC script, bat scripts (DOS bat files) and other text scripts. The engine matches the input content only with signatures for script files. Script scanning is applicable only for HTML content over the HTTP protocol. There are two criteria for this scan-type. First, the content-type field of this HTML document must be text or HTML. Second, there is no content encoding in the HTTP header. If those two criteria are met, an HTML parser is used to parse the HTML document.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Full Antivirus Protection Overview](#) on page 869
- [Understanding Full Antivirus Application Protocol Scanning](#) on page 886
- [Enabling HTTP Scanning \(CLI Procedure\)](#) on page 889

### Enabling HTTP Scanning (CLI Procedure)

To enable antivirus scanning for HTTP traffic, enter the following CLI configuration statement:

```
user@host# set security utm utm-policy policy-name anti-virus http
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Understanding Full Antivirus Application Protocol Scanning on page 886
- Understanding HTTP Scanning on page 888

### Understanding HTTP Trickling

HTTP trickling is a mechanism used to prevent the HTTP client or server from timing-out during a file transfer or during antivirus scanning. On some slow link transferring, a large file could timeout if too much time is taken for the antivirus scanner to scan a complex file.

HTTP trickling is the forwarding of specified amounts of unscanned HTTP traffic to the requesting HTTP client to prevent the browser window from timing out while the scan manager examines downloaded HTTP files. (The security device forwards small amounts of data in advance of transferring an entire scanned file.)

HTTP Trickling is time-based and there is only one parameter, the time-out interval, to configure for this feature. By default, trickling is disabled.



**NOTE:** The timeout based trickling is packet driven. This means, if no packet is received within a certain time frame, HTTP trickling is discontinued. This setting is only supported for HTTP connections.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Configuring HTTP Trickling to Prevent Timeouts During Antivirus Scanning (CLI Procedure) on page 889

### Configuring HTTP Trickling to Prevent Timeouts During Antivirus Scanning (CLI Procedure)

To configure HTTP trickling, use the following CLI configuration statements:

```
security utm feature-profile anti-virus kaspersky-lab-engine {
  profile name {
    trickling timeout seconds;
  }
}
```

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Full Antivirus Protection Overview on page 869
  - Understanding HTTP Tricking on page 889

### Understanding MIME Whitelists

The gateway device uses MIME (Multipurpose Internet Mail Extension) types to decide which traffic may bypass antivirus scanning. The MIME whitelist defines a list of MIME types and can contain one or many MIME entries.

A MIME entry is case-insensitive. An empty MIME is an invalid entry and should never appear in the MIME list. If the MIME entry ends with a / character, prefix matching takes place. Otherwise, exact matching occurs.

There are two types of MIME lists used to configure MIME type antivirus scan bypassing:

- mime-whitelist list—This is the comprehensive list for those MIME types that can bypass antivirus scanning.
- exception list—The exception list is a list for excluding some MIME types from the mime-whitelist list. This list is a subset of MIME types found in the mime-whitelist.

For example, if the mime-whitelist includes the entry, **video/** and the exception list includes the entry **video/x-shockwave-flash**, by using these two lists, you can bypass objects with “video/” MIME type but not bypass “video/x-shockwave-flash” MIME type.

You should note that there are limits for mime-whitelist entries as follows:

- The maximum number of MIME items in a MIME list is 50.
- The maximum length of each MIME entry is restricted to 40 bytes.
- The maximum length of a MIME list name string is restricted to 40 bytes.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Full Antivirus Protection Overview on page 869
  - Example: Configuring MIME Whitelists to Bypass Antivirus Scanning on page 890
  - Understanding URL Whitelists on page 891
  - Configuring URL Whitelists to Bypass Antivirus Scanning (CLI Procedure) on page 892

### Example: Configuring MIME Whitelists to Bypass Antivirus Scanning

This example shows how to configure MIME whitelists to bypass antivirus scanning.

- Requirements on page 891
- Overview on page 891
- Configuration on page 891
- Verification on page 891



**Requirements**

Before you begin, decide the type of MIME lists used to configure MIME type antivirus scan bypassing. See “Understanding MIME Whitelists” on page 890.

**Overview**

In this example, you create MIME lists called avmime2 and ex-avmime2 and add patterns to them.

**Configuration**

**Step-by-Step Procedure** To configure MIME whitelists to bypass antivirus scanning:

1. Create MIME lists and add patterns to the lists.

```
[edit]
user@host# set security utm custom-objects mime-pattern avmime2 value
[video/quicktime image/x-portable-anymap x-world/x-vrml]
user@host# set security utm custom-objects mime-pattern ex-avmime2 value
[video/quicktime-inappropriate]
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

**Verification**

To verify the configuration is working properly, enter the **show security utm** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Understanding URL Whitelists on page 891
- Configuring URL Whitelists to Bypass Antivirus Scanning (CLI Procedure) on page 892

**Understanding URL Whitelists**

A URL whitelist is a unique custom list that you define in which all the URLs or IP addresses in that list for a specified category are always bypassed for scanning. Because antivirus scanning is CPU and memory intensive action, if there are URLs or IP addresses that you are sure do not require scanning, you might want to create this custom list and add them to it.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Understanding MIME Whitelists on page 890
- Example: Configuring MIME Whitelists to Bypass Antivirus Scanning on page 890
- Configuring URL Whitelists to Bypass Antivirus Scanning (CLI Procedure) on page 892

## Configuring URL Whitelists to Bypass Antivirus Scanning (CLI Procedure)

To configure URL whitelists, use the following CLI configuration statements:

```
security utm custom-objects {
  custom-url-category { ; set of list
    name url-category-name; #mandatory
    value url-pattern-name;
  }
}
```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Full Antivirus Protection Overview on page 869](#)
- [Understanding MIME Whitelists on page 890](#)
- [Example: Configuring MIME Whitelists to Bypass Antivirus Scanning on page 890](#)
- [Understanding URL Whitelists on page 891](#)

## FTP Full Antivirus Scanning

- [Understanding FTP Antivirus Scanning on page 892](#)
- [Enabling FTP Antivirus Scanning \(CLI Procedure\) on page 893](#)

### Understanding FTP Antivirus Scanning

If antivirus scanning is enabled for File Transfer Protocol (FTP) traffic in a content security profile, the security device monitors the control channel and, when it detects one of the FTP commands for transferring data, it scans the data sent over the data channel.

This is a general description of how FTP traffic is intercepted, scanned, and acted upon by the antivirus scanner:

1. A local FTP client opens an FTP control channel to an FTP server and requests the transfer of some data.
2. The FTP client and server negotiate a data channel over which the server sends the requested data. The security device intercepts the data and passes it to the antivirus scan engine, which scans it for viruses.
3. After completing the scan, the device follows one of two courses:
  - If there is no virus, the device forwards the data to the client.
  - If there is a virus, the device replaces the data with a drop message in the data channel and sends a message reporting the infection in the control channel.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Full Antivirus Protection Overview on page 869](#)
- [Enabling FTP Antivirus Scanning \(CLI Procedure\) on page 893](#)

### Enabling FTP Antivirus Scanning (CLI Procedure)

To enable antivirus scanning for File Transfer Protocol (FTP) traffic, enter the following CLI configuration statement:

```
user@host# security utm utm-policy policy-name anti-virus ftp
```



**NOTE:** In order to scan FTP traffic, the FTP ALG must be enabled.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Full Antivirus Protection Overview on page 869](#)
- [Understanding FTP Antivirus Scanning on page 892](#)

### SMTP Full Antivirus Scanning

- [Understanding SMTP Antivirus Scanning on page 893](#)
- [Enabling SMTP Antivirus Scanning \(CLI Procedure\) on page 895](#)

#### Understanding SMTP Antivirus Scanning

If SMTP (Simple Mail Transfer Protocol) antivirus scanning is enabled in a content security profile, the security device redirects traffic from local SMTP clients to the antivirus scanner before sending it to the local mail server.



**NOTE:** Chunking is an alternative to the data command. It provides a mechanism to transmit a large message in small chunks. It is not supported. Messages using chunking are bypassed and are not scanned.

This is a general description of how SMTP traffic is intercepted, scanned, and acted upon by the antivirus scanner:

1. An SMTP client sends an e-mail message to a local mail server or a remote mail server forwards an e-mail message via SMTP to the local mail server.
2. The security device intercepts the e-mail message and passes the data to the antivirus scanner, which scans it for viruses.
3. After completing the scan, the device follows one of two courses:
  - If there is no virus, the device forwards the message to the local server.
  - If there is a virus, the device sends a replacement message to the client.

This topic includes the following sections:

- [Understanding SMTP Antivirus Mail Message Replacement on page 894](#)
- [Understanding SMTP Antivirus Sender Notification on page 894](#)
- [Understanding SMTP Antivirus Subject Tagging on page 894](#)

### *Understanding SMTP Antivirus Mail Message Replacement*

If the antivirus scanner finds a virus in an e-mail message, the original message is dropped, the message body is truncated, and the content is replaced by a message that may appear as follows:

```
nContent-Type: text/plain
Your mail <src_ip> : <src_port> — <dst_port>: <dst_port> contains contaminated file
<filename> with virus <virusname>, so it is dropped.
```

If a scan error is returned and the fail mode is set to drop, the original message is dropped and the entire message body is truncated. The content is replaced by a message that may appear as follows:

```
nContent-Type: text/plain
Your mail <src_ip> : <src_port> — <dst_port>: <dst_port> is dropped for <reason>.
```

### *Understanding SMTP Antivirus Sender Notification*

If **notify-sender-on-virus** is set and the message is dropped due to a detected virus, an e-mail is sent to the mail sender. The content of the notification may appear as follows:

```
From: <admin>@<gateway_ip>
To: <sender_e-mail>
Subject: Mail Delivery Failure
This message is created automatically by mail delivery software. A message that you sent
could not be delivered to one or more of its recipients for the reason:
<src_ip> : <src_port> — <dst_port>: <dst_port> <ENVID> contaminated file <filename>
with virus <virusname>.
e-mail Header is:
<header of scanned e-mail>
```

If **notify-sender-on-error-drop** is set and the message is dropped due to a scan error, an e-mail is sent to the mail sender of the scanned message. The content of the e-mail may appear as follows:

```
From: <admin>@<gateway_ip>
To: <sender_e-mail>
Subject: Mail Delivery Failure
This message is created automatically by mail delivery software. A message that you sent
could not be delivered to one or more of its recipients for the reason:
<src_ip> : <src_port> — <dst_port>: <dst_port> <ENVID> <reason>.
e-mail Header is:
<header of scanned e-mail>
```



NOTE: For information on the ENVID parameter, refer to RFC 3461.

### *Understanding SMTP Antivirus Subject Tagging*

If a scan error is returned and the fail mode is set to **pass**, the antivirus module passes the message through to the server. If **notify-recipient-on-error-pass** is set, the following string is appended to the end of the subject field:

```
(No virus check: <reason>)
```

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Full Antivirus Protection Overview on page 869
  - Enabling SMTP Antivirus Scanning (CLI Procedure) on page 895

### Enabling SMTP Antivirus Scanning (CLI Procedure)

To enable antivirus scanning for SMTP traffic, enter the following CLI configuration statement:

```
user@host# set security utm utm-policy policy-name anti-virus smtp-profile
```

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Full Antivirus Protection Overview on page 869
  - Understanding SMTP Antivirus Scanning on page 893

## POP3 Full Antivirus Scanning

- Understanding POP3 Antivirus Scanning on page 895
- Enabling POP3 Antivirus Scanning (CLI Procedure) on page 896

### Understanding POP3 Antivirus Scanning

If Post Office Protocol 3 (POP3) antivirus scanning is enabled in a content security profile, the security device redirects traffic from a local mail server to antivirus scanner before sending it to the local POP3 client.

This is a general description of how POP3 traffic is intercepted, scanned, and acted upon by the antivirus scanner.

1. The POP3 client downloads an e-mail message from the local mail server.
2. The security device intercepts the e-mail message and passes the data to the antivirus scanner, which scans it for viruses.
3. After completing the scan, the security device follows one of two courses:
  - If there is no virus, the device forwards the message to the client.
  - If there is a virus, the device sends a message reporting the infection to the client.



**NOTE:** See “Understanding Protocol-Only Virus-Detected Notifications” on page 900 for information on protocol-only notifications for IMAP.

This topic includes the following sections:

- Understanding POP3 Antivirus Mail Message Replacement on page 896
- Understanding POP3 Antivirus Sender Notification on page 896
- Understanding POP3 Antivirus Subject Tagging on page 896

### ***Understanding POP3 Antivirus Mail Message Replacement***

If the antivirus scanner finds a virus in an e-mail message, the original message is dropped, the message body is truncated, and the content is replaced by a message that may appear as follows:

```
nContent-Type: text/plain
Your mail <src_ip> : <src_port> — <dst_port>: <dst_port> contains contaminated file
<filename> with virus <virusname>, so it is dropped.
```

### ***Understanding POP3 Antivirus Sender Notification***

If **notify-sender-on-virus** is set and the message is dropped due to a detected virus, an e-mail is sent to the mail sender.

```
From: <admin>@<gateway_ip>
To: <sender_e-mail>
Subject: Mail Delivery Failure
This message is created automatically by mail delivery software. A message that you sent
could not be delivered to one or more of its recipients for the reason:
<src_ip> : <src_port> — <dst_port>: <dst_port> contaminated file <filename> with virus
<virusname>.
e-mail Header is:
<header of scanned e-mail>
```

If **notify-sender-on-error-drop** is set and the message is dropped due to a scan error, an e-mail is sent to the mail sender of the scanned message. The content of the e-mail may appear as follows:

```
From: <admin>@<gateway_ip>
To: <sender_e-mail>
Subject: Mail Delivery Failure
This message is created automatically by mail delivery software. A message that you sent
could not be delivered to one or more of its recipients for the reason:
<src_ip> : <src_port> — <dst_port>: <dst_port> <reason>.
e-mail Header is:
<header of scanned e-mail>
```

### ***Understanding POP3 Antivirus Subject Tagging***

If a scan error is returned and the fail mode is set to **pass**, the antivirus module passes the message through to the server. If **notify-recipient-on-error-pass** is set, the following string is appended to the end of subject field:

```
(No virus check: <reason>)
```

#### **Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Enabling POP3 Antivirus Scanning (CLI Procedure) on page 896

### ***Enabling POP3 Antivirus Scanning (CLI Procedure)***

To enable antivirus scanning for POP3 traffic, enter the following CLI configuration statement:

```
user@host# set security utm utm-policy policy-name anti-virus pop3-profile
```

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Full Antivirus Protection Overview on page 869
  - Understanding POP3 Antivirus Scanning on page 895

## IMAP Full Antivirus Scanning

- Understanding IMAP Antivirus Scanning on page 897
- Enabling IMAP Antivirus Scanning (CLI Procedure) on page 899

### Understanding IMAP Antivirus Scanning

If IMAP (Internet Message Access Protocol) antivirus scanning is enabled in a content security profile, the security device redirects traffic from a local mail server to the internal antivirus scanner before sending it to the local IMAP client.

This is a general description of how IMAP traffic is intercepted, scanned, and acted upon by the antivirus scanner.

1. The IMAP client downloads an e-mail message from the local mail server.
2. The security device intercepts the e-mail message and passes the data to the antivirus scanner, which scans it for viruses.
3. After completing the scan, the security device follows one of two courses:
  - If there is no virus, the device forwards the message to the client.
  - If there is a virus, the device sends a message reporting the infection to the client.



**NOTE:** See “Understanding Protocol-Only Virus-Detected Notifications” on page 900 for information on protocol-only notifications for IMAP.

This topic includes the following sections:

- Understanding IMAP Antivirus Mail Message Replacement on page 897
- Understanding IMAP Antivirus Sender Notification on page 898
- Understanding IMAP Antivirus Subject Tagging on page 898
- Understanding IMAP Antivirus Scanning Limitations on page 898

#### ***Understanding IMAP Antivirus Mail Message Replacement***

If the antivirus scanner finds a virus in an e-mail message, the original message is dropped, the message body is truncated, and the content is replaced by a message that may appear as follows:

```
nContent-Type: text/plain
Your mail <src_ip> : <src_port> — <dst_port>: <dst_port> contains contaminated file
<filename> with virus <virusname>, so it is dropped.
```

### ***Understanding IMAP Antivirus Sender Notification***

If **notify-sender-on-virus** is set and the message is dropped due to a detected virus, an e-mail is sent to the mail sender.

```
From: <admin>@<gateway_ip>
To: <sender_e-mail>
Subject: Mail Delivery Failure
This message is created automatically by mail delivery software. A message that you sent
could not be delivered to one or more of its recipients for the reason:
<src_ip> : <src_port> — <dst_port>: <dst_port> contaminated file <filename> with virus
<virusname>.
e-mail Header is:
<header of scanned e-mail>
```

If **notify-sender-on-error-drop** is set and the message is dropped due to a scan error, an e-mail is sent to the mail sender of the scanned message. The content of the e-mail may appear as follows:

```
From: <admin>@<gateway_ip>
To: <sender_e-mail>
Subject: Mail Delivery Failure
This message is created automatically by mail delivery software. A message that you sent
could not be delivered to one or more of its recipients for the reason:
<src_ip> : <src_port> — <dst_port>: <dst_port> <reason>.
e-mail Header is:
<header of scanned e-mail>
```

### ***Understanding IMAP Antivirus Subject Tagging***

If a scan error is returned and the fail mode is set to **pass**, the antivirus module passes the message through to the server. If **notify-recipient-on-error-pass** is set, the following string is appended to the end of subject field:

```
(No virus check: <reason>)
```

### ***Understanding IMAP Antivirus Scanning Limitations***

**Mail Fragments** — It is possible to chop one e-mail into multiple parts and to send each part through a different response. This is called mail fragmenting and most popular mail clients support it in order to send and receive large e-mails. Scanning of mail fragments is not supported by the antivirus scanner and in such cases, the message body is not scanned.

**Partial Content** — Some mail clients treat e-mail of different sizes differently. For example, small e-mails (less than 10 KB) are downloaded as a whole. Large e-mails (for example, less than 1 MB) are chopped into 10 KB pieces upon request from the IMAP server. Scanning of any partial content requests is not supported by the antivirus scanner.

**IMAP Uploads** — Only antivirus scanning of IMAP downloads is supported. IMAP upload traffic is not scanned.

#### **Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Enabling IMAP Antivirus Scanning (CLI Procedure) on page 899



### Enabling IMAP Antivirus Scanning (CLI Procedure)

To enable antivirus scanning for IMAP traffic, enter the following CLI configuration statement:

```
user@host# security utm utm-policy policy-name anti-virus imap-profile
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Understanding IMAP Antivirus Scanning on page 897

## Full Antivirus Scan Results and Notification Options

- Understanding Full Antivirus Scan Result Handling on page 899
- Protocol-Only Virus-Detected Notifications on page 899
- E-Mail Virus-Detected Notifications on page 900
- Custom Message Virus-Detected Notifications on page 901
- Full Antivirus Scanning Fallback Options on page 902

### Understanding Full Antivirus Scan Result Handling

Different antivirus scan results are handled in different manners. For example, if a scan result is clean, the traffic is forwarded to the receiver. If the scan result is infected, the traffic is dropped. If the scan results in an error, the result handling depends on the cause of the failure and the configuration (fallback settings).

The following is a list of actions based on scan results:

- Scan Result = Pass

The scan result handling action is to pass the message. In this case, no virus is detected and no error code is returned. Or, an error code is returned, but the fallback option for this error code is set to log-and-permit.

- Scan Result = Block

The scan result handling action is to block the message. In this case, either a virus is detected or an error code is returned and the fallback option for this error code is BLOCK.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Understanding Full Antivirus Scan Level Settings on page 876

### Protocol-Only Virus-Detected Notifications

- Understanding Protocol-Only Virus-Detected Notifications on page 900
- Configuring Protocol-Only Virus-Detected Notifications (CLI Procedure) on page 900

### Understanding Protocol-Only Virus-Detected Notifications

When content is blocked because a virus is found or a scan error occurs, the client generally still receives a successful response code but with modified content (file replacement) containing a warning message. But with protocol-only notifications, a protocol-specific error code may be returned to the client. This way, the client determines that a virus was detected rather than interpreting that a file transfer succeeded.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Configuring Protocol-Only Virus-Detected Notifications (CLI Procedure) on page 900

### Configuring Protocol-Only Virus-Detected Notifications (CLI Procedure)

To configure protocol-only virus-detected notifications, use the following CLI configuration statements:

```
security utm feature-profile anti-virus kaspersky-lab-engine profile name {
  notification-options {
    virus-detection {
      type { protocol-only | message }
    }
    fallback-block {
      type { protocol-only | message }
    }
  }
}
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Understanding Protocol-Only Virus-Detected Notifications on page 900

## E-Mail Virus-Detected Notifications

- Understanding E-Mail Virus-Detected Notifications on page 900
- Configuring E-Mail Virus-Detected Notifications (CLI Procedure) on page 901

### Understanding E-Mail Virus-Detected Notifications

For mail protocols (SMTP, POP3, IMAP), e-mail notification is used to notify the sender or the recipient about the detected viruses or the scanning errors. There are three settings for e-mail notifications:

- `virus-detection/notify-mail-sender` — This setting is used when a virus is detected. If it is enabled, an e-mail is sent to the sender upon virus detection.
- `fallback-block/notify-mail-sender` — This setting is used when other scan codes or scanning errors are returned and the message is dropped. If it is enabled, an e-mail is sent to the sender when an error code is returned.

- `fallback-non-block/notify-mail-recipient` — This setting is used when other scan codes or scanning errors are returned and the message is passed. If it is enabled, the e-mail sent to the recipient is tagged when an error code is returned.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Configuring E-Mail Virus-Detected Notifications (CLI Procedure) on page 901

### Configuring E-Mail Virus-Detected Notifications (CLI Procedure)

To configure the system to send e-mail notifications when viruses are detected, use the following CLI configuration statements:

```
security utm feature-profile anti-virus kaspersky-lab-engine profile name {
  notification-options {
    virus-detection {
      notify-mail-sender
    }
  }
  fallback-block {
    notify-mail-sender
  }
  fallback-non-block {
    notify-mail-recipient
  }
}
}
```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Understanding E-Mail Virus-Detected Notifications on page 900

## Custom Message Virus-Detected Notifications

- Understanding Custom Message Virus-Detected Notifications on page 901
- Configuring Custom Message Virus-Detected Notifications (CLI Procedure) on page 902

### Understanding Custom Message Virus-Detected Notifications

Custom message notifications are mainly used in file replacement or in a response message when the antivirus scan result is to drop the file. When using custom messages, you can provide a customized message in the message content you can define customized subject tags.



**NOTE:** Custom-message in `fallback-nonblock` is used only by mail protocols.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869

- [Configuring Custom Message Virus-Detected Notifications \(CLI Procedure\) on page 902](#)

### [Configuring Custom Message Virus-Detected Notifications \(CLI Procedure\)](#)

To configure the system to send custom messages when viruses are detected, use the following CLI configuration statements:

```
security utm feature-profile anti-virus kaspersky-lab-engine profile name {
  notification-options {
    virus-detection {
      custom-message msg
      custom-message-subject subject-msg
    }
  }
  fallback-block {
    custom-message msg
    custom-message-subject subject-msg
  }
  fallback-non-block {
    custom-message msg
    custom-message-subject subject-msg
  }
}
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Full Antivirus Protection Overview on page 869](#)
- [Understanding Custom Message Virus-Detected Notifications on page 901](#)

## Full Antivirus Scanning Fallback Options

- [Understanding Antivirus Scanning Fallback Options on page 902](#)
- [Example: Configuring Antivirus Scanning Fallback Options on page 903](#)

### [Understanding Antivirus Scanning Fallback Options](#)

Fallback options tell the system how to handle the errors returned by either the scan engine or the scan manager. The following is a list of possible errors and the default fallback actions for those error types:

- Scan engine is not ready (engine-not-ready)
 

The scan engine is initializing itself, for example, loading the signature database. During this phase, it is not ready to scan a file. A file could either pass or be blocked according to this setting. The default action is BLOCK.
- Corrupt file (corrupt-file)
 

Corrupt file is the error returned by the scan engine when engine detects a corrupted file. The default action is PASS.
- Decompression layer (decompress-layer)

Decompress layer error is the error returned by the scan engine when the scanned file has too many compression layers. The default action is BLOCK.

- Password protected file (password-file)

Password protected file is the error returned by the scan engine when the scanned file is protected by a password. The default action is PASS.

- Max content size (content-size)

If the content size exceeds a set limit, the content is passed or blocked depending on the max-content-size fallback option. The default action is BLOCK.

- Too many requests (too-many-requests)

If the total number of messages received concurrently exceeds the device limits, the content is passed or blocked depending on the too-many-request fallback option. The default action is BLOCK. (The allowed request limit is not configurable.)

- Timeout

Scanning a complex file could consume resources and time. If the time it is taking to scan exceeds the timeout setting in the antivirus profile, the processing is aborted and the content is passed or blocked without completing the virus checking. The decision is made based on the timeout fallback option. The default action is BLOCK.

- Out of resources (out-of-resources)

Virus scanning requires a great deal of memory and CPU resources. Due to resource constraints, memory allocation requests can be denied by the system. This failure could be returned by either scan engine (as a scan-code) or scan manager. When out-of-resources occurs, scanning is aborted. The default action is BLOCK.

- Default

All the errors other than those in the above list fall into this category. This could include either unhandled system exceptions (internal errors) or other unknown errors. The default action is BLOCK.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Example: Configuring Antivirus Scanning Fallback Options on page 903

#### Example: Configuring Antivirus Scanning Fallback Options

This example shows how to configure antivirus scanning fallback options.

- Requirements on page 904
- Overview on page 904
- Configuration on page 904
- Verification on page 906

### Requirements

Before you begin, understand the possible error types and the default fallback actions for those error types. See “Understanding Antivirus Scanning Fallback Options” on page 902.

### Overview

In this example, you configure a feature profile called `kasprof`, and set the fallback scanning options for default, content-size, corrupt-file, decompress-layer, engine-not-ready, out-of-resources, password-file, timeout, too-many-requests, as block.



**NOTE:** The command for changing the URL for the pattern database is:

```
[edit]
user@host# edit security utm feature-profile anti-virus kaspersky-lab-engine
[edit security utm feature-profile anti-virus kaspersky-lab-engine]
user@host# set pattern-update url http://.
```

The default URL is `http://update.juniper-updates.net/AV/<device-version>`. You should not change this URL unless you are experiencing problems with it and have called for support.

### Configuration

**CLI Quick Configuration** To quickly configure scanning fallback options, copy the following commands and paste them into the CLI:

```
[edit]
set security utm feature-profile anti-virus type kaspersky-lab-engine
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options content-size block
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options corrupt-file block
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options decompress-layer block
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options default block
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options engine-not-ready block
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options out-of-resources block
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options password-file block
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options timeout block
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options too-many-requests block
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure scanning fallback options:

1. Select and configure the engine type.

```
[edit]
user@host# set security utm feature-profile anti-virus type kaspersky-lab-engine
```

2. Create a profile for the Kaspersky Lab engine and configure a list of fallback options as block or log-and-permit.

```
[edit security utm feature-profile anti-virus kaspersky-lab-engine]
user@host# set profile kasprof1 fallback-options content-size block
user@host# set profile kasprof1 fallback-options corrupt-file block
user@host# set profile kasprof1 fallback-options decompress-layer block
user@host# set profile kasprof1 fallback-options default block
user@host# set profile kasprof1 fallback-options engine-not-ready block
user@host# set profile kasprof1 fallback-options out-of-resources block
user@host# set profile kasprof1 fallback-options password-file block
user@host# set profile kasprof1 fallback-options timeout block
user@host# set profile kasprof1 fallback-options too-many-requests block
```

**Results** From configuration mode, confirm your configuration by entering the **show security utm feature-profile anti-virus** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
user@host#show security utm feature-profile anti-virus
kaspersky-lab-engine {
  profile kasprof1 {
    fallback-options {
      default block;
      corrupt-file block;
      password-file block;
      decompress-layer block;
      content-size block;
      engine-not-ready block;
      timeout block;
      out-of-resources block;
      too-many-requests block;
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

**Verification**

To confirm that the configuration is working properly, perform this task:

- Verifying the Antivirus Scanning Fallback Options on page 906

**Verifying the Antivirus Scanning Fallback Options**

- Purpose** Verify the antivirus scanning fallback options.
- Action** From operational mode, enter the **show configuration security utm** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Full Antivirus Protection Overview on page 869
  - Understanding Full Antivirus Scan Level Settings on page 876

## Full Antivirus Configuration Overview

---

When configuring antivirus protection, you must first create the antivirus custom objects you are using. Those custom objects may include the MIME pattern list, MIME exception list, and the filename extension list. Once you have created your custom objects, you can configure full antivirus protection, including intelligent prescreening, and content size limits.

To configure full file-based antivirus protection:

1. Configure UTM custom objects for the UTM feature. The following example enables the mime-pattern, filename-extension, url-pattern, and custom-url-category custom-objects:

```
user@host# set security utm custom-objects mime-pattern
user@host# set security utm custom-objects filename-extension
user@host# set security utm custom-objects url-pattern
user@host# set security utm custom-objects custom-url-category
```

2. Configure the main feature parameters using feature profiles. The following example enables options using the anti-virus feature profile:

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine
pattern-update
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine profile
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine profile
fallback-options
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine profile
notification-options
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine profile
scan-options
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine profile
trickling
user@host# set security utm feature-profile anti-virus mime-whitelist
user@host# set security utm feature-profile anti-virus url-whitelist
```



3. Configure a UTM policy for each protocol and attach this policy to a profile. The following example configure the utmp2 UTM policy for the HTTP protocol:

```
user@host# set security utm utm-policy utmp2 anti-virus http-profile http1
```

4. Attach the UTM policy to a security policy. The following example attaches the utmp2 UTM policy to the p2 security policy:

```
user@host# set security policies from-zone trust to-zone untrust policy p2 then permit application-services utm-policy utmp2
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869
- Configuring Full Antivirus (J-Web Procedure) on page 907

## Configuring Full Antivirus (J-Web Procedure)

- Configuring Full Antivirus Custom Objects (J-Web Procedure) on page 907
- Configuring Full Antivirus Feature Profiles (J-Web Procedure) on page 909
- Configuring Full Antivirus UTM Policies (J-Web Procedure) on page 912
- Attaching Full Antivirus UTM Policies to Security Policies (J-Web Procedure) on page 913

### Configuring Full Antivirus Custom Objects (J-Web Procedure)

To configure antivirus protection, you must first create your custom objects (MIME Pattern List, Filename Extension List, URL Pattern List, and Custom URL Category List).

Configure a MIME pattern list custom object:

1. Select **Configure>Security>UTM>Custom Objects**.
2. From the MIME Pattern List tab, click the **Add** button to create MIME pattern lists.
3. In the Add MIME Pattern pop-up window, next to **MIME Pattern Name**, enter a unique name.



**NOTE:** Keep in mind that you are creating a MIME whitelist and a MIME exception list (if necessary). Both MIME lists appear in the MIME Whitelist and Exception MIME Whitelist fields when you configure antivirus. Therefore, the MIME list names you create should be as descriptive as possible.

4. Next to **MIME Pattern Value**, enter the MIME pattern.
5. Click **Add** to add your MIME pattern to the Values list box. Within this box, you can also select an entry and use the Delete button to delete it from the list. Continue to add MIME patterns in this manner.
6. Optionally, create a new MIME list to act as an exception list. The exception list is generally a subset of the main MIME list.

7. Click **OK** to check your configuration and save the selected values as part of the MIME list, then click **Commit Options>Commit**.
8. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

Configure a filename extension list custom object:

1. Select **Configure>Security>UTM>Custom Objects**.
2. From the Filename Extension List tab, click the **Add** button to create filename extension lists.
3. Next to **File Extension Name**, enter a unique name. This name appears in the Scan Option By Extension list when you configure an antivirus profile.
4. In the **Available Values** box, select one or more default values (press Shift to select multiple concurrent items or press Ctrl to select multiple separate items) and click the right arrow button to move the value or values to the Selected Values box.
5. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.
6. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If the profile is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

Configure a URL pattern list custom object:



**NOTE:** Because you use URL pattern lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure a custom URL category list.

1. Select **Configure>Security>UTM>Custom Objects**.
2. From the URL Pattern List tab, click the **Add** button to create URL pattern lists.
3. Next to URL Pattern Name, enter a unique name. This name appears in the Custom URL Category List Custom Object page for selection.
4. Next to **URL Pattern Value**, enter the URL or IP address you want added to the list for bypassing scanning.

When entering the URL pattern, note the following wildcard character support:

- The `\*\.[\]\?*` wildcard characters are supported.
- You must precede all wildcard URLs with `http://`.
- You can only use the asterisk `*` wildcard character if it is at the beginning of the URL and is followed by a period.
- You can only use the question mark `?` wildcard character at the end of the URL.

- The following wildcard syntax IS supported: **http://\*.juniper.net**, **http://www.juniper.ne?**, **http://www.juniper.n??**.
  - The following wildcard syntax is NOT supported: **\*juniper.net**, **www.juniper.ne?**, **http://\*juniper.net**, **http://\***.
5. Click **Add** to add your URL pattern to the Values list box. The list can contain up to 8192 items. You can also select an entry and use the Delete button to delete it from the list. Continue to add URLs or IP addresses in this manner.
  6. Click **OK** to check your configuration and save the selected values as part of the URL pattern list you have created, then click **Commit Options>Commit**.
  7. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

Configure a custom URL category list custom object:



**NOTE:** Because you use URL Pattern Lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure a custom URL category list.

1. Select **Configure>Security>UTM>Custom Objects**.
2. In the URL Category List tab, click **Add** to create URL category lists.
3. Next to **URL Category Name**, enter a unique name. This name appears in the URL Whitelist list when you configure antivirus global options.
4. In the **Available Values** box, select a URL Pattern List name from the list for bypassing scanning and click the right arrow button to move it to the Selected Values box.
5. Click **OK** to check your configuration and save the selected values as part of the URL list that you have created, then click **Commit Options>Commit**.  
Click **OK** to save the selected values as part of the custom URL list you have created.
6. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

## Configuring Full Antivirus Feature Profiles (J-Web Procedure)

After you have created your custom object, configure an antivirus feature profile:

1. Select **Configure>Security>UTM>Global options**.
2. In the Anti-Virus tab, next to **MIME whitelist**, select the custom object you created from the list.
3. Next to **Exception MIME whitelist**, select the custom object you created from the list.
4. Next to **URL Whitelist**, select the custom object you created from the list.

5. In the **Engine Type** section, select the type of engine you are using. For full antivirus protection, you should select **Kaspersky Lab**.
6. In the Kaspersky Lab Engine Option section, in the **Pattern update URL** box, enter the URL for the pattern database.



**NOTE:** The URL is `http://update.juniper-updates.net/AV/<device version>` and you should not change it.

7. Next to **Pattern update interval**, enter the time interval, in seconds, for automatically updating the pattern database in the box. The default interval is 60.
8. Select whether you want the pattern file to update automatically (**Auto update**) or not (**No Auto update**).
9. Click **OK** to save the selected values.
10. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in a pop-up window that appears to discover why.
11. Under Security, in the left pane, select **Anti-Virus**.
12. In the right window, click **Add** to create a profile for the antivirus Kaspersky Lab Engine. (To edit an existing item, select it and click the **Edit** button.)
13. Next to **Profile name**, enter a unique name for this antivirus profile.
14. Select the **Profile Type**. In this case, select **Kaspersky**.
15. Next to **Trickling timeout**, enter timeout parameters.



**NOTE:** Trickling applies only to HTTP. HTTP trickling is a mechanism used to prevent the HTTP client or server from timing out during a file transfer or during antivirus scanning.

16. Next to Intelligent prescreening, select **Yes** or **No**.



**NOTE:** Intelligent prescreening is only intended for use with non-encoded traffic. It is not applicable for mail protocols (SMTP, POP3, IMAP, and HTTP POST).

17. In the Scan Options section, next to Intelligent prescreening, select **Yes** if you are using it.



**NOTE:** Intelligent prescreening is only intended for use with non-encoded traffic. It is not applicable for mail protocols (SMTP, POP3, IMAP, and HTTP POST).

18. Next to **Content Size Limit**, enter content size parameters. The content size check occurs before the scan request is sent. The content size refers to accumulated TCP payload size.
19. Next to **Scan engine timeout**, enter scanning timeout parameters.
20. Next to **Decompress Layer Limit**, enter decompression layer limit parameters.
21. In the Scan mode section, select either **Scan all files**, if you are scanning all content, or **Scan files with specified extension**, if you are scanning by file extensions.



**NOTE:** If you select Scan files with specified extension, you must select a filename extension list custom object from the Scan engine filename extension list that appears.

22. Select the **Fallback settings** tab.
23. Next to Default (fallback option), select **Log and permit** or **Block** from the list. In most cases, Block is the default fallback option.
24. Next to Corrupt File (fallback option), select **Log and permit** or **Block** from the list.
25. Next to Password File (fallback option), select **Log and permit** or **Block** from the list.
26. Next to Decompress Layer (fallback option), select **Log and permit** or **Block** from the list.
27. Next to Content Size (fallback option), select **Log and permit** or **Block** from the list.
28. Next to Engine Not Ready (fallback option), select **Log and permit** or **Block** from the list.
29. Next to Timeout (fallback option), select **Log and permit** or **Block** from the list.
30. Next to Out Of Resources (fallback option), select **Log and permit** or **Block** from the list.
31. Next to Too Many Request (fallback option), select **Log and permit** or **Block** from the list.
32. Select the **Notification options** tab.
33. In the Fallback block section, next to Notification type, select **Protocol Only** or **Message** to select the type of notification that is sent when a fallback option of block is triggered.
34. Next to Notify mail sender, select **Yes** or **No**.
35. If you selected Yes, next to **Custom Message**, enter text for the message body of your custom message for this notification (if you are using a custom message).
36. Next to **Custom message subject**, enter text to appear in the subject line of your custom message for this notification (if you are using a custom message).
37. In the Fallback non block section, next to Notify mail recipient, select **Yes** or **No**.
38. If you selected Yes, next to **Custom Message**, enter text for the message body of your custom message for this notification (if you are using a custom message).

39. Next to **Custom message subject**, enter text to appear in the subject line of your custom message for this notification (if you are using a custom message).
40. Select the **Notification options cont** tab.
41. In the Virus detection section, next to Notification type, select **Protocol Only** or **Message** to select the type of notification that is sent when a fallback option of block is triggered.
42. Next to Notify mail sender, select **Yes** or **No**.
43. If you selected Yes, next to **Custom Message**, enter text for the message body of your custom message for this notification (if you are using a custom message).
44. Next to **Custom message subject**, enter text to appear in the subject line of your custom message for this notification (if you are using a custom message). The limit is 255 characters.
45. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.
46. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.



**NOTE:** You create a separate antivirus profile for each antivirus protocol. These profiles may basically contain the same configuration information, but when you are creating your UTM policy for an antivirus profile, the UTM policy configuration page provides separate antivirus profile selection fields for each supported protocol.

## Configuring Full Antivirus UTM Policies (J-Web Procedure)

After you have created an antivirus feature profile, configure a UTM policy to which you can attach the feature profile:

1. Select **Configure>Security>Policy>UTM Policies**.
2. From the UTM policy configuration window, click **Add** to configure a UTM policy. This action takes you to the policy configuration pop-up window.
3. Select the **Main** tab in pop-up window.
4. In the **Policy name** box, enter a unique name for the UTM policy.
5. In the **Session per client limit** box, enter a session per client limit from 0 to 20000 for this UTM policy.
6. In the **Session per client over limit** list, select the action that the device should take when the session per client limit for this UTM policy is exceeded. Options include **Log and permit** and **Block**.
7. Select the **Anti-Virus profiles** tab in the pop-up window.

8. Select the appropriate profile you have configured from the list for the corresponding protocol listed.
9. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.
10. If the policy is saved successfully, you receive a confirmation and you must click **OK** again. If the profile is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

### Attaching Full Antivirus UTM Policies to Security Policies (J-Web Procedure)

After you create a UTM policy, create a security policy and attach the UTM policy to the security policy:

1. Select **Configure>Security>Policy>FW Policies**.
2. From the Security Policy window, click **Add** to configure a security policy with UTM. This action takes you to the policy configuration pop-up window.
3. In the Policy tab, enter a name in the **Policy Name** box.
4. Next to **From Zone**, select a zone from the list.
5. Next to **To Zone**, select a zone from the list.
6. Choose a **Source Address**.
7. Choose a **Destination Address**.
8. Choose an application by selecting **junos-protocol** (for all protocols that support antivirus scanning) in the Application Sets box and clicking the —> button to move it to the Matched box.
9. Next to Policy Action, select **Permit**.



**NOTE:** When you select Permit for Policy Action, several additional fields become available in the Applications Services tab, including UTM Policy.

10. Select the **Application Services** tab in the pop-up window.
11. Next to **UTM Policy**, select the appropriate policy from the list. This action attaches your UTM policy to the security policy.
12. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.
13. If the policy is saved successfully, you receive a confirmation and you must click **OK** again. If the profile is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

You must activate your new policy to apply it.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Protection Overview on page 869

- [Configuring Full Antivirus \(J-Web Procedure\) on page 907](#)
- [Configuring Full Antivirus Custom Objects \(J-Web Procedure\) on page 907](#)
- [Configuring Full Antivirus Feature Profiles \(J-Web Procedure\) on page 909](#)

## Example: Configuring Full Antivirus (CLI)

---

- [Example: Configuring Full Antivirus Custom Objects on page 914](#)
- [Example: Configuring Full Antivirus Feature Profiles on page 917](#)
- [Example: Configuring Full Antivirus UTM Policies on page 922](#)
- [Example: Attaching Full Antivirus UTM Policies to Security Policies on page 923](#)

## Example: Configuring Full Antivirus Custom Objects

This example shows how to configure full antivirus custom objects.

- [Requirements on page 914](#)
- [Overview on page 914](#)
- [Configuration on page 914](#)
- [Verification on page 916](#)

### Requirements

---

Before you begin:

- Decide the type of full antivirus protection you require. See “Full Antivirus Protection Overview” on page 869.
- Understand the order in which full antivirus parameters are configured. See “Full Antivirus Pattern Update Configuration Overview” on page 871.

### Overview

---

In this example, you define custom objects that are used to create full antivirus feature profiles. You perform the following tasks to define custom objects:

1. Configure a filename extension list called `extlist1` and add extensions such as `.zip`, `.js`, and `.vbs` to the list.
2. Create two MIME lists called `avmime1` and `ex-avmime1` and add patterns to the list.
3. Configure a URL pattern list called `urllist1`.
4. Configure a custom URL category list called `custurl1` using the `urllist1` URL pattern list.

### Configuration

---

#### CLI Quick Configuration

To quickly configure full antivirus custom objects, copy the following commands and paste them into the CLI.

```
[edit]  
set security utm custom-objects filename-extension extlist1 value [zip js vbs]
```



```

set security utm custom-objects mime-pattern avmime1 value [video/quicktime
image/x-portable-anymap x-world/x-vrml]
set security utm custom-objects mime-pattern ex-avmime1 value
[video/quicktime-inappropriate]
set security utm custom-objects url-pattern urllist1 value [http://www.url.com 5.6.7.8]
set security utm custom-objects custom-url-category custurl1 value urllist1

```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure full antivirus filtering custom objects:

1. Configure the filename extension list and add extensions to it.

```

[edit security utm]
user@host# set custom-objects filename-extension extlist1 value [zip js vbs]

```



**NOTE:** The Kaspersky scan engine ships with a read-only default extension list that you can use.

2. Create MIME lists and add MIME patterns to the lists.

```

[edit security utm]
user@host# set custom-objects mime-pattern avmime1 value [video/quicktime
image/x-portable-anymap x-world/x-vrml]
user@host# set custom-objects mime-pattern ex-avmime1 value
[video/quicktime-inappropriate]

```

3. Configure a URL pattern list.

```

[edit security utm]
user@host# set custom-objects url-pattern urllist1 value [http://www.url.com
5.6.7.8]

```

When entering the URL pattern, note the following wildcard character support:

- The `\*\.[ ]\?*` wildcard characters are supported.
- You must precede all wildcard URLs with `http://`.
- You can only use the asterisk `*` wildcard character if it is at the beginning of the URL and is followed by a period.
- You can only use the question mark `?` wildcard character at the end of the URL.
- The following wildcard syntax is supported: `http://*juniper.net`, `http://www.juniper.ne?`, `http://www.juniper.n??`.
- The following wildcard syntax is not supported: `*juniper.net`, `www.juniper.ne?`, `http://*juniper.net`, `http://*`.



**NOTE:** Because you use URL pattern lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure custom URL category lists.

4. Configure a custom URL category list.

```
[edit security utm]
user@host# set custom-objects custom-url-category custurl1 value urllist1
```

**Results** From configuration mode, confirm your configuration by entering the **show security utm** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
userhost# show security utm
custom-objects {
  mime-pattern {
    avmime1 {
      value [ video/quicktime image/x-portable-anymap x-world/x-vrml ];
    }
    ex-avmime1 {
      value video/quicktime-inappropriate;
    }
  }
  filename-extension {
    extlist1 {
      value [ zip js vbs ];
    }
  }
  url-pattern {
    urllist1 {
      value [ http://www.url.com 5.6.7.8 ];
    }
  }
  custom-url-category {
    custurl1 {
      value urllist1;
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Full Antivirus Custom Objects on page 916

#### *Verifying Full Antivirus Custom Objects*

**Purpose** Verify the full antivirus custom objects.

**Action** From operational mode, enter the **show configuration security utm** command.

- Related Documentation**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Example: Configuring Full Antivirus Feature Profiles on page 917
  - Configuring Full Antivirus Feature Profiles (J-Web Procedure) on page 909

## Example: Configuring Full Antivirus Feature Profiles

This example shows how to configure a full antivirus feature profile.

- Requirements on page 917
- Overview on page 917
- Configuration on page 919
- Verification on page 922

### Requirements

Before you begin:

- Decide the type of full antivirus protection you require. See “Full Antivirus Protection Overview” on page 869.
- Understand the order in which full antivirus parameters are configured. See “Full Antivirus Configuration Overview” on page 906.
- MIME patterns must be defined for lists and exception lists. See “Example: Configuring MIME Whitelists to Bypass Antivirus Scanning” on page 890.

### Overview

In this example, you configure a feature profile called `kasprof1` and specify custom objects to be used for filtering content:

- Select and configure the engine type as Kaspersky Lab Engine.
- Select 120 as the time interval for updating the pattern database. The default full file-based antivirus pattern-update interval is 60 minutes.



**NOTE:** The command for changing the URL for the pattern database is:

```
[edit]
user@host# edit security utm feature-profile anti-virus kaspersky-lab-engine
[edit security utm feature-profile anti-virus kaspersky-lab-engine]
user@host# set pattern-update url http://.
```

The default URL is `http://update.juniper-updates.net/AV/<device-version>`. You should not change this URL unless you are experiencing problems with it and have called for support.

- Enable an e-mail notification with a custom message as pattern file was updated and a custom subject line as AV pattern file updated.
- Configure a list of fallback options as block.
- Configure the notification options for fallback blocking for virus detection. Configure a custom message for the fallback blocking action.
- Configure a notification for protocol-only virus detection.
- Configure scan options. For this example, configure the device to perform a TCP payload content size check before the scan request is sent.
- Configure the decompression layer limit. For this example configure the device to decompress three layers of nested compressed files before it executes the virus scan.
- Configure content size parameters as 20000.



**NOTE:** For SRX100, SRX110, SRX210, SRX220, and SRX240 devices the content size is 20000. For SRX650 device the content size is 40000.

- Configure scan extension settings. The default list is junos-default-extension. For this example, you select extlist1, which you created as a custom object.
- Configure the scan mode setting to configure the device to use a custom extension list. Although you can choose to scan all files, for this example you select only files with the extensions that you specify.
- Enable intelligent prescreening and set its timeout setting to 1800 seconds and trickling setting (applicable only to HTTP) to 600 seconds. This means that if the device receives a packet within a 600-second period during a file transfer or while performing an antivirus scan, it should not time out.



**NOTE:** Intelligent prescreening is only intended for use with non-encoded traffic. It is not applicable for mail protocols (SMTP, POP3, IMAP) and HTTP POST.

The following example disables intelligent prescreening for the kasprof1 profile:

```
[edit security utm feature-profile anti-virus kaspersky-lab-engine]
user@host# set profile kasprof1 scan-options no-intelligent-prescreening
```

- Configure the antivirus scanner to use MIME bypass lists and exception lists. You can use your own custom object lists, or you can use the default list that ships with the device called junos-default-bypass-mime. For this example, you use the avmime1 and ex-avmime1 lists.
- Configure the antivirus module to use URL bypass lists. If you are using a URL whitelist (valid only for HTTP traffic), this is a custom URL category that you have previously configured as a custom object. For this example, you enable the custurl1 bypass list.

## Configuration

**CLI Quick Configuration** To quickly configure an antivirus feature profile, copy the following commands and paste them into the CLI.

```
[edit]
set security utm feature-profile anti-virus type kaspersky-lab-engine
set security utm feature-profile anti-virus kaspersky-lab-engine pattern-update interval
  120
set security utm feature-profile anti-virus kaspersky-lab-engine pattern-update
  email-notify admin-email administrator@juniper.net custom-message
  patternfilewasupdated custom-message-subject AVpatternfileupdated
set security utm feature-profile anti-virus type kaspersky-lab-engine
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options content-size block
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options corrupt-file block
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options decompress-layer block
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options default block
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options engine-not-ready block
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options out-of-resources block
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options password-file block
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options timeout block
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  fallback-options too-many-requests block
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  notification-options fallback-block custom-message "Dropped due to fallback settings"
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  notification-options virus-detection type protocol-only
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  scan-options content-size-limit 20000
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  scan-options decompress-layer-limit 3
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  scan-options intelligent-prescreening
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  scan-options scan-extension extlist1
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  scan-options scan-mode by-extension
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
  scan-options timeout 1800
set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1 trickling
  timeout 600
set security utm feature-profile anti-virus mime-whitelist list avmime1
set security utm feature-profile anti-virus mime-whitelist list avmime1 exception
  ex-avmime1
set security utm feature-profile anti-virus url-whitelist custurl1
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure full antivirus feature profiles:

1. Select and configure the engine type.

```
[edit]
user@host# set security utm feature-profile anti-virus type kaspersky-lab-engine
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine
pattern-update interval 120
```

2. Configure the device to notify a specified administrator when patterns are updated.

```
[edit security utm feature-profile anti-virus kaspersky-lab-engine]
user@host# set pattern-update email-notify admin-email administrator@juniper.net
custom-message patternfilewasupdated custom-message-subject
AVpatternfileupdated
```

3. Create a profile for the Kaspersky Lab engine and configure fallback options as block.

```
[edit security utm feature-profile anti-virus kaspersky-lab-engine]
user@host# set profile kasprof1 fallback-options content-size block
user@host# set profile kasprof1 fallback-options corrupt-file block
user@host# set profile kasprof1 fallback-options decompress-layer block
user@host# set profile kasprof1 fallback-options default block
user@host# set profile kasprof1 fallback-options engine-not-ready block
user@host# set profile kasprof1 fallback-options out-of-resources block
user@host# set profile kasprof1 fallback-options password-file block
user@host# set profile kasprof1 fallback-options timeout block
user@host# set profile kasprof1 fallback-options too-many-requests block
```

4. Configure a custom notification for the fallback blocking action and send a notification.

```
[edit security utm feature-profile anti-virus kaspersky-lab-engine]
user@host# set profile kasprof1 notification-options fallback-block custom-message
"Dropped due to fallback settings"
```

5. Configure a notification for protocol-only virus detection.

```
[edit security utm feature-profile anti-virus kaspersky-lab-engine]
user@host# set profile kasprof1 notification-options virus-detection type
protocol-only
```

6. Configure content size parameter.

```
[edit security utm feature-profile anti-virus kaspersky-lab-engine]
user@host# set profile kasprof1 scan-options content-size-limit 20000
```

7. Configure the decompression layer limit.

```
[edit security utm feature-profile anti-virus kaspersky-lab-engine]
user@host# set profile kasprof1 scan-options decompress-layer-limit 3
```

8. Configure intelligent prescreening.

```
[edit security utm feature-profile anti-virus kaspersky-lab-engine]
user@host# set profile kasprof1 scan-options intelligent-prescreening
```

9. Configure scan extension setting.
 

```
[edit security utm feature-profile anti-virus kaspersky-lab-engine]
user@host# set profile kasprof1 scan-options scan-extension extlist1
```
10. Configure the scan mode setting.
 

```
[edit security utm feature-profile anti-virus kaspersky-lab-engine]
user@host# set profile kasprof1 scan-options scan-mode by-extension
```
11. Configure the timeout setting.
 

```
[edit security utm feature-profile anti-virus kaspersky-lab-engine]
user@host# set profile kasprof1 scan-options timeout 1800
```
12. Configure trickling setting.
 

```
[edit security utm feature-profile anti-virus kaspersky-lab-engine]
user@host# set profile kasprof1 trickling timeout 600
```
13. Configure the antivirus scanner to use MIME bypass lists and exception lists.
 

```
[edit security utm feature-profile anti-virus]
user@host# set mime-whitelist list avmime1
user@host# set mime-whitelist list avmime1 exception ex-avmime1
```
14. Configure the antivirus module to use URL bypass lists.
 

```
[edit security utm feature-profile anti-virus]
user@host# set url-whitelist custurl1
```

**Results** From configuration mode, confirm your configuration by entering the **show security utm feature-profile anti-virus** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security utm feature-profile anti-virus
mime-whitelist {
  list avmime1;
  exception ex-avmime1;
}
url-whitelist custurl1;
kaspersky-lab-engine {
  pattern-update {
    email-notify {
      admin-email "administrator@juniper.net";
      custom-message patternfilewasupdated;
      custom-message-subject AVpatternfileupdated;
    }
    interval 120;
  }
  profile kasprof1 {
    fallback-options {
      default block;
      corrupt-file block;
      password-file block;
      decompress-layer block;
      content-size block;
      engine-not-ready block;
```

```

        timeout block;
        out-of-resources block;
        too-many-requests block;
    }
    scan-options {
        intelligent-prescreening;
        scan-mode by-extension;
        scan-extension extlist1;
        content-size-limit 20000;
        timeout 1800;
        decompress-layer-limit 3;
    }
    trickling timeout 600;
    notification-options {
        virus-detection {
            type protocol-only;
            custom-message ***virus-found***;
        }
        fallback-block {
            custom-message "Dropped due to fallback settings";
        }
    }
}
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- [Verifying the Configuration of Full Antivirus Feature Profile on page 922](#)

#### **Verifying the Configuration of Full Antivirus Feature Profile**

**Purpose** Verify the full antivirus feature profile.

**Action** From operational mode, enter the **show configuration security utm** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Full Antivirus Configuration Overview on page 906](#)
- [Example: Configuring Full Antivirus UTM Policies on page 922](#)
- [Example: Attaching Full Antivirus UTM Policies to Security Policies on page 923](#)

### Example: Configuring Full Antivirus UTM Policies

This example shows how to create a UTM policy to attach to a feature profile.

- [Requirements on page 923](#)
- [Overview on page 923](#)



- Configuration on page 923
- Verification on page 923

### Requirements

---

Before you begin, create an antivirus feature profile. See “Example: Configuring Full Antivirus Feature Profiles” on page 917.

### Overview

---

In this example, you configure a full antivirus UTM policy called utmp2 and attach the policy to an HTTP profile called kasprofile1 HTTP.

### Configuration

---

#### Step-by-Step Procedure

To configure a full antivirus UTM policy:

1. Create a UTM policy for HTTP antivirus scanning and attach the policy to the profile.

[edit]

```
user@host# set security utm utm-policy utmp2 anti-virus http-profile kasprofile1
```

2. If you are done configuring the device, commit the configuration.

[edit]

```
user@host# commit
```

### Verification

---

To verify the configuration is working properly, enter the **show security utm** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Configuration Overview on page 906
- Example: Configuring Full Antivirus Feature Profiles on page 917
- Example: Attaching Full Antivirus UTM Policies to Security Policies on page 923

## Example: Attaching Full Antivirus UTM Policies to Security Policies

This example shows how to attach a UTM policy to a security policy.

- Requirements on page 923
- Overview on page 923
- Configuration on page 924
- Verification on page 924

### Requirements

---

Before you begin, create a UTM policy. See “Example: Configuring Full Antivirus UTM Policies” on page 922.

### Overview

---

In this example, you attach the UTM policy called utmp2 to the security policy called p2.

## Configuration

### Step-by-Step Procedure

To attach a full antivirus UTM policy to a security policy:

1. Enable and configure the security policy.
 

```
[edit]
user@host# set security policies from-zone trust to-zone untrust policy p2 match
source-address any
user@host# set security policies from-zone trust to-zone untrust policy p2 match
destination-address any
user@host# set security policies from-zone trust to-zone untrust policy p2 match
application junos-http
```
2. Attach the UTM policy to the security policy.
 

```
[edit]
user@host# set security policies from-zone trust to-zone untrust policy p2 then
permit application-services utm-policy utmp2
```
3. If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

## Verification

To verify the configuration is working properly, enter the **show security policies** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Full Antivirus Configuration Overview on page 906
- Example: Configuring Full Antivirus UTM Policies on page 922

## Monitoring Antivirus Sessions and Scan Results

The antivirus module provides functions which allow you to use the CLI to check the system settings and the status of scan engine. It also provides functions to check the ongoing antivirus sessions and antivirus statistics.

- [Monitoring Antivirus Scan Engine Status on page 924](#)
- [Monitoring Antivirus Session Status on page 925](#)
- [Monitoring Antivirus Scan Results on page 925](#)

### Monitoring Antivirus Scan Engine Status

**Purpose** Using the CLI, you can view the following scan engine status items:

Antivirus license key status

- View license expiration dates.

Scan engine status and settings

- View last action result.

- View default file extension list.

Antivirus pattern update server settings

- View update URL (HTTP or HTTPS-based).
- View update interval.

Antivirus pattern database status

- View auto update status.
- View last result of database loading.
- If the download completes, view database version timestamp virus record number.
- If the download fails, view failure reason.

**Action** In the CLI, enter the `user@host> show security utm anti-virus status` command.

Example status result:

```
AV Key Expire Date: 03/01/2010 00:00:00
Update Server: http://update.juniper-updates.net/AV/SRX210
interval: 60 minutes
auto update status: next update in 12 minutes
last result: new database loaded
AV signature version: 12/21/2008 00:35 GMT, virus records: 154018
Scan Engine Info: last action result: No error(0x00000000)
```

## Monitoring Antivirus Session Status

**Purpose** Using the CLI, you can view the following session status items:

Antivirus session status displays a snapshot of current antivirus sessions. It includes

- Maximum supported antivirus session numbers.
- Total allocated antivirus session numbers.
- Total freed antivirus session numbers.
- Current active antivirus session numbers.

**Action** In the CLI, enter the `user@host> show security utm session status` command.

## Monitoring Antivirus Scan Results

**Purpose** View statistics for antivirus requests, scan results, and fallback counters.

Scan requests provide

- The total number of scan request forwarded to the engine.
- The number of scan request being pre-windowed.

- The number of scan requests using scan-all mode.
- The number of scan requests using scan-by-extension mode.

Scan code counters provide

- Number of clean files.
- Number of infected files.
- Number of password protected files.
- Number of decompress layers.
- Number of corrupt files.
- When the engine is out of resources.
- When there is an internal error.

Fallback applied status provides either a log-and-permit or block result when the following has occurred

- Scan engine not ready.
- Maximum content size reached.
- Too many requests.
- Password protected file found.
- Decompress layer too large.
- Corrupt file found.
- Timeout occurred.
- Out of resources.
- Other.

**Action** To view antivirus scan results using the CLI editor, enter the `user@host> show security utm anti-virus statistics status` command.

To view antivirus scan results using J-Web:

1. Select **Monitor>Security>UTM>Anti-Virus**.

The following information becomes viewable in the right pane.

Antivirus license key status

- View license expiration dates.

Antivirus pattern update server settings

- View update URL (HTTP or HTTPS-based).
- View update interval.

#### Antivirus pattern database status

- View auto update status.
- View last result of database loading.
- If the download completes, view database version timestamp virus record number.
- If the download fails, view failure reason.

#### Antivirus statistics provide

- The number of scan request being pre-windowed.
- The total number of scan request forwarded to the engine.
- The number of scan requests using scan-all mode.
- The number of scan requests using scan-by-extension mode.

#### Scan code counters provide

- Number of clean files.
- Number of infected files.
- Number of password protected files.
- Number of decompress layers.
- Number of corrupt files.
- When the engine is out of resources.
- When there is an internal error.

Fallback applied status provides either a log-and-permit or block result when the following has occurred

- Scan engine not ready.
- Password protected file found.
- Decompress layer too large.
- Corrupt file found.
- Out of resources.
- Timeout occurred.
- Maximum content size reached.
- Too many requests.
- Other.

2. You can click the **Clear Anti-Virus Statistics** button to clear all current viewable statistics and begin collecting new statistics.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Full Antivirus Configuration Overview on page 906](#)
- [Monitoring Antivirus Sessions and Scan Results on page 924](#)
- [Monitoring Antivirus Scan Engine Status on page 924](#)
- [Monitoring Antivirus Session Status on page 925](#)

# Express Antivirus Protection

- Express Antivirus Protection Overview on page 929
- Express Antivirus Scanner Pattern Database on page 931
- Express Antivirus Configuration Overview on page 934
- Configuring Express Antivirus (J-Web Procedure) on page 935
- Example: Configuring Express Antivirus (CLI) on page 941

## Express Antivirus Protection Overview

---

Express antivirus scanning is offered as a less CPU intensive alternative to the full file-based antivirus feature. Express antivirus supports the same protocols as full antivirus and functions in much the same manner, however, it has a smaller memory footprint, compatible with the smaller system memory present on lower end devices.



**NOTE:** If you switch from express antivirus protection to full file-based antivirus protection, you must reboot the device in order for full file-based antivirus to begin working.

This topic includes the following sections:

- Express Antivirus Packet-Based Scanning Versus File-Based Scanning on page 929
- Express Antivirus Expanded MIME Decoding Support on page 930
- Express Antivirus Scan Result Handling on page 930
- Express Antivirus Intelligent Prescreening on page 930
- Express Antivirus Limitations on page 930

## Express Antivirus Packet-Based Scanning Versus File-Based Scanning

Express antivirus uses a different antivirus scan engine than the full file-based antivirus feature and a different back-end hardware engine to accelerate pattern matching for higher data throughput.

The packet based scanning done by express antivirus provides virus scanning data buffers without waiting for entire file to be received by the firewall, whereas the file-based scanning done by full antivirus can only start virus scanning when entire file is received.

## Express Antivirus Expanded MIME Decoding Support

Express antivirus offers MIME decoding support for HTTP, POP3, SMTP, and IMAP. MIME decoding support includes the following for each supported protocol:

- Multi-part and nested header decoding
- Base64 decoding, printed quote decoding, and encoded word decoding (in the subject field)

## Express Antivirus Scan Result Handling

With express antivirus, the TCP traffic is closed gracefully when a virus is found and the data content is dropped.



.....

**NOTE:** Express antivirus supports the following fail mode options: default, engine-not-ready, out-of-resource, and too-many-requests. Fail mode handling of supported options with express antivirus is much the same as with full antivirus.

.....

## Express Antivirus Intelligent Prescreening

Intelligent prescreening functionality is identical in both express antivirus and full antivirus.

## Express Antivirus Limitations

Express antivirus has the following limitations when compared to full antivirus functionality:

- Express antivirus provides limited support for the scanning of file archives and compressed file formats. Express antivirus can only support gzip, deflate and compressed compressing formats.
- Express antivirus provides limited support for decompression. Decompression is only supported with HTTP (supports only gzip, deflate, and compress for HTTP and only supports one layer of compression) and POP3 (supports only gzip for POP3 and only supports one layer of compression).
- Express antivirus does not support scanning by extension.
- Express antivirus scanning is interrupted when the scanning database is loading.
- Express antivirus may truncate a warning message if a virus has been detected and the replacement warning message that is sent is longer than the original content it is replacing.





**NOTE:** Because express antivirus does only packet based string matching, if you use the standard EICAR file to test express antivirus, you will see false positives. To avoid these false positives, Juniper has disabled scanning on the standard EICAR file to create a modified EICAR file for testing express antivirus. You can download this modified EICAR file from the following links:

<http://www.juniper.net/security/avtest/ss-eicar.txt>

<http://www.juniper.net/security/avtest/ss-eicar.com>

<http://www.juniper.net/security/avtest/ss-eicar.zip>

#### Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Express Antivirus Scanner Pattern Updates on page 931
- Express Antivirus Configuration Overview on page 934
- Example: Automatically Updating Express Antivirus Patterns on page 933

## Express Antivirus Scanner Pattern Database

- Understanding Express Antivirus Scanner Pattern Updates on page 931
- Example: Automatically Updating Express Antivirus Patterns (J-Web) on page 932
- Example: Automatically Updating Express Antivirus Patterns on page 933
- Manually Updating, Reloading, and Deleting Express Antivirus Patterns (CLI Procedure) on page 934

## Understanding Express Antivirus Scanner Pattern Updates

Express antivirus uses a different signature database than the full antivirus signature database. The express antivirus signature database is called Juniper Express antivirus database and it is compatible with the hardware engine. The express signature database targets only critical viruses and malware, including worms, Trojans, and spyware. This is a smaller sized database, providing less coverage than the full antivirus signature database.

The express antivirus pattern database is updated over HTTP or HTTPS and can occur automatically or manually. This is similar functionality to that found in full antivirus with some minor differences:

- With express antivirus, the signature database auto-update interval, is once a day.
- With express antivirus, there is no support for the downloading of multiple database types.
- With express antivirus, during database loading, all scan operations are interrupted. Scan operations for existing traffic flows are stopped and no new scan operations are initiated for newly established traffic flows. You can specify the desired action for this

interruption period using the **fall-back** parameter for **engine-busy-loading-database**. The available actions are **block** or **log-and-permit**.

- By default, the URL for express antivirus is <http://update.juniper-updates.net/EAV/SRX210>. "SRX210" in the URL is the platform name. This part of the URL is different and platform specific for each platform. (Other than the platform name, you should not change this URL unless you are experiencing problems with it and have called for support.)



**NOTE:** Once your subscription expires, you have a 30 day grace period during which you can continue to update the antivirus pattern file. Once that grace period expires, the update server no longer permits antivirus pattern file updates.

The express Antivirus scanning feature is a separately licensed subscription service. When your antivirus license key expires, you can continue to use locally stored antivirus signatures. But in that case, if the local database is deleted, antivirus scanning is disabled.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Express Antivirus Protection Overview on page 929
- Example: Automatically Updating Express Antivirus Patterns (J-Web) on page 932
- Example: Automatically Updating Express Antivirus Patterns on page 933
- Manually Updating, Reloading, and Deleting Express Antivirus Patterns (CLI Procedure) on page 934

### Example: Automatically Updating Express Antivirus Patterns (J-Web)

In this example, you configure the security device to update the pattern file automatically every 120 minutes. (The default antivirus pattern-update interval is once a day.)

To automatically update antivirus patterns:

1. Select **Configure>Security>UTM>Anti-Virus**.
2. Next to Interval, in the Juniper Express Engine section, enter **120** in the box.
3. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Express Antivirus Protection Overview on page 929
- Understanding Express Antivirus Scanner Pattern Updates on page 931
- Example: Automatically Updating Express Antivirus Patterns on page 933

- Manually Updating, Reloading, and Deleting Express Antivirus Patterns (CLI Procedure) on page 934

## Example: Automatically Updating Express Antivirus Patterns

This example shows how to update the pattern file automatically on a security device.

- Requirements on page 933
- Overview on page 933
- Configuration on page 933
- Verification on page 933

### Requirements

Before you begin:

- Obtain a valid antivirus scanner license. See “Full Antivirus Protection Overview” on page 869.
- Get network connectivity and access to the pattern database server. See “Understanding Full Antivirus Pattern Updates” on page 870.
- Configure your DNS settings and port settings (port 80) correctly. See “DNS Overview” on page 127.

### Overview

In this example, you configure the security device to update the pattern file automatically every 120 minutes. (The default antivirus pattern-update interval is once a day.)

### Configuration

#### Step-by-Step Procedure

To configure the security device to update the pattern file automatically:

1. Set the interval.

```
[edit]
user@host# set security utm feature-profile anti-virus juniper-express-engine
pattern-update interval 120
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security utm** command.

#### Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Express Antivirus Protection Overview on page 929
- Understanding Express Antivirus Scanner Pattern Updates on page 931
- Example: Automatically Updating Express Antivirus Patterns (J-Web) on page 932

- Manually Updating, Reloading, and Deleting Express Antivirus Patterns (CLI Procedure) on page 934

## Manually Updating, Reloading, and Deleting Express Antivirus Patterns (CLI Procedure)

To manually update antivirus patterns, enter the following CLI statement:

```
user@host> request security utm anti-virus juniper-express-engine pattern-update
```

To manually reload antivirus patterns, enter the following CLI statement:

```
user@host> request security utm anti-virus juniper-express-engine pattern-reload
```

To manually delete antivirus patterns, enter the following CLI statement:

```
user@host> request security utm anti-virus juniper-express-engine pattern-delete
```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Express Antivirus Protection Overview on page 929
- Understanding Express Antivirus Scanner Pattern Updates on page 931
- Example: Automatically Updating Express Antivirus Patterns (J-Web) on page 932
- Example: Automatically Updating Express Antivirus Patterns on page 933

## Express Antivirus Configuration Overview

For each UTM feature, you should configure feature parameters in the following order:

1. Configure UTM custom objects for the UTM features. The following example enables the mime-pattern, url-pattern, and custom-url-category custom objects:

```
user@host# set security utm custom-objects mime-pattern
user@host# set security utm custom-objects url-pattern
user@host# set security utm custom-objects custom-url-category
```

2. Configure main feature parameters using feature profiles. The following examples enables the anti-virus feature profile:

```
user@host# set security utm feature-profile anti-virus juniper-express-engine
```

3. Configure a UTM policy for each protocol and attach this policy to a profile. The following example creates the utmp3 UTM policy for the HTTP protocol:

```
user@host# set security utm utm-policy utmp3 anti-virus http-profile http1
```

4. Attach the UTM policy to a security policy. The following example attaches the utmp3 UTM policy to the p3 security policy:

```
user@host# set security policies from-zone trust to-zone untrust policy p3 then permit
application-services utm-policy utmp3
```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Express Antivirus Protection Overview on page 929

- Configuring Express Antivirus (J-Web Procedure) on page 935
- Configuring Express Antivirus Custom Objects (J-Web Procedure) on page 935
- Configuring Express Antivirus Feature Profiles (J-Web Procedure) on page 937
- Configuring Express Antivirus UTM Policies (J-Web Procedure) on page 939

## Configuring Express Antivirus (J-Web Procedure)

- Configuring Express Antivirus Custom Objects (J-Web Procedure) on page 935
- Configuring Express Antivirus Feature Profiles (J-Web Procedure) on page 937
- Configuring Express Antivirus UTM Policies (J-Web Procedure) on page 939
- Attaching Express Antivirus UTM Policies to Security Policies (J-Web Procedure) on page 940

## Configuring Express Antivirus Custom Objects (J-Web Procedure)

To configure express antivirus protection using the J-Web configuration editor, you must first create your custom objects (MIME pattern list, URL pattern list, and custom URL category list).

Configure a MIME pattern list custom object as follows:

1. Select **Configure>Security>UTM Custom Objects**.
2. From the MIME Pattern List tab, click **Add** to create MIME pattern lists.
3. In the Add MIME Pattern pop-up window, next to **MIME Pattern Name**, enter a unique name.



**NOTE:** Keep in mind that you are creating a MIME whitelist and a MIME exception list (if necessary). Both MIME lists appear in the MIME Whitelist and Exception MIME Whitelist fields when you configure antivirus. Therefore, the MIME list names you create should be as descriptive as possible.

4. Next to **MIME Pattern Value**, enter the MIME pattern.
5. Click **Add** to add your MIME pattern to the Values list box. Within this box, you can also select an entry and use the Delete button to delete it from the list. Continue to add MIME patterns in this manner.
6. Optionally, create a new MIME list to act as an exception list. The exception list is generally a subset of the main MIME list.
7. Click **OK** to check your configuration and save the selected values as part of the MIME list, then click **Commit Options>Commit**.
8. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

Configure a URL pattern list custom object as follows:



**NOTE:** Because you use URL pattern lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure a custom URL category list.

1. Select **Configure>Security>UTM>Custom Objects**.
2. From the URL Pattern List tab, click **Add** to create URL pattern lists.
3. Next to **URL Pattern Name**, enter a unique name. This name appears in the Custom URL Category List Custom Object page for selection.
4. Next to **URL Pattern Value**, enter the URL or IP address you want added to list for bypassing scanning.

When entering the URL pattern, note the following wildcard character support:

- The `\*\.[\]\?*` wildcard characters are supported.
  - You must precede all wildcard URLs with `http://`.
  - You can only use the asterisk `*` wildcard character if it is at the beginning of the URL and is followed by a period.
  - You can only use the question mark `?` wildcard character at the end of the URL.
  - The following wildcard syntax IS supported: `http://*.juniper.net`, `http://www.juniper.ne?`, `http://www.juniper.n??`.
  - The following wildcard syntax is NOT supported: `*juniper.net`, `www.juniper.ne?`, `http://*juniper.net`, `http://*`.
5. Click **Add** to add your URL pattern to the Values list box. The list can contain up to 8192 items. You can also select an entry and use the Delete button to delete it from the list. Continue to add URLs or IP addresses in this manner.
  6. Click **OK** to check your configuration and save the selected values as part of the URL pattern list, then click **Commit Options>Commit**.
  7. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

Configure a custom URL category list custom object using the URL pattern list that you created:

1. Select **Configure>Security>UTM>Custom Objects**.
2. From the URL Category List tab, click **Add** to create URL category lists.
3. Next to **URL Category Name**, enter a unique name. This name appears in the URL Whitelist list when you configure antivirus global options.

4. In the Available Values box, select a **URL Pattern List** name from the list for bypassing scanning and click the right arrow button to move it to the Selected Values box.
5. Click **OK** to check your configuration and save the selected values as part of the URL list, then click **Commit Options>Commit**.
6. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

## Configuring Express Antivirus Feature Profiles (J-Web Procedure)

After you create your custom objects, configure the antivirus feature profile:

1. Select **Configure>Security>UTM>Global options**.
2. In the Anti-Virus tab, next to **MIME whitelist**, select the custom object you created from the list.
3. Next to **Exception MIME whitelist**, select the custom object you created from the list.
4. Next to **URL Whitelist**, select the custom object you created from the list.
5. In the Engine Type section, select the type of engine you are using. For express antivirus protection, you should select **Juniper Express**.
6. Next to **Pattern update URL**, enter the URL for the pattern database in the box. Note that the URL is `http://update.juniper-updates.net/EAV/<device version>` and you should not change it.
7. Next to **Pattern update interval**, enter the time interval for automatically updating the pattern database in the box. The default for express antivirus checking is once per day.
8. Select whether you want the pattern file to update automatically (**Auto update**) or not (**No Auto update**).
9. Click **OK** to save the selected values.
10. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.
11. Under Security, in the left pane, select **Anti-Virus**.
12. Click **Add** in the right window to create a profile for the antivirus Juniper Express Engine. To edit an existing item, select it and click **Edit**.
13. In the Main tab, next to **Profile name**, enter a unique name for this antivirus profile.
14. Select the Profile Type. In this case, select **Juniper Express**.
15. Next to **Trickling timeout**, enter timeout parameters.



**NOTE:** Trickling applies only to HTTP. HTTP trickling is a mechanism used to prevent the HTTP client or server from timing out during a file transfer or during antivirus scanning.

16. Next to Intelligent prescreening, select **Yes** or **No**.



**NOTE:** Intelligent prescreening is only intended for use with non-encoded traffic. It is not applicable for mail protocols (SMTP, POP3, IMAP, and HTTP POST).

17. Next to **Content Size Limit**, enter content size parameters. The content size check occurs before the scan request is sent. The content size refers to accumulated TCP payload size.
18. Next to **Scan engine timeout**, enter scanning timeout parameters.
19. Select the **Fallback settings** tab.
20. Next to Default (fallback option), select **Log and permit** or **Block** from the list. In most cases, Block is the default fallback option.
21. Next to Decompress Layer (fallback option), select **Log and permit** or **Block** from the list.
22. Next to Content Size (fallback option), select **Log and permit** or **Block** from the list.
23. Next to Engine Not Ready (fallback option), select **Log and permit** or **Block** from the list.
24. Next to Timeout (fallback option), select **Log and permit** or **Block** from the list.
25. Next to Out of Resource (fallback option), select **Log and permit** or **Block** from the list.
26. Next to Too Many Requests (fallback option), select **Log and permit** or **Block** from the list.
27. Select the **Notification options** tab.
28. In the Fallback block section, next to Notification type, select **Protocol Only** or **Message** to select the type of notification that is sent when a fallback option of block is triggered.
29. Next to Notify mail sender, select **Yes** or **No**.
30. If you selected Yes, next to **Custom Message**, enter text for the message body of your custom message for this notification (if you are using a custom message).
31. Next to **Custom message subject**, enter text to appear in the subject line of your custom message for this notification (if you are using a custom message).
32. In the Fallback non block section, next to Notify mail recipient, select **Yes** or **No**.
33. If you selected Yes, next to **Custom Message**, enter text for the message body of your custom message for this notification (if you are using a custom message).
34. Next to **Custom message subject**, enter text to appear in the subject line of your custom message for this notification (if you are using a custom message).
35. Select the **Notification options cont** tab.



36. In the Virus detection section, next to Notification type, select **Protocol Only** or **Message** to select the type of notification that is sent when a fallback option of block is triggered.
37. Next to Notify mail sender, select **Yes** or **No**.
38. If you selected Yes, next to **Custom Message**, enter text for the message body of your custom message for this notification (if you are using a custom message).
39. Next to **Custom message subject**, enter text to appear in the subject line of your custom message for this notification (if you are using a custom message). The limit is 255 characters.
40. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.
41. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up that appears window to discover why.



**NOTE:** You create a separate antivirus profile for each antivirus protocol. These profiles may basically contain the same configuration information, but when you are creating your UTM policy for antivirus, the UTM policy configuration page provides separate antivirus profile selection fields for each supported protocol.

## Configuring Express Antivirus UTM Policies (J-Web Procedure)

After you have created an antivirus feature profile, configure a UTM policy to which you can attach the feature profile:

1. Select **Configure>Security>Policy>UTM Policies**.
2. From the UTM policy configuration window, click **Add** to configure a UTM policy. The policy configuration pop-up window appears.
3. Select the **Main** tab.
4. In the **Policy name** box, enter a unique name.
5. In the **Session per client limit** box, enter a session per client limit from 0 to 20000 for this UTM policy.
6. In the Session per client over limit list, select the action that the device should take when the session per client limit for this UTM policy is exceeded. Options include **Log and permit** and **Block**.
7. Select the **Anti-Virus profiles** tab.
8. Select the appropriate profile you have configured from the list for the corresponding protocol listed.

9. Click **OK**.
10. If the policy is saved successfully, you receive a confirmation and you must click **OK** again. If the profile is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

### Attaching Express Antivirus UTM Policies to Security Policies (J-Web Procedure)

After you create a UTM policy, create a security policy and attach the UTM policy to the security policy:

1. Select **Configure>Security>Policy>FW Policies**.
2. From the Security Policy window, click **Add** to configure a security policy with UTM. The policy configuration pop-up window appears.
3. In the Policy tab, enter a name in the **Policy Name** box.
4. Next to Default Policy Action, select one of the following: **Deny-All** or **Permit-All**.
5. Next to **From Zone**, select a zone from the list.
6. Next to **To Zone**, select a zone from the list.
7. Under Zone Direction, click **Add a Policy**.
8. Choose a **Source Address**.
9. Choose a **Destination Address**.
10. Choose an application by selecting **junos-protocol** (for all protocols that support antivirus scanning) in the Application Sets box and clicking the —> button to move it to the Matched box.
11. Next to Policy Action, select **Permit**.



**NOTE:** When you select Permit for Policy Action, several additional fields become available in the Applications Services tab, including UTM Policy.

12. Select the **Application Services** tab.
13. Next to **UTM Policy**, select the appropriate policy from the list. This action attaches your UTM policy to the security policy.
14. Click **OK**.
15. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.
16. If the policy is saved successfully, you receive a confirmation and you must click **OK** again. If the profile is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

You must activate your new policy to apply it.

## Example: Configuring Express Antivirus (CLI)

---

- Example: Configuring Express Antivirus Custom Objects on page 941
- Example: Configuring Express Antivirus Feature Profiles on page 943
- Example: Configuring Express Antivirus UTM Policies on page 948
- Example: Attaching Express Antivirus UTM Policies to Security Policies on page 949

## Example: Configuring Express Antivirus Custom Objects

This example shows how to configure express antivirus custom objects.

- Requirements on page 941
- Overview on page 941
- Configuration on page 942
- Verification on page 943

### Requirements

---

Before you begin:

- Decide the type of express antivirus protection you require. See “Express Antivirus Protection Overview” on page 929.
- Understand the order in which express antivirus parameters are configured. See “Express Antivirus Configuration Overview” on page 934.

### Overview

---

In this example, you define custom objects that are used to create express antivirus feature profiles. You perform the following tasks to define custom objects:

- Create two MIME lists called avmime2 and ex-avmime2, and add patterns to the list.
- Configure a URL pattern list called urllist2.

When entering the URL pattern, note the following wildcard character support:

- The `\*\.[\]\?*` wildcard characters are supported.
- You must precede all wildcard URLs with `http://`.
- You can use the asterisk `*` wildcard character only if it is at the beginning of the URL and is followed by a period.
- You can use the question mark `?` wildcard character only at the end of the URL.
- The following wildcard syntax is supported: `http://*juniper.net`, `http://www.juniper.ne?`, `http://www.juniper.n??`.
- The following wildcard syntax is not supported: `*juniper.net`, `www.juniper.ne?`, `http://*juniper.net`, `http://*`.

- Configure a custom URL category list called `custurl2`, using the `urllist2` URL pattern list.

### Configuration

#### CLI Quick Configuration

To quickly configure express antivirus custom objects, copy the following commands and paste them into the CLI.

```
[edit]
set security utm custom-objects mime-pattern avmime2 value [video/quicktime
image/x-portable-anymap x-world/x-vrml]
set security utm custom-objects mime-pattern ex-avmime2 value
[video/quicktime-inappropriate]
set security utm custom-objects url-pattern urllist2 value [http://www.juniper.net 1.2.3.4]
set security utm custom-objects custom-url-category custurl2 value urllist2
```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure express antivirus filtering custom objects:

1. Create MIME lists, and add MIME patterns to the lists.

```
[edit security utm]
user@host# set custom-objects mime-pattern avmime2 value [video/quicktime
image/x-portable-anymap x-world/x-vrml]
user@host# set custom-objects mime-pattern ex-avmime2 value
[video/quicktime-inappropriate]
```



**NOTE:** Because you use URL pattern lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure custom URL category list.

2. Configure a URL pattern list custom object.

```
[edit security utm]
user@host# set custom-objects url-pattern urllist2 value [http://www.juniper.net
1.2.3.4]
```

3. Configure a custom URL category list.

```
[edit security utm]
user@host# set custom-objects custom-url-category custurl2 value urllist2
```

#### Results

From configuration mode, confirm your configuration by entering the `show security utm` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security utm
custom-objects {
  mime-pattern {
```

```

avmime2 {
  value [ video/quicktime image/x-portable-anymap x-world/x-vrml ];
}
ex-avmime2 {
  value video/quicktime-inappropriate;
}
}
url-pattern {
  urllist2 {
    value [ http://www.juniper.net 1.2.3.4 ];
  }
}
custom-url-category {
  custurl2 {
    value urllist2;
  }
}
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Express Antivirus Custom Objects on page 943

#### *Verifying Express Antivirus Custom Objects*

**Purpose** Verify the express antivirus custom objects.

**Action** From operational mode, enter the **show configuration security utm** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Configuring Express Antivirus Custom Objects \(J-Web Procedure\) on page 935](#)
  - [Attaching Express Antivirus UTM Policies to Security Policies \(J-Web Procedure\) on page 940](#)
  - [Configuring Express Antivirus \(J-Web Procedure\) on page 935](#)
  - [Configuring Express Antivirus Feature Profiles \(J-Web Procedure\) on page 937](#)

### Example: Configuring Express Antivirus Feature Profiles

This example shows how to configure an express antivirus feature profile.

- Requirements on page 944
- Overview on page 944
- Configuration on page 945
- Verification on page 948

## Requirements

Before you begin:

- Decide the type of express antivirus protection you require. See “Express Antivirus Protection Overview” on page 929.
- Understand the order in which express antivirus parameters are configured. See “Express Antivirus Configuration Overview” on page 934.
- MIME patterns must be defined for lists and exception lists. See “Example: Configuring MIME Whitelists to Bypass Antivirus Scanning” on page 890.
- Custom objects must be defined. See “Example: Configuring Express Antivirus Custom Objects” on page 941
- SMTP must be configured on the device. See “Understanding SMTP Antivirus Scanning” on page 893

## Overview

In this example, you configure a feature profile called `junexprof1` and specify custom objects to be used for filtering content.

- Select and configure the Juniper Express Engine as the engine type.
- Select 120 as the time interval for updating the pattern database. The default antivirus pattern-update interval is once a day.



**NOTE:** The command for changing the URL for the pattern database is:

```
[edit]
user@host# set security utm feature-profile anti-virus juniper-express-engine
pattern-update url http://...
```

Under most circumstances, you should not need to change the default URL.

- Enable an e-mail notification with a custom message as pattern file was updated and a custom subject line as AV pattern file updated.
- Configure a list of fallback options as block.
- Configure the notification options for fallback blocking for virus detection. Configure a custom message for the fallback blocking action, and send a notification.
- Configure a notification for protocol-only virus detection, and send a notification as Antivirus Alert.
- Configure content size parameters as 20000.



**NOTE:** For SRX100, SRX110, SRX210, SRX220, and SRX240 devices the maximum value for content size is 20000. For SRX650 device the maximum value for content size is 40000.

- Enable intelligent prescreening and set its timeout setting to 1800 seconds and trickling setting (applicable only to HTTP) to 600 seconds. This means that if the device receives a packet within a 600-second period during a file transfer or while performing an antivirus scan, it should not time out



**NOTE:** Intelligent prescreening is intended only for use with non-encoded traffic. It is not applicable to mail protocols (SMTP, POP3, IMAP) or HTTP POST.

- Configure the antivirus scanner to use MIME bypass lists and exception lists. You can use your own custom object lists, or you can use the default list, called `junos-default-bypass-mime`, which ships with the device. The following example enables the `avmime2` and `ex-avmime2` lists.
- Configure the antivirus module to use URL bypass lists. If you are using a URL whitelist (valid only for HTTP traffic), this is a custom URL category that you previously configured as a custom object. For this example, you enable the `custurl1` bypass list.

### Configuration

#### CLI Quick Configuration

To quickly configure an antivirus feature profile, copy the following commands and paste them into the CLI.

```
[edit]
set security utm feature-profile anti-virus juniper-express-engine pattern-update interval
  120
set security utm feature-profile anti-virus juniper-express-engine pattern-update
  email-notify admin-email administrator@juniper.net custom-message "pattern file
  was updated" custom-message-subject "AV pattern file updated"
set security utm feature-profile anti-virus juniper-express-engine profile junexprof1
  fallback-options content-size block
set security utm feature-profile anti-virus juniper-express-engine profile junexprof1
  fallback-options default block
set security utm feature-profile anti-virus juniper-express-engine profile junexprof1
  fallback-options engine-not-ready block
set security utm feature-profile anti-virus juniper-express-engine profile junexprof1
  fallback-options out-of-resources block
set security utm feature-profile anti-virus juniper-express-engine profile junexprof1
  fallback-options timeout block
set security utm feature-profile anti-virus juniper-express-engine profile junexprof1
  fallback-options too-many-requests block
set security utm feature-profile anti-virus juniper-express-engine profile junexprof1
  notification-options fallback-block custom-message "Dropped due to fallback condition"
set security utm feature-profile anti-virus juniper-express-engine profile junexprof1
  notification-options virus-detection type protocol-only
set security utm feature-profile anti-virus juniper-express-engine profile junexprof1
  notification-options virus-detection custom-message ***virus-found***
set security utm feature-profile anti-virus juniper-express-engine profile junexprof1
  scan-options content-size-limit 20000
set security utm feature-profile anti-virus juniper-express-engine profile junexprof1
  scan-options intelligent-prescreening
set security utm feature-profile anti-virus juniper-express-engine profile junexprof1
  scan-options timeout 1800
```

```

set security utm feature-profile anti-virus juniper-express-engine profile junexprof1 trickling
  timeout 600
set security utm feature-profile anti-virus mime-whitelist list avmime2
set security utm feature-profile anti-virus mime-whitelist list avmime2 exception
  ex-avmime2
set security utm feature-profile anti-virus url-whitelist custurl2

```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure express antivirus feature profiles:

1. Select and configure the engine type.

```

[edit]
user@host# set security utm feature-profile anti-virus type juniper-express-engine

```

2. Select a time interval for updating the pattern database.

```

[edit security utm feature-profile anti-virus juniper-express-engine]
user@host# set pattern-update interval 120

```

3. Configure the device to notify a specified administrator when patterns are updated.

```

[edit security utm feature-profile anti-virus juniper-express-engine]
user@host# set pattern-update email-notify admin-email administrator@juniper.net
  custom-message "pattern file was updated" custom-message-subject "AV pattern
  file updated"

```

4. Create a profile for the Juniper Express Engine, and configure fallback options as block.

```

[edit security utm feature-profile anti-virus juniper-express-engine]
user@host# set profile junexprof1 fallback-options content-size block
user@host# set profile junexprof1 fallback-options default block
user@host# set profile junexprof1 fallback-options engine-not-ready block
user@host# set profile junexprof1 fallback-options out-of-resources block
user@host# set profile junexprof1 fallback-options timeout block
user@host# set profile junexprof1 fallback-options too-many-requests block

```

5. Configure a custom notification for the fallback blocking action, and send a notification.

```

[edit security utm feature-profile anti-virus juniper-express-engine]
user@host# set profile junexprof1 notification-options fallback-block
  custom-message "Dropped due to fallback condition"

```

6. Configure a notification for protocol-only virus detection, and send a notification.

```

[edit security utm feature-profile anti-virus juniper-express-engine]
user@host# set profile junexprof1 notification-options virus-detection type
  protocol-only

```

7. Configure a custom notification for virus detection.

```

[edit security utm feature-profile anti-virus juniper-express-engine]
set profile junexprof1 notification-options virus-detection custom-message
  ***virus-found***

```



8. Configure content size parameter.
 

```
[edit security utm feature-profile anti-virus juniper-express-engine]
user@host# set profile junexprof1 scan-options content-size-limit 20000
```
9. Configure intelligent prescreening.
 

```
[edit security utm feature-profile anti-virus juniper-express-engine]
user@host# set profile junexprof1 scan-options intelligent-prescreening
```
10. Configure the timeout setting.
 

```
[edit security utm feature-profile anti-virus juniper-express-engine]
user@host# set profile junexprof1 scan-options timeout 1800
```
11. Configure trickling setting.
 

```
[edit security utm feature-profile anti-virus juniper-express-engine]
user@host# set profile junexprof1 trickling timeout 600
```
12. Configure the antivirus scanner to use MIME bypass lists and exception lists.
 

```
[edit security utm feature-profile anti-virus]
user@host# set mime-whitelist list avmime2
user@host# set mime-whitelist list avmime2 exception ex-avmime2
```
13. Configure the antivirus module to use URL bypass lists.
 

```
[edit security utm feature-profile anti-virus]
user@host# set url-whitelist custurl2
```

**Results** From configuration mode, confirm your configuration by entering the **show security utm feature-profile anti-virus** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security utm feature-profile anti-virus
mime-whitelist {
  list avmime2;
  exception ex-avmime2;
}
url-whitelist custurl2;
juniper-express-engine {
  pattern-update {
    email-notify {
      admin-email "administrator@juniper.net";
      custom-message "pattern file was updated";
      custom-message-subject "AV pattern file updated";
    }
    interval 120;
  }
}
profile junexprof1 {
  fallback-options {
    default block;
    content-size block;
    engine-not-ready block;
    timeout block;
    out-of-resources block;
    too-many-requests block;
  }
}
```

```

    }
    scan-options {
      intelligent-prescreening;
      content-size-limit 20000;
      timeout 1800;
    }
    trickling timeout 600;
    notification-options {
      virus-detection {
        type protocol-only;
        custom-message ***virus-found***;
      }
      fallback-block {
        custom-message "Dropped due to fallback condition";
      }
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task.

- Verifying the Configuration of Express Antivirus Feature Profile on page 948

#### *Verifying the Configuration of Express Antivirus Feature Profile*

**Purpose** Verify the express antivirus feature profile.

**Action** From operational mode, enter any of the following commands:

- **show configuration security utm**
- **show security utm anti-virus status**
- **show security utm anti-virus statistics**

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Configuring Express Antivirus UTM Policies \(J-Web Procedure\) on page 939](#)
  - [Example: Configuring Express Antivirus UTM Policies on page 948](#)
  - [Example: Attaching Express Antivirus UTM Policies to Security Policies on page 949](#)

### Example: Configuring Express Antivirus UTM Policies

This example shows how to create an express antivirus UTM policy to attach to your feature profile.

- [Requirements on page 949](#)
- [Overview on page 949](#)

- Configuration on page 949
- Verification on page 949

### Requirements

---

Before you begin, create an antivirus feature profile. See “Example: Configuring Express Antivirus Feature Profiles” on page 943.

### Overview

---

In this example, you configure an express antivirus UTM policy called `utmp3` and attach the policy to the antivirus profile called `junexprof1`.

### Configuration

---

#### Step-by-Step Procedure

To configure an express antivirus UTM policy:

1. Create a UTM policy for HTTP antivirus scanning and attach the policy to the profile.

```
[edit]
user@host# set security utm utm-policy utmp3 anti-virus http-profile junexprof1
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

---

To verify the configuration is working properly, enter the **show security utm** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Express Antivirus Protection Overview on page 929
- Express Antivirus Configuration Overview on page 934
- Example: Attaching Express Antivirus UTM Policies to Security Policies on page 949

## Example: Attaching Express Antivirus UTM Policies to Security Policies

This example shows how to attach an express antivirus UTM policy to a security policy.

- Requirements on page 949
- Overview on page 949
- Configuration on page 950
- Verification on page 950

### Requirements

---

Before you begin, create a UTM policy. See “Example: Configuring Express Antivirus UTM Policies” on page 948.

### Overview

---

In this example, you attach the express antivirus UTM policy called `utmp3` to the security policy called `p3`.

## Configuration

---

### Step-by-Step Procedure

To attach an express antivirus UTM policy to a security policy:

1. Enable and configure the security policy.

```
[edit]
user@host# set security policies from-zone trust to-zone untrust policy p3 match
source-address any
user@host# set security policies from-zone trust to-zone untrust policy p3 match
destination-address any
user@host# set security policies from-zone trust to-zone untrust policy p3 match
application junos-http
```

2. Attach the UTM policy to the security policy.

```
[edit]
user@host# set security policies from-zone trust to-zone untrust policy p3 then
permit application-services utm-policy utmp3
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

## Verification

---

To verify the configuration is working properly, enter **show security policies detail** from operational mode.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Express Antivirus Protection Overview on page 929
- Express Antivirus Configuration Overview on page 934
- Example: Configuring Express Antivirus Feature Profiles on page 943
- Attaching Express Antivirus UTM Policies to Security Policies (J-Web Procedure) on page 940

# Sophos Antivirus Protection

This chapter includes the following topics:

- Sophos Antivirus Protection Overview on page 951
- Sophos Antivirus Features on page 952
- Comparison of Sophos Antivirus to Kaspersky Antivirus on page 953
- Understanding Sophos Antivirus Data File Update on page 953
- Managing Sophos Antivirus Data Files on page 954
- Sophos Antivirus Configuration Overview on page 955
- Example: Configuring Sophos Antivirus Custom Objects on page 955
- Example: Configuring Sophos Antivirus Feature Profile on page 959
- Example: Configuring Sophos Antivirus UTM Policies on page 965
- Example: Configuring Sophos Antivirus Firewall Security Policies on page 966

## Sophos Antivirus Protection Overview

---

Sophos antivirus scanning is offered as a less CPU-intensive alternative to the full file-based antivirus feature. Sophos supports the same protocols as full antivirus and functions in much the same manner; however, it has a smaller memory footprint and is compatible with lower end devices that have less memory.

Sophos antivirus is as an in-the-cloud antivirus solution. The virus pattern and malware database is located on external servers maintained by Sophos (Sophos Extensible List) servers, thus there is no need to download and maintain large pattern databases on the Juniper device. The Sophos antivirus scanner also uses a local internal cache to maintain query responses from the external list server to improve lookup performance.

Because a significant amount of traffic processed by Juniper Unified Threat Management (UTM) is HTTP based, Uniform Resource Identifier (URI) checking is used to effectively prevent malicious content from reaching the endpoint client or server. The following checks are performed for HTTP traffic: URI lookup, true file type detection, and file checksum lookup. The following application layer protocols are supported: HTTP, FTP, SMTP, POP3 and IMAP.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

- [Sophos Antivirus Features](#) on page 952
- [Sophos Antivirus Configuration Overview](#) on page 955

## Sophos Antivirus Features

Sophos Antivirus has the following main features:

- **Sophos Antivirus Expanded MIME Decoding Support**—Sophos antivirus offers decoding support for HTTP, POP3, SMTP, and IMAP. MIME decoding support includes the following for each supported protocol:
  - Multipart and nested header decoding
  - Base64 decoding, printed quote decoding, and encoded word decoding in the subject field
- **Sophos Antivirus Scan Result Handling**—With Sophos antivirus, the TCP traffic is closed gracefully when a virus is found and the data content is dropped.

The following fail mode options are supported: content-size, default, engine-not-ready, out-of-resource, timeout, and too-many-requests. You can set the following actions: block, log-and-permit, and permit. Fail mode handling of supported options with Sophos is much the same as with full antivirus.

- **Sophos Uniform Resource Identifier Checking**—Sophos provides Uniform Resource Identifier (URI) checking, which is similar to anti-spam realtime blackhole list (RBL) lookups. URI checking is a way of analyzing URI content in HTTP traffic against the Sophos database to identify malware or malicious content. Because malware is predominantly static, a checksum mechanism is used to identify malware to improve performance. Files that are capable of using a checksum include: .exe, .zip, .rar, .swf, .pdf, and .ole2 (doc and xls).



**NOTE:** If you have a Juniper device protecting an internal network that has no HTTP traffic, or has Web servers that are not accessible to the outside world, you may want to turn off URI checking. If the Web servers are not accessible to the outside world, it is unlikely that they contain URI information that is in the Sophos URI database. URI checking is on by default.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Sophos Antivirus Protection Overview](#) on page 951
- [Sophos Antivirus Configuration Overview](#) on page 955
- [Example: Configuring Sophos Antivirus Feature Profile](#) on page 959

## Comparison of Sophos Antivirus to Kaspersky Antivirus

Sophos Antivirus is much like Juniper Express Antivirus and also has similarities to the Full Antivirus feature:

- Unlike the Juniper Express and Full Antivirus solutions, the antivirus and malware database for Sophos is stored on a group of remote Sophos Extensible List servers. Queries are performed using the DNS protocol. Sophos maintains these servers, so there is no need to download and maintain large pattern databases on the Juniper device. Because the database is remote, there is no size limitation and there is a quicker response to new virus outbreaks.



**NOTE:** Sophos antivirus uses a set of data files that need to be updated on a regular basis. These are not typical virus pattern files; they are a set of small files that help guide virus scanning logic. You can manually download the data files or set up automatic download.

- Sophos does not provide the same prescreening detection as Kaspersky Antivirus. Sophos does provide a similar solution that is part of the Sophos engine and cannot be turned on and off.
- The Sophos antivirus scanning feature is a separately licensed subscription service. Also, the pattern lookup database is located on remote servers maintained by Sophos, so when your antivirus license key expires, functionality will no longer work. You have a 30-day grace period in which to update your license.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Sophos Antivirus Protection Overview on page 951](#)
- [Sophos Antivirus Configuration Overview on page 955](#)

## Understanding Sophos Antivirus Data File Update

Sophos antivirus uses a small set of data files that need to be updated periodically. These data files only contain information on guiding scanning logic and do not contain the full pattern database. The main pattern database, which includes protection against critical viruses, URI checks, malware, worms, Trojans, and spyware, is located on remote Sophos Extensible List servers maintained by Sophos.

The Sophos data files are updated over HTTP or HTTPS and can be updated manually or scheduled to update automatically. This is similar functionality to that found in the Full Antivirus solution, but with some minor differences:

- With Sophos antivirus, the signature database auto-update interval is once a day by default. This interval can be changed.
- With Sophos, there is no interruption in virus scanning capability during the data file update. If the update fails, the existing data files will continue to be used.

- By default, the URL for Sophos antivirus data file update is <http://update.juniper-updates.net/SAV/>.



**NOTE:** The Sophos antivirus scanning feature is a separately licensed subscription service. When your antivirus license key expires, functionality will no longer work because the pattern lookup database is located on remote Sophos servers. You have a 30-day grace period in which to update your license.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Sophos Antivirus Protection Overview on page 951](#)
- [Managing Sophos Antivirus Data Files on page 954](#)
- [Sophos Antivirus Configuration Overview on page 955](#)

## Managing Sophos Antivirus Data Files

Before you begin:

- Install a Sophos antivirus license. See the “Installing and Managing Licenses” information in the [Junos OS Administration Guide for Security Devices](#).
- Configure Sophos as the antivirus feature for the device. See “Example: Configuring Sophos Antivirus Feature Profile” on page 959. To set the antivirus engine type, you run the `set security utm feature-profile anti-virus type sophos-engine` statement.

In this example, you configure the security device to update the data files automatically every 4320 minutes (every 3 days). The default data file update interval is 1440 minutes (every 24 hours).

To automatically update Sophos data files:

```
[edit security utm feature-profile anti-virus]
user@host# set sophos-engine pattern-update interval 4320
```



**NOTE:** The following commands are performed from CLI operational mode.

To manually update data files:

```
user@host> request security utm anti-virus sophos-engine pattern-update
```

To manually reload data files:

```
user@host> request security utm anti-virus sophos-engine pattern-reload
```

To manually delete data files:

```
user@host> request security utm anti-virus sophos-engine pattern-delete
```

To check the status of antivirus, which also shows the data files version:



```
user@host> show security utm anti-virus status
```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Sophos Antivirus Protection Overview](#) on page 951
- [Understanding Sophos Antivirus Data File Update](#) on page 953
- [Sophos Antivirus Configuration Overview](#) on page 955

## Sophos Antivirus Configuration Overview

---

Sophos antivirus is part of the Unified Threat Management (UTM) feature set, so you first configure UTM options (custom objects), configure the Sophos Feature, then create a UTM policy and a security policy. The security policy controls all traffic that is forwarded by the device, and the UTM policy specifies which parameters to use to scan traffic. The UTM policy is also used to bind a set of protocols to one or more UTM feature profiles, including Sophos antivirus in this case.

You must complete the following tasks to configure Sophos antivirus:

1. Configure UTM custom objects and MIME lists. See “Example: Configuring Sophos Antivirus Custom Objects” on page 955,
2. Configure the Sophos antivirus feature profile. See “Example: Configuring Sophos Antivirus Feature Profile” on page 959.
3. Configure a UTM policy. See “Example: Configuring Sophos Antivirus UTM Policies” on page 965
4. Configure a security policy. See “Example: Configuring Sophos Antivirus Firewall Security Policies” on page 966.

## Example: Configuring Sophos Antivirus Custom Objects

---

This example shows you how to create UTM global custom objects to be used with Sophos antivirus.

- [Requirements](#) on page 955
- [Overview](#) on page 956
- [Configuration](#) on page 956
- [Verification](#) on page 958

### Requirements

Before you begin, read about UTM custom objects. See “Understanding UTM Custom Objects” on page 844.

## Overview

Configure MIME lists. This includes creating a MIME whitelist and a MIME exception list for antivirus scanning. In this example, you bypass scanning of QuickTime videos, unless if they contain the MIME type quicktime-inappropriate.



**WARNING:** When you configure the MIME whitelist feature, be aware that, because header information in HTTP traffic can be spoofed, you cannot always trust HTTP headers to be legitimate. When a Web browser is determining the appropriate action for a given file type, it detects the file type without checking the MIME header contents. However, the MIME whitelist feature does refer to the MIME encoding in the HTTP header. For these reasons, it is possible in certain cases for a malicious website to provide an invalid HTTP header. For example, a network administrator might inadvertently add a malicious website to a MIME whitelist, and, because the site is in the whitelist, it will not be blocked by Sophos even though Sophos has identified the site as malicious in its database. Internal hosts would then be able to reach this site and could become infected.

## Configuration

### J-Web Quick Configuration

To configure a MIME list:

1. Click the **Configure** tab from the taskbar, and then select **Security>UTM>Custom Objects**.
2. Click the **MIME Pattern List** tab and then click **Add**
3. In the MIME Pattern Name box, type **avmime2**.
4. In the MIME Pattern Value box, type **video/quicktime**, and click **Add**.
5. In the MIME Pattern Value box, type **image/x-portable-anympa**, and click **Add**.
6. In the MIME Pattern Value box, type **x-world/x-vrml**, and click **Add**.

To configure a MIME exception list:

1. Click the **Configure** tab from the taskbar, and then select **Security>UTM>Custom Objects**.
2. Click the **MIME Pattern List** tab and then select **Add**
3. In the MIME Pattern Name box, type **exception-avmime2**.
4. In the MIME Pattern Value box, type **video/quicktime-inappropriate** and click **Add**.

Configure a URL pattern list (whitelist) of URLs or addresses that will be bypassed by antivirus scanning. After you create the URL pattern list, you will create a custom URL category list and add the pattern list to it.



**NOTE:** Because you use URL pattern lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure custom URL category lists.

To configure a URL pattern whitelist:

1. Click the **Configure** tab from the taskbar, and then select **Security>UTM>Custom Objects**.
2. Click the **URL Pattern List** tab, and then click **Add**
3. In the URL Pattern Name box, enter **urlist2**.
4. In the URL Pattern Value box, enter **http://juniper.net**. (You can also use the IP address of the server instead of the URL.)

Save your configuration:

1. Click **OK** to check your configuration and save it as a candidate configuration.
2. If you are done configuring the device, click **Actions>Commit**.



**NOTE:** URL pattern wildcard support—The wildcard rule is as follows: `\*\.[]\?*` and you must precede all wildcard URLs with `http://`. You can use “\*” only if it is at the beginning of the URL and is followed by a “.”. You can only use “?” at the end of the URL.

The following wildcard syntax is supported: `http://*.juniper.net`, `http://www.juniper.ne?`, `http://www.juniper.n??`. The following wildcard syntax is not supported: `*juniper.net`, `www.juniper.ne?`, `http://*juniper.net`, `http://*`

#### Step-by-Step Procedure

To configure antivirus protection using the CLI, you must create your custom objects in the following order:

1. Create the MIME whitelist.

```
[edit security utm]
user@host# set custom-objects mime-pattern avmime2 value [video/quicktime
image/x-portable-anymap x-world/x-vrml]
```

Create the MIME exception list.

```
[edit security utm]
user@host# set custom-objects mime-pattern exception-avmime2 value
[video/quicktime-inappropriate]
```

- Configure a URL pattern list (whitelist) of URLs or addresses that you want to bypass. After you create the URL pattern list, you create a custom URL category list and add the pattern list to it. Configure a URL pattern list custom object by creating the list name and adding values to it as follows.



**NOTE:** Because you use URL pattern lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure custom URL category lists.

```
[edit security utm]
user@host# set custom-objects url-pattern urllist2 value [http://www.juniper.net
192.168.1.5]
```



**NOTE:** URL pattern wildcard support—The wildcard rule is as follows: `\*\.[]\?*` and you must precede all wildcard URLs with `http://`. You can only use “\*” if it is at the beginning of the URL and is followed by a “.”. You can only use “?” at the end of the URL.

The following wildcard syntax is supported: `http://*.juniper.net`, `http://www.juniper.ne?`, `http://www.juniper.n??`. The following wildcard syntax is not supported: `*.juniper.net`, `www.juniper.ne?`, `http://*juniper.net`, `http://*`.

- Configure a custom URL category list custom object by using the URL pattern list `urllist2` that you created earlier:

```
[edit security utm]
user@host# set custom-objects custom-url-category custurl2 value urllist2
```

## Verification

To verify the configuration, enter the `show security utm custom-objects` command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Sophos Antivirus Protection Overview on page 951](#)
- [Sophos Antivirus Configuration Overview on page 955](#)
- [Example: Configuring Sophos Antivirus Feature Profile on page 959](#)
- [Understanding UTM Custom Objects on page 844](#)

## Example: Configuring Sophos Antivirus Feature Profile

This example shows you how to configure a Sophos antivirus profile that defines the parameters that will be used for virus scanning.

- Requirements on page 959
- Overview on page 959
- Configuration on page 959
- Verification on page 964

### Requirements

Before you begin:

- Install a Sophos antivirus license. See the “Installing and Managing Licenses” information in the *Junos OS Administration Guide for Security Devices*.
- Configure custom objects for UTM. See “Example: Configuring Sophos Antivirus Custom Objects” on page 955.

### Overview

The following configuration defines Sophos as the antivirus engine and sets parameters, such as the data file update interval, notification options for administrators, fallback options, and file size limits.

### Configuration

#### J-Web Quick Configuration



**NOTE:** The following example shows you how to create a custom Sophos profile. If you want to use the Juniper Networks preconfigured profile, use the profile named `junos-sophos-av-defaults` in your UTM policy. See “Example: Configuring Sophos Antivirus UTM Policies” on page 965.

1. Select and configure the engine type. Because you are configuring Sophos antivirus, you configure `sophos-engine`:
  - a. Click the **Configure** tab from the taskbar, and then select **Security>UTM>Anti-Virus**.
  - b. Click the **Global Options** tab and then click **Sophos**.

- c. Click **OK** and commit your changes.
- d. Restart the device to enable Sophos as the antivirus engine.
2. Return to the antivirus Global Options screen as you did in step 1, and set the following parameters:
  - a. In the MIME whitelist list, select **exception-avmime2**.
  - b. In the URL whitelist list, select **custurl2**.
  - c. In the Pattern update interval (sec) box, type **2880**.
  - d. In the box, type the e-mail address that will receive SophosAdmin email data file update notifications. For example - **admin@juniper.net**.
  - e. In the Custom Message box, type **The Sophos data file update on the SRX240 has been completed**. In the Custom message subject box, type **Sophos Data File Updated**.
  - f. Click **OK** to check your configuration and save it as a candidate configuration.
3. Configure a profile for the sophos-engine and set parameters.
  - a. Click the **Configure** tab from the taskbar and then select **Security>UTM>Anti-Virus**. Click **Add**.
  - b. In the Add profile box, click the **Main** tab.
  - c. In the Profile name box, type **sophos-profl**.
  - d. In the Trickling timeout box, type **180**.



.....

**WARNING:** When enabling the trickling option, it's important to understand that trickling may send part of the file to the client during the antivirus scan. It is possible that some of the content could be received by the client and the client may become infected before the file is fully scanned.

.....

- e. URI checking is on by default. To turn it off, clear **yes** in the URI check box.
- f. In the Content size Limit box, type **20000**.
- g. In the Scan engine timeout box, type **1800**.
4. Configure fallback settings by clicking the **Fallback settings** tab. In this example, all fallback options are set to log and permit. Click **Log and permit** for the following items: Default action, Content size, Engine not ready, Timeout, Out of resource, Too many requests.

5. Configure notification options by clicking the **Notification options** tab. You can configure notifications for both fallback blocking and fallback nonblocking actions and for virus detection.

To configure notifications for Fallback settings:

- a. For Notification type, click **Protocol**.
  - b. For Notify mail sender, click **yes**.
  - c. In the Custom message box, type **Fallback block action occurred**.
  - d. In the Custom message subject box, type **\*\*\*Antivirus fallback Alert\*\*\***.
6. To configure notification options for virus detection, click the **Notification options cont...** tab.
    - a. For the Notification type option button, select **Protocol**.
    - b. For the Notify mail sender option button, select **yes**.
    - c. In the Custom message box, type **Virus has been detected**.
    - d. In the Custom message subject box, type **\*\*\*Virus detected\*\*\***.
  7. Click **OK** to check your configuration and save it as a candidate configuration.
  8. If you are done configuring the device, click **Actions>Commit**.

### Step-by-Step Procedure

To configure the Sophos antivirus feature profile using the CLI:



**NOTE:** The following example shows you how to create a custom Sophos profile. If you want to use the Juniper Networks preconfigured profile, use the profile named `junos-sophos-av-defaults` in your UTM policy. See “Example: Configuring Sophos Antivirus UTM Policies” on page 965.

1. Select and configure the engine type. Because you are configuring Sophos antivirus, you configure `sophos-engine`.
 

```
[edit]
user@host# set security utm feature-profile anti-virus type sophos-engine
```
2. Commit the configuration, and restart the device. After the device restarts, enter configuration mode again.
3. Select a time interval for updating the data files. The default antivirus pattern-update interval is 1440 minutes (every 24 hours). You can choose to leave this default, or you can change it. You can also force a manual update, if needed. To change the default from every 24 hours to every 48 hours:

```
[edit security utm feature-profile anti-virus]
user@host# set sophos-engine pattern-update interval 2880
```

4. In most circumstances, you will not need to change the URL to update the pattern database. If you do need to change this option, use the following command:

```
[edit security utm feature-profile anti-virus]
user@host# set sophos-engine pattern-update url
http://www.juniper.net/test-download
```

5. You can configure the device to notify a specified administrator when data files are updated. This is an e-mail notification with a custom message and a custom subject line.

```
[edit security utm feature-profile anti-virus]
user@host# set sophos-engine pattern-update email-notify admin-email
admin@juniper.net custom-message "Sophos antivirus data file was updated"
custom-message-subject "AV data file updated"
```

6. Configure a list of fallback options as block, log and permit, or permit. The default setting is log-and-permit. You can use the default settings, or you can change them.

Configure the content size action. In this example, if the content size is exceeded, the action taken is block.

First create the profile named sophos-profl.

```
[edit security utm feature-profile anti-virus]
user@host# edit sophos-engine profile sophos-profl
```

Configure the content size fallback-option to block.

```
[edit security utm feature-profile anti-virus sophos-engine profile sophos-profl]
user@host# set fallback-options content-size block
```

Configure the default fallback option to log-and-permit.

```
[edit security utm feature-profile anti-virus sophos-engine profile sophos-profl]
user@host# set fallback-options default log-and-permit
```

Configure log-and-permit if the antivirus engine is not ready.

```
[edit security utm feature-profile anti-virus sophos-engine profile sophos-profl]
user@host# set fallback-options engine-not-ready log-and-permit
```

Configure log-and-permit if the device is out of resources.

```
[edit security utm feature-profile anti-virus sophos-engine profile sophos-profl]
user@host# set fallback-options out-of-resources log-and-permit
```

Configure log-and-permit if a virus scan timeout occurs.

```
[edit security utm feature-profile anti-virus sophos-engine profile sophos-profl]
user@host# set fallback-options timeout log-and-permit
```

Configure log-and-permit if there are too many requests for the virus engine to handle.

```
[edit security utm feature-profile anti-virus sophos-engine profile sophos-profl]
user@host# set fallback-options too-many-requests log-and-permit
```

7. Configure notification options. You can configure notifications for fallback blocking, fallback nonblocking actions, and virus detection.



In this step, configure a custom message for the fallback blocking action and send a notification for protocol-only actions to the administrator and the sender.

```
[edit security utm feature-profile anti-virus sophos-engine profile sophos-prof1]
user@host# set notification-options fallback-block custom-message ***Fallback
block action occurred*** custom-message-subject Antivirus Fallback Alert
notify-mail-sender type protocol-only allow email administrator-email
admin@juniper.net
```

8. Configure a notification for protocol-only virus detection, and send a notification.

```
[edit security utm feature-profile anti-virus sophos-engine profile sophos-prof1]
user@host#set notification-options virus-detection type protocol-only
notify-mail-sender custom-message-subject ***Virus detected***
custom-message Virus has been detected
```

9. Configure content size parameters.



**NOTE:** When you configure the content-size value, keep in mind that in certain cases, content size is available in the protocol headers, so the max-content-size fallback is applied before a scan request is sent. However, in many cases, content size is not provided in the protocol headers. In these cases, the TCP payload is sent to the antivirus scanner and accumulates until the end of the payload. If the accumulated payload exceeds the maximum content size value, then max-content-size fallback is applied. The default fallback action is log and permit, so you may want to change this option to block, in which case such a packet is dropped and a block message is sent to the client.

In this example, if the content size exceeds 20 MB, the packet is dropped.

```
[edit security utm feature-profile anti-virus sophos-engine profile sophos-prof1]
user@host# set scan-options content-size-limit 20000
```

10. URI checking is on by default. To turn off URI checking:

```
[edit security utm feature-profile anti-virus sophos-engine profile sophos-prof1]
user@host# set scan-options no-uri-check
```

11. Configure the timeout setting for the scanning operation to 1800 seconds.

```
[edit security utm feature-profile anti-virus sophos-engine profile sophos-prof1]
user@host# set scan-options timeout 1800
```

12. The Sophos Extensible List servers contain the virus and malware database for scanning operations. Set the response timeout for these servers to 3 seconds (the default is 2 seconds).

```
[edit security utm feature-profile anti-virus sophos-engine profile sophos-prof1]
user@host# set scan-options sxl-timeout 3
```

13. Configure the Sophos Extensible List server retry option to 2 retries (the default is 1).

```
[edit security utm feature-profile anti-virus sophos-engine profile sophos-prof1]
user@host# set scan-options sxl-retry 2
```

14. Configure the trickling setting to 180 seconds. If you use trickling, you can also set timeout parameters. Trickling applies only to HTTP. HTTP trickling is a mechanism used to prevent the HTTP client or server from timing out during a file transfer or during antivirus scanning.



**WARNING:** When you enable the trickling option, keep in mind that trickling might send part of a file to the client during its antivirus scan. It is therefore possible that some of the content could be received by the client before the file has been fully scanned.

```
[edit security utm feature-profile anti-virus]
user@host# set sophos-engine profile sophos-profl trickling timeout 180
```

15. Configure the antivirus module to use MIME bypass lists and exception lists. You can use your own custom object lists, or you can use the default list that ships with the device called junos-default-bypass-mime. In this example, you use the lists that you setup earlier.

```
[edit security utm feature-profile anti-virus]
user@host# set mime-whitelist list avmime2
[edit security utm feature-profile anti-virus]
user@host# set mime-whitelist list exception-avmime2
```

16. Configure the antivirus module to use URL bypass lists. If you are using a URL whitelist, this is a custom URL category you have previously configured as a custom object. URL whitelists are valid only for HTTP traffic. In this example you use the lists that you setup earlier.

```
[edit security utm feature-profile anti-virus]
user@host# set url-whitelist custurl2
```

## Verification

To verify your feature profile configuration, run the **show security utm feature-profile anti-virus** command.

### Obtaining Information About the Current Antivirus Status

**Action** From operational mode, enter the **show security utm anti-virus status** command to view the antivirus status.

```
user@host>show security utm anti-virus status
```

- Meaning**
- Antivirus key expire date—The license key expiration date.
  - Update server—URL for the data file update server.
    - Interval—The time period, in minutes, when the device will update the data file from the update server.
    - Pattern update status—When the data file will be updated next, displayed in minutes.

- Last result—Result of the last update. If you already have the latest version, this will display **already have latest database**.
- Antivirus signature version—Version of the current data file.
- Scan engine type—The antivirus engine type that is currently running.
- Scan engine information—Result of the last action that occurred with the current scan engine.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Sophos Antivirus Protection Overview on page 951
- Sophos Antivirus Configuration Overview on page 955

## Example: Configuring Sophos Antivirus UTM Policies

This example shows how to create a UTM policy for Sophos antivirus.

- Requirements on page 965
- Overview on page 965
- Configuration on page 965
- Verification on page 966

### Requirements

Before you create the UTM policy, create custom objects and the Sophos feature profile.

1. Configure UTM custom objects and MIME lists. See “Example: Configuring Sophos Antivirus Custom Objects” on page 955.
2. Configure the Sophos antivirus feature profile. See “Example: Configuring Sophos Antivirus Feature Profile” on page 959.

### Overview

After you have created an antivirus feature profile, you configure a UTM policy for an antivirus scanning protocol and attach this policy to a feature profile. In this example, HTTP will be scanned for viruses, as indicated by the **http-profile** statement. You can scan other protocols as well by creating different profiles or adding other protocols to the profile, such as: **imap-profile**, **pop3-profile**, and **smtp-profile**.

### Configuration

#### J-Web Quick Configuration

To configure a UTM policy for Sophos antivirus:

1. Click the **Configure** tab from the taskbar, and then select **Security>Policy>UTM Policies**. Then click **Add**.
2. Click the **Main** tab. In the Policy name box, type **utmp3**.
3. Click the **Anti-Virus profiles** tab. In the HTTP profile list, select **sophos-profl**.

4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, select **Actions>Commit**.

### Step-by-Step Procedure

To configure a UTM policy for Sophos antivirus:

1. Go to the edit security utm hierarchy.
 

```
[edit]
user@host# edit security utm
```
2. Create the UTM policy utmp3 and attach it to the http-profile sophos-profl.
 

```
[edit security utm]
user@host# set utm-policy utmp3 anti-virus http-profile sophos-profl
```



**NOTE:** You can use the default Sophos feature profile settings by replacing `sophos-profl` in the above statement with `junos-sophos-av-defaults`.

## Verification

To verify the configuration, enter the `show security utm utm-policy utmp3` command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Sophos Antivirus Protection Overview on page 951](#)
- [Sophos Antivirus Configuration Overview on page 955](#)
- [Example: Configuring Sophos Antivirus Feature Profile on page 959](#)

## Example: Configuring Sophos Antivirus Firewall Security Policies

This example shows how to create a security policy for Sophos antivirus.

- [Requirements on page 966](#)
- [Overview on page 967](#)
- [Configuration on page 967](#)
- [Verification on page 968](#)

## Requirements

Before you create the security policy, create custom objects, the Sophos feature profile, and the UTM policy.

1. Configure UTM custom objects and MIME lists. See “Example: Configuring Sophos Antivirus Custom Objects” on page 955.
2. Configure the Sophos antivirus feature profile. See “Example: Configuring Sophos Antivirus Feature Profile” on page 959.

3. Configure a UTM policy. See “Example: Configuring Sophos Antivirus UTM Policies” on page 965.

## Overview

Create a firewall security policy that will cause traffic from the untrust zone to the trust zone to be scanned by Sophos antivirus using the feature profile settings defined in “Example: Configuring Sophos Antivirus Feature Profile” on page 959. Because the match application configuration is set to any, all application types will be scanned.

## Configuration

### J-Web Quick Configuration

To configure a security policy for Sophos antivirus:

1. Configure the untrust to trust policy to match any source address or destination address, and select the applications to be scanned to **any**.
  - a. Click the **Configure** tab from the taskbar, and then select **Security>Policy>FW Policies**. Then select **Add**.
  - b. In the Policy Name box, type **p3**.
  - c. In the Policy Action box, select **permit**.
  - d. In the From Zone list, select **untrust**.
  - e. In the To Zone list, select **trust**.
  - f. In the Source Address and Destination Address boxes, make sure that Matched is set to **any**.
  - g. In the Applications boxes, select **any** from the Application/Sets list, and move it to the Matched list.
2. Attach the UTM policy named **utmp3** to the firewall security policy. This will cause matched traffic to be scanned by the Sophos antivirus feature.
  - a. From the Edit Policy box, click the **Application Services** tab.
  - b. In the UTM Policy list, select **utmp3**.
3. Click **OK** to check your configuration and save it as a candidate configuration.
4. If you are done configuring the device, select **Actions>Commit**.

### Step-by-Step Procedure

To configure a security policy for Sophos antivirus:

1. Configure the untrust to trust policy to match any source-address.
 

```
[edit security]
user@host# set policies from-zone untrust to-zone trust policy p3 match
source-address any
```
2. Configure the untrust to trust policy to match any destination-address.
 

```
[edit security]
```

```
user@host# set policies from-zone untrust to-zone trust policy p3 match
destination-address any
```

3. Configure the untrust to trust policy to match any application type.

```
[edit security]
```

```
user@host# set policies from-zone untrust to-zone trust policy p3 match application
any
```

4. Attach the UTM policy named utmp3 to the firewall security policy. This will cause matched traffic to be scanned by the Sophos antivirus feature.

```
[edit security]
```

```
user@host# set policies from-zone untrust to-zone trust policy p3 then permit
application-services utm-policy utmp3
```

## Verification

To verify the configuration, enter the **show security policies** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Sophos Antivirus Protection Overview on page 951](#)
- [Sophos Antivirus Configuration Overview on page 955](#)
- [Example: Configuring Sophos Antivirus Feature Profile on page 959](#)

# Content Filtering

- Content Filtering Overview on page 969
- Content Filtering Protocol Support on page 970
- Example: Configuring Content Filtering on page 972
- Monitoring Content Filtering Configurations on page 982

## Content Filtering Overview

---

Content filtering blocks or permits certain types of traffic based on the MIME type, file extension, and protocol command. The content filter controls file transfers across the gateway by checking traffic against configured filter lists.

The content filter module evaluates traffic before all other UTM modules, except Web Filtering. Therefore, if traffic meets criteria configured in the content-filter, the content-filter acts first upon this traffic.

You can configure the following types of content filters:

- MIME Pattern Filter — MIME patterns are used to identify the type of traffic in HTTP and MAIL protocols. There are two lists of MIME patterns that are used by the content filter to determine the action to be taken. The block MIME list contains a list of MIME type traffic that is to be blocked by the content filter. The MIME exception list contains MIME patterns that are not to be blocked by the content filter and are generally subsets of items on the block list. Note that the exception list has a higher priority than the block list. If you have MIME entries that appear on both lists, those MIME types are not blocked by the content filter because the exception list takes priority. Therefore, when adding items to the exception list, it is to your advantage to be specific.
- Block Extension List — Because the name of a file is available during file transfers, using file extensions is a highly practical way to block or allow file transfers. The content filter list contains a list of file extensions to be blocked. All protocols support the use of the block extension list.
- Protocol Command Block and Permit Lists — Different protocols use different commands to communicate between servers and clients. By blocking or allowing certain commands, traffic can be controlled on the protocol command level.

The block and permit command lists are intended to be used in combination, with the permit list acting as an exception list to the block list.



**NOTE:** If a protocol command appears on the both the permit list and the block list, that command is permitted.

---

Because not all harmful files or components can be controlled by the MIME type or by the file extension, you can also use the content filter module to block ActiveX, Java Applets, and other types of content. The following types of content blocking are supported only for HTTP:

- Block ActiveX
- Block Java applets
- Block cookies
- Block EXE files
- Block ZIP files

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Content Filtering Protocol Support on page 970](#)
- [Content Filtering Configuration Overview on page 972](#)
- [Monitoring Content Filtering Configurations on page 982](#)

## Content Filtering Protocol Support

---

- [Understanding Content Filtering Protocol Support on page 970](#)
- [Specifying Content Filtering Protocols \(CLI Procedure\) on page 971](#)

### Understanding Content Filtering Protocol Support

Each supported protocol may implement available content filters differently. Not all filtering capabilities are supported for each protocol.

This topic contains the following sections:

- [HTTP Support on page 970](#)
- [FTP Support on page 971](#)
- [E-Mail Support on page 971](#)

#### HTTP Support

---

The HTTP protocol supports all content filtering features. With HTTP, the content filter remains in the gateway, checking every request and response between the HTTP client and server.



If an HTTP request is dropped due to content filtering, the client receives a response such as:

```
<custom drop message/user-configured drop
message>.<src_port><dst_ip>:<dst_port>Download request was dropped due to
<reason>
```

Therefore, a message may appear as follows:

```
Juniper Networks Firewall Content Filtering blocked request. 5.5.5.1:80->4.4.4.1:55247
Download request was dropped due to file extension block list
```

### FTP Support

The FTP protocol does not support all content filtering features. It supports only the following: Block Extension List and Protocol Command Block List.

When content filtering blocks an FTP request, the following response is sent through the control channel:

```
550 <src_ip>:<src_port>-<dst_ip>:<dst_port><custom drop message/user-configured
drop message> for Content Filtering file extension block list.>
```

Therefore, a message may appear as follows:

```
550 5.5.5.1:21->4.4.4.1:45237 Requested action not taken and the request is dropped for
Content Filtering file extension block list
```

### E-Mail Support

E-mail protocols (SMTP, IMAP, POP3) have limited content filtering support for the following features: Block Extension List, Protocol Command Block List, and MIME Pattern Filtering. Support is limited for e-mail protocols for the following reasons:

- The content filter scans only one level of an e-mail header. Therefore recursive e-mail headers and encrypted attachments are not scanned.
- If an entire e-mail is MIME encoded, the content filter can only scan for the MIME type.
- If any part of an e-mail is blocked due to content filtering, the original e-mail is dropped and replaced by a text file with an explanation for why the e-mail was blocked.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Content Filtering Overview on page 969
- Specifying Content Filtering Protocols (CLI Procedure) on page 971
- Content Filtering Configuration Overview on page 972
- Monitoring Content Filtering Configurations on page 982

## Specifying Content Filtering Protocols (CLI Procedure)

To configure content filtering protocols, use the following CLI configuration statements:

```
content-filtering {
  profile name {
    permit-command cmd-list
```

```

    block-command cmd-list
    block-extension file-ext-list
    block-mime {
        list mime-list
        exception ex-mime-list
    }
    block-content-type {
        activex
        java-applet
        exe
        zip
        http-cookie
    }
    notification-options {
        type { message }
        notify-mail-sender
        custom-message msg
    }
}
traceoptions {
    flag {
        all
        basic
        detail
    }
}
}

```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Content Filtering Overview on page 969](#)
- [Content Filtering Configuration Overview on page 972](#)
- [Example: Configuring Content Filtering Custom Objects on page 973](#)
- [Example: Configuring Content Filtering Feature Profiles on page 976](#)
- [Example: Configuring Content Filtering UTM Policies on page 979](#)

## Example: Configuring Content Filtering

- [Content Filtering Configuration Overview on page 972](#)
- [Example: Configuring Content Filtering Custom Objects on page 973](#)
- [Example: Configuring Content Filtering Feature Profiles on page 976](#)
- [Example: Configuring Content Filtering UTM Policies on page 979](#)
- [Example: Attaching Content Filtering UTM Policies to Security Policies on page 980](#)

## Content Filtering Configuration Overview

Content security filter is a new feature that blocks or allows certain type of traffic base on the mime type, file extension, protocol commands and embedded object type. The content filter controls file transfers across the gateway by checking traffic against configured filter lists. The content filtering module evaluates traffic before all other UTM

modules, if traffic meets the criteria configured in the content filter, the content filter acts first upon this traffic. The following procedure lists the recommended order in which you should configure content filters:

1. Configure UTM custom objects for the feature. See Example: Configuring Content Filtering Custom Objects.
2. Configure the main feature parameters using feature profiles. See Example: Configuring Content Filtering Feature Profiles.
3. Configure a UTM policy for each protocol and attach this policy to a profile. See Example: Configuring Content Filtering UTM Policies.
4. Attach the UTM policy to a security policy. See Example: Attaching Content Filtering UTM Policies to Security Policies.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Configuring Content Filtering Custom Objects on page 973](#)
- [Example: Configuring Content Filtering Feature Profiles on page 976](#)
- [Example: Configuring Content Filtering UTM Policies on page 979](#)
- [Example: Attaching Content Filtering UTM Policies to Security Policies on page 980](#)

### Example: Configuring Content Filtering Custom Objects

This example shows how to configure content filtering custom objects.

- [Requirements on page 973](#)
- [Overview on page 973](#)
- [Configuration on page 974](#)
- [Verification on page 975](#)

#### Requirements

---

Before you begin:

1. Decide on the type of content filter you require. See “Content Filtering Overview” on page 969.
2. Understand the order in which content filtering parameters are configured. See “Content Filtering Configuration Overview” on page 972.

#### Overview

---

In this example, you define custom objects that are used to create content filtering profiles. You perform the following tasks to define custom objects:

1. Create two protocol command lists called `ftpprotocom1` and `ftpprotocom2`, and add `user`, `pass`, `port`, and `type` commands to it.
2. Create a filename extension list called `extlist2`, and add the `.zip`, `.js`, and `.vbs` extensions to it.

3. Define block-mime list call cfmime1 and add patterns to the list.

### Configuration

#### CLI Quick Configuration

To quickly configure content filtering custom objects, copy the following commands and paste them into the CLI.

```
[edit]
set security utm custom-objects protocol-command ftpprotocom1 value [user pass port
type]
set security utm custom-objects protocol-command ftpprotocom2 value [user pass port
type]
set security utm custom-objects filename-extension extlist2 value [zip js vbs]
set security utm custom-objects mime-pattern cfmime1 value [video/quicktime
image/x-portable-anymap x-world/x-vrml]
set security utm custom-objects mime-pattern ex-cfmime1 value
[video/quicktime-inappropriate]
```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure content filtering custom objects:

1. Create two protocol command lists.

```
[edit security utm]
user@host# set custom-objects protocol-command ftpprotocom1
[edit security utm]
user@host# set custom-objects protocol-command ftpprotocom2
```

2. Add protocol commands to the list.

```
[edit security utm]
user@host# set custom-objects protocol-command ftpprotocom1 value [user pass
port type]
[edit security utm]
user@host# set custom-objects protocol-command ftpprotocom2 value [user pass
port type]
```

3. Create a filename extension list.

```
[edit security utm]
user@host# set custom-objects filename-extension extlist2
```

4. Add extensions to the list.

```
[edit security utm]
user@host# set custom-objects filename-extension extlist2 value [zip js vbs]
```

5. Create antivirus scanning lists.

```
[edit security utm]
user@host# set custom-objects mime-pattern cfmime1
user@host# set custom-objects mime-pattern ex-cfmime1
```

6. Add patterns to the lists.

```
[edit security utm]
```

```

user@host# set custom-objects mime-pattern cfmime1 value [video/quicktime
image/x-portable-anymap x-world/x-vrml]
user@host# set custom-objects mime-pattern ex-cfmime1 value
[video/quicktime-inappropriate]

```

**Results** From configuration mode, confirm your configuration by entering the **show security utm** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
userhost#show security utm
  custom-objects {
    mime-pattern {
      cfmime1 {
        value [ video/quicktime image/x-portable-anymap x-world/x-vrml ];
      }
      ex-cfmime1 {
        value video/quicktime-inappropriate;
      }
    }
  }
  filename-extension {
    extlist2 {
      value [ zip js vbs ];
    }
  }
  protocol-command {
    ftpprotocom1 {
      value [ user pass port type ];
    }
  }
  protocol-command {
    ftpprotocom2 {
      value [ user pass port type ];
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Content Filtering Custom Objects on page 975

#### **Verifying Content Filtering Custom Objects**

**Purpose** Verify the content filtering custom objects.

**Action** From operational mode, enter the **show configuration security utm** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Content Filtering Overview on page 969](#)

- Content Filtering Configuration Overview on page 972
- Example: Configuring Content Filtering Feature Profiles on page 976
- Example: Configuring Content Filtering UTM Policies on page 979
- Example: Attaching Content Filtering UTM Policies to Security Policies on page 980

## Example: Configuring Content Filtering Feature Profiles

This example describes how to configure the content filtering feature profiles.

- Requirements on page 976
- Overview on page 976
- Configuration on page 976
- Verification on page 978

### Requirements

---

Before you begin:

1. Decide on the type of content filter you require. See “Content Filtering Overview” on page 969.
2. Create custom objects. See “Content Filtering Configuration Overview” on page 972.

### Overview

---

In this example, you configure a feature profile called `confilter1` and specify the following custom objects to be used for filtering content:

1. Apply the `ftpprotocol1` protocol command list custom object to `confilter1`.
2. Apply blocks to Java applets, executable files, and HTTP cookies.
3. Apply the extension list `extlist2` custom object to `confilter1` for blocking extensions.
4. Apply the MIME pattern list custom objects `cfmime1` and `ex-cfmime1` to the `confilter1` for blocking MIME types.
5. Apply the protocol permit command custom object `ftpprotocol2` to `confilter1`. (The permit protocol command list acts as an exception list for the block protocol command list.)



**NOTE:** Protocol command lists, both permit and block, are created by using the same custom object.

6. Configure a custom message to send a notification.

### Configuration

---

#### CLI Quick Configuration

To quickly configure the content filtering feature profile, copy the following commands and paste them into the CLI.

```
[edit]
set security utm feature-profile content-filtering profile confilter1
set security utm feature-profile content-filtering profile confilter1 block-command
  ftpprotocom1
set security utm feature-profile content-filtering profile confilter1 block-content-type
  java-applet exe http-cookie
set security utm feature-profile content-filtering profile confilter1 block-extension extlist2
set security utm feature-profile content-filtering profile confilter1 block-mime list cfmime1
  exception ex-cfmime1
set security utm feature-profile content-filtering profile confilter1 permit-command
  ftpprotocom2
set security utm feature-profile content-filtering profile confilter1 notification-options
  custom-message "the action is not taken" notify-mail-sender type message
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a content filtering feature profiles:

1. Create a content filtering profile.

```
[edit security utm]
user@host# set feature-profile content-filtering profile confilter1
```

2. Apply a protocol command list custom object to the profile.

```
[edit security utm]
user@host# set feature-profile content-filtering profile confilter1 block-command
  ftpprotocom1
```

3. Apply blocks to available content.

```
[edit security utm]
user@host# set feature-profile content-filtering profile confilter1 block-content-type
  java-applet exe http-cookie
```

4. Apply an extension list custom object to the profile.

```
[edit security utm]
user@host# set feature-profile content-filtering profile confilter1 block-extension
  extlist2
```

5. Apply pattern list custom objects to the profile.

```
[edit security utm]
user@host# set feature-profile content-filtering profile confilter1 block-mime list
  cfmime1 exception ex-cfmime1
```

6. Apply the protocol permit command custom object to the profile.

```
[edit security utm]
user@host# set feature-profile content-filtering profile confilter1 permit-command
  ftpprotocom2
```

7. Configure the notification options.

```
[edit security utm]
```

```

user@host# set feature-profile content-filtering profile confilter1m
notification-options custom-message "the action is not taken" notify-mail-sender
type message

```

**Results** From configuration mode, confirm your configuration by entering the **show security utm** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show security utm
feature-profile {
  content-filtering {
    profile contentfilter1;
    profile confilter1 {
      permit-command ftpprotocom2;
      block-command ftpprotocom1;
      block-extension extlist2;
      block-mime {
        list cfmime1;
        exception ex-cfmime1;
      }
      block-content-type {
        java-applet;
        exe;
        http-cookie;
      }
      notification-options {
        type message;
        notify-mail-sender;
        custom-message " the action is not taken";
      }
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying the Configuration of Content Filtering Feature Profile on page 978

#### *Verifying the Configuration of Content Filtering Feature Profile*

**Purpose** Verify the content filtering feature profile.

**Action** From operational mode, enter the **show configuration security utm** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Content Filtering Overview on page 969
- Content Filtering Configuration Overview on page 972



- Example: Configuring Content Filtering Custom Objects on page 973
- Example: Configuring Content Filtering UTM Policies on page 979
- Example: Attaching Content Filtering UTM Policies to Security Policies on page 980

## Example: Configuring Content Filtering UTM Policies

This example describes how to create a content filtering UTM policy to attach to your feature profile.

- Requirements on page 979
- Overview on page 979
- Configuration on page 979
- Verification on page 980

### Requirements

Before you begin:

1. Decide on the type of content filter you require. See “Content Filtering Overview” on page 969.
2. Configure UTM custom objects for each feature and define the content-filtering profile. See “Content Filtering Configuration Overview” on page 972.

### Overview

You configure UTM policies to selectively enforce various UTM solutions on network traffic passing through a UTM-enabled device. Through feature profiles you associate custom objects to these policies and specify blocking or permitting certain types of traffic.

In this example, you configure a UTM policy called `utmp4`, and then assign the preconfigured feature profile `confilter1` to this policy.

### Configuration

#### Step-by-Step Procedure

To configure a content filtering UTM policy:

You can configure different protocol applications in `utm-policy`. The example only show `http` and not other protocols. Earlier we configure custom objects for `ftp` (`ftpprotocolm1` and `ftpprotocolm2`). We should add content filter policy for `ftp`. e.g.

```
set security utm utm-policy utmp4 content-filtering ftp upload-profile confilter1
```

```
set security utm utm-policy utmp4 content-filtering ftp download-profile confilter1
```

1. Create a UTM policy.

```
[edit security utm]
user@host# set utm-policy utmp4
```

2. Attach the UTM policy to the profile.

```
[edit security utm]
user@host# set utm-policy utmp4 content-filtering http-profile contentfilter1
```

3. If you are done configuring the device, commit the configuration.

```
[edit]  
user@host# commit
```

### Verification

---

To verify the configuration is working properly, enter the **show security utm** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Content Filtering Overview on page 969](#)
- [Content Filtering Configuration Overview on page 972](#)
- [Example: Configuring Content Filtering Custom Objects on page 973](#)
- [Example: Configuring Content Filtering Feature Profiles on page 976](#)
- [Example: Attaching Content Filtering UTM Policies to Security Policies on page 980](#)

### Example: Attaching Content Filtering UTM Policies to Security Policies

This example shows how to create a security policy and attach the UTM policy to the security policy.

- [Requirements on page 980](#)
- [Overview on page 980](#)
- [Configuration on page 980](#)
- [Verification on page 982](#)

### Requirements

---

Before you begin:

1. Configure UTM custom objects, define the content filtering profile, and create a UTM policy. See “Content Filtering Configuration Overview” on page 972.
2. Enable and configure a security policy. See “Example: Configuring a Security Policy to Permit or Deny All Traffic” on page 152.

### Overview

---

By attaching content filtering UTM policies to security policies, you can filter traffic transiting from one security zone to another.

In this example, you create a security policy called p4 and specify that traffic from any source address to any destination address with an HTTP application matches the criteria. You then assign a UTM policy called utmp4 to the security policy p4. This UTM policy applies to any traffic that matches the criteria specified in the security policy p4.

### Configuration

---

#### CLI Quick Configuration

To quickly attach a content filtering UTM policy to a security policy, copy the following commands and paste them into the CLI.

```
[edit]
set security policies from-zone trust to-zone untrust policy p4 match source-address any
set security policies from-zone trust to-zone untrust policy p4 match destination-address
  any
set security policies from-zone trust to-zone untrust policy p4 match application
  junos-http
set security from-zone trust to-zone untrust policy p4 then permit application-services
  utm-policy utmp4
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To attach a UTM policy to a security policy:

1. Create a security policy.

```
[edit]
user@host# edit security policies from-zone trust to-zone untrust policy p4
```

2. Specify the match conditions for the policy.

```
[edit security policies from-zone trust to-zone untrust policy p4]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application junos-http
```

3. Attach the UTM policy to the security policy.

```
[edit security policies from-zone trust to-zone untrust policy p4]
user@host# set then permit application-services utm-policy utmp4
```

**Results** From configuration mode, confirm your configuration by entering the `show security policies` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security policies
from-zone trust to-zone untrust {
  policy p4 {
    match {
      source-address any;
      destination-address any;
      application junos-http;
    }
    then {
      permit {
        application-services {
          utm-policy utmp4;
        }
      }
    }
  }
}
default-policy {
  permit-all;
```

```
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Attaching Content Filtering UTM Policies to Security Policies on page 982

#### *Verifying Attaching Content Filtering UTM Policies to Security Policies*

**Purpose** Verify the attachment of the content filtering UTM policy to the security policy.

**Action** From operational mode, enter the **show security policy** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Content Filtering Overview on page 969
- Content Filtering Configuration Overview on page 972
- Example: Configuring Content Filtering Custom Objects on page 973
- Example: Configuring Content Filtering Feature Profiles on page 976
- Example: Configuring Content Filtering UTM Policies on page 979

## Monitoring Content Filtering Configurations

**Purpose** View content filtering statistics.

**Action** To view content filtering statistics in the CLI, enter the **user@host > show security utm content-filtering statistics** command.

The content filtering show statistics command displays the following information:

```
Base on command list: # Blocked
Base on mime list: # Blocked
Base on extension list: # Blocked
ActiveX plugin: # Blocked
Java applet: # Blocked
EXE files: # Blocked
ZIP files: # Blocked
HTTP cookie: # Blocked
```

To view content filtering statistics using J-Web:

1. Select **Monitor>Security>UTM>Content Filtering**.

The following statistics becomes viewable in the right pane.

```
Base on command list: # Passed # Blocked
Base on mime list: # Passed # Blocked
Base on extension list: # Passed # Blocked
ActiveX plugin: # Passed # Blocked
Java applet: # Passed # Blocked
```

EXE files: # Passed # Blocked  
ZIP files: # Passed # Blocked  
HTTP cookie: # Passed # Blocked

2. You can click **Clear Content filtering statistics** to clear all current viewable statistics and begin collecting new statistics.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Content Filtering Overview on page 969](#)
- [Understanding Content Filtering Protocol Support on page 970](#)
- [Content Filtering Configuration Overview on page 972](#)
- [Example: Attaching Content Filtering UTM Policies to Security Policies on page 980](#)



# Web Filtering

- Web Filtering Overview on page 985
- Integrated Web Filtering on page 986
- Redirect Web Filtering on page 997
- Local Web Filtering on page 1005
- Monitoring Web Filtering Configurations on page 1012

## Web Filtering Overview

---

Web filtering lets you manage Internet usage by preventing access to inappropriate Web content. There are three types of Web filtering solutions:

- Integrated Web filtering—The integrated Web filtering solution intercepts every HTTP request in a TCP connection. In this case, the decision making is done on the device after it identifies the category for a URL either from user-defined categories or from a category server (SurfControl Content Portal Authority provided by Websense).



**NOTE:** The integrated Web filtering feature is a separately licensed subscription service. When the license key for Web filtering has expired, no URLs are sent to the category server for checking, only local user-defined categories are checked.

- Redirect Web filtering—The redirect Web filtering solution intercepts HTTP requests and sends them to an external URL filtering server, provided by Websense, to determine whether to block the requests.



**NOTE:** Redirect Web filtering does not require a license.

- Local Web filtering—The local Web filtering solution intercepts every HTTP request in a TCP connection. In this case, the decision making is done on the device after it looks up a URL to determine if it is in the whitelist or blacklist based on its user-defined category.



**NOTE:** Local Web filtering does not require a license or a remote category server.

You can bind either Web filtering profiles or antivirus profiles, or both, to a firewall policy. When both are bound to a firewall policy, Web filtering is applied first, then antivirus is applied. If a URL is blocked by Web filtering, the TCP connection is closed and no antivirus scanning is necessary. If a URL is permitted, the content of the transaction is then passed to the antivirus scanning process.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Integrated Web Filtering on page 986](#)
- [Understanding Redirect Web Filtering on page 997](#)
- [Understanding Local Web Filtering on page 1005](#)
- [Monitoring Web Filtering Configurations on page 1012](#)

## Integrated Web Filtering

- [Understanding Integrated Web Filtering on page 986](#)
- [Example: Configuring Integrated Web Filtering on page 989](#)
- [Displaying Global SurfControl URL categories on page 996](#)

## Understanding Integrated Web Filtering

With integrated Web filtering, the firewall intercepts every HTTP request in a TCP connection and extracts the URL from the HTTP request. Each individual HTTP request is blocked or permitted based on URL filtering profiles defined by you. The decision making is done on the device after it identifies a category for a URL.

A URL category is a list of URLs grouped by content. URL categories are predefined and maintained by SurfControl or are defined by you. SurfControl maintains about 40 predefined categories. When defining your own URL categories, you can group URLs and create categories specific to your needs.

You define your own categories using URL pattern list and custom URL category list custom objects. Once defined, you can select your categories when you configure your Web filtering profile. Each category can have a maximum of 20 URLs. When you create a category, you can add either the URL or the IP address of a site. When you add a URL to a user-defined category, the device performs DNS lookup, resolves the host name into IP addresses, and caches this information. When a user tries to access a site with the IP address of the site, the device checks the cached list of IP addresses and tries to resolve the hostname. Many sites have dynamic IP addresses, meaning that their IP addresses change periodically. A user attempting to access a site can type an IP address that is not in the cached list on the device. Therefore, if you know the IP addresses of sites you are adding to a category, enter both the URL and the IP address(es) of the site.





**NOTE:** If a URL appears in both a user-defined category and a predefined category, the device matches the URL to the user-defined category.



**NOTE:** Web filtering is performed on all the methods defined in HTTP1.0 and HTTP 1.1.

This topic contains the following sections:

- Integrated Web Filtering Process on page 987
- Integrated Web Filtering Cache on page 987
- Integrated Web Filtering Profiles on page 988
- Profile Matching Precedence on page 988

### Integrated Web Filtering Process

This is a general description of how Web traffic is intercepted and acted upon by the Web filtering module.

1. The device intercepts a TCP connection.
2. The device intercepts each HTTP request in the TCP connection.
3. The device extracts each URL in the HTTP request and checks its URL filter cache.
4. Global Web filtering white and blacklists are checked first for block or permit.
5. If the HTTP request URL is allowed based on cached parameters, it is forwarded to the webserver. If there is no cache match, a request for categorization is sent to the SurfControl server. (If the HTTP request URL is blocked, the request is not forwarded and a notification message is logged.)
6. In the allowed case, the SurfControl server responds with the corresponding category.
7. Based on the identified category, if the URL is permitted, the device forwards the HTTP request to the webserver. If the URL is not permitted, then a deny page is sent to the HTTP client.

### Integrated Web Filtering Cache

By default, the device retrieves and caches the URL categories from the SurfControl CPA server. This process reduces the overhead of accessing the SurfControl CPA server each time the device receives a new request for previously requested URLs. You can configure the size and duration of the cache, according to the performance and memory requirements of your networking environment. The lifetime of cached items is configurable between 1 and 1800 seconds with a default value of 300 seconds.



**NOTE:** Caches are not preserved across device reboots or power losses.

## Integrated Web Filtering Profiles

You configure Web filtering profiles that permit or block URLs according to defined categories. A Web filtering profile consists of a group of URL categories assigned one of the following actions:

- Permit — The device always allows access to the websites in this category.
- Block — The device blocks access to the websites in this category. When the device blocks access to this category of websites, it displays a message in your browser indicating the URL category.
- Blacklist — The device always blocks access to the websites in this list. You can create a user-defined category.
- Whitelist — The device always allows access to the websites in this list. You can create a user-defined category.



**NOTE:** A predefined profile is provided and can be used if you choose not to define your own profile.

A Web filtering profile may contain one blacklist or one whitelist, multiple user-defined and/or predefined categories each with a permit or block action, and an *Other* category with a permit or block action. You can define an action for all *Other* categories in a profile to specify what to do when the incoming URL does not belong to any of the categories defined in the profile. If the action for the *Other* category is block, the incoming URL is blocked if it does not match any of the categories explicitly defined in the profile. If an action for the *Other* category is not specified, the default action of permit is applied to the incoming URL not matching any category.

## Profile Matching Precedence

When a profile employs several categories for URL matching, those categories are checked for matches in the following order:

1. If present, the global blacklist is checked first. If a match is made, the URL is blocked. If no match is found...
2. The global whitelist is checked next. If a match is made, the URL is permitted. If no match is found...
3. User-defined categories are checked next. If a match is made, the URL is blocked or permitted as specified. If no match is found...
4. Predefined categories are checked next. If a match is made, the URL is blocked or permitted as specified. If no match is found...
5. The *Other* category is checked next. If a match is made, the URL is blocked or permitted as specified.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Web Filtering Overview on page 985](#)

- Understanding Redirect Web Filtering on page 997
- Understanding Local Web Filtering on page 1005
- Example: Configuring Integrated Web Filtering on page 989

## Example: Configuring Integrated Web Filtering

This example shows how to configure integrated Web filtering.

- Requirements on page 989
- Overview on page 989
- Configuration on page 990
- Verification on page 996

### Requirements

---

Before you begin, learn more about Web filtering. See “Web Filtering Overview” on page 985.

### Overview

---

In this example you configure integrated Web filtering custom objects, integrated Web filtering feature profiles, and integrated Web filtering UTM policies. You also attach integrated Web filtering UTM policies to security policies.

In the first example configuration you create a custom object called `urllist3` that contains the pattern `http://www.juniper.net 1.2.3.4`. The `urllist3` custom object is then added to the custom URL category `custurl3`.

In the second example configuration, you configure the Web filtering feature profile. You set the URL blacklist filtering category to `custblacklist`, set the whitelist filtering category to `custwhitelist` and the type of Web filtering engine to `surf-control-integrated`. Then you set the cache size parameters for Web filtering to 500 KB, which is the default, and the cache timeout parameters to 1800.

You name the Surf Control server as `surfcontrolserver` and enter 8080 as the port number for communicating with it. (Default ports are 80, 8080, and 8081.) Then you create a `surf-control-integrated` profile name called `surfprofile1`.

Next you select a category from the included whitelist and blacklist categories or select a custom URL category list you created for filtering against. Then you enter an action (permit, log and permit, block) to go with the filter. You do this as many times as necessary to compile your whitelists and blacklists and their accompanying actions. This example blocks URLs in the `custurl3` category.

Then you enter a custom message to be sent when HTTP requests are blocked. This example configures the device to send an `***access denied***` message. You select a default action (permit, log and permit, block) for this profile for requests that experience errors. This example sets the default action to block. You select fallback settings (block or log and permit) for this profile, in case errors occur in each configured category. This example sets fallback settings to block.

Finally, you enter a timeout value in seconds. Once this limit is reached, fail mode settings are applied. The default is 10 seconds, and you can enter a value from 10 to 240 seconds. This example sets the timeout value to 10.

In the third example configuration, you create UTM policy utmp5 and attach it to profile surfprofile1.

In the final example configuration, you attach the UTM policy utmp5 to the security policy p5.

### Configuration

- Configuring Integrated Web Filtering Custom Objects on page 990
- Configuring the Integrated Web Filtering Feature Profiles on page 991
- Configuring Integrated Web Filtering UTM Policies on page 994
- Attaching Integrated Web Filtering UTM Policies to Security Policies on page 995

#### *Configuring Integrated Web Filtering Custom Objects*

#### CLI Quick Configuration

To quickly configure integrated Web filtering custom objects, copy the following commands and paste them into the CLI:

```
[edit]
set security utm custom-objects url-pattern urllist3 value http://www.juniper.net
set security utm custom-objects url-pattern urllist3 value 1.2.3.4
set security utm custom-objects url-pattern urllistblack value http://www.untrusted.com
set security utm custom-objects url-pattern urllistblack value 13.13.13.13
set security utm custom-objects url-pattern urllistwhite value http://www.trusted.com
set security utm custom-objects url-pattern urllistwhite value 7.7.7.7
set security utm custom-objects custom-url-category custurl3 value urllist3
set security utm custom-objects custom-url-category custblacklist value urllistblack
set security utm custom-objects custom-url-category custwhitelist value urllistwhite
```

#### Step-by-Step Procedure

To configure integrated Web filtering:

1. Create custom objects and create the URL pattern list.

```
[edit security utm]
user@host# set custom-objects url-pattern urllist3 value [http://www.juniper.net
1.2.3.4]
```

2. Configure the custom URL category list custom object using the URL pattern list.

```
[edit security utm]
user@host# set custom-objects custom-url-category custurl3 value urllist3
```

3. Create a list of untrusted sites

```
[edit security utm]
user@host# set custom-objects url-pattern urllistblack value
[http://www.untrusted.com 13.13.13.13]
```

4. Configure the custom URL category list custom object using the URL pattern list of untrusted sites.

```
[edit security utm]
user@host# set custom-objects custom-url-category custblacklist value urllistblack
```

5. Create a list of trusted sites.

```
[edit security utm]
user@host# set custom-objects url-pattern urllistwhite value
[http://www.trusted.com 7.7.7.7]
```

6. Configure the custom URL category list custom object using the URL pattern list of trusted sites.

```
[edit security utm]
user@host# set custom-objects custom-url-category custwhitelist value urllistwhite
```

**Results** From configuration mode, confirm your configuration by entering the **show security utm custom-objects** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this show command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
userhost#show security utm custom-objects
url-pattern {
  urllist3 {
    value [ http://www.juniper.net ];
  }
  urllistblack {
    value [ http://www.untrusted.com 13.13.13.13 ];
  }
  urllistwhite {
    value [ http://www.trusted.com 7.7.7.7 ];
  }
}
custom-url-category {
  custurl3 {
    value urllist3;
  }
  custblacklist {
    value urllistblack;
  }
  custwhitelist {
    value urllistwhite;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### *Configuring the Integrated Web Filtering Feature Profiles*

**CLI Quick Configuration** To quickly configure Integrated Web filtering feature profiles, copy the following commands and paste them into the CLI:

```
[edit]
set security utm feature-profile web-filtering url-whitelist custwhitelist
set security utm feature-profile web-filtering url-blacklist custblacklist
set security utm feature-profile web-filtering surf-control-integrated cache timeout 1800
```

```

set security utm feature-profile web-filtering surf-control-integrated cache size 500
set security utm feature-profile web-filtering surf-control-integrated server host
  surfcontrolserver
set security utm feature-profile web-filtering surf-control-integrated server port 8080
set security utm feature-profile web-filtering surf-control-integrated profile surfprofile1
  category custurl3 action block
set security utm feature-profile web-filtering surf-control-integrated profile surfprofile1
  default block
set security utm feature-profile web-filtering surf-control-integrated profile surfprofile1
  custom-block-message "***access denied ***"
set security utm feature-profile web-filtering surf-control-integrated profile surfprofile1
  fallback-settings default block
set security utm feature-profile web-filtering surf-control-integrated profile surfprofile1
  fallback-settings server-connectivity block
set security utm feature-profile web-filtering surf-control-integrated profile surfprofile1
  fallback-settings timeout block
set security utm feature-profile web-filtering surf-control-integrated profile surfprofile1
  fallback-settings too-many-requests block
set security utm feature-profile web-filtering surf-control-integrated profile surfprofile1
  timeout 10
set security utm feature-profile content-filtering profile contentfilter1

```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure integrated Web filtering feature profiles:

1. Configure the Web filtering URL Black List.
 

```
[edit security utm feature-profile web-filtering]
user@host# set url-blacklist custblacklist
```
2. Configure the Web filtering URL White List.
 

```
[edit security utm feature-profile web-filtering]
user@host# set url-whitelist custwhitelist
```
3. Specify the surf-control-integrated Web filtering engine and set the cache size parameters.
 

```
[edit security utm feature-profile web-filtering]
user@host# set surf-control-integrated cache size 500
```
4. Set the cache timeout parameters.
 

```
[edit security utm feature-profile web-filtering]
user@host# set surf-control-integrated cache timeout 1800
```
5. Set the server name or IP address.
 

```
[edit security utm feature-profile web-filtering]
user@host# set surf-control-integrated server host surfcontrolserver
```
6. Enter the port number for communicating with the server.
 

```
[edit security utm feature-profile web-filtering]
user@host# set surf-control-integrated server port 8080
```

7. Create a profile name and select a category from the included whitelist and blacklist categories.

```
[edit security utm feature-profile web-filtering]
user@host# set surf-control-integrated profile surfprofile1 category custurl3 action
block
```

8. Enter a custom message to be sent when HTTP requests are blocked.

```
[edit security utm feature-profile web-filtering]
user@host# set surf-control-integrated profile surfprofile1 custom-block-message
"***access denied***"
```

9. Select a default action (permit, log and permit, block) for this profile for requests that experience errors.

```
[edit security utm feature-profile web-filtering]
user@host# set surf-control-integrated profile surfprofile1 default block
```

10. Select fallback settings (block or log and permit) for this profile.

```
[edit security utm feature-profile web-filtering]
user@host# set surf-control-integrated profile surfprofile1 fallback-settings default
block
```

```
user@host# set surf-control-integrated profile surfprofile1 fallback-settings
server-connectivity block
```

```
user@host# set surf-control-integrated profile surfprofile1 fallback-settings timeout
block
```

```
user@host# set surf-control-integrated profile surfprofile1 fallback-settings
too-many-requests block
```

11. Enter a timeout value, in seconds.

```
[edit security utm feature-profile web-filtering]
user@host# set surf-control-integrated profile surfprofile1 timeout 10
```

**Results** From configuration mode, confirm your configuration by entering the **show security utm feature-profile** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
userhost#show security utm feature-profile
web-filtering {
  url-whitelist custwhitelist;
  url-blacklist custblacklist;
  type juniper-local;
  surf-control-integrated {
    cache {
      timeout 1800;
      size 500;
    }
    server {
      host surfcontrolserver;
      port 8080;
    }
  }
  profile surfprofile1 {
    category {
```

```

        custurl3 {
            action block;
        }
    }
    default block;
    custom-block-message "***access denied ***";
    fallback-settings {
        default block;
        server-connectivity block;
        timeout block;
        too-many-requests block;
    }
    timeout 10;
}
}
content-filtering {
    profile contentfilter1;
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### *Configuring Integrated Web Filtering UTM Policies*

**CLI Quick Configuration** To quickly configure integrated Web filtering UTM policies, copy the following commands and paste them into the CLI:

```

[edit]
set security utm utm-policy utmp5 web-filtering http-profile surfprofile1

```

**Step-by-Step Procedure** To configure a UTM policy:

1. Create the UTM policy referencing a profile.

```

[edit]
user@host# set security utm utm-policy utmp5 web-filtering http-profile surfprofile1

```

**Results** From configuration mode, confirm your configuration by entering the **show security utm utm-policy** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
userhost#show security utm utm-policy
...
  utm-policy utmp5 {
    content-filtering {
      http-profile contentfilter1;
    }
    web-filtering {
      http-profile surfprofile1;
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.



*Attaching Integrated Web Filtering UTM Policies to Security Policies*

**CLI Quick Configuration** To quickly attach integrated Web filtering UTM policies to security policies, copy the following commands and paste them into the CLI:

```
[edit]
set security policies from-zone trust to-zone untrust policy p5 match source-address any
set security policies from-zone trust to-zone untrust policy p5 match destination-address
  any
set security policies from-zone trust to-zone untrust policy p5 match application junos-http
set security policies from-zone trust to-zone untrust policy p5 then permit
  application-services utm-policy utmp5
```

**Step-by-Step Procedure** To attach a UTM policy to a security policy:

1. Create and configure the security policy.

```
[edit security policies from-zone trust to-zone untrust policy p5]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application junos-http
```

2. Attach the UTM policy to the security policy.

```
[edit security policies from-zone trust to-zone untrust policy p5]
user@host# set then permit application-services utm-policy utmp5
```

**Results** From configuration mode, confirm your configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
userhost#show security policies
from-zone trust to-zone untrust {
  policy p5 {
    match {
      source-address any;
      destination-address any;
      application junos-http;
    }
    then {
      permit {
        application-services {
          utm-policy utmp5;
        }
      }
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

---

To confirm that the configuration is working properly, perform these tasks:

- Verifying the Configuration of Integrated Web Filtering Custom Objects on page 996
- Verifying the Configuration of Integrated Web Filtering Feature Profiles on page 996
- Verifying the Configuration of Integrated Web Filtering UTM Policies on page 996
- Verifying the Attachment of Integrated Web Filtering UTM Policies to Security Policies on page 996

### *Verifying the Configuration of Integrated Web Filtering Custom Objects*

**Purpose** Verify the configuration of integrated Web filtering custom objects.

**Action** From the top of the configuration in configuration mode, enter the **show security utm custom-objects** command.

### *Verifying the Configuration of Integrated Web Filtering Feature Profiles*

**Purpose** Verify the configuration of integrated Web filtering feature profiles.

**Action** From the top of the configuration in configuration mode, enter the **show security utm feature-profile** command.

### *Verifying the Configuration of Integrated Web Filtering UTM Policies*

**Purpose** Verify the configuration of integrated Web filtering UTM policies.

**Action** From the top of the configuration in configuration mode, enter the **show security utm** command.

### *Verifying the Attachment of Integrated Web Filtering UTM Policies to Security Policies*

**Purpose** Verify the attachment of integrated Web filtering UTM policies to security policies.

**Action** From the top of the configuration in configuration mode, enter the **show security policies** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Web Filtering Overview on page 985
- Understanding Redirect Web Filtering on page 997
- Understanding Local Web Filtering on page 1005
- Understanding Integrated Web Filtering on page 986

## Displaying Global SurfControl URL categories

**Purpose** View global URL categories defined and maintained by SurfControl.

**Action** Enter the `user@host# show groups junos-defaults` CLI command. You can also look for `custom-url-category`.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Web Filtering Overview on page 985](#)
  - [Understanding Redirect Web Filtering on page 997](#)
  - [Understanding Local Web Filtering on page 1005](#)
  - [Understanding Integrated Web Filtering on page 986](#)

## Redirect Web Filtering

- [Understanding Redirect Web Filtering on page 997](#)
- [Example: Configuring Redirect Web Filtering on page 998](#)

### Understanding Redirect Web Filtering

With redirect Web filtering, the Web filtering module intercepts an HTTP request. The URL in the request is then sent to the external Websense server, which makes a permit or a deny decision. If access is permitted to the URL in question, the original HTTP request and all the subsequent requests are sent to the intended HTTP server. But if access is denied to the URL in question, a blocking message is sent to the client.

This is a general description of how Web traffic is intercepted, redirected, and acted upon by the Web filtering module:

1. A Web client establishes a TCP connection with the webserver.
2. The Web client then sends an HTTP request.
3. The device intercepts the requests and extract URL. The URL is checked against Global Web filtering white and blacklists. If no match is made, the Websense server configuration parameters are utilized. Otherwise go to step 6.
4. The URL is sent to the Websense server for checking,
5. The Websense server returns a response indicating whether or not the URL is to be permitted or blocked.
6. If access is allowed, then the original HTTP request is sent to the webserver. If access is denied, the device sends a blocking message to the client and tears down the TCP connection.



**NOTE:** Web filtering is performed on all the methods defined in HTTP1.0 and HTTP 1.1. However, redirect Web filtering does not support HTTPS traffic because it cannot be decrypted to obtain the URL.



**NOTE:** Decision making from real-time options provides a higher level of accuracy, therefore caching for redirect Web filtering is not supported.



**NOTE:** Redirect Web filtering does not require a subscription license.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Web Filtering Overview](#) on page 985
- [Understanding Integrated Web Filtering](#) on page 986
- [Understanding Local Web Filtering](#) on page 1005
- [Example: Configuring Redirect Web Filtering](#) on page 998

## Example: Configuring Redirect Web Filtering

This example shows how to configure redirect Web filtering.

- [Requirements](#) on page 998
- [Overview](#) on page 998
- [Configuration](#) on page 999
- [Verification](#) on page 1004

### Requirements

Before you begin, learn more about Web filtering. See “Web Filtering Overview” on page 985.

### Overview

In this example you configure redirect Web filtering custom objects, redirect Web filtering feature profiles, and redirect Web filtering UTM policies. You also attach redirect Web filtering UTM policies to security policies.

In the first example configuration you create a custom object called `urllist4` that contains the pattern `http://www.juniper.net 1.2.3.4`. The `urllist4` custom object is then added to the custom URL category `custurl4`.

In the second example configuration, you configure the Web filtering feature profile. You set the URL blacklist filtering category to `custblacklist` and the type of Web filtering engine to `Websense-redirect`.

You create a `Websense-redirect` profile name called `websenseprofile1`. Then you name the Web sense server as `Websenseserver` and enter 8080 as the port number for communicating with it. (Default ports are 80, 8080, and 8081.)

You select fallback settings (block or log and permit) for this profile, in case errors occur in each configured category. This example sets fallback settings to block. You enter the number of sockets used for communicating between the client and the server. The default is 1.

Finally, you enter a timeout value in seconds. Once this limit is reached, fail mode settings are applied. The default is 10 seconds, and you can enter a value from 10 to 240 seconds. This example sets the timeout value to 10.

In the third example configuration, you create UTM policy utmp6 and attach it to profile websenseprofile1.

In the final example configuration, you attach the UTM policy utmp6 to the security policy p6.

### Configuration

- Configuring Redirect Web Filtering Custom Objects on page 999
- Configuring the Redirect Web Filtering Feature Profiles on page 1000
- Configuring Redirect Web Filtering UTM Policies on page 1002
- Attaching Redirect Web Filtering UTM Policies to Security Policies on page 1003

#### *Configuring Redirect Web Filtering Custom Objects*

#### CLI Quick Configuration

To quickly configure redirect Web filtering custom objects, copy the following commands and paste them into the CLI:

```
[edit]
set security utm custom-objects url-pattern urllist4 value 1.2.3.4
set security utm custom-objects url-pattern urllistblack value http://www.untrusted.com
set security utm custom-objects url-pattern urllistblack value 13.13.13.13
set security utm custom-objects url-pattern urllistwhite value http://www.trusted.com
set security utm custom-objects url-pattern urllistwhite value 7.7.7.7
set security utm custom-objects custom-url-category custurl4 value urllist4
set security utm custom-objects custom-url-category custblacklist value urllistblack
set security utm custom-objects custom-url-category custwhitelist value urllistwhite
```

#### Step-by-Step Procedure

To configure redirect Web filtering:

1. Create custom objects and create the URL pattern list.

```
[edit security utm]
user@host# set custom-objects url-pattern urllist4 value [http://www.juniper.net
1.2.3.4]
```

2. Configure the custom URL category list custom object using the URL pattern list.

```
[edit security utm]
user@host# set custom-objects custom-url-category custurl4 value urllist4
```

3. Create a list of untrusted sites

```
[edit security utm]
user@host# set custom-objects url-pattern urllistblack value
[http://www.untrusted.com 13.13.13.13]
```

4. Configure the custom URL category list custom object using the URL pattern list of untrusted sites.

```
[edit security utm]
user@host# set custom-objects custom-url-category custblacklist value urllistblack
```

5. Create a list of trusted sites.

```
[edit security utm]
user@host# set custom-objects url-pattern urllistwhite value
[http://www.trusted.com 7.7.7]
```

6. Configure the custom URL category list custom object using the URL pattern list of trusted sites.

```
[edit security utm]
user@host# set custom-objects custom-url-category custwhitelist value urllistwhite
```

**Results** From configuration mode, confirm your configuration by entering the **show security utm custom-objects** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this show command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
userhost#show security utm custom-objects
url-pattern {
  urllist4 {
    value [ http://www.juniper.net 1.2.3.4 ];
  }
  urllistblack {
    value [ http://www.untrusted.com 13.13.13.13 ];
  }
  urllistwhite {
    value [ http://www.trusted.com 7.7.7 ];
  }
}
custom-url-category {
  custurl4 {
    value urllist4;
  }
  custblacklist {
    value urllistblack;
  }
  custwhitelist {
    value urllistwhite;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### *Configuring the Redirect Web Filtering Feature Profiles*

**CLI Quick Configuration** To quickly configure redirect Web filtering feature profiles, copy the following commands and paste them into the CLI:

```
[edit]
set security utm feature-profile web-filtering url-whitelist custwhitelist
set security utm feature-profile web-filtering url-blacklist custblacklist
```

```

set security utm feature-profile web-filtering type juniper-local set security utm
  feature-profile web-filtering websense-redirect profile websenseprofile1 server host
  Websenseserver
set security utm feature-profile web-filtering websense-redirect profile websenseprofile1
  server port 8080
set security utm feature-profile web-filtering websense-redirect profile websenseprofile1
  fallback-settings server-connectivity block
set security utm feature-profile web-filtering websense-redirect profile websenseprofile1
  fallback-settings timeout block
set security utm feature-profile web-filtering websense-redirect profile websenseprofile1
  fallback-settings too-many-requests block
set security utm feature-profile web-filtering websense-redirect profile websenseprofile1
  timeout 10
set security utm feature-profile web-filtering websense-redirect profile websenseprofile1
  sockets 1

```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure redirect Web filtering feature profiles:

1. Configure the Web filtering URL Black List.
 

```
[edit security utm feature-profile web-filtering]
user@host# set url-blacklist custblacklist
```
2. Configure the Web filtering URL White List.
 

```
[edit security utm feature-profile web-filtering]
user@host# set url-whitelist custwhitelist
```
3. Specify the web-filtering type, create a profile name, and set the server name or IP address.
 

```
[edit security utm feature-profile web-filtering]
user@host# set websense-redirect profile websenseprofile1 server host
  Websenseserver
```
4. Enter the port number for communicating with the server.
 

```
[edit security utm feature-profile web-filtering]
user@host# set websense-redirect profile websenseprofile1 server port 8080
```
5. Select fallback settings (block or log and permit) for this profile.
 

```
[edit security utm feature-profile web-filtering]
user@host# set websense-redirect profile websenseprofile1 fallback-settings default
  block

user@host# set websense-redirect profile websenseprofile1 fallback-settings
  server-connectivity block
user@host# set websense-redirect profile websenseprofile1 fallback-settings
  timeout block
user@host# set websense-redirect profile websenseprofile1 fallback-settings
  too-many-requests block
```
6. Enter the number of sockets used for communicating between the client and server.
 

```
[edit security utm feature-profile web-filtering]
```

```
user@host# set websense-redirect profile websenseprofile1 sockets 1
```

7. Enter a timeout value, in seconds.

```
[edit security utm feature-profile web-filtering]
user@host# set .websense-redirect profile websenseprofile1 timeout 10
```

**Results** From configuration mode, confirm your configuration by entering the **show security utm feature-profile** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
userhost#show security utm feature-profile
web-filtering {
  url-whitelist custwhitelist;
  url-blacklist custblacklist;
  type websense-redirect {
    profile websenseprofile1 {
      server {
        host Websenseserver;
        port 8080;
      }
      fallback-settings {
        server-connectivity block;
        timeout block;
        too-many-requests block;
      }
      timeout 10;
      sockets 1;
    }
  }
}
content-filtering {
  profile contentfilter1;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### *Configuring Redirect Web Filtering UTM Policies*

**CLI Quick Configuration** To quickly configure redirect Web filtering UTM Policies, copy the following commands and paste them into the CLI:

```
[edit security utm]
set utm-policy utmp6 web-filtering http-profile websenseprofile1
```

**Step-by-Step Procedure** To configure a UTM policy:

1. Create the UTM policy referencing a profile.

```
[edit security utm]
user@host# set utm-policy utmp6 web-filtering http-profile websenseprofile1
```



**Results** From configuration mode, confirm your configuration by entering the **show security utm** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this show command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
userhost#show security utm
utm-policy utmp6 {
  web-filtering {
    http-profile websenseprofile;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### *Attaching Redirect Web Filtering UTM Policies to Security Policies*

**CLI Quick Configuration** To quickly attach redirect Web filtering UTM policies to security policies, copy the following commands and paste them into the CLI:

```
[edit]
set security policies from-zone trust to-zone untrust policy p6 match source-address any
set security policies from-zone trust to-zone untrust policy p6 match destination-address
any
set security policies from-zone trust to-zone untrust policy p6 match application junos-http
set security policies from-zone trust to-zone untrust policy p6 then permit
application-services utm-policy utmp6
```

**Step-by-Step Procedure** To attach a UTM policy to a security policy:

1. Create and configure the security policy.

```
[edit security policies from-zone trust to-zone untrust policy p6]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application junos-http
```

2. Attach the UTM policy to the security policy.

```
[edit security policies from-zone trust to-zone untrust policy p6]
user@host# set then permit application-services utm-policy utmp6
```

**Results** From configuration mode, confirm your configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
userhost#show security policies
from-zone trust to-zone untrust {
  policy p6 {
    match {
      source-address any;
      destination-address any;
```

```
        application junos-http;
    }
    then {
        permit {
            application-services {
                utm-policy utmp6;
            }
        }
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying the Configuration of Redirect Web Filtering Custom Objects on page 1004
- Verifying the Configuration of Redirect Web Filtering Feature Profiles on page 1004
- Verifying the Configuration of Redirect Web Filtering UTM Policies on page 1004
- Verifying the Attachment of Redirect Web Filtering UTM Policies to Security Policies on page 1004

### *Verifying the Configuration of Redirect Web Filtering Custom Objects*

**Purpose** Verify the configuration of redirect Web filtering custom objects.

**Action** From the top of the configuration in configuration mode, enter the **show security utm custom-objects** command.

### *Verifying the Configuration of Redirect Web Filtering Feature Profiles*

**Purpose** Verify the configuration of redirect Web filtering feature profiles.

**Action** From the top of the configuration in configuration mode, enter the **show security utm feature-profile** command.

### *Verifying the Configuration of Redirect Web Filtering UTM Policies*

**Purpose** Verify the configuration of redirect Web filtering UTM policies.

**Action** From the top of the configuration in configuration mode, enter the **show security utm** command.

### *Verifying the Attachment of Redirect Web Filtering UTM Policies to Security Policies*

**Purpose** Verify the attachment of redirect Web filtering UTM policies to security policies.

**Action** From the top of the configuration in configuration mode, enter the **show security policies** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Web Filtering Overview on page 985](#)
  - [Understanding Integrated Web Filtering on page 986](#)
  - [Understanding Local Web Filtering on page 1005](#)

## Local Web Filtering

---

- [Understanding Local Web Filtering on page 1005](#)
- [Example: Configuring Local Web Filtering on page 1007](#)

### Understanding Local Web Filtering

With local Web filtering, the firewall intercepts every HTTP request in a TCP connection and extracts the URL. The decision making is done on the device after it looks up a URL to determine if it is in the whitelist or blacklist based on its user-defined category. If the URL is in the url-blacklist, the request is blocked; if it's in the url-whitelist, the request is permitted. If the URL is not in either list, the defined default action will occur (block, log-and-permit, or permit). You can permit or block access to a requested site by binding a Web filtering profile to a firewall policy. Local Web filtering provides basic Web filtering without requiring an additional license or external category server.

This topic contains the following sections:

- [User-Defined URL Categories on page 1005](#)
- [Local Web Filtering Process on page 1006](#)
- [Local Web Filtering Profiles on page 1006](#)
- [Profile Matching Precedence on page 1006](#)

#### User-Defined URL Categories

---

When defining your own URL categories, you can group URLs and create categories specific to your needs. Each category can have a maximum of 20 URLs. When you create a category, you can add either the URL or the IP address of a site. When you add a URL to a user-defined category, the device performs DNS lookup, resolves the hostname into IP addresses, and caches this information. When a user tries to access a site with the IP address of the site, the device checks the cached list of IP addresses and tries to resolve the hostname. Many sites have dynamic IP addresses, meaning that their IP addresses change periodically. A user attempting to access a site can type an IP address that is not in the cached list on the device. Therefore, if you know the IP addresses of sites you are adding to a category, enter both the URL and the IP address(es) of the site.

You define your own categories using URL pattern list and custom URL category list custom objects. Once defined, you assign your categories to the global user-defined url-blacklist (block) or url-whitelist (permit) categories.



**NOTE:** Web filtering is performed on all the methods defined in HTTP1.0 and HTTP 1.1.

### Local Web Filtering Process

---

This is a general description of how Web traffic is intercepted and acted upon by the Web filtering module.

1. The device intercepts a TCP connection.
2. The device intercepts each HTTP request in the TCP connection.
3. The device extracts each URL in the HTTP request and checks its URL against the user-defined whitelist and blacklist.
4. If the URL is found in the blacklist, the request is not permitted and a deny page is sent to the http client. If the URL is found in the whitelist, the request is permitted.
5. If the URL is not found in the whitelist or blacklist, the configured default fallback action is applied. If no fallback action is defined, then the request is permitted.

### Local Web Filtering Profiles

---

You configure Web filtering profiles that permit or block URLs according to defined custom categories. A Web filtering profile consists of a group of URL categories assigned one of the following actions:

- Blacklist — The device always blocks access to the websites in this list. Only user-defined categories are used with local Web filtering.
- Whitelist — The device always allows access to the websites in this list. Only user-defined categories are used with local Web filtering.

A Web filtering profile can contain one blacklist or one whitelist with multiple user-defined categories each with a permit or block action. You can define a default fallback action when the incoming URL does not belong to any of the categories defined in the profile. If the action for the default category is block, the incoming URL is blocked if it does not match any of the categories explicitly defined in the profile. If an action for the default action is not specified, the default action of permit is applied to the incoming URL not matching any category.

### Profile Matching Precedence

---

When a profile employs several categories for URL matching, those categories are checked for matches in the following order:

1. If present, the global blacklist is checked first. If a match is made, the URL is blocked. If no match is found...
2. The global whitelist is checked next. If a match is made, the URL is permitted. If no match is found...
3. User-defined categories are checked next. If a match is made, the URL is blocked or permitted as specified.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Web Filtering Overview on page 985](#)

- Understanding Integrated Web Filtering on page 986
- Understanding Redirect Web Filtering on page 997
- Example: Configuring Local Web Filtering on page 1007

## Example: Configuring Local Web Filtering

This example shows how to configure local Web filtering.

- Requirements on page 1007
- Overview on page 1007
- Configuration on page 1008
- Verification on page 1012

### Requirements

---

Before you begin, learn more about Web filtering. See “Web Filtering Overview” on page 985.

### Overview

---

In this example you configure local Web filtering custom objects, local Web filtering feature profiles, and local Web filtering UTM policies. You also attach local Web filtering UTM policies to security policies.

In the first example configuration you create custom objects called `urllist5` and `urllist6` that contains the patterns `http://www.juniper.net 1.2.3.4` and `http://www.acmegizmo.com 1.2.3.4` respectively. The `urllist5` and `urllist6` custom objects are then added to the custom URL category `custurl5` and `custurl6`.

In the second example configuration, you configure the Web filtering feature profile. You set the URL blacklist filtering category to `custurl5` and URL whitelist filtering category to `custurl6`. You set the type of Web filtering engine to `juniper-local`.

Then you create a `juniper-local` profile name called `localprofile1`. You select a default action (permit, log and permit, block) for this profile for requests that experience errors. This example sets the default action to permit.

Then you enter a custom message to be sent when HTTP requests are blocked. This example configures the device to send an `***Access to this site is not permitted***` message. You select fallback settings (block or log and permit) for this profile, in case errors occur in each configured category. This example sets fallback settings to block.

In the third example configuration, you create UTM policy `utmp5` and attach it to profile `localprofile1`.

In the final example configuration, you attach the UTM policy `utmp5` to the security policy `p5`.

## Configuration

- Configuring Local Web Filtering Custom Objects on page 1008
- Configuring the Local Web Filtering Feature Profiles on page 1009
- Configuring Local Web Filtering UTM Policies on page 1010
- Attaching Local Web Filtering UTM Policies to Security Policies on page 1011

### Configuring Local Web Filtering Custom Objects

**CLI Quick Configuration** To quickly configure local Web filtering custom objects, copy the following commands and paste them into the CLI:

```
[edit]
set security utm custom-objects url-pattern urllist5 value http://www.juniper.net
set security utm custom-objects url-pattern urllist5 value 1.2.3.4
set security utm custom-objects url-pattern urllist6 value http://www.acmegizmo.com
set security utm custom-objects url-pattern urllist6 value 1.2.3.4
set security utm custom-objects custom-url-category custurl5 value urllist5
set security utm custom-objects custom-url-category custurl6 value urllist6
```

**Step-by-Step Procedure** To configure local Web filtering using the CLI:

1. Create custom objects and URL pattern lists.

```
[edit]
user@host# set security utm custom-objects url-pattern urllist5 value
[http://www.juniper.net 1.2.3.4]
user@host# set security utm custom-objects url-pattern urllist6 value
[http://www.acmegizmo.com 1.2.3.4]
```

2. Configure the custom URL category list custom object using the URL pattern list.

```
[edit]
user@host# set security utm custom-objects custom-url-category custurl5 value
urllist5
user@host# set security utm custom-objects custom-url-category custurl6 value
urllist6
```

**Results** From configuration mode, confirm your configuration by entering the **show security utm custom-objects** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
userhost#show security utm custom-objects
url-pattern {
  urllist5 {
    value [ http://www.juniper.net 1.2.3.4 ];
  }
  urllist6 {
    value [ http://www.acmegizmo.com 1.2.3.4 ];
  }
}
custom-url-category {
  custurl5 {
    value urllist5;
```

```

    }
    custurl6 {
        value urllist6;
    }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### *Configuring the Local Web Filtering Feature Profiles*

**CLI Quick Configuration** To quickly configure local Web filtering feature profiles, copy the following commands and paste them into the CLI:

```

[edit]
set security utm feature-profile web-filtering url-whitelist custurl4
set security utm feature-profile web-filtering url-blacklist custurl4
set security utm feature-profile web-filtering type juniper-local
set security utm feature-profile web-filtering juniper-local profile localprofile1 default
  permit
set security utm feature-profile web-filtering juniper-local profile localprofile1
  custom-block-message "Access to this site is not permitted."
set security utm feature-profile web-filtering juniper-local profile localprofile1
  fallback-settings default block
set security utm feature-profile web-filtering juniper-local profile localprofile1
  fallback-settings too-many-requests block
set security utm feature-profile content-filtering profile contentfilter1

```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure local Web filtering feature profiles:

1. Configure the Web filtering feature profiles.
 

```

[edit security utm feature-profile web-filtering]
user@host# set url-blacklist custurl3
user@host# set url-blacklist custurl4

```
2. Select the Web filtering engine.
 

```

[edit security utm feature-profile web-filtering]
user@host# set type juniper-local

```
3. Select a default action (permit, log and permit, block) for this profile for requests that experience errors.
 

```

[edit security utm feature-profile web-filtering]
user@host# set juniper-local profile localprofile1 default permit

```
4. Enter a custom message to be sent when HTTP requests are blocked.
 

```

[edit security utm feature-profile web-filtering]
user@host# set juniper-local profile localprofile1 custom-block-message "Access
to this site is not permitted"

```
5. Select fallback settings (block or log and permit) for this profile.
 

```

[edit security utm feature-profile web-filtering]

```

```

user@host# set juniper-local profile localprofile1 fallback-settings default block
set security utm feature-profile web-filtering juniper-local profile localprofile1
fallback-settings too-many-requests block

```

**Results** From configuration mode, confirm your configuration by entering the **show security utm feature-profile** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
userhost#show security utm feature-profile
web-filtering {
  url-whitelist custurl4;
  url-blacklist custurl4;
  type juniper-local;
  juniper-local {
    profile localprofile1 {
      default permit;
      custom-block-message "Access to this site is not permitted.";
      fallback-settings {
        default block;
        too-many-requests block;
      }
    }
  }
}
content-filtering {
  profile contentfilter1;
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### *Configuring Local Web Filtering UTM Policies*

**CLI Quick Configuration** To quickly configure local Web filtering UTM policies, copy the following commands and paste them into the CLI:

```

[edit]
et security utm utm-policy utmp5 web-filtering http-profile localprofile1

```

**Step-by-Step Procedure** To configure a UTM policy:

1. Create the UTM policy referencing a profile.

```

[edit]
user@host# set security utm utm-policy utmp5 web-filtering http-profile localprofile1

```

**Results** From configuration mode, confirm your configuration by entering the **show security utm** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this show command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```

[edit]

```



```

userhost#show security utm
utm-policy utmp5 {
  web-filtering {
    http-profile localprofile1;
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### *Attaching Local Web Filtering UTM Policies to Security Policies*

**CLI Quick Configuration** To quickly attach local Web filtering UTM policies to security policies, copy the following commands and paste them into the CLI.

```

[edit ]
set security policies from-zone trust to-zone untrust policy p5 match source-address any
set security policies from-zone trust to-zone untrust policy p5 match destination-address
any
set security policies from-zone trust to-zone untrust policy p5 match application junos-http
set security policies from-zone trust to-zone untrust policy p5 then permit
application-services utm-policy utmp5

```

**Step-by-Step Procedure** To attach a UTM policy to a security policy:

1. Create and configure the security policy.

```

[edit security policies from-zone trust to-zone untrust policy p5]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application junos-http

```

2. Attach the UTM policy to the security policy.

```

[edit security policies from-zone trust to-zone untrust policy p5]
user@host# set then permit application-services utm-policy utmp5

```

**Results** From configuration mode, confirm your configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
userhost#show security policies
from-zone trust to-zone untrust {
  policy p5 {
    match {
      source-address any;
      destination-address any;
      application junos-http;
    }
    then {
      permit {
        application-services {
          utm-policy utmp5;
        }
      }
    }
  }
}

```

```
    }  
  }
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

---

To confirm that the configuration is working properly, perform these tasks:

- Verifying the Configuration of Local Web Filtering Custom Objects on page 1012
- Verifying the Configuration of Local Web Filtering Feature Profiles on page 1012
- Verifying the Configuration of Local Web Filtering UTM Policies on page 1012
- Verifying the Attachment of Local Web Filtering UTM Policies to Security Policies on page 1012

#### *Verifying the Configuration of Local Web Filtering Custom Objects*

**Purpose** Verify the configuration of local Web filtering custom objects.

**Action** From operational mode, enter the **show security utm custom-objects** command.

#### *Verifying the Configuration of Local Web Filtering Feature Profiles*

**Purpose** Verify the configuration of local Web filtering feature profiles.

**Action** From operational mode, enter the **show security utm feature-profile** command.

#### *Verifying the Configuration of Local Web Filtering UTM Policies*

**Purpose** Verify the configuration of local Web filtering UTM policies.

**Action** From operational mode, enter the **show security utm** command.

#### *Verifying the Attachment of Local Web Filtering UTM Policies to Security Policies*

**Purpose** Verify the attachment of local Web filtering UTM policies to security policies.

**Action** From operational mode, enter the **show security policies** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding Integrated Web Filtering on page 986
  - Understanding Local Web Filtering on page 1005
  - Monitoring Web Filtering Configurations on page 1012

## Monitoring Web Filtering Configurations

---

**Purpose** View Web filtering statistics.

**Action** To view Web filtering statistics using the CLI, enter the following commands:

```
user@host> show security utm web-filtering status
user@host> show security utm web-filtering statistics
```

To view Web filtering statistics using J-Web:

1. Select **Monitor>Security>UTM>Web Filtering**.

The following information becomes viewable in the right pane.

```
Total Requests: #
White List Hit: #
Black List Hit: #
Queries to Server: #
Server Reply Permit: #
Server Reply Block: #
Custom Category Permit: #
Custom Category Block: #
Cache Hit Permit: #
Cache Hit Block: #
Web Filtering Session Total: #
Web Filtering Session Inuse: #
Fall Back: Log-and-Permit Block
Default # #
Timeout # #
Server-Connectivity # #
Too-Many-Requests # #
```

2. You can click the **Clear Web Filtering Statistics** button to clear all current viewable statistics and begin collecting new statistics.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Web Filtering Overview](#) on page 985
- [Understanding Integrated Web Filtering](#) on page 986
- [Example: Configuring Local Web Filtering](#) on page 1007



## PART 9

# Attack Detection and Prevention

- [Attack Detection and Prevention on page 1017](#)
- [Reconnaissance Deterrence on page 1019](#)
- [Suspicious Packet Attributes on page 1051](#)
- [Denial-of-Service Attacks on page 1065](#)



# Attack Detection and Prevention

- Attack Detection and Prevention Overview on page 1017

## Attack Detection and Prevention Overview

---

The Juniper Networks Intrusion Detection and Prevention (IDP) feature, also known as a *stateful firewall*, detects and prevents attacks in network traffic.

An exploit can be either an information-gathering probe or an attack to compromise, disable, or harm a network or network resource. In some cases, the distinction between the two objectives of an exploit can be unclear. For example, a barrage of TCP SYN segments might be an IP address sweep with the intent of triggering responses from active hosts, or it might be a SYN flood attack with the intent of overwhelming a network so that it can no longer function properly. Furthermore, because an attacker usually precedes an attack by performing reconnaissance on the target, we can consider information-gathering efforts as a precursor to an impending attack—that is, they constitute the first stage of an attack. Thus, the term *exploit* encompasses both reconnaissance and attack activities, and the distinction between the two is not always clear.

Juniper Networks provides various detection and defense mechanisms at the zone and policy levels to combat exploits at all stages of their execution:

- Screen options at the zone level.
- Firewall policies at the inter-, intra-, and super-zone policy levels (*super-zone* here means in global policies, where no security zones are referenced).

To secure all connection attempts, Junos OS uses a dynamic packet-filtering method known as *stateful inspection*. Using this method, Junos OS identifies various components in the IP packet and TCP segment headers—source and destination IP addresses, source and destination port numbers, and packet sequence numbers—and maintains the state of each TCP session and pseudo UDP session traversing the firewall. (Junos OS also modifies session states based on changing elements such as dynamic port changes or session termination.) When a responding TCP packet arrives, Junos OS compares the information reported in its header with the state of its associated session stored in the inspection table. If they match, the responding packet is allowed to pass the firewall. If the two do not match, the packet is dropped.

Junos OS screen options secure a zone by inspecting, then allowing or denying, all connection attempts that require crossing an interface bound to that zone. Junos OS then applies firewall policies, which can contain content filtering and IDP components, to the traffic that passes the screen filters.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)



# Reconnaissance Deterrence

- Reconnaissance Deterrence Overview on page 1019
- IP Address Sweeps on page 1019
- Port Scanning on page 1022
- Network Reconnaissance Using IP Options on page 1025
- Operating System Probes on page 1030
- Attacker Evasion Techniques on page 1038

## Reconnaissance Deterrence Overview

---

Attackers can better plan their attack when they first know the layout of the targeted network (which IP addresses have active hosts), the possible entry points (which port numbers are active on the active hosts), and the constitution of their victims (which operating system the active hosts are running). To gain this information, attackers must perform reconnaissance.

Juniper Networks provides several screen options for deterring attackers' reconnaissance efforts and thereby hindering them from obtaining valuable information about the protected network and network resources.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

## IP Address Sweeps

---

- Understanding IP Address Sweeps on page 1019
- Example: Blocking IP Address Sweeps on page 1020

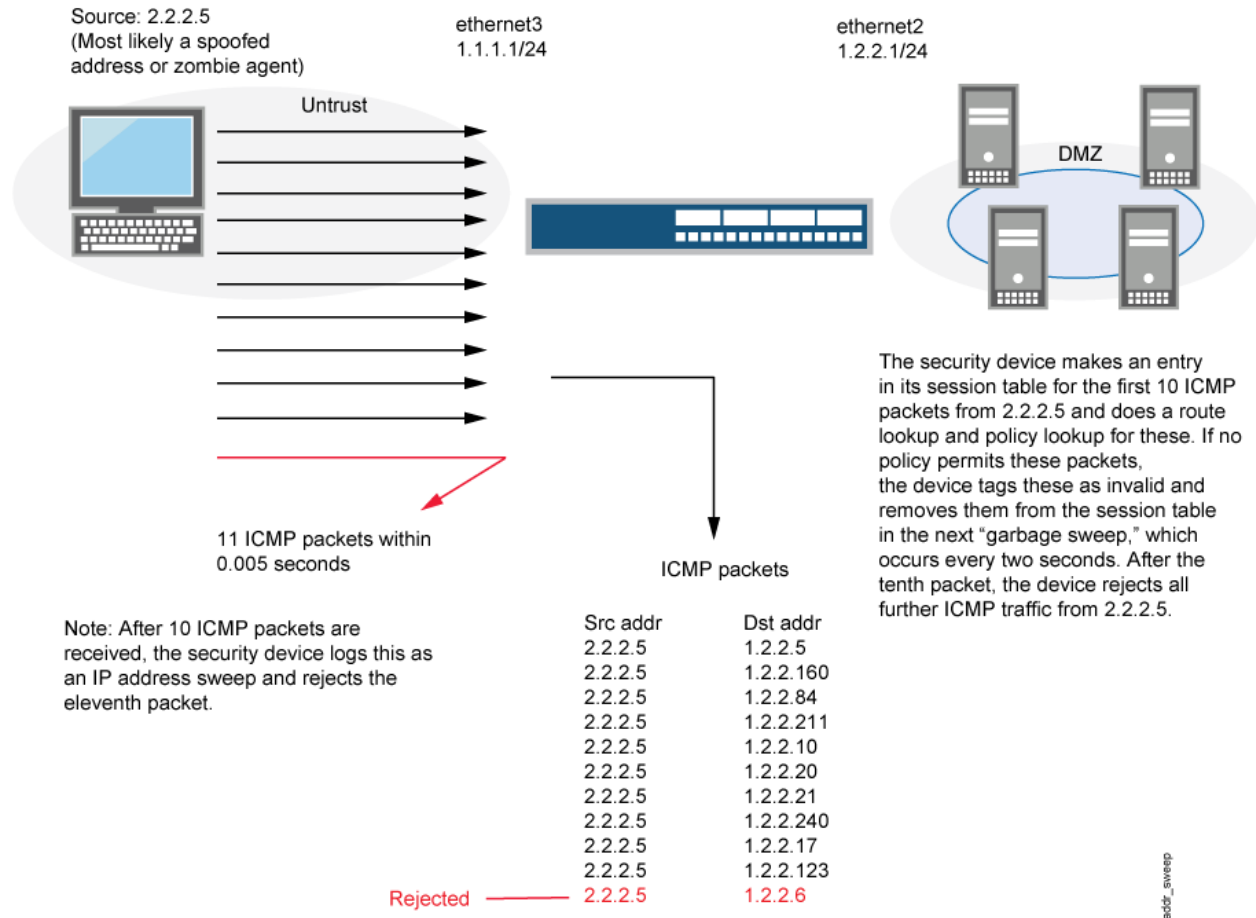
### Understanding IP Address Sweeps

An address sweep occurs when one source IP address sends a defined number of ICMP packets sent to different hosts within a defined interval (5000 microseconds is the default). The purpose of this attack is to send ICMP packets—typically echo requests—to various hosts in the hopes that at least one replies, thus uncovering an address to target.

Junos OS internally logs the number of ICMP packets to different addresses from one remote source. Using the default settings, if a remote host sends ICMP traffic to 10

addresses in 0.005 seconds (5000 microseconds), then the device flags this as an address sweep attack and rejects all further ICMP packets from that host for the remainder of the specified threshold time period. See Figure 63 on page 1020.

Figure 63: Address Sweep



Consider enabling this screen option for a security zone only if there is a policy permitting ICMP traffic from that zone. Otherwise, you do not need to enable the screen option. The lack of such a policy denies all ICMP traffic from that zone, precluding an attacker from successfully performing an IP address sweep anyway.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

**Example: Blocking IP Address Sweeps**

This example describes how to configure a screen to block an IP address sweep originating from a security zone.

- Requirements on page 1021
- Overview on page 1021

- Configuration on page 1021
- Verification on page 1021

### Requirements

---

Before you begin:

1. Understand how IP address sweeps work. See “Understanding IP Address Sweeps” on page 1019.
2. Configure security zones. “Security Zones and Interfaces Overview” on page 111.

### Overview

---

You need to enable a screen for a security zone if you have configured a policy that permits ICMP traffic from that zone. If you have not configured such a policy, then your system denies all ICMP traffic from that zone, and the attacker cannot perform an IP address sweep successfully anyway.

In this example you configure a **5000-ip-sweep** screen to block IP address sweeps originating in the zone-1 security zone.

### Configuration

---

#### Step-by-Step Procedure

To configure a screen to block IP address sweeps:

1. Configure a screen.

```
[edit]
user@host# set security screen ids-option 5000-ip-sweep icmp ip-sweep threshold
5000
```

2. Enable the screen in the security zone.

```
[edit]
user@host# set security zones security-zone zone-1 screen 5000-ip-sweep
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

---

To confirm that the configuration is working properly, perform these tasks:

#### *Verifying the Screens in the Security Zone*

**Purpose** Verify that the screen is enabled in the security zone.

**Action** From operational mode, enter the **show security zones** command.

```
[edit]
user@host> show security zones
Security zone: zone-1
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Screen: 5000-ip-sweep
  Interfaces bound: 1
```

Interfaces:  
ge-1/0/0.0

### *Verifying the Security Screen Configuration*

**Purpose** Display the configuration information about the security screen.

**Action** From operational mode, enter the `show security screen ids-option screen-name` command.

```
[edit]
user@host> show security screen ids-option 5000-id-sweep
Screen object status:
```

Name	Value
ICMP address sweep threshold	5000

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

---

## Port Scanning

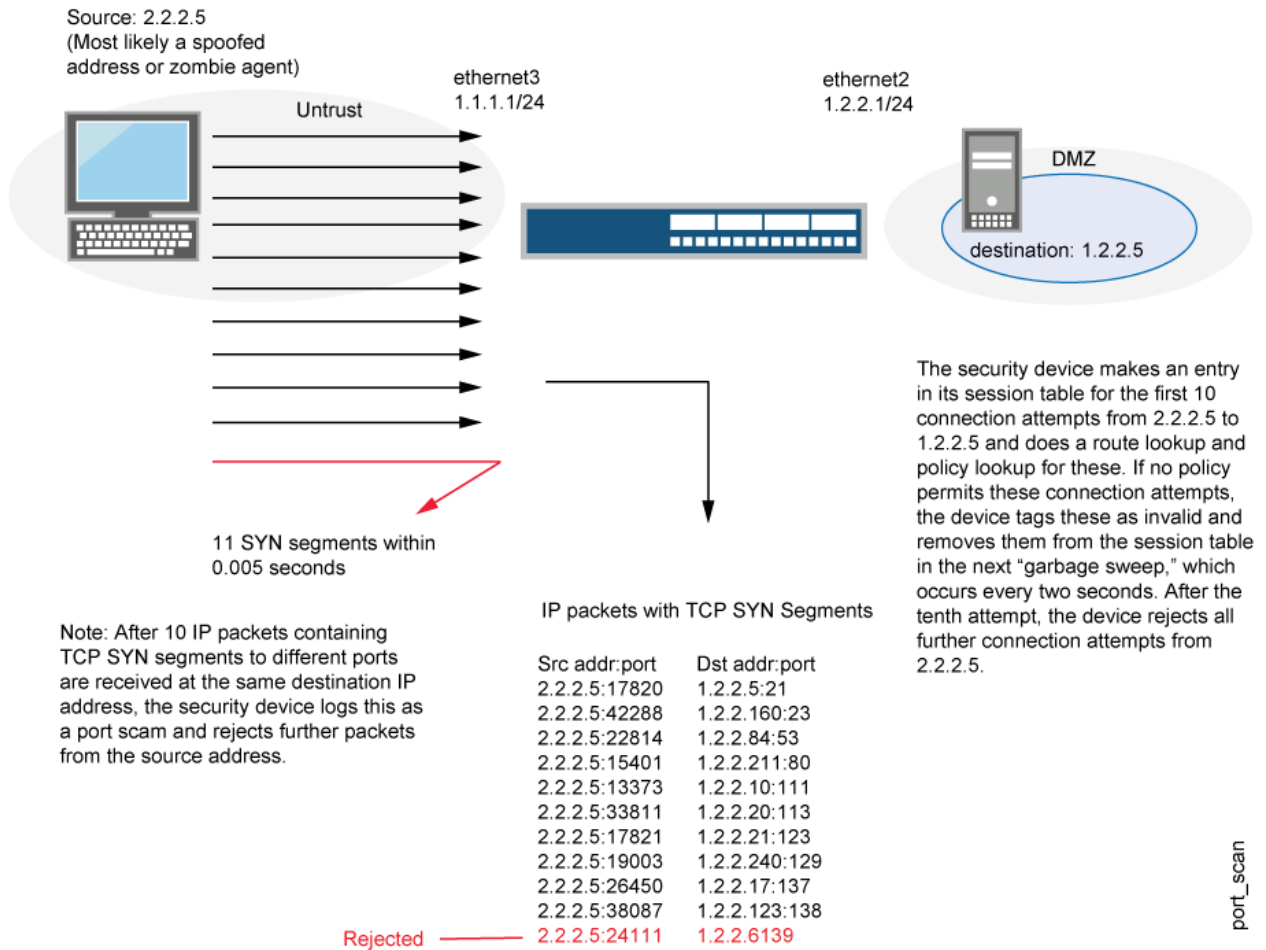
- [Understanding Port Scanning on page 1022](#)
- [Example: Blocking Port Scans on page 1023](#)

### Understanding Port Scanning

A port scan occurs when one source IP address sends IP packets containing TCP SYN segments to a defined number of different ports at the same destination IP address within a defined interval (5000 microseconds is the default). The purpose of this attack is to scan the available services in the hopes that at least one port will respond, thus identifying a service to target.

Junos OS internally logs the number of different ports scanned from one remote source. Using the default settings, if a remote host scans 10 ports in 0.005 seconds (5000 microseconds), then the device flags this as a port scan attack and rejects all further packets from the remote source, regardless of the destination IP address, for the remainder of the specified timeout period. See Figure 64 on page 1023.

Figure 64: Port Scan



**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

### Example: Blocking Port Scans

This example shows how to configure a screen to block port scans originating from a particular security zone.

- Requirements on page 1024
- Overview on page 1024
- Configuration on page 1024
- Verification on page 1024

## Requirements

Before you begin, understand how port scanning works. See “Understanding Port Scanning” on page 1022.

## Overview

You can use a port scan to block IP packets containing TCP SYN segments sent to different ports from the same source address within a defined interval. The purpose of this attack is to scan the available services in the hopes that at least one port will respond. Once a port responds, it is identified as a service to target.

In this example, you configure a 5000-port-scan screen to block port scans originating from a particular security zone and then assign the screen to the zone called zone-1.

## Configuration

### Step-by-Step Procedure

To configure a screen to block port scans:

1. Configure the screen.

```
[edit]
user@host# set security screen ids-option 5000-port-scan tcp port-scan threshold
5000
```

2. Enable the screen in the security zone.

```
[edit]
user@host# set security zones security-zone zone-1 screen 5000-port-scan
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

## Verification

To confirm that the configuration is working properly, perform these tasks:

### Verifying the Screens in the Security Zone

**Purpose** Verify that the screen is enabled in the security zone.

**Action** From operational mode, enter the **show security zones** command.

```
[edit]
user@host> show security zones
Security zone: zone-1
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Screen: 5000-port-scan
  Interfaces bound: 1
  Interfaces:
    ge-1/0/0.0
```

### Verifying the Security Screen Configuration

**Purpose** Display the configuration information about the security screen.

**Action** From operational mode, enter the `show security screen ids-option screen-name` command.

```
[edit]
user@host> show security screen ids-option 5000-port-scan
Screen object status:

Name                               Value
TCP port scan threshold            5000
```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

## Network Reconnaissance Using IP Options

- Understanding Network Reconnaissance Using IP Options on page 1025
- Example: Detecting Packets That Use IP Screen Options for Reconnaissance on page 1028

### Understanding Network Reconnaissance Using IP Options

The IP standard RFC 791, *Internet Protocol*, specifies a set of options for providing special routing controls, diagnostic tools, and security.

RFC 791 states that these options are “unnecessary for the most common communications” and, in reality, they rarely appear in IP packet headers. These options appear after the destination address in an IP packet header, as shown in Figure 65 on page 1025. When they do appear, they are frequently being put to some illegitimate use.

**Figure 65: Routing Options**

Version	Header	Type of Service	Total Packet Length (in Bytes)			
Identification			O	D	M	Fragment Offset
Time to Live (TTL)	Protocol		Header Checksum			
Source Address						
Destination Address						
Options						
Payload						

9030007

This topic contains the following sections:

- Uses for IP Packet Header Options on page 1025
- Screen Options for Detecting IP Options Used for Reconnaissance on page 1027

### [Uses for IP Packet Header Options](#)

Table 104 on page 1026 lists the IP options and their accompanying attributes.

Table 104: IP Options and Attributes

Type	Class	Number	Length	Intended Use	Nefarious Use
End of Options	0*	0	0	Indicates the end of one or more IP options.	None.
No Options	0	1	0	Indicates there are no IP options in the header.	None.
Security	0	2	11 bits	<p>Provides a way for hosts to send security, TCC (closed user group) parameters, and Handling Restriction Codes compatible with Department of Defense (DoD) requirements. (This option, as specified in RFC 791, <i>Internet Protocol</i>, and RFC 1038, <i>Revised IP Security Option</i>, is obsolete.)</p> <p>Currently, this screen option is applicable only to IPv4.</p>	Unknown. However, because it is obsolete, its presence in an IP header is suspect.
Loose Source Route	0	3	Varies	Specifies a partial route list for a packet to take on its journey from source to destination. The packet must proceed in the order of addresses specified, but it is allowed to pass through other devices in between those specified.	Evasion. The attacker can use the specified routes to hide the true source of a packet or to gain access to a protected network.
Record Route	0	7	Varies	<p>Records the IP addresses of the network devices along the path that the IP packet travels. The destination machine can then extract and process the route information. (Due to the size limitation of 40 bytes for both the option and storage space, this can only record up to 9 IP addresses.)</p> <p>Currently, this screen option is applicable only to IPv4.</p>	Reconnaissance. If the destination host is a compromised machine in the attacker's control, he or she can glean information about the topology and addressing scheme of the network through which the packet passed.
Stream ID	0	8	4 bits	<p>(Obsolete) Provided a way for the 16-bit SATNET stream identifier to be carried through networks that did not support the stream concept.</p> <p>Currently, this screen option is applicable only to IPv4.</p>	Unknown. However, because it is obsolete, its presence in an IP header is suspect.



Table 104: IP Options and Attributes (*continued*)

Type	Class	Number	Length	Intended Use	Nefarious Use
Strict Source Route	0	9	Varies	Specifies the complete route list for a packet to take on its journey from source to destination. The last address in the list replaces the address in the destination field.  Currently, this screen option is applicable only to IPv4.	Evasion. An attacker can use the specified routes to hide the true source of a packet or to gain access to a protected network.
Timestamp	2**	4		Records the time (in coordinated universal time [UTC]***) when each network device receives the packet during its trip from the point of origin to its destination. The network devices are identified by IP address.  This option develops a list of IP addresses of the devices along the path of the packet and the duration of transmission between each one.  Currently, this screen option is applicable only to IPv4.	Reconnaissance. If the destination host is a compromised machine in the attacker's control, he or she can glean information about the topology and addressing scheme of the network through which the packet has passed.

\* The class of options identified as 0 was designed to provide extra packet or network control.

\*\* The class of options identified as 2 was designed for diagnostics, debugging, and measurement.

\*\*\* The timestamp uses the number of milliseconds since midnight UTC. UTC is also known as Greenwich Mean Time (GMT), which is the basis for the international time standard.

### Screen Options for Detecting IP Options Used for Reconnaissance

The following screen options detect IP options that an attacker can use for reconnaissance or for some unknown but suspect purpose:

- Record Route—Junos OS detects packets where the IP option is 7 (record route) and records the event in the screen counters list for the ingress interface. Currently, this screen option is applicable only to IPv4.
- Timestamp—Junos OS detects packets where the IP option list includes option 4 (Internet timestamp) and records the event in the screen counters list for the ingress interface. Currently, this screen option is applicable only to IPv4.
- Security—Junos OS detects packets where the IP option is 2 (security) and records the event in the screen counters list for the ingress interface. Currently, this screen option is applicable only to IPv4.
- Stream ID—Junos OS detects packets where the IP option is 8 (stream ID) and records the event in the screen counters list for the ingress interface. Currently, this screen option is applicable only to IPv4.

If a packet with any of the previous IP options is received, Junos OS flags this as a network reconnaissance attack and records the event for the ingress interface.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

## Example: Detecting Packets That Use IP Screen Options for Reconnaissance

This example shows how to detect packets that use IP screen options for reconnaissance.

- Requirements on page 1028
- Overview on page 1028
- Configuration on page 1028
- Verification on page 1029

### Requirements

Before you begin, understand how network reconnaissance works. See “Understanding Network Reconnaissance Using IP Options” on page 1025.

### Overview

RFC 791, *Internet Protocol*, specifies a set of options for providing special routing controls, diagnostic tools, and security. The screen options detect IP options that an attacker can use for reconnaissance, including record route, timestamp, security, and stream ID.

In this example, you configure an IP screen screen-1 and enable it in a security zone called zone-1.



**NOTE:** You can enable only one screen in one security zone.

### Configuration

**CLI Quick Configuration** To quickly detect packets with the record route, timestamp, security, and stream ID IP screen options, copy the following commands and paste them into the CLI.

```
[edit]
set security screen ids-option screen-1 ip record-route-option
set security screen ids-option screen-1 ip timestamp-option
set security screen ids-option screen-1 ip security-option
set security screen ids-option screen-1 ip stream-option
set security zones security-zone zone-1 screen screen-1
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see [Using the CLI Editor in Configuration Mode](#).

To detect packets that use IP screen options for reconnaissance:

1. Configure IP screen options.



**NOTE:** Currently, these screen options support IPv4 only.

```
[edit security screen]
user@host# set ids-option screen-1 ip-record-route-option
user@host# set ids-option screen-1 ip-timestamp-option
user@host# set ids-option screen-1 ip-security-option
user@host# set ids-option screen-1 ip-stream-option
```

2. Enable the screen in the security zone.

```
[edit security zones ]
user@host# set security-zone zone-1 screen screen-1
```

**Results** From configuration mode, confirm your configuration by entering the **show security screen** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
[user@host]show security screen
ids-option screen-1 {
  ip {
    record-route-option;
    timestamp-option;
    security-option;
    stream-option;
  }
}
[edit]
[user@host]show security zones
zones {
  security-zone zone-1 {
    screen screen-1;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying the Screens in the Security Zone on page 1029
- Verifying the Security Screen Configuration on page 1030

#### **Verifying the Screens in the Security Zone**

**Purpose** Verify that the screen is enabled in the security zone.

**Action** From operational mode, enter the **show security zones** command.

```
[edit]
user@host> show security zones
```

```

Security zone: zone-1
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Screen: screen-1
  Interfaces bound: 1
  Interfaces:
    ge-1/0/0.0

```

### Verifying the Security Screen Configuration

**Purpose** Display the configuration information about the security screen.

**Action** From operational mode, enter the `show security screen ids-option screen-name` command.

[edit]

```
user@host> show security screen ids-option screen-1
```

Screen object status:

Name	Value
IP record route option	enabled
IP timestamp option	enabled
IP security option	enabled
IP stream option	enabled

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

## Operating System Probes

- [Understanding Operating System Probes on page 1030](#)
- [TCP Headers with SYN and FIN Flags Set on page 1030](#)
- [TCP Headers With FIN Flag Set and Without ACK Flag Set on page 1033](#)
- [TCP Header with No Flags Set on page 1035](#)

### Understanding Operating System Probes

Before launching an exploit, attackers might try to probe the targeted host to learn its operating system (OS). With that knowledge, they can better decide which attack to launch and which vulnerabilities to exploit. Junos OS can block reconnaissance probes commonly used to gather information about OS types.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

### TCP Headers with SYN and FIN Flags Set

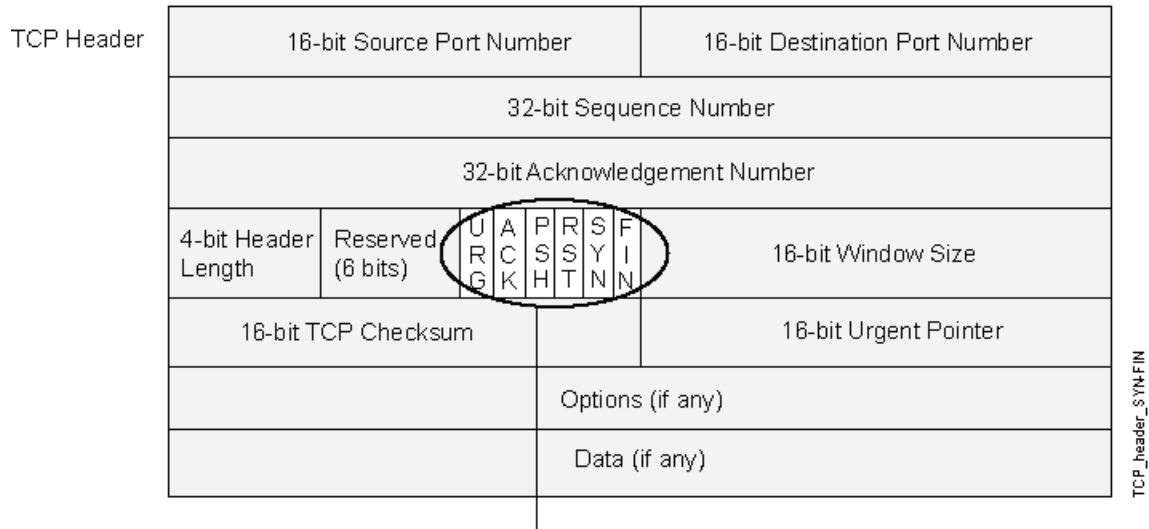
- [Understanding TCP Headers with SYN and FIN Flags Set on page 1030](#)
- [Example: Blocking Packets with SYN and FIN Flags Set on page 1031](#)

#### [Understanding TCP Headers with SYN and FIN Flags Set](#)

Both the SYN and FIN control flags are not normally set in the same TCP segment header. The SYN flag synchronizes sequence numbers to initiate a TCP connection. The FIN flag

indicates the end of data transmission to finish a TCP connection. Their purposes are mutually exclusive. A TCP header with the SYN and FIN flags set is anomalous TCP behavior, causing various responses from the recipient, depending on the OS. See Figure 66 on page 1031.

**Figure 66: TCP Header with SYN and FIN Flags Set**



The SYN and FIN flags are set.

An attacker can send a segment with both flags set to see what kind of system reply is returned and thereby determine what kind of OS is on the receiving end. The attacker can then use any known system vulnerabilities for further attacks.

When you enable this screen option, Junos OS checks if the SYN and FIN flags are set in TCP headers. If it discovers such a header, it drops the packet.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

#### Example: Blocking Packets with SYN and FIN Flags Set

This example shows how to create a screen to block packets with the SYN and FIN flags set.

- Requirements on page 1031
- Overview on page 1032
- Configuration on page 1032
- Verification on page 1032

#### **Requirements**

Before you begin, understand how TCP headers with SYN and FIN flags work. See “Understanding TCP Headers with SYN and FIN Flags Set” on page 1030.

**Overview**

The TCP header with the SYN and FIN flags set cause different responses from a targeted device depending on the OS it is running. The syn-fin screen is enabled for the security zone.

In this example, you create a screen called screen-1 in a security zone to block packets with the SYN and FIN flags set.

**Configuration****Step-by-Step Procedure**

To block packets with both the SYN and FIN flags set:

1. Configure the screen.

```
[edit]
user@host# set security screen ids-option screen-1 tcp syn-fin
```

2. Enable the screen in the security zone.

```
[edit ]
user@host# set security zones security-zone zone-1 screen screen-1
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

**Verification**

To confirm that the configuration is working properly, perform these tasks:

**Verifying the Screens in the Security Zone**

**Purpose** Verify that the screen is enabled in the security zone.

**Action** From operational mode, enter the **show security zones** command.

```
[edit]
user@host> show security zones

Security zone: zone-1
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Screen: screen-1
  Interfaces bound: 1
  Interfaces:
    ge-1/0/0.0
```

**Verifying the Security Screen Configuration**

**Purpose** Display the configuration information about the security screen.

**Action** From operational mode, enter the **show security screen ids-option screen-name** command.

```
[edit]
user@host> show security screen ids-option screen-1
Screen object status:
```

Name	Value
------	-------

TCP SYN FIN                      enabled

## TCP Headers With FIN Flag Set and Without ACK Flag Set

- Understanding TCP Headers With FIN Flag Set and Without ACK Flag Set on page 1033
- Example: Blocking Packets With FIN Flag Set and Without ACK Flag Set on page 1034

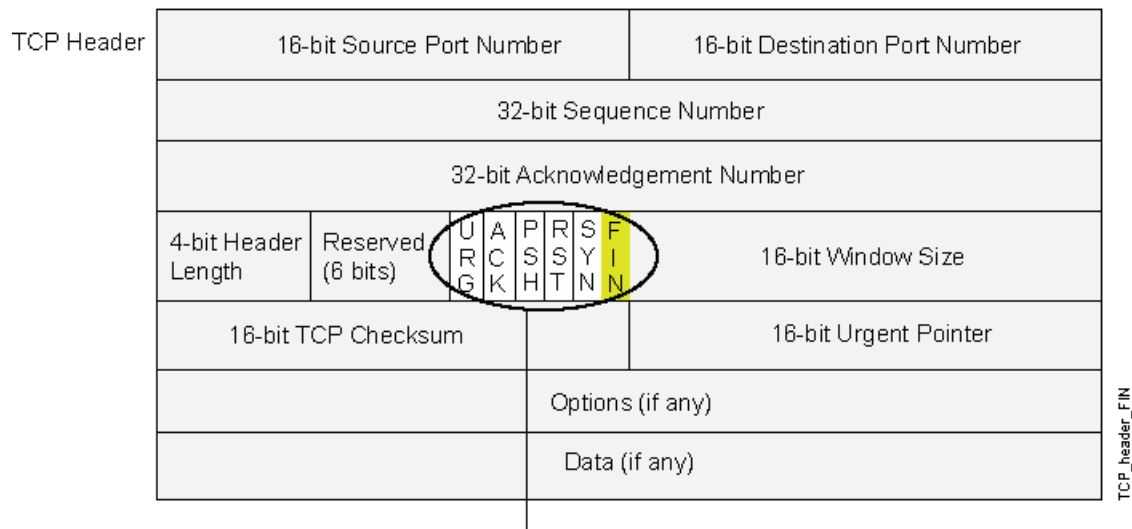
### Understanding TCP Headers With FIN Flag Set and Without ACK Flag Set

Figure 67 on page 1033 shows TCP segments with the FIN control flag set (to signal the conclusion of a session and terminate the connection). Normally, TCP segments with the FIN flag set also have the ACK flag set (to acknowledge the previous packet received). Because a TCP header with the FIN flag set but not the ACK flag is anomalous TCP behavior, there is no uniform response to this. The OS might respond by sending a TCP segment with the RST flag set. Another might completely ignore it. The victim's response can provide the attacker with a clue as to its OS. (Other purposes for sending a TCP segment with the FIN flag set are to evade detection while performing address and port scans and to evade defenses on guard for a SYN flood by performing a FIN flood instead.)



**NOTE:** Vendors have interpreted RFC 793, *Transmission Control Protocol*, variously when designing their TCP/IP implementations. When a TCP segment arrives with the FIN flag set but not the ACK flag, some implementations send RST segments, while others drop the packet without sending an RST.

Figure 67: TCP Header with FIN Flag Set



Only the FIN flag is set.

When you enable this screen option, Junos OS checks if the FIN flag is set but not the ACK flag in TCP headers. If it discovers a packet with such a header, it drops the packet.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

**Example: Blocking Packets With FIN Flag Set and Without ACK Flag Set**

This example shows how to create a screen to block packets with the FIN flag set but the ACK flag not set.

- Requirements on page 1034
- Overview on page 1034
- Configuration on page 1034
- Verification on page 1034

**Requirements**

Before you begin, understand how TCP headers work. See “Understanding TCP Headers With FIN Flag Set and Without ACK Flag Set” on page 1033.

**Overview**

The TCP segments with the FIN flag set also have the ACK flag set to acknowledge the previous packet received. Because a TCP header with the FIN flag set but the ACK flag not set is anomalous TCP behavior, there is no uniform response to this. When you enable the fin-no-ack screen option, Junos OS checks if the FIN flag is set but not the ACK flag in TCP headers. If it discovers a packet with such a header, it drops the packet.

In this example, you create a screen called screen-1 to block packets with the FIN flag set but the ACK flag not set.

**Configuration****Step-by-Step Procedure**

To block packets with the FIN flag set but the ACK flag not set:

1. Configure the screen.

```
[edit ]
user@host# set security screen ids-option screen-1 tcp fin-no-ack
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

**Verification**

To confirm that the configuration is working properly, perform these tasks:

**Verifying the Screens in the Security Zone**

**Purpose** Verify that the screen is enabled in the security zone.

**Action** From operational mode, enter the **show security zones** command.

```
[edit]
user@host> show security zones
```

```
Security zone: zone-1
  Send reset for non-SYN session TCP packets: Off
```



```

Policy configurable: Yes
Screen: screen-1
Interfaces bound: 1
Interfaces:
  ge-1/0/0.0

```

### Verifying the Security Screen Configuration

**Purpose** Display the configuration information about the security screen.

**Action** From operational mode, enter the `show security screen ids-option screen-name` command.

```

[edit]
user@host> show security screen ids-option screen-1
Screen object status:

```

Name	Value
TCP FIN no ACK	enabled

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

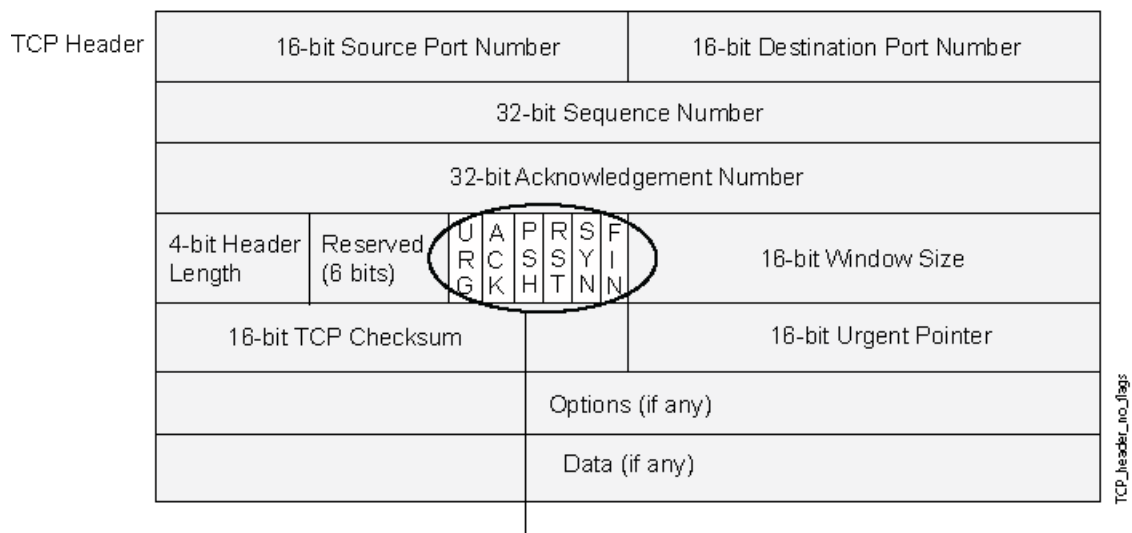
## TCP Header with No Flags Set

- [Understanding TCP Header with No Flags Set](#) on page 1035
- [Example: Blocking Packets with No Flags Set](#) on page 1036

### Understanding TCP Header with No Flags Set

A normal TCP segment header has at least one flag control set. A TCP segment with no control flags set is an anomalous event. Because different operating systems respond differently to such anomalies, the response (or lack of response) from the targeted device can provide a clue as to the type of OS it is running. See Figure 68 on page 1036.

Figure 68: TCP Header with No Flags Set



When you enable the device to detect TCP segment headers with no flags set, the device drops all TCP packets with a missing or malformed flags field.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

**Example: Blocking Packets with No Flags Set**

This example shows how to create a screen to block packets with no flags set.

- Requirements on page 1036
- Overview on page 1036
- Configuration on page 1037
- Verification on page 1037

**Requirements**

Before you begin, understand how a TCP header with no flags set works. See “Understanding TCP Header with No Flags Set” on page 1035.

**Overview**

A normal TCP segment header has at least one flag control set. A TCP segment with no control flags set is an anomalous event. Because different operating systems respond differently to such anomalies, the response (or lack of response) from the targeted device can provide a clue as to the type of OS it is running.

When you enable the device to detect TCP segment headers with no flags set, the device drops all TCP packets with a missing or malformed flags field.

In this example, you create a screen called screen-1 to block packets with no flags set.

**Configuration****Step-by-Step Procedure**

To block packets with no flags set:

1. Configure the screen.

```
[edit ]
user@host# set security screen ids-option screen-1 tcp tcp-no-flag
```

2. Enable the screen in the security zone.

```
[edit ]
user@host# set security zones security-zone zone-1 screen screen-1
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

**Verification**

To confirm that the configuration is working properly, perform these tasks:

**Verifying the Screens in the Security Zone**

**Purpose** Verify that the screen is enabled in the security zone.

**Action** From operational mode, enter the **show security zones** command.

```
[edit]
user@host> show security zones

Security zone: zone-1
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Screen: screen-1
  Interfaces bound: 1
  Interfaces:
    ge-1/0/0.0
```

**Verifying the Security Screen Configuration**

**Purpose** Display the configuration information about the security screen.

**Action** From operational mode, enter the **show security screen ids-option screen-name** command.

```
[edit]
user@host> show security screen ids-option screen-1
Screen object status:
```

Name	Value
TCP no flag	enabled

## Attacker Evasion Techniques

---

- Understanding Attacker Evasion Techniques on page 1038
- Fin Scanning on page 1038
- TCP SYN Checking on page 1039
- IP Spoofing on page 1041
- IP Source Route Options on page 1043

### Understanding Attacker Evasion Techniques

Whether gathering information or launching an attack, it is generally expected that the attacker avoids detection. Although some IP address and port scans are blatant and easily detectable, more wily attackers use a variety of means to conceal their activity. Techniques such as using FIN scans instead of SYN scans—which attackers know most firewalls and intrusion detection programs detect—indicate an evolution of reconnaissance and exploit techniques for evading detection and successfully accomplishing their tasks.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

### Fin Scanning

- Understanding FIN Scans on page 1038
- Thwarting a FIN Scan (CLI Procedure) on page 1038

#### Understanding FIN Scans

---

A FIN scan sends TCP segments with the FIN flag set in an attempt to provoke a response (a TCP segment with the RST flag set) and thereby discover an active host or an active port on a host. Attackers might use this approach rather than perform an address sweep with ICMP echo requests or an address scan with SYN segments, because they know that many firewalls typically guard against the latter two approaches but not necessarily against FIN segments. The use of TCP segments with the FIN flag set might evade detection and thereby help the attackers succeed in their reconnaissance efforts.

#### Thwarting a FIN Scan (CLI Procedure)

---

To thwart FIN scans, take either or both of the following actions:

- Enable the screen option that specifically blocks TCP segments with the FIN flag set but not the ACK flag, which is anomalous for a TCP segment:

```
user@host#set security screen fin-no-ack tcp fin-no-ack
user@host#set security zones security-zone name screen fin-no-ack
```

where *name* is the name of the zone to which you want to apply this screen option .

- Change the packet processing behavior to reject all non-SYN packets that do not belong to an existing session. The SYN check flag is set as the default.



**NOTE:** Changing the packet flow to check that the SYN flag is set for packets that do not belong to existing sessions also thwarts other types of non-SYN scans, such as a null scan (when no TCP flags are set).

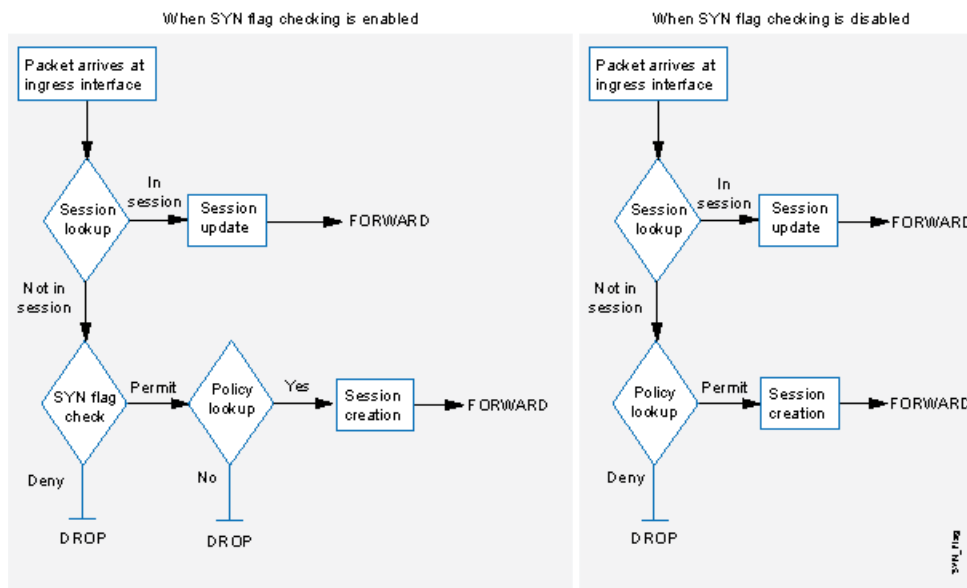
## TCP SYN Checking

- Understanding TCP SYN Checking on page 1039
- Setting TCP SYN Checking (CLI Procedure) on page 1041
- Setting Strict SYN Checking (CLI Procedure) on page 1041

### Understanding TCP SYN Checking

By default, Junos OS checks for SYN flags in the first packet of a session and rejects any TCP segments with non-SYN flags attempting to initiate a session. You can leave this packet flow as is or change it so that Junos OS does not enforce SYN flag checking before creating a session. Figure 69 on page 1039 illustrates packet flow sequences both when SYN flag checking is enabled and when it is disabled.

**Figure 69: SYN Flag Checking**



When Junos OS with SYN flag checking enabled receives a non-SYN TCP segment that does not belong to an existing session, it drops the packet. By default, Junos OS does not send a TCP RST to the source host on receiving the non-SYN segment. You can configure the device to send TCP RST to the source host by using the **set security zones security-zone trust tcp-rst** command. If the code bit of the initial non-SYN TCP packet is RST, the device does not send a TCP-RST.

Not checking for the SYN flag in the first packets offers the following advantages:

- NSRP with Asymmetric Routing—In an active/active NSRP configuration in a dynamic routing environment, a host might send the initial TCP segment with the SYN flag set

to one device (Device-A), but the SYN/ACK might be routed to the other device in the cluster (Device-B). If this asymmetric routing occurs after Device-A has synchronized its session with Device-B, all is well. On the other hand, if the SYN/ACK response reaches Device-B before Device-A synchronizes the session and SYN checking is enabled, Device-B rejects the SYN/ACK, and the session cannot be established. With SYN checking disabled, Device-B accepts the SYN/ACK response—even though there is no existing session to which it belongs—and creates a new session table entry for it.

- **Uninterrupted Sessions**—If you reset the device or even change a component in the core section of a policy and SYN checking is enabled, all existing sessions or those sessions to which the policy change applies are disrupted and must be restarted. Disabling SYN checking avoids such disruptions to network traffic flows.



**NOTE:** A solution to this scenario is to install the device with SYN checking disabled initially. Then, after a few hours—when established sessions are running through the device—enable SYN checking. The core section in a policy contains the following main components: source and destination zones, source and destination addresses, one or more services, and an action.

However, the previous advantages exact the following security sacrifices:

- **Reconnaissance Holes**—When an initial TCP segment with a non-SYN flag—such as ACK, URG, RST, FIN—arrives at a closed port, many operating systems (Windows, for example) respond with a TCP segment that has the RST flag set. If the port is open, then the recipient does not generate any response.

By analyzing these responses or lack thereof, an intelligence gatherer can perform reconnaissance on the protected network and also on the Junos OS policy set. If a TCP segment is sent with a non-SYN flag set and the policy permits it through, the destination host receiving such a segment might drop it and respond with a TCP segment that has the RST flag set. Such a response informs the perpetrator of the presence of an active host at a specific address and that the targeted port number is closed. The intelligence gatherer also learns that the firewall policy permits access to that port number on that host.

By enabling SYN flag checking, Junos OS drops TCP segments without a SYN flag if they do not belong to an existing session. It does not return a TCP RST segment. Consequently, the scanner gets no replies regardless of the policy set or whether the port is open or closed on the targeted host.

- **Session Table Floods**—If SYN checking is disabled, an attacker can bypass the Junos OS SYN flood protection feature by flooding a protected network with a barrage of TCP segments that have non-SYN flags set. Although the targeted hosts drop the packets—and possibly send TCP RST segments in reply—such a flood can fill up the session table of the Juniper Networks device. When the session table is full, the device cannot process new sessions for legitimate traffic.

By enabling SYN checking and SYN flood protection, you can thwart this kind of attack. Checking that the SYN flag is set on the initial packet in a session forces all new sessions

to begin with a TCP segment that has the SYN flag set. SYN flood protection then limits the number of TCP SYN segments per second so that the session table does not become overwhelmed.

If you do not need SYN checking disabled, Juniper Networks strongly recommends that it be enabled (its default state for an initial installation of Junos OS). You can enable it with the `set flow tcp-syn-check` command. With SYN checking enabled, the device rejects TCP segments with non-SYN flags set unless they belong to an established session.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

### Setting TCP SYN Checking (CLI Procedure)

With SYN checking enabled, the device rejects TCP segments with non-SYN flags set unless they belong to an established session. Enabling SYN checking can help prevent attacker reconnaissance and session table floods. TCP SYN checking is enabled by default.

To disable SYN checking:

```
user@host#set security flow tcp-session no-syn-check
```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

### Setting Strict SYN Checking (CLI Procedure)

With strict SYN checking enabled, the device enables the strict three-way handshake check for the TCP session. It enhances security by dropping data packets before the three-way handshake is done. TCP strict SYN checking is disabled by default.



**NOTE:** The `strict-syn-check` option cannot be enabled if `no-syn-check` or `no-syn-check-in-tunnel` is enabled.

To enable strict SYN checking:

```
user@host#set security flow tcp-session strict-syn-check
```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

## IP Spoofing

- Understanding IP Spoofing on page 1041
- Example: Blocking IP Spoofing on page 1042

### Understanding IP Spoofing

One method of attempting to gain access to a restricted area of the network is to insert a false source address in the packet header to make the packet appear to come from a trusted source. This technique is called IP spoofing. The mechanism to detect IP spoofing relies on route table entries. For example, if a packet with source IP address 10.1.1.6 arrives

at ge-0/0/1, but Junos OS has a route to 10.1.1.0/24 through ge-0/0/0, a check for IP spoofing discovers that this address arrived at an invalid interface as defined in the route table. A valid packet from 10.1.1.6 can only arrive via ge-0/0/0, not ge-0/0/1. Therefore, Junos OS concludes that the packet has a spoofed source IP address and discards it.



**NOTE:** IP Spoofing allows Junos OS to detect and drop IPv6 spoofed packets.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

#### Example: Blocking IP Spoofing

This example shows how to configure a screen to block IP spoof attacks.

- Requirements on page 1042
- Overview on page 1042
- Configuration on page 1042
- Verification on page 1042

#### Requirements

Before you begin, understand how IP Spoofing works. See “Understanding IP Spoofing” on page 1041.

#### Overview

One method of attempting to gain access to a restricted area of a network is to insert a bogus source address in the packet header to make the packet appear to come from a trusted source. This technique is called IP spoofing.

In this example, you configure a screen called screen-1 to block IP spoof attacks and enable the screen in the zone-1 security zone.

#### Configuration

#### Step-by-Step Procedure

To block IP spoofing:

1. Configure the screen.

```
[edit ]
user@host# set security screen ids-option screen-1 ip spoofing
```

2. Enable the screen in the security zone.

```
[edit]
user@host# set security zone security-zone zone-1 screen screen-1
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

#### Verification

To confirm that the configuration is working properly, perform these tasks:



**Verifying the Screens in the Security Zone**

**Purpose** Verify that the screen is enabled in the security zone.

**Action** From operational mode, enter the **show security zones** command.

```
[edit]
user@host> show security zones

Security zone: zone-1
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Screen: screen-1
  Interfaces bound: 1
  Interfaces:
    ge-1/0/0.0
```

**Verifying the Security Screen Configuration**

**Purpose** Display the configuration information about the security screen.

**Action** From operational mode, enter the **show security screen ids-option screen-name** command.

```
[edit]
user@host> show security screen ids-option screen-1
Screen object status:
```

Name	Value
IP spoofing	enabled

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

## IP Source Route Options

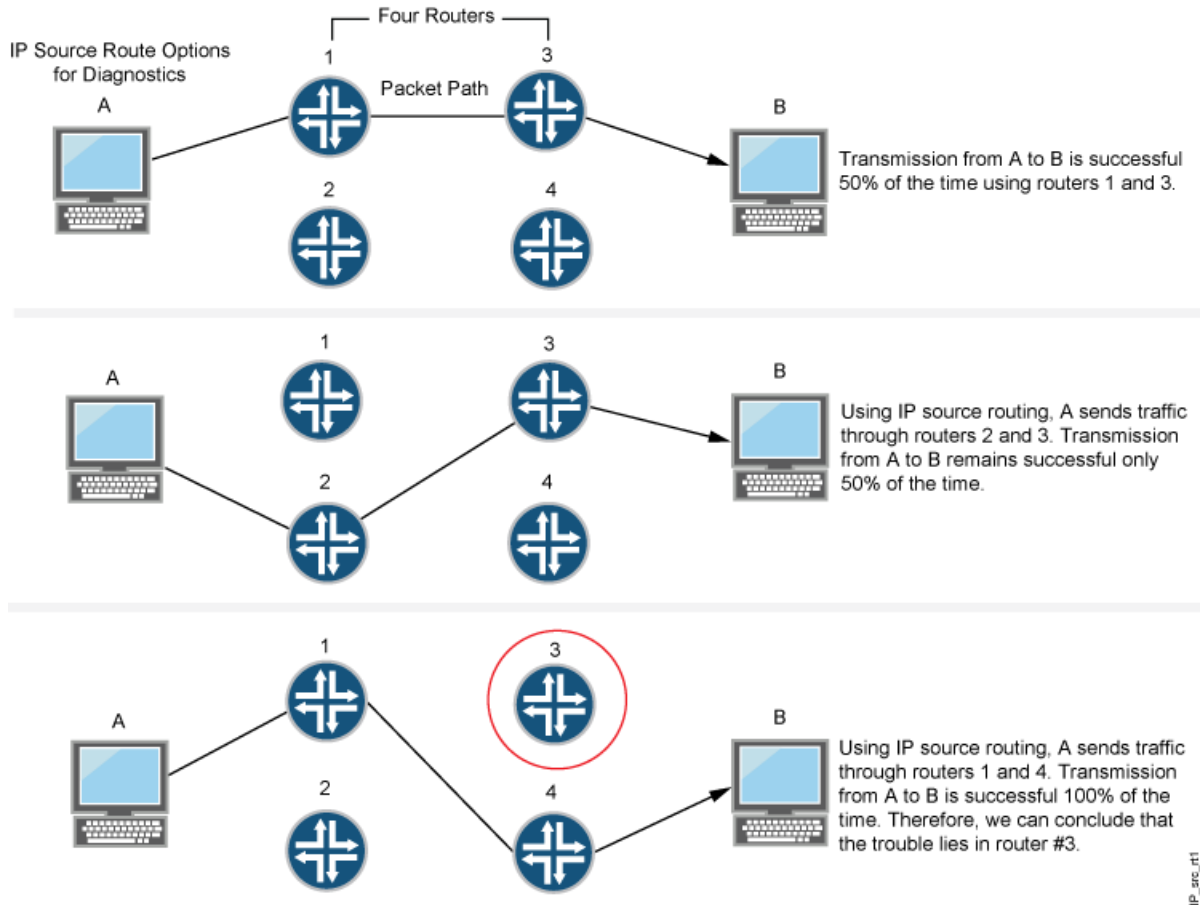
- [Understanding IP Source Route Options on page 1043](#)
- [Example: Blocking Packets with Either a Loose or a Strict Source Route Option Set on page 1046](#)
- [Example: Detecting Packets with Either a Loose or a Strict Source Route Option Set on page 1047](#)

### Understanding IP Source Route Options

Source routing was designed to allow users at the source of an IP packet transmission to specify the IP addresses of the devices (also referred to as “hops” ) along the path that they want an IP packet to take on its way to its destination. The original intent of the IP source route options was to provide routing control tools to aid diagnostic analysis. If, for example, the transmission of a packet to a particular destination meets with irregular success, you might first use either the record route or the timestamp IP option to discover the addresses of devices along the path or paths that the packet takes. You can then use either the loose or the strict source route option to direct traffic along a specific path, using the addresses you learned from the results that the record route or timestamp

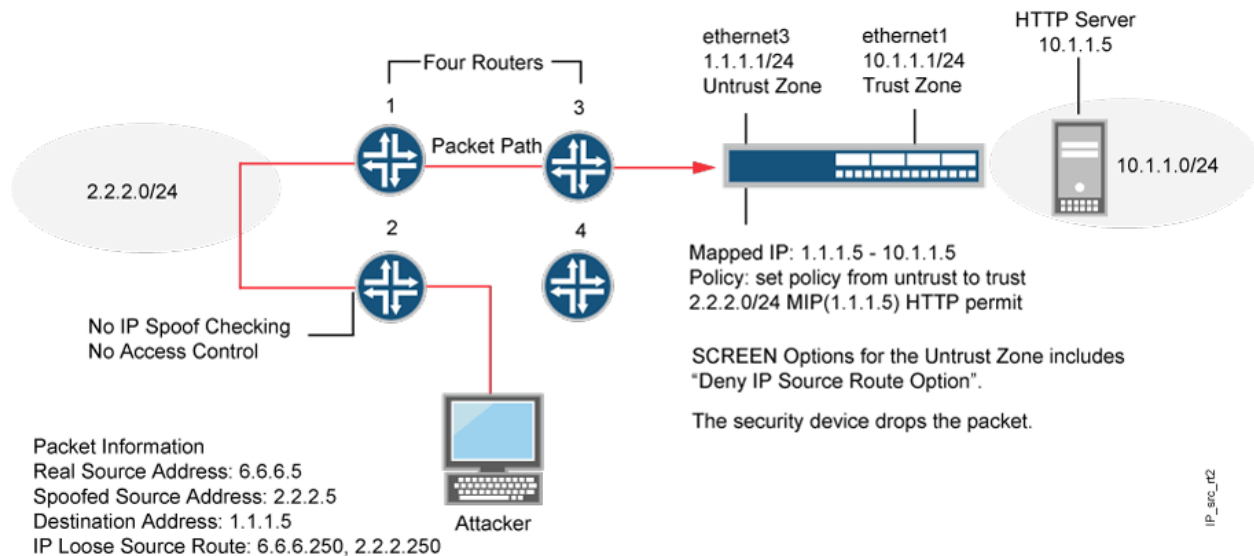
options produced. By changing device addresses to alter the path and sending several packets along different paths, you can note changes that either improve or lessen the success rate. Through analysis and the process of elimination, you might be able to deduce where the trouble lies. See Figure 70 on page 1044.

Figure 70: IP Source Routing



Although the uses of IP source route options were originally benign, attackers have learned to put them to more devious uses. They can use IP source route options to hide their true address and access restricted areas of a network by specifying a different path. For an example showing how an attacker can put both deceptions to use, consider the following scenario as illustrated in Figure 71 on page 1045.

Figure 71: Loose IP Source Route Option for Deception



Junos OS only allows traffic 2.2.2.0/24 if it comes through ethernet1, an interface bound to zone\_external. Devices 3 and 4 enforce access controls but devices 1 and 2 do not. Furthermore, device 2 does not check for IP spoofing. The attacker spoofs the source address and, by using the loose source route option, directs the packet through device 2 to the 2.2.2.0/24 network and from there out device 1. Device 1 forwards it to device 3, which forwards it to the Juniper Networks device. Because the packet came from the 2.2.2.0/24 subnet and has a source address from that subnet, it seems to be valid. However, one remnant of the earlier chicanery remains: the loose source route option. In this example, you have enabled the deny IP source route screen option for zone\_external. When the packet arrives at ethernet3, the device rejects it.

You can enable the device to either block any packets with loose or strict source route options set or detect such packets and then record the event in the counters list for the ingress interface. The screen options are as follows:

- Deny IP Source Route Option—Enable this option to block all IP traffic that employs the loose or strict source route option. Source route options can allow an attacker to enter a network with a false IP address.
- Detect IP Loose Source Route Option—The device detects packets where the IP option is 3 (Loose Source Routing) and records the event in the screen counters list for the ingress interface. This option specifies a partial route list for a packet to take on its journey from source to destination. The packet must proceed in the order of addresses specified, but it is allowed to pass through other devices in between those specified.
- Detect IP Strict Source Route Option—The device detects packets where the IP option is 9 (Strict Source Routing) and records the event in the screen counters list for the ingress interface. This option specifies the complete route list for a packet to take on its journey from source to destination. The last address in the list replaces the address in the destination field. Currently, this screen option is applicable to IPv4 only.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

**Example: Blocking Packets with Either a Loose or a Strict Source Route Option Set**

---

This example shows how to block packets with either a loose or a strict source route option set.

- Requirements on page 1046
- Overview on page 1046
- Configuration on page 1046
- Verification on page 1046

**Requirements**

Before you begin, understand how IP source route options work. See “Understanding IP Source Route Options” on page 1043.

**Overview**

Source routing allows users at the source of an IP packet transmission to specify the IP addresses of the devices (also referred to as “hops” ) along the path that they want an IP packet to take on its way to its destination. The original intent of the IP source route options was to provide routing control tools to aid diagnostic analysis.

You can enable the device to either block any packets with loose or strict source route options set or detect such packets and then record the event in the counters list for the ingress interface.

In this example you create the screen called screen-1 to block packets with either a loose or a strict source route option set and enable the screen in the zone-1 security zone.

**Configuration****Step-by-Step  
Procedure**

To block packets with either the loose or the strict source route option set:

1. Configure the screen.

```
[edit ]
```

```
user@host# set security screen ids-option screen-1 ip source-route-option
```

2. Enable the screen in the security zone.

```
[edit ]
```

```
user@host# set security zones security-zone zone-1 screen screen-1
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
```

```
user@host# commit
```

**Verification**

To confirm that the configuration is working properly, perform these tasks:

**Verifying the Screens in the Security Zone**

**Purpose** Verify that the screen is enabled in the security zone.

**Action** From operational mode, enter the **show security zones** command.

```
[edit]
user@host> show security zones
Security zone: zone-1
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Screen: screen-1
  Interfaces bound: 1
  Interfaces:
    ge-1/0/0.0
```

**Verifying the Security Screen Configuration**

**Purpose** Display the configuration information about the security screen.

**Action** From operational mode, enter the **show security screen ids-option screen-name** command.

```
[edit]
user@host> show security screen ids-option screen-1
Screen object status:

Name                               Value
IP source route option             enabled
```

**Example: Detecting Packets with Either a Loose or a Strict Source Route Option Set**

This example shows how to detect packets with either a loose or a strict source route option set.

- Requirements on page 1047
- Overview on page 1047
- Configuration on page 1048
- Verification on page 1048

**Requirements**

Before you begin, understand how IP source route options work. See “Understanding IP Source Route Options” on page 1043.

**Overview**

Source routing allows users at the source of an IP packet transmission to specify the IP addresses of the devices (also referred to as “hops”) along the path that they want an IP packet to take on its way to its destination. The original intent of the IP source route options was to provide routing control tools to aid diagnostic analysis.

You can enable the device to either block any packets with loose or strict source route options set or detect such packets and then record the event in the counters list for the ingress interface.

In this example, you create two screens called screen-1 and screen-2 to detect and record, but not block, packets with a loose or strict source route option set and enable the screens in zones zone-1 and zone-2.

### Configuration

**Step-by-Step Procedure** To detect and record, but not block, packets with a loose or strict source route option set:

1. Configure the loose source screen.

```
[edit]
user@host# set security screen ids-option screen-1 ip loose-source-route-option
```

2. Configure the strict source route screen.

```
[edit]
user@host# set security screen ids-option screen-2 ip strict-source-route-option
```



**NOTE:** Currently, this screen option supports IPv4 only.

3. Enable the screens in the security zones.

```
[edit]
user@host# set security zones security-zone zone-1 screen screen-1
user@host# set security zones security-zone zone-2 screen screen-2
```

4. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

To confirm that the configuration is working properly, perform these tasks:

#### Verifying the Screens in the Security Zone

**Purpose** Verify that the screen is enabled in the security zone.

**Action** From operational mode, enter the **show security zones** command.

```
[edit]
user@host> show security zones

Security zone: zone-1
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Screen: screen-1
  Interfaces bound: 1
  Interfaces:
    ge-1/0/0.0
Security zone: zone-2
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Screen: screen-2
  Interfaces bound: 1
```

Interfaces:  
ge-2/0/0.0

### *Verifying the Security Screen Configuration*

**Purpose** Display the configuration information about the security screen.

**Action** From operational mode, enter the `show security screen ids-option screen-name` command.

[edit]

```
user@host> show security screen ids-option screen-1  
Screen object status:
```

Screen object status:

Name	Value
IP loose source route option	enabled

[edit]

```
user@host> show security screen ids-option screen-2  
Screen object status:
```

Screen object status:

Name	Value
IP strict source route option	enabled





## CHAPTER 39

# Suspicious Packet Attributes

- [Suspicious Packet Attributes Overview on page 1051](#)
- [ICMP Fragment Protection on page 1051](#)
- [Large ICMP Packet Protection on page 1053](#)
- [Bad IP Option Protection on page 1055](#)
- [Unknown Protocol Protection on page 1057](#)
- [IP Packet Fragment Protection on page 1059](#)
- [SYN Fragment Protection on page 1061](#)

## Suspicious Packet Attributes Overview

---

Attackers can craft packets to perform reconnaissance or launch denial-of-service (DoS) attacks. Sometimes it is unclear what the intent of a crafted packet is, but the very fact that it is crafted suggests that it is being put to some kind of insidious use.

The following topics describe screen options that block suspicious packets that might contain hidden threats:

- [Understanding ICMP Fragment Protection on page 1052](#)
- [Understanding Large ICMP Packet Protection on page 1053](#)
- [Understanding Bad IP Option Protection on page 1055](#)
- [Understanding Unknown Protocol Protection on page 1057](#)
- [Understanding IP Packet Fragment Protection on page 1059](#)
- [Understanding SYN Fragment Protection on page 1061](#)

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

## ICMP Fragment Protection

---

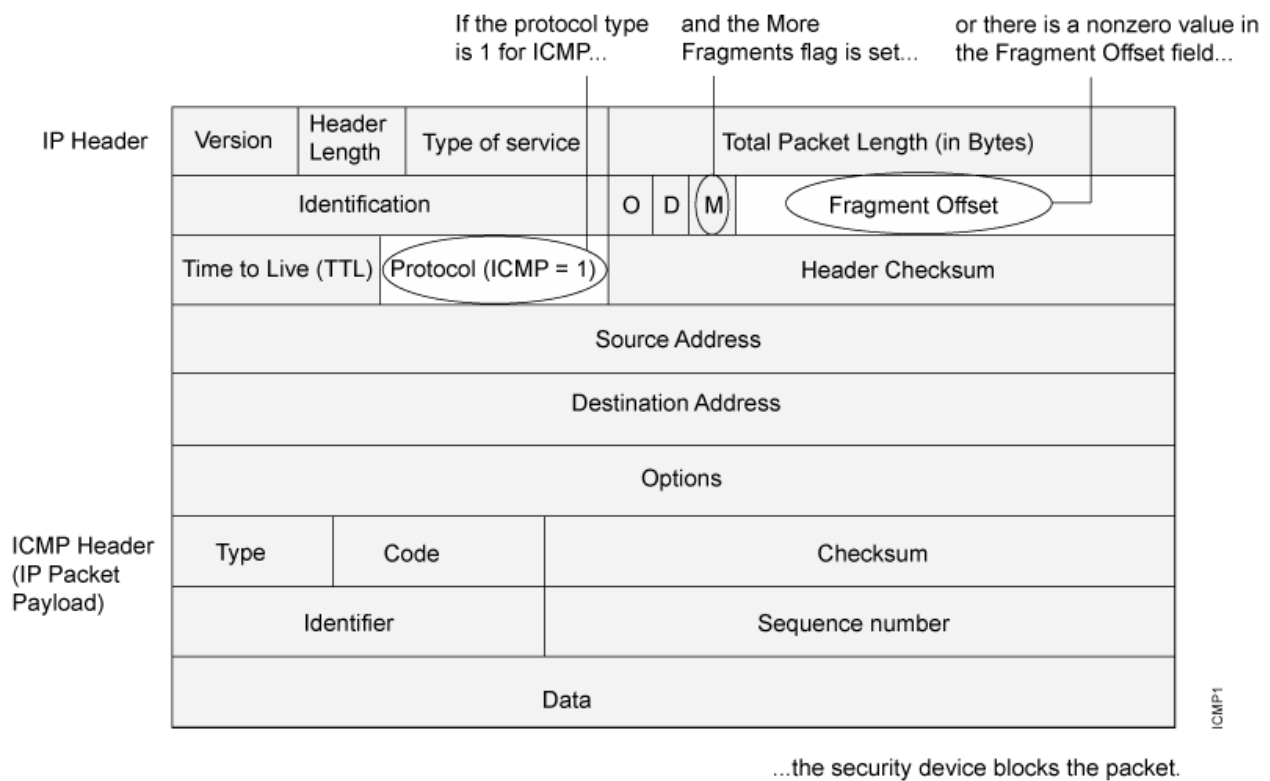
- [Understanding ICMP Fragment Protection on page 1052](#)
- [Example: Blocking Fragmented ICMP Packets on page 1052](#)

## Understanding ICMP Fragment Protection

Internet Control Message Protocol (ICMP) provides error reporting and network probe capabilities. Because ICMP packets contain very short messages, there is no legitimate reason for ICMP packets to be fragmented. If an ICMP packet is so large that it must be fragmented, something is amiss.

When you enable the ICMP fragment protection screen option, Junos OS blocks any ICMP packet that has the More Fragments flag set or that has an offset value indicated in the offset field. See Figure 72 on page 1052.

Figure 72: Blocking ICMP Fragments



**Related Documentation** • [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

### Example: Blocking Fragmented ICMP Packets

This example shows how to block fragmented ICMP packets.

---

## Requirements

Before you begin, Understand ICMP fragment protection. See “Suspicious Packet Attributes Overview” on page 1051.

---

## Overview

When you enable the ICMP fragment protection screen option, Junos OS blocks any ICMP packet that has the more fragments flag set or that has an offset value indicated in the offset field.

In this example, you configure the ICMP fragment screen to block fragmented ICMP packets originating from the zone1 security zone.

---

## Configuration

### Step-by-Step Procedure

To block fragmented ICMP packets:

1. Configure the screen.
 

```
[edit]
user@host# set security screen ids-option icmp-fragment icmp fragment
```
2. Configure a security zone.
 

```
[edit]
user@host# set security zones security-zone zone1 screen icmp-fragment
```
3. If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

---

## Verification

To verify the configuration is working properly, enter the **show security screen statistics zone zone-name** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

---

## Large ICMP Packet Protection

- [Understanding Large ICMP Packet Protection](#) on page 1053
- [Example: Blocking Large ICMP Packets](#) on page 1054

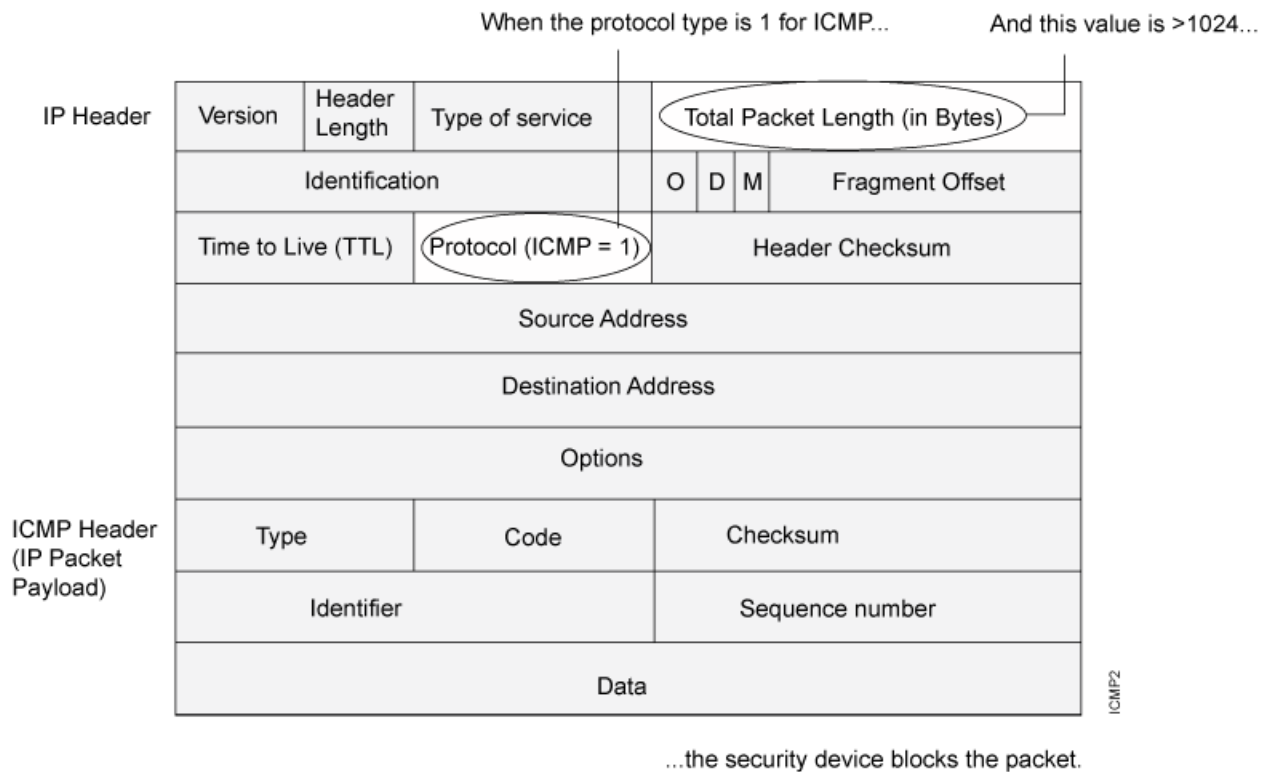
## Understanding Large ICMP Packet Protection

Internet Control Message Protocol (ICMP) provides error reporting and network probe capabilities. Because ICMP packets contain very short messages, there is no legitimate reason for large ICMP packets. If an ICMP packet is unusually large, something is amiss.

For example, the SRX 210 uses ICMP as a channel for transmitting covert messages. The presence of large ICMP packets might expose a compromised machine acting as a SRX

210 agent. It also might indicate some other kind of questionable activity. See Figure 73 on page 1054.

Figure 73: Blocking Large ICMP Packets



When you enable the large size ICMP packet protection screen option, Junos OS drops ICMP packets with a length greater than 1024 bytes.

**Related Documentation** • [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

### Example: Blocking Large ICMP Packets

This example shows how to block large ICMP packets.

#### Requirements

Before you begin, Understand large ICMP packet protection. See “Suspicious Packet Attributes Overview” on page 1051.

#### Overview

When you enable the large ICMP packet protection screen option, Junos OS drops ICMP packets that are larger than 1024 bytes.

In this example, you configure the ICMP large screen to block large ICMP packets originating from the zone1 security zone.

---

## Configuration

---

### Step-by-Step Procedure

To block large ICMP packets:

1. Configure the screen.

```
[edit]
user@host# set security screen ids-option icmp-large icmp large
```

2. Configure a security zone.

```
[edit]
user@host# set security zones security-zone zone1 screen icmp-large
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

---

## Verification

---

To verify the configuration is working properly, enter the **show security screen statistics zone *zone-name*** command.

---

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

---

## Bad IP Option Protection

---

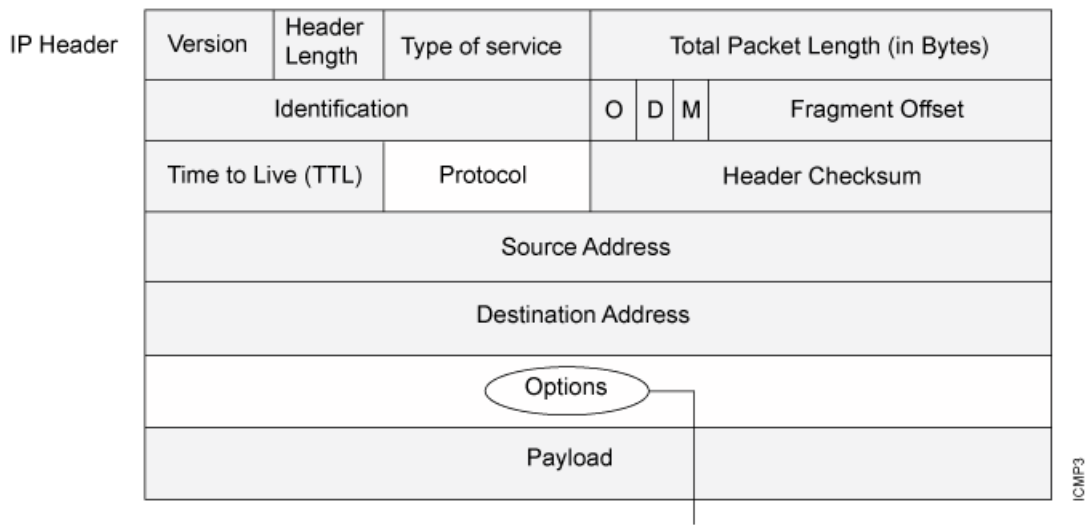
- [Understanding Bad IP Option Protection on page 1055](#)
- [Example: Blocking IP Packets with Incorrectly Formatted Options on page 1056](#)

### Understanding Bad IP Option Protection

The IP standard RFC 791, *Internet Protocol*, specifies a set of eight options that provide special routing controls, diagnostic tools, and security. Although the original, intended uses for these options served worthy ends, people have figured out ways to twist these options to accomplish less commendable objectives.

Either intentionally or accidentally, attackers sometimes configure IP options incorrectly, producing either incomplete or malformed fields. Regardless of the intentions of the person who crafted the packet, the incorrect formatting is anomalous and potentially harmful to the intended recipient. See Figure 74 on page 1056.

Figure 74: Incorrectly Formatted IP Options



If the IP options are incorrectly formatted, the security device records the event in the screen counters for the ingress interface.

When you enable the bad IP option protection screen option, Junos OS blocks packets when any IP option in the IP packet header is incorrectly formatted. Additionally, Junos OS records the event in the event log.

### Example: Blocking IP Packets with Incorrectly Formatted Options

This example shows how to block large ICMP packets with incorrectly formatted options.

#### Requirements

Before you begin, Understand bad IP option protection. See “Suspicious Packet Attributes Overview” on page 1051.

#### Overview

When you enable the bad IP option protection screen option, Junos OS blocks packets when any IP option in the IP packet header is incorrectly formatted. Additionally, Junos OS records the event in the event log.

In this example, you configure the IP bad option screen to block large ICMP packets originating from the zone1 security zone.

#### Configuration

##### Step-by-Step Procedure

To detect and block IP packets with incorrectly formatted IP options:

1. Configure the screen.

[edit]

```
user@host# set security screen ids-option ip-bad-option ip bad-option
```



NOTE: Currently this screen option is applicable only to IPv4.

2. Configure a security zone.

```
[edit]  
user@host# set security zones security-zone zone1 screen ip-bad-option
```

3. If you are done configuring the device, commit the configuration.

```
[edit]  
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security screen statistics zone zone-name** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

## Unknown Protocol Protection

- Understanding Unknown Protocol Protection on page 1057
- Example: Dropping Packets Using an Unknown Protocol on page 1058

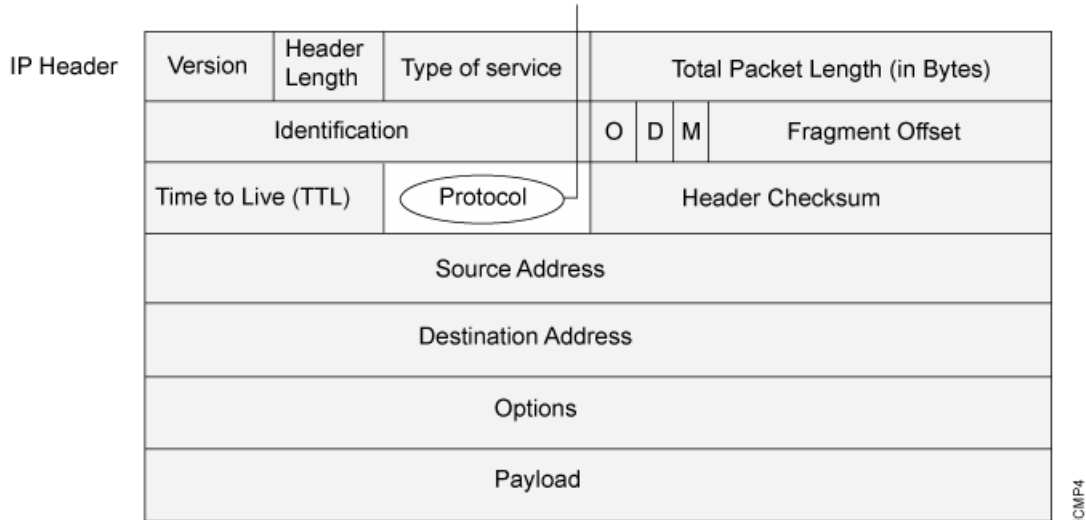
### Understanding Unknown Protocol Protection

Based on RFC 1700, the protocol types with ID numbers of 137 or greater are reserved and undefined at this time. Precisely because these protocols are undefined, there is no way to know in advance if a particular unknown protocol is benign or malicious.

Unless your network makes use of a nonstandard protocol with an ID number of 137 or greater, a cautious stance is to block such unknown elements from entering your protected network. See Figure 75 on page 1058.

Figure 75: Unknown Protocols

If the ID number of the protocol is 137 or greater, the security device blocks the packet.



If the IP options are incorrectly formatted, the security device records the event in the screen counters for the ingress interface.

When you enable the unknown protocol protection screen option, Junos OS drops packets when the protocol field contains a protocol ID number of 137 or greater by default.



**NOTE:** When you enable the unknown protocol protection screen option for IPv6 protocol, Junos OS drops packets when the protocol field contains a protocol ID number of 139 or greater by default.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

### Example: Dropping Packets Using an Unknown Protocol

This example shows how to drop packets using an unknown protocol.

#### Requirements

Before you begin, Understand unknown protocol protection. See “Suspicious Packet Attributes Overview” on page 1051.

#### Overview

When you enable the unknown protocol protection screen option, Junos OS drops packets when the protocol field contains a protocol ID number of 137 or greater by default.

In this example, you configure the unknown protocol screen to block packets with an unknown protocol originating from the zone1 security zone.



---

## Configuration

---

### Step-by-Step Procedure

To drop packets that use an unknown protocol:

1. Configure the unknown protocol screen.  

```
[edit]  
user@host# set security screen ids-option unknown-protocol ip unknown-protocol
```
2. Configure a security zone.  

```
[edit]  
user@host# set security zones security-zone zone1 screen unknown-protocol
```
3. If you are done configuring the device, commit the configuration.  

```
[edit]  
user@host# commit
```

## Verification

---

To verify the configuration is working properly, enter the **show security screen statistics zone *zone-name*** command.

---

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

---

## IP Packet Fragment Protection

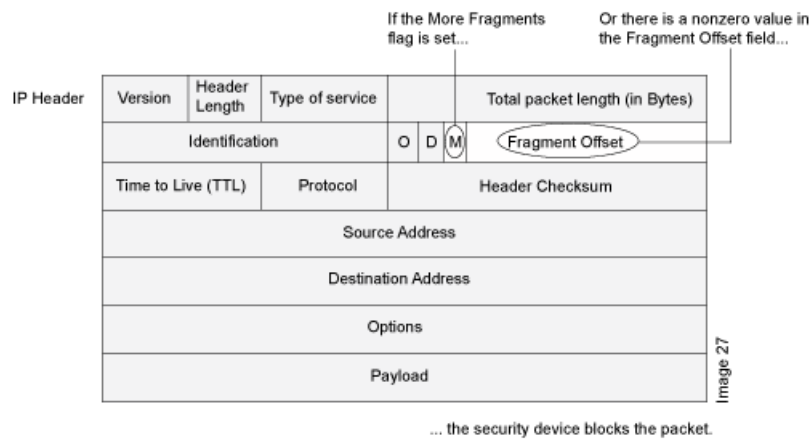
---

- [Understanding IP Packet Fragment Protection on page 1059](#)
- [Example: Dropping Fragmented IP Packets on page 1060](#)

### Understanding IP Packet Fragment Protection

As packets traverse different networks, it is sometimes necessary to break a packet into smaller pieces (fragments) based upon the maximum transmission unit (MTU) of each network. IP fragments might contain an attacker's attempt to exploit the vulnerabilities in the packet reassembly code of specific IP stack implementations. When the victim receives these packets, the results can range from processing the packets incorrectly to crashing the entire system. See Figure 76 on page 1060.

Figure 76: IP Packet Fragments



When you enable Junos OS to deny IP fragments on a security zone, it blocks all IP packet fragments that it receives at interfaces bound to that zone.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

## Example: Dropping Fragmented IP Packets

This example shows how to drop fragmented IP packets.

### Requirements

Before you begin, Understand IP packet fragment protection. See “Suspicious Packet Attributes Overview” on page 1051.

### Overview

When this feature is enabled, Junos OS denies IP fragments on a security zone and blocks all IP packet fragments that are received at interfaces bound to that zone.

In this example, you configure the block fragment screen to drop fragmented IP packets originating from the zone1 security zone.

### Configuration

**Step-by-Step Procedure**

To drop fragmented IP packets:

1. Configure the screen.
 

```
[edit]
user@host# set security screen ids-option block-frag ip block-frag
```
2. Configure the security zone.
 

```
[edit]
user@host# set security zones security-zone zone1 screen block-frag
```
3. If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

### Verification

---

To verify the configuration is working properly, enter the **show security screen statistics zone *zone-name*** command.

---

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

## SYN Fragment Protection

---

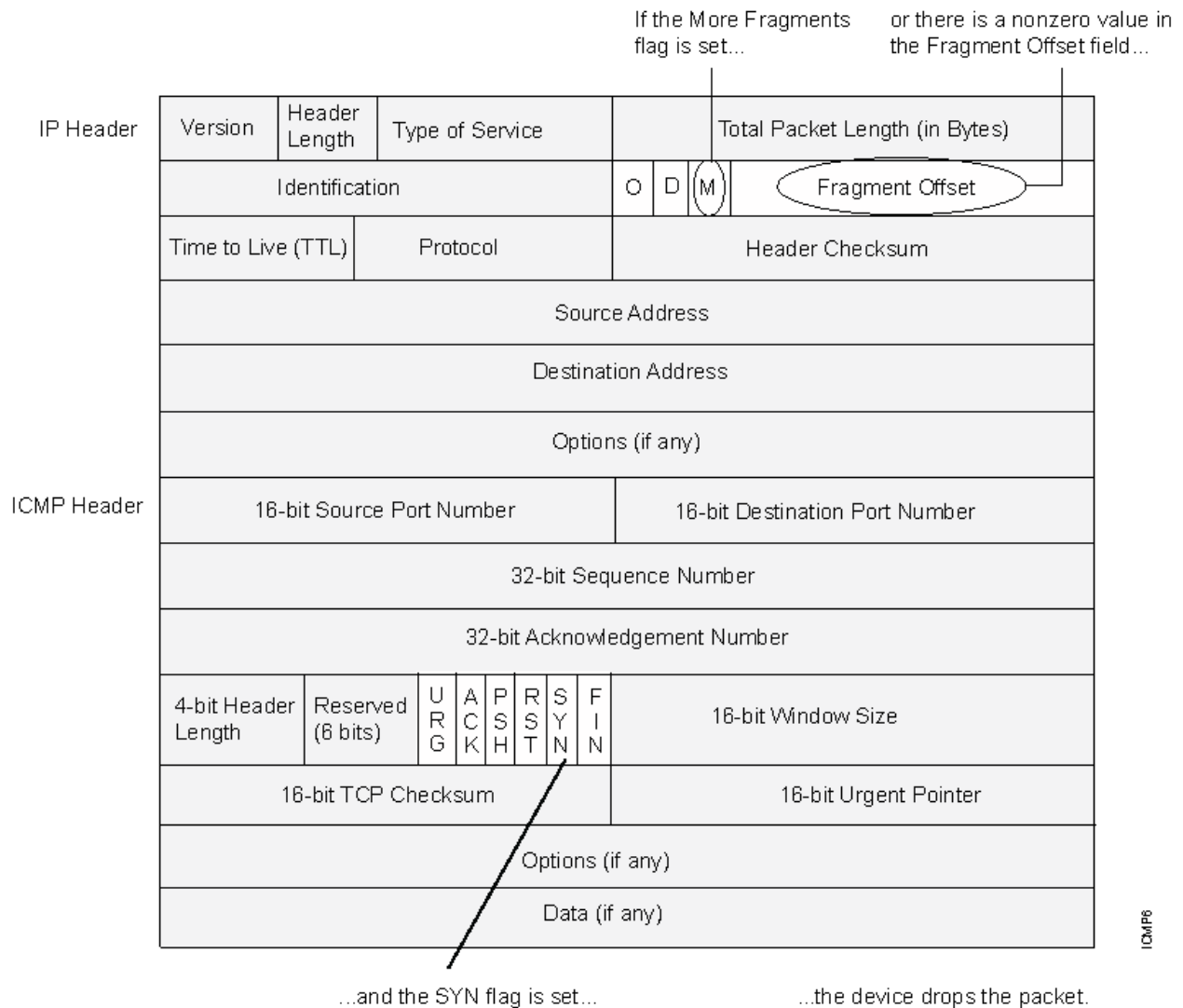
- [Understanding SYN Fragment Protection on page 1061](#)
- [Example: Dropping IP Packets Containing SYN Fragments on page 1062](#)

### Understanding SYN Fragment Protection

The IP encapsulates a TCP SYN segment in the IP packet that initiates a TCP connection. Because the purpose of this packet is to initiate a connection and invoke a SYN/ACK segment in response, the SYN segment typically does not contain any data. Because the IP packet is small, there is no legitimate reason for it to be fragmented.

A fragmented SYN packet is anomalous, and, as such, it is suspect. To be cautious, block such unknown elements from entering your protected network. See Figure 77 on page 1062.

Figure 77: SYN Fragments



When you enable the SYN fragment detection screen option, Junos OS detects packets when the IP header indicates that the packet has been fragmented and the SYN flag is set in the TCP header. Junos OS records the event in the screen counters list for the ingress interface.

**Related Documentation** • [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

### Example: Dropping IP Packets Containing SYN Fragments

This example shows how to drop IP packets containing SYN fragments.

## Requirements

---

Before you begin, Understand IP packet fragment protection. See “Suspicious Packet Attributes Overview” on page 1051.

## Overview

---

When you enable the SYN fragment detection screen option, Junos OS detects packets when the IP header indicates that the packet has been fragmented and the SYN flag is set in the TCP header. Also, Junos OS records the event in the screen counters list for the ingress interface.

In this example, you configure the SYN fragment screen to drop fragmented SYN packets originating from the zone1 security zone.

## Configuration

---

### Step-by-Step Procedure

To drop IP packets containing SYN fragments:

1. Configure the screen.  
[edit]  
user@host# set security screen ids-option syn-frag tcp syn-frag
2. Configure the security zone.  
[edit]  
user@host# set security zones security-zone zone1 screen syn-frag
3. If you are done configuring the device, commit the configuration.  
[edit]  
user@host# commit

## Verification

---

To verify the configuration is working properly, enter the **show security screen statistics zone zone-name** command.

---

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)



# Denial-of-Service Attacks

- DoS Attack Overview on page 1065
- Firewall DoS Attacks on page 1065
- Network DoS Attacks on page 1073
- OS-Specific DoS Attacks on page 1094

## DoS Attack Overview

---

The intent of a denial-of-service (DoS) attack is to overwhelm the targeted victim with a tremendous amount of bogus traffic so that the victim becomes so preoccupied processing the bogus traffic that legitimate traffic cannot be processed. The target can be the firewall, the network resources to which the firewall controls access, or the specific hardware platform or operating system of an individual host.

If a DoS attack originates from multiple source addresses, it is known as a distributed denial-of-service (DDoS) attack. Typically, the source address of a DoS attack is spoofed. The source addresses in a DDoS attack might be spoofed, or the actual addresses of compromised hosts might be used as “zombie agents” to launch the attack.

The device can defend itself and the resources it protects from DoS and DDoS attacks.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Firewall DoS Attacks Overview on page 1066
- Network DoS Attacks Overview on page 1073
- OS-Specific DoS Attacks Overview on page 1094

## Firewall DoS Attacks

---

- Firewall DoS Attacks Overview on page 1066
- Session Table Flood Attacks on page 1066
- SYN-ACK-ACK Proxy Flood Attacks on page 1072

## Firewall DoS Attacks Overview

The intent of a denial-of-service (DoS) attack is to overwhelm the targeted victim with a tremendous amount of bogus traffic so that the victim becomes so preoccupied processing the bogus traffic that legitimate traffic cannot be processed.

If attackers discover the presence of the Juniper Networks firewall, they might launch a DoS attack against it instead of the network behind it. A successful DoS attack against a firewall amounts to a successful DoS attack against the protected network in that it thwarts attempts of legitimate traffic to traverse the firewall.

An attacker might use session table floods and SYN-ACK-ACK proxy floods to fill up the session table of Junos OS and thereby produce a DoS.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [DoS Attack Overview on page 1065](#)
- [Network DoS Attacks Overview on page 1073](#)
- [OS-Specific DoS Attacks Overview on page 1094](#)
- [Understanding Session Table Flood Attacks on page 1066](#)

## Session Table Flood Attacks

- [Understanding Session Table Flood Attacks on page 1066](#)
- [Understanding Source-Based Session Limits on page 1067](#)
- [Example: Setting Source-Based Session Limits on page 1068](#)
- [Understanding Destination-Based Session Limits on page 1070](#)
- [Example: Setting Destination-Based Session Limits on page 1071](#)

### Understanding Session Table Flood Attacks

---

A successful DoS attack overwhelms its victim with such a massive barrage of false simulated traffic that it becomes unable to process legitimate connection requests. DoS attacks can take many forms—SYN flood, SYN-ACK-ACK flood, UDP flood, ICMP flood, and so on—but they all seek the same objective, which is to fill up their victim's session table.

When the session table is full, that host cannot create any new sessions and begins rejecting new connection requests. The source-based session limits screen option and the destination-based session limit screen option help mitigate such attacks.

### Related Documentation

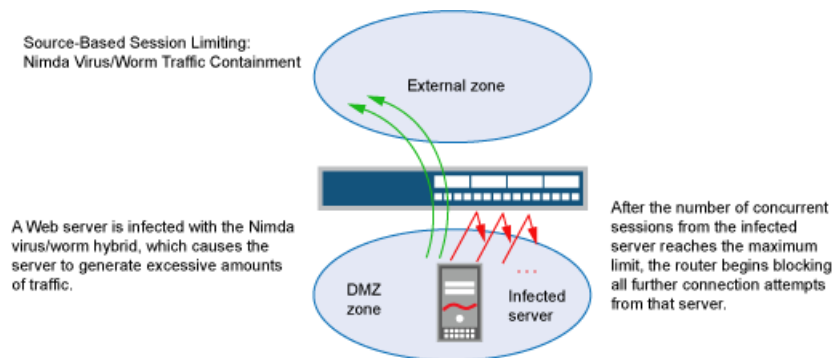
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [DoS Attack Overview on page 1065](#)
- [Understanding Source-Based Session Limits on page 1067](#)
- [Understanding SYN Flood Attacks on page 1074](#)
- [Understanding SYN-ACK-ACK Proxy Flood Attacks on page 1072](#)



### Understanding Source-Based Session Limits

In addition to limiting the number of concurrent sessions from the same source IP address, you can also limit the number of concurrent sessions to the same destination IP address. One benefit of setting a source-based session limit is that it can stem an attack such as the Nimda virus (which is actually both a virus and a worm) that infects a server and then begins generating massive amounts of traffic from that server. Because all the virus-generated traffic originates from the same IP address, a source-based session limit ensures that the firewall can curb such excessive amounts of traffic. See Figure 78 on page 1067.

Figure 78: Limiting Sessions Based on Source IP Address



Another benefit of source-based session limiting is that it can mitigate attempts to fill up the firewall's session table if all the connection attempts originate from the same source IP address.

Determining what constitutes an acceptable number of connection requests requires a period of observation and analysis to establish a baseline for typical traffic flows. You also need to consider the maximum number of concurrent sessions required to fill up the session table of the particular Juniper Networks platform you are using. To see the maximum number of sessions that your session table supports, use the CLI command **get session**, and then look at the first line in the output, which lists the number of current (allocated) sessions, the maximum number of sessions, and the number of failed session allocations:

```
alloc 420/max 128000, alloc failed 0
```

The default maximum for source-based session limits is 128 concurrent sessions, a value that might need adjustment to suit the needs of your network environment and the platform.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [DoS Attack Overview on page 1065](#)
- [Example: Setting Source-Based Session Limits on page 1068](#)

### Example: Setting Source-Based Session Limits

This example shows how to limit the amount of sessions based on source IP.

- Requirements on page 1068
- Overview on page 1068
- Configuration on page 1068
- Verification on page 1069

#### Requirements

No special configuration beyond device initialization is required before configuring this feature.

#### Overview

The following example shows how to limit the number of sessions that any one server in the DMZ and in zone\_a can initiate. Because the DMZ contains only web servers, none of which should initiate traffic, you set the source-session limit at the lowest possible value, which is one session. On the other hand, zone\_a contains personal computers, servers, printers, and so on, many of which do initiate traffic. For zone\_a, you set the source-session limit to a maximum of 80 concurrent sessions.

#### Configuration

#### CLI Quick Configuration

To quickly configure the source-based session limits, copy the following commands and paste them into the CLI:

```
[edit]
set security screen ids-option 1-limit-session limit-session source-ip-based 1
set security zones security-zone dmz screen 1-limit-session
set security screen ids-option 80-limit-session limit-session source-ip-based 80
set security zones security-zone zone_a screen 80-limit-session
```

#### Step-by-Step Procedure

To configure the source-based session limits:

1. Specify the number of concurrent sessions based on source IP for the DMZ zone.

```
[edit]
user@host# set security screen ids-option 1-limit-session limit-session
source-ip-based 1
```

2. Set the security zone for the DMZ to the configuration limit.

```
[edit]
user@host# set security zones security-zone dmz screen 1-limit-session
```

3. Specify the number of concurrent sessions based on source IP for the zone\_a zone.

```
[edit]
user@host# set security screen ids-option 80-limit-session limit-session
source-ip-based 80
```

4. Set the security zone for zone\_a to the configuration limit.

```
[edit]
user@host# set security zones security-zone zone_a screen 80-limit-session
```

**Results** From configuration mode, confirm your configuration by entering the **show security screen ids-option 1-limit-session**, **show security screen ids-option 1-limit-session**, and **show security zones** commands in operational mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show security** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
user@host# show security
...
ids-option 1-limit-session {
  limit-session {
    source-ip-based 1;
  }
}
ids-option 80-limit-session {
  limit-session {
    source-ip-based 80;
  }
}
...
security-zone dmz {
  screen 1-limit-session;
}
security-zone zone_a {
  screen 80-limit-session;
}
...
```

If you are done configuring the device, enter **commit** from configuration mode.

#### **Verification**

To confirm that the configuration is working properly, perform this task:

- Verifying Source-Based Session Limits on page 1069

#### **Verifying Source-Based Session Limits**

**Purpose** Verify source-based session limits.

**Action** From operational mode, enter the **show security** command.

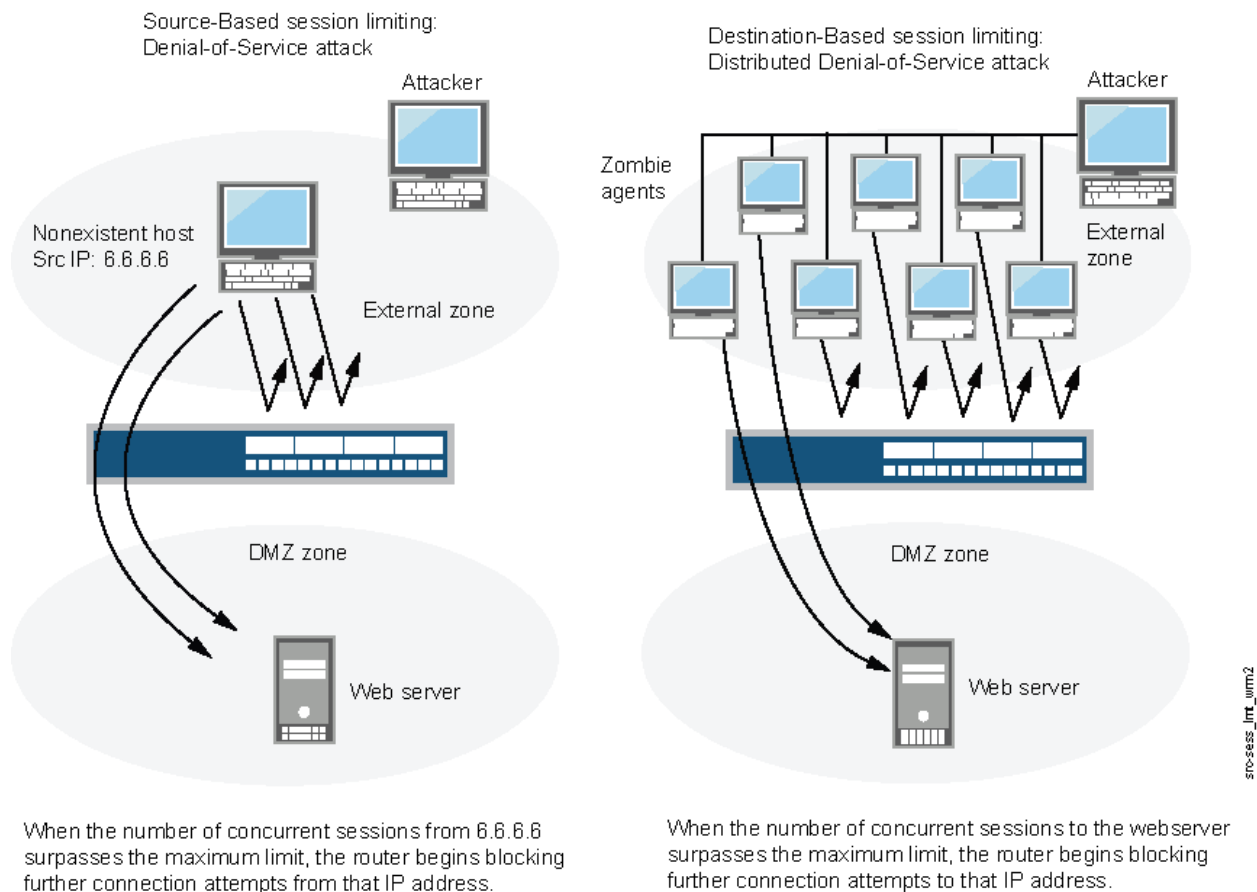
#### **Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Session Table Flood Attacks on page 1066
- Understanding Source-Based Session Limits on page 1067
- Example: Setting Destination-Based Session Limits on page 1071

## Understanding Destination-Based Session Limits

In addition to limiting the number of concurrent sessions from the same source IP address, you can also limit the number of concurrent sessions to the same destination IP address. A wily attacker can launch a distributed denial-of-service (DDoS) attack. In a DDoS attack, the malicious traffic can come from hundreds of hosts, known as “zombie agents,” that are surreptitiously under the control of an attacker. In addition to the SYN, UDP, and ICMP flood detection and prevention screen options, setting a destination-based session limit can ensure that Junos OS allows only an acceptable number of concurrent connection requests—no matter what the source—to reach any one host. See Figure 79 on page 1070.

**Figure 79: Distributed DOS Attack**



The default maximum for destination-based session limits is 128 concurrent sessions, a value that might need adjustment to suit the needs of your network environment and the platform.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [DoS Attack Overview](#) on page 1065
- [Example: Setting Destination-Based Session Limits](#) on page 1071
- [Understanding Source-Based Session Limits](#) on page 1067

### Example: Setting Destination-Based Session Limits

This example shows how to set the destination-based session limits.

- Requirements on page 1071
- Overview on page 1071
- Configuration on page 1071
- Verification on page 1071

#### Requirements

No special configuration beyond device initialization is required before configuring this feature.

#### Overview

In this example, you limit the amount of traffic to a webserver at 1.2.2.5. The server is in the DMZ. The example assumes that after observing the traffic flow from the external zone to this server for a month, you have determined that the average number of concurrent sessions it receives is 2000. Also, you set the new session limit at 2000 concurrent sessions. Although traffic spikes might sometimes exceed that limit, the example assumes that you are opting for firewall security over occasional server inaccessibility.

#### Configuration

#### Step-by-Step Procedure

To set the destination-based session limits:

1. Specify the number of concurrent sessions.

[edit]

```
user@host# set security screen ids-option 2000-limit-session limit-session
destination-ip-based 2000
```

2. Set the security zone for the external zone.

[edit]

```
user@host# set security zones security-zone external_zone screen
2000-limit-session
```

3. If you are done configuring the device, commit the configuration.

[edit]

```
user@host# commit
```

#### Verification

To verify the configuration is working properly, enter the **show security screen ids-option 2000-limit-session** and **show security zones** commands in operational mode.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Destination-Based Session Limits on page 1070
- DoS Attack Overview on page 1065

## SYN-ACK-ACK Proxy Flood Attacks

- Understanding SYN-ACK-ACK Proxy Flood Attacks on page 1072
- Example: Protecting Against a SYN-ACK-ACK Proxy Flood Attack on page 1072

### Understanding SYN-ACK-ACK Proxy Flood Attacks

---

When an authentication user initiates a Telnet or an FTP connection, the user sends a SYN segment to the Telnet or FTP server. Junos OS intercepts the SYN segment, creates an entry in its session table, and proxies a SYN-ACK segment to the user. The user then replies with an ACK segment. At this point, the initial three-way handshake is complete. Junos OS sends a login prompt to the user. If the user, with malicious intent, does not log in but instead continues initiating SYN-ACK-ACK sessions, the firewall session table can fill up to the point where the device begins rejecting legitimate connection requests.

To prevent such an attack, you can enable the SYN-ACK-ACK proxy protection screen option. After the number of connections from the same IP address reaches the SYN-ACK-ACK proxy threshold, Junos OS rejects further connection requests from that IP address. By default, the threshold is 512 connections from any single IP address. You can change this threshold (to any number between 1 and 250,000) to better suit the requirements of your network environment.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- DoS Attack Overview on page 1065
- Example: Protecting Against a SYN-ACK-ACK Proxy Flood Attack on page 1072

### Example: Protecting Against a SYN-ACK-ACK Proxy Flood Attack

---

This example shows how to protect against a SYN-ACK-ACK proxy flood attack.

- Requirements on page 1072
- Overview on page 1072
- Configuration on page 1072
- Verification on page 1073

#### **Requirements**

No special configuration beyond device initialization is required before configuring this feature.

#### **Overview**

In this example, you enable protection against a SYN-ACK-ACK proxy flood. The value unit is connections per source address. The default value is 512 connections from any single address.

#### **Configuration**

#### Step-by-Step Procedure

To protect against a SYN-ACK-ACK proxy flood attack:

1. Specify the source session limits.

[edit]

```
user@host# set security screen ids-option 1000-syn-ack-ack-proxy tcp
syn-ack-ack-proxy threshold 1000
```

2. Set the security zone for zone screen.

```
[edit]
user@host# set security zones security-zone zone screen 1000-syn-ack-ack-proxy
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security screen ids-option 1000-syn-ack-ack-proxy** and **show security zones** commands in operational mode.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding SYN-ACK-ACK Proxy Flood Attacks on page 1072](#)
- [DoS Attack Overview on page 1065](#)

## Network DoS Attacks

- [Network DoS Attacks Overview on page 1073](#)
- [SYN Flood Attacks on page 1074](#)
- [SYN Cookie Protection on page 1085](#)
- [ICMP Flood Protection on page 1088](#)
- [UDP Flood Attacks on page 1090](#)
- [Land Attacks on page 1092](#)

### Network DoS Attacks Overview

A denial-of-service (DoS) attack directed against one or more network resources floods the target with an overwhelming number of SYN, ICMP, or UDP packets or with an overwhelming number of SYN fragments.

Depending on the attackers' purpose and the extent and success of previous intelligence gathering efforts, the attackers might single out a specific host, such as a device or server or they might aim at random hosts across the targeted network. Either approach has the potential of upsetting service to a single host or to the entire network, depending on how critical the role of the victim is to the rest of the network.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [DoS Attack Overview on page 1065](#)
- [Firewall DoS Attacks Overview on page 1066](#)
- [OS-Specific DoS Attacks Overview on page 1094](#)

## SYN Flood Attacks

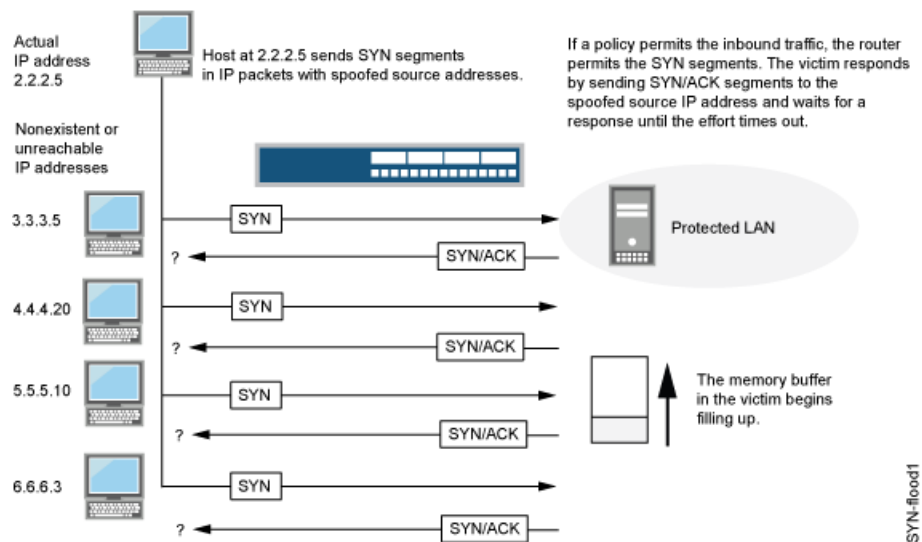
- Understanding SYN Flood Attacks on page 1074
- Example: Enabling SYN Flood Protection on page 1078
- Configuring SYN Flood Protection Options (CLI Procedure) on page 1079
- Example: Enabling SYN Flood Protection for Web servers in the DMZ on page 1079

### Understanding SYN Flood Attacks

A SYN flood occurs when a host becomes so overwhelmed by SYN segments initiating incomplete connection requests that it can no longer process legitimate connection requests.

Two hosts establish a TCP connection with a triple exchange of packets known as a *three-way handshake*: A sends a SYN segment to B; B responds with a SYN/ACK segment; and A responds with an ACK segment. A SYN flood attack inundates a site with SYN segments containing forged (spoofed) IP source addresses with nonexistent or unreachable addresses. B responds with SYN/ACK segments to these addresses and then waits for responding ACK segments. Because the SYN/ACK segments are sent to nonexistent or unreachable IP addresses, they never elicit responses and eventually time out. See Figure 80 on page 1074.

Figure 80: SYN Flood Attack



By flooding a host with incomplete TCP connections, the attacker eventually fills the memory buffer of the victim. Once this buffer is full, the host can no longer process new TCP connection requests. The flood might even damage the victim's operating system. Either way, the attack disables the victim and its normal operations.

This topic includes the following sections:

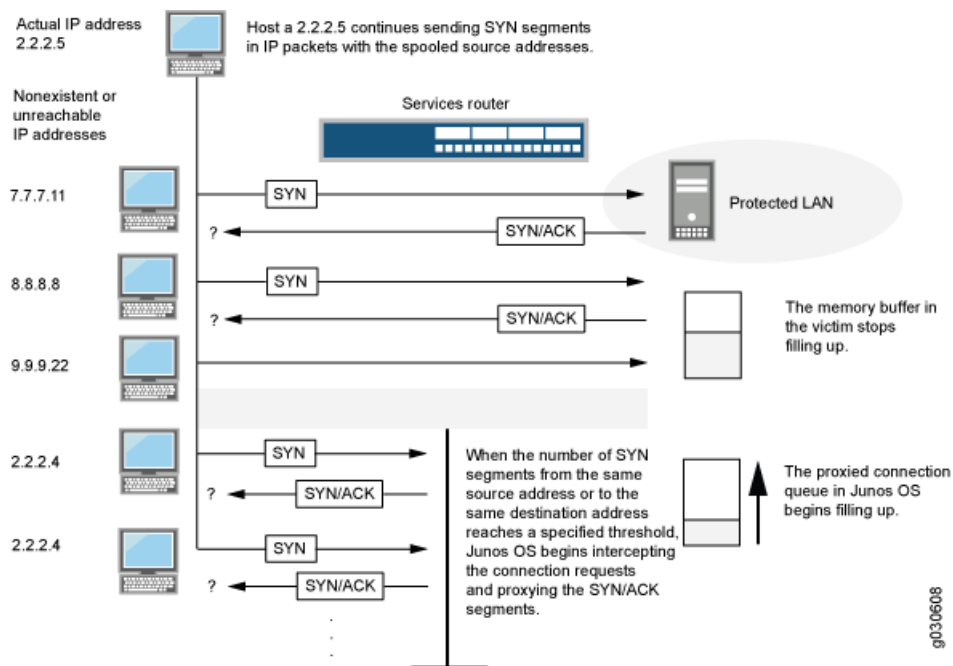
- SYN Flood Protection on page 1075
- SYN Flood Options on page 1076



### SYN Flood Protection

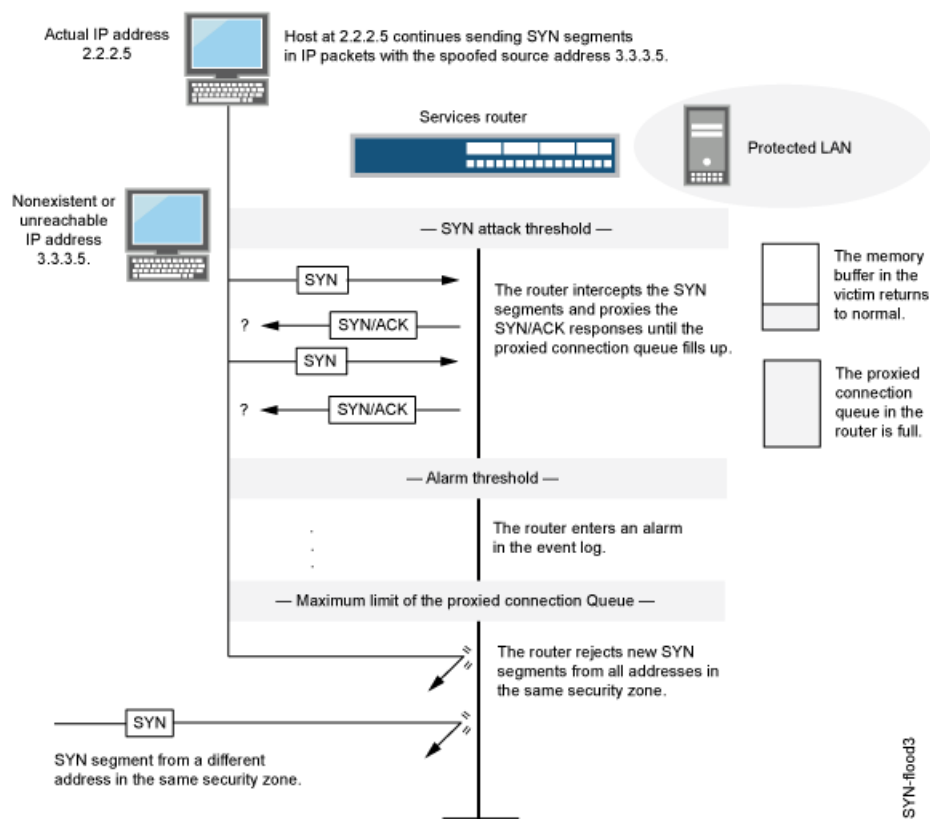
Junos OS can impose a limit on the number of SYN segments permitted to pass through the firewall per second. You can base the attack threshold on the destination address and port, the destination address only, or the source address only. When the number of SYN segments per second exceeds one of these thresholds, Junos OS starts proxying incoming SYN segments, replying with SYN/ACK segments and storing the incomplete connection requests in a connection queue. The incomplete connection requests remain in the queue until the connection is completed or the request times out. In Figure 81 on page 1075, the SYN attack threshold has passed, and Junos OS has started proxying SYN segments.

Figure 81: Proxying SYN Segments



In Figure 82 on page 1076, the proxied connection queue has completely filled up, and Junos OS is rejecting new incoming SYN segments. This action shields hosts on the protected network from the bombardment of incomplete three-way handshakes.

Figure 82: Rejecting New SYN Segments



The device starts receiving new SYN packets when the proxy queue drops below the maximum limit.



**NOTE:** The procedure of proxying incomplete SYN connections above a set threshold pertains only to traffic permitted by existing policies. Any traffic for which a policy does not exist is automatically dropped.

### SYN Flood Options

You can set the following parameters for proxying uncompleted TCP connection requests:

- **Attack Threshold**—This option allows you to set the number of SYN segments (that is, TCP segments with the SYN flag set) to the same destination address and port number per second required to activate the SYN proxying mechanism. Although you can set the threshold to any number, you need to know the normal traffic patterns at your site to set an appropriate threshold for it. For example, if it is an e-business site that normally gets 20,000 SYN segments per second, you might want to set the threshold to 30,000 per second. If a smaller site normally gets 20 SYN segments per second, you might consider setting the threshold to 40.
- **Alarm Threshold**—This option allows you to set the number of proxied, half-complete TCP connection requests per second after which Junos OS enters an alarm in the event log. The value you set for an alarm threshold triggers an alarm when the number of

proxied, half-completed connection requests to the same destination address and port number per second exceeds that value. For example, if you set the SYN attack threshold at 2000 SYN segments per second and the alarm at 1000, then a total of 3001 SYN segments to the same destination address and port number per second is required to trigger an alarm entry in the log. More precisely:

1. The firewall passes the first 2000 SYN segments per second that meet policy requirements.
2. The firewall proxies the next 1000 SYN segments in the same second.
3. The 1001st proxied connection request (or 3001st connection request in that second) triggers the alarm.

For each SYN segment to the same destination address and port number in excess of the alarm threshold, the attack detection module generates a message. At the end of the second, the logging module compresses all similar messages into a single log entry that indicates how many SYN segments to the same destination address and port number arrived after exceeding the alarm threshold. If the attack persists beyond the first second, the event log enters an alarm every second until the attack stops.

- **Source Threshold**—This option allows you to specify the number of SYN segments received per second from a single source IP address—regardless of the destination IP address and port number—before Junos OS begins dropping connection requests from that source.

Tracking a SYN flood by source address uses different detection parameters from tracking a SYN flood by destination address and destination port number. When you set a SYN attack threshold and a source threshold, you put both the basic SYN flood protection mechanism and the source-based SYN flood tracking mechanism in effect.

- **Destination Threshold**—This option allows you to specify the number of SYN segments received per second for a single destination IP address before Junos OS begins dropping connection requests to that destination. If a protected host runs multiple services, you might want to set a threshold based on destination IP address only—regardless of the destination port number.

When you set a SYN attack threshold and a destination threshold, you put both the basic SYN flood protection mechanism and the destination-based SYN flood tracking mechanism in effect.

Tracking a SYN flood by destination address uses different detection parameters from tracking a SYN flood by destination address and destination port number. Consider the following case where Junos OS has policies permitting FTP requests (port 21) and HTTP requests (port 80) to the same server. If the SYN flood attack threshold is 1000 packets per second (pps) and an attacker sends 999 FTP packets and 999 HTTP pps, neither set of packets (where a set is defined as having the same destination address and port number) activates the SYN proxying mechanism. The basic SYN flood attack mechanism tracks destination address and port number, and neither set exceeds the attack threshold of 1000 pps. However, if the destination threshold is 1000 pps, Junos

OS treats both FTP and HTTP packets with the same destination address as members of a single set and rejects the 1001st packet—FTP or HTTP—to that destination.

- **Timeout**—This option allows you to set the maximum length of time before a half-completed connection is dropped from the queue. The default is 20 seconds, and you can set the timeout from 0–50 seconds. You might try decreasing the timeout value to a shorter length until you begin to see any dropped connections during normal traffic conditions. Twenty seconds is a very conservative timeout for a three-way handshake ACK response.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Enabling SYN Flood Protection on page 1078](#)
- [Configuring SYN Flood Protection Options \(CLI Procedure\) on page 1079](#)
- [Example: Enabling SYN Flood Protection for Webservers in the DMZ on page 1079](#)

#### Example: Enabling SYN Flood Protection

This example shows how to enable SYN flood protection.

- [Requirements on page 1078](#)
- [Overview on page 1078](#)
- [Configuration on page 1078](#)
- [Verification on page 1079](#)

#### Requirements

No special configuration beyond device initialization is required before configuring this feature.

#### Overview

In this example, you enable the zone-syn-flood protection screen option and set the timeout value to 20. You also specify the zone where the flood might originate.

#### Configuration

#### Step-by-Step Procedure

To enable SYN flood protection:

1. Specify the screen object name.

```
[edit]
user@host# set security screen ids-option zone-syn-flood tcp syn-flood timeout
20
```

2. Set the security zone for the zone screen.

```
[edit]
user@host# set security zones security-zone zone screen zone-syn-flood
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

**Verification**

To verify the configuration is working properly, enter the **show security screen ids-option zone-syn-flood** and **show security zones** commands in operational mode.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding SYN Flood Attacks on page 1074
- Configuring SYN Flood Protection Options (CLI Procedure) on page 1079
- Example: Enabling SYN Flood Protection for Webserver in the DMZ on page 1079

**Configuring SYN Flood Protection Options (CLI Procedure)**

To set syn-flood parameters, use the following commands:

```
user@host# set security screen zone-syn-flood tcp syn-flood attack-threshold number
user@host# set security screen zone-syn-flood tcp syn-flood alarm-threshold number
user@host# set security screen zone-syn-flood tcp syn-flood source-threshold number
user@host# set security screen zone-syn-flood tcp syn-flood destination-threshold number
user@host# set security screen zone-syn-flood tcp syn-flood timeout number
```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding SYN Flood Attacks on page 1074
- Example: Enabling SYN Flood Protection on page 1078
- Example: Enabling SYN Flood Protection for Webserver in the DMZ on page 1079

**Example: Enabling SYN Flood Protection for Webserver in the DMZ**

This example shows how to enable SYN flood protection for webserver in the DMZ.

- Requirements on page 1079
- Overview on page 1079
- Configuration on page 1082
- Verification on page 1085

**Requirements**

No special configuration beyond device initialization is required before configuring this feature.

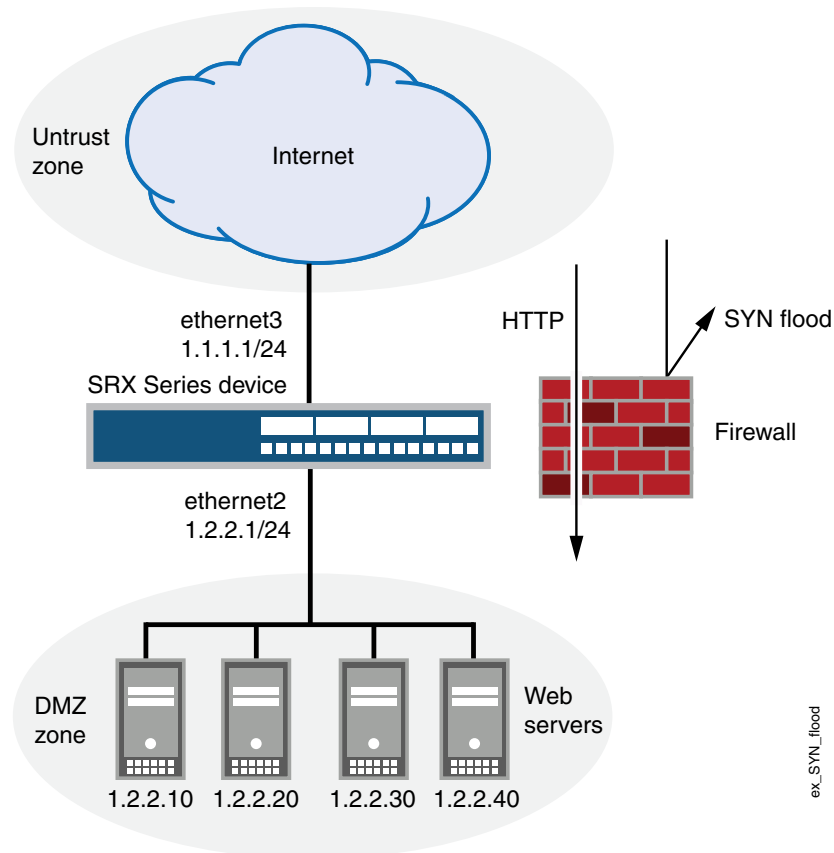
**Overview**

This example shows how to protect four webserver in the DMZ from SYN flood attacks originating in the external zone, by enabling the SYN flood protection screen option for the external zone.



**NOTE:** We recommend that you augment the SYN flood protection that Junos OS provides with device-level SYN flood protection on each webserver. In this example, the web servers are running UNIX, which also provides some SYN flood defenses, such as adjusting the length of the connection request queue and changing the timeout period for incomplete connection requests.

Figure 83: Device-Level SYN Flood Protection



To configure the SYN flood protection parameters with appropriate values for your network, you must first establish a baseline of typical traffic flows. For example, for one week, you run a sniffer on ethernet3—the interface bound to zone\_external—to monitor the number of new TCP connection requests arriving every second for the four web servers in the DMZ. Your analysis of the data accumulated from one week of monitoring produces the following statistics:

- Average number of new connection requests per server: 250 per second
- Average peak number of new connection requests per server: 500 per second



**NOTE:** A sniffer is a network-analyzing device that captures packets on the network segment to which you attach it. Most sniffers allow you to define filters to collect only the type of traffic that interests you. Later, you can view and evaluate the accumulated information. In this example, you want the sniffer to collect all TCP packets with the SYN flag set arriving at ethernet3 and destined for one of the four webservers in the DMZ. You might want to continue running the sniffer at regular intervals to see whether there are traffic patterns based on the time of day, day of the week, time of the month, or season of the year. For example, in some organizations, traffic might increase dramatically during a critical event. Significant changes probably warrant adjusting the various thresholds.

Based on this information, you set the following SYN flood protection parameters for zone\_external as shown in Table 105 on page 1081.

**Table 105: SYN Flood Protection Parameters**

Parameter	Value	Reason for Each Value
Attack threshold	625 pps	This is 25% higher than the average peak number of new connection requests per second per server, which is unusual for this network environment. When the number of SYN packets per second for any one of the four webservers exceeds this number, the device begins proxying new connection requests to that server. (In other words, beginning with the 626th SYN packet to the same destination address and port number in one second, the device begins proxying connection requests to that address and port number.)
Alarm threshold	250 pps	When the device proxies 251 new connection requests in one second, it makes an alarm entry in the event log. By setting the alarm threshold somewhat higher than the attack threshold, you can avoid alarm entries for traffic spikes that only slightly exceed the attack threshold.
Source threshold	25 pps	When you set a source threshold, the device tracks the source IP address of SYN packets, regardless of the destination address and port number. (Note that this source-based tracking is separate from the tracking of SYN packets based on destination address and destination port number, which constitutes the basic SYN flood protection mechanism.)  In the one week of monitoring activity, you observed that no more than 1/25 of new connection requests for all servers came from any one source within a one-second interval. Therefore, connection requests exceeding this threshold are unusual and provide sufficient cause for the device to execute its proxying mechanism. (Note that 25 pps is 1/25 of the attack threshold, which is 625 pps.)  If the device tracks 25 SYN packets from the same source IP address, then, beginning with the 26th packet, it rejects all further SYN packets from that source for the remainder of that second and for the next second as well.
Destination threshold	0 pps	When you set a destination threshold, the device runs a separate tracking of only the destination IP address, regardless of the destination port number. Because the four webservers receive only HTTP traffic (destination port 80)—no traffic to any other destination port number reaches them—setting another destination threshold offers no additional advantage.

Table 105: SYN Flood Protection Parameters (*continued*)

Parameter	Value	Reason for Each Value
Timeout	20 seconds	The default value of 20 seconds is a reasonable length of time to hold incomplete connection requests.

**Configuration**

**CLI Quick Configuration** To quickly configure SYN flood protection for web servers in the DMZ, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet address 1.2.2.1/24
set interfaces fe-1/0/0 unit 0 family inet address 1.1.1.1/24
set security zones security-zone zone_dmz interfaces ge-0/0/0.0
set security zones security-zone zone_external interfaces fe-1/0/0.0
set security zones security-zone zone_dmz address-book address ws1 1.2.2.10/32
set security zones security-zone zone_dmz address-book address ws2 1.2.2.20/32
set security zones security-zone zone_dmz address-book address ws3 1.2.2.30/32
set security zones security-zone zone_dmz address-book address ws4 1.2.2.40/32
set security zones security-zone zone_dmz address-book address-set web_servers address
ws1
set security zones security-zone zone_dmz address-book address-set web_servers address
ws2
set security zones security-zone zone_dmz address-book address-set web_servers address
ws3
set security zones security-zone zone_dmz address-book address-set web_servers address
ws4
set security policies from-zone zone_external to-zone zone_dmz policy id_1 match
source-address any destination-address web_servers application junos-http
set security policies from-zone zone_external to-zone zone_dmz policy id_1 then permit
set security screen ids-option zone_external-syn-flood tcp syn-flood alarm-threshold
250 attack-threshold 625 source-threshold 25 timeout 20
set security zones security-zone zone_external screen zone_external-syn-flood
```

**Step-by-Step Procedure** To configure SYN flood protection parameters:

1. Set interfaces.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 1.2.2.1/24
user@host# set interfaces fe-1/0/0 unit 0 family inet address 1.1.1.1/24
user@host# set security zones security-zone zone_dmz interfaces ge-0/0/0.0
user@host# set security zones security-zone zone_external interfaces fe-1/0/0.0
```

2. Define addresses.

```
[edit]
user@host# set security zones security-zone zone_dmz address-book address ws1
1.2.2.10/32
user@host# set security zones security-zone zone_dmz address-book address ws2
1.2.2.20/32
user@host# set security zones security-zone zone_dmz address-book address ws3
1.2.2.30/32
```



```

user@host# set security zones security-zone zone_dmz address-book address ws4
1.2.2.40/32
user@host# set security zones security-zone zone_dmz address-book address-set
web_servers address ws1
user@host# set security zones security-zone zone_dmz address-book address-set
web_servers address ws2
user@host# set security zones security-zone zone_dmz address-book address-set
web_servers address ws3
user@host# set security zones security-zone zone_dmz address-book address-set
web_servers address ws4

```

3. Configure the policy.

```

[edit]
user@host# set security policies from-zone zone_external to-zone zone_dmz policy
id_1 match source-address any
user@host# set security policies from-zone zone_external to-zone zone_dmz policy
id_1 match destination-address web_servers
user@host# set security policies from-zone zone_external to-zone zone_dmz policy
id_1 match application junos-http
user@host# set security policies from-zone zone_external to-zone zone_dmz policy
id_1 then permit

```

4. Configure the screen options.

```

[edit]
user@host# set security screen ids-option zone_external-syn-flood tcp syn-flood
alarm-threshold 250
user@host# set security screen ids-option zone_external-syn-flood tcp syn-flood
attack-threshold 625
user@host# set security screen ids-option zone_external-syn-flood tcp syn-flood
source-threshold 25
user@host# set security screen ids-option zone_external-syn-flood tcp syn-flood
timeout 20
user@host# set security zones security-zone zone_external screen
zone_external-syn-flood

```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show security zones**, **show security policies**, and **show security screen** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```

[edit]
user@host# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 1.2.2.1/24;
    }
  }
}
fe-1/0/0 {

```

```
unit 0 {
  family inet {
    address 1.1.1.1/24;
  }
}
...
[edit]
user@host# show security zones
...
security-zone zone_dmz {
address-book {
address ws1 1.2.2.10/32;
  address ws2 1.2.2.20/32;
  address ws3 1.2.2.30/32;
  address ws4 1.2.2.40/32;
address-set web_servers {
  address ws1;
  address ws2;
  address ws3;
  address ws4;
}
}
interfaces {
  ge-0/0/0.0;
}
}
security-zone zone_external {
  screen zone_external-syn-flood;
  interfaces {
    fe-1/0/0.0;
  }
}
[edit]
user@host# show security policies
from-zone zone_external to-zone zone_dmz {
  policy id_1 {
  match {
source-address any;
  destination-address web_servers;
  application junos-http;
  }
then {
  permit;
  }
}
}
[edit]
user@host# show security screen
...
ids-option zone_external-syn-flood {
  tcp {
syn-flood {
alarm-threshold 250;
  attack-threshold 625;
  source-threshold 25;
}
```

```

    timeout 20;
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### **Verification**

To confirm that the configuration is working properly, perform this task:

- Verifying SYN Flood Protection for Webservers in the DMZ on page 1085

### **Verifying SYN Flood Protection for Webservers in the DMZ**

**Purpose** Verify SYN flood protection for webservers in the DMZ.

**Action** From operational mode, enter the **show interfaces**, **show security zones**, **show security policies**, and **show security screen ids-option zone\_external-syn-flood** commands.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding SYN Flood Attacks on page 1074
- Example: Enabling SYN Flood Protection on page 1078
- Configuring SYN Flood Protection Options (CLI Procedure) on page 1079

## **SYN Cookie Protection**

- Understanding SYN Cookie Protection on page 1085
- Example: Enabling SYN Cookie Protection on page 1087

### **Understanding SYN Cookie Protection**

SYN cookie is a stateless SYN proxy mechanism you can use in conjunction with other defenses against a SYN flood attack.

As with traditional SYN proxying, SYN cookie is activated when the SYN flood attack threshold is exceeded. However, because SYN cookie is stateless, it does not set up a session or policy and route lookups upon receipt of a SYN segment, and it maintains no connection request queues. This dramatically reduces CPU and memory usage and is the primary advantage of using SYN cookie over the traditional SYN proxying mechanism.

When SYN cookie is enabled on Junos OS and becomes the TCP-negotiating proxy for the destination server, it replies to each incoming SYN segment with a SYN/ACK containing an encrypted cookie as its initial sequence number (ISN). The cookie is an MD5 hash of the original source address and port number, destination address and port number, and ISN from the original SYN packet. After sending the cookie, Junos OS drops the original SYN packet and deletes the calculated cookie from memory. If there is no response to the packet containing the cookie, the attack is noted as an active SYN attack and is effectively stopped.

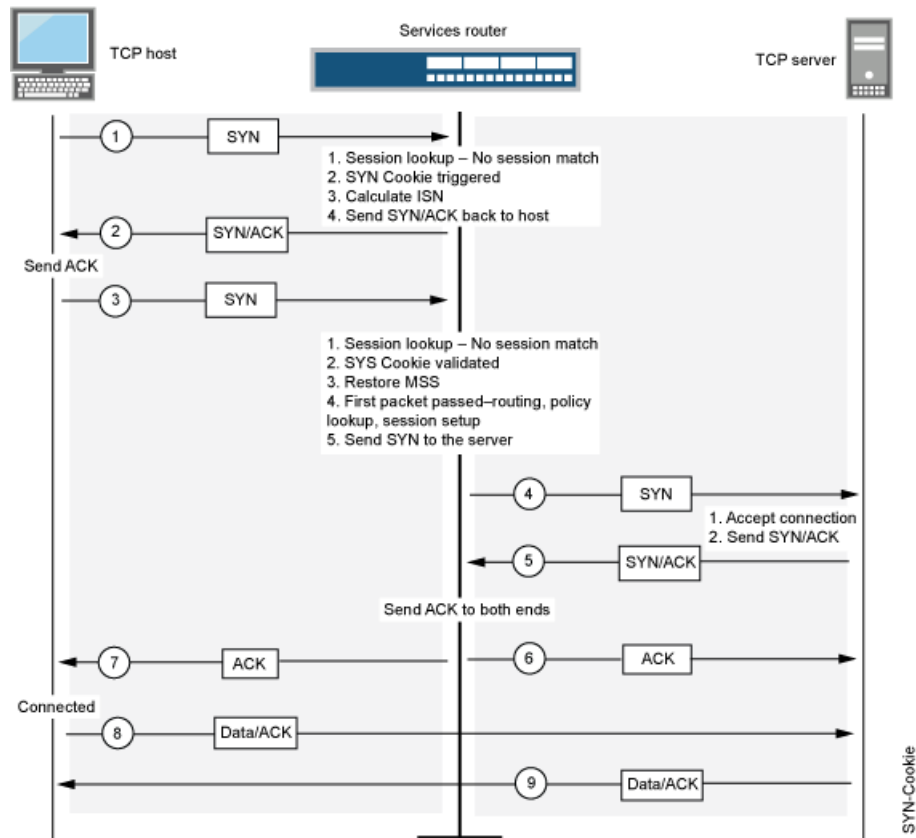
If the initiating host responds with a TCP packet containing the cookie +1 in the TCP ACK field, Junos OS extracts the cookie, subtracts 1 from the value, and recomputes the cookie to validate that it is a legitimate ACK. If it is legitimate, Junos OS starts the TCP proxy process by setting up a session and sending a SYN to the server containing the source information from the original SYN. When Junos OS receives a SYN/ACK from the server, it sends ACKs to the server and to the initiation host. At this point the connection is established and the host and server are able to communicate directly.



**NOTE:** The use of SYN cookie or SYN proxy enables the SRX Series device to protect the TCP servers behind it from SYN flood attacks in IPv6 flows.

Figure 84 on page 1086 shows how a connection is established between an initiating host and a server when SYN cookie is active on Junos OS.

Figure 84: Establishing a Connection with SYN Cookie Active



**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Enabling SYN Cookie Protection on page 1087](#)
- [DoS Attack Overview on page 1065](#)

### Example: Enabling SYN Cookie Protection

This example shows how to enable the SYN Cookie protection.

- Requirements on page 1087
- Overview on page 1087
- Configuration on page 1087
- Verification on page 1087

#### Requirements

No special configuration beyond device initialization is required before configuring this feature.

#### Overview

In this example, you set the external-syn-flood timeout value to 20 and set the security zone for external screen to external-syn-flood. Also, you set the protection mode to syn-cookie.



**NOTE:** The SYN Cookie feature can detect and protect only against spoofed SYN flood attacks, thus minimizing the negative impact on hosts that are secured by Junos OS. If an attacker uses a legitimate IP source address, rather than a spoofed IP source, then the SYN Cookie mechanism does not stop the attack.

#### Configuration

#### Step-by-Step Procedure

To enable the SYN Cookie protection:

1. Specify the external-syn-flood timeout value.
 

```
[edit]
user@host# set security screen ids-option external-syn-flood tcp syn-flood timeout
20
```
2. Set the security-zone for external screen.
 

```
[edit]
user@host# set security zones security-zone external screen external-syn-flood
```
3. Set the protection mode.
 

```
[edit]
user@host# set security flow syn-flood-protection-mode syn-cookie
```
4. If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

#### Verification

To verify the configuration is working properly, enter the **show security screen ids-option external-syn-flood** and **show security zones** commands in operational mode.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding SYN Cookie Protection on page 1085](#)
  - [DoS Attack Overview on page 1065](#)

## ICMP Flood Protection

- [Understanding ICMP Flood Attacks on page 1088](#)
- [Example: Enabling ICMP Flood Protection on page 1089](#)

### Understanding ICMP Flood Attacks

An ICMP flood typically occurs when ICMP echo requests overload the victim with so many requests that the victim expends all its resources responding until it can no longer process valid network traffic.

When enabling the ICMP flood protection feature, you can set a threshold that, once exceeded, invokes the ICMP flood attack protection feature. (The default threshold value is 1000 packets per second.) If the threshold is exceeded, Junos OS ignores further ICMP echo requests for the remainder of that second plus the next second as well. See Figure 85 on page 1089.



**NOTE:** An ICMP flood can consist of any type of ICMP message. Therefore, Junos OS monitors all ICMP message types, not just echo requests.

---

Figure 85: ICMP Flooding

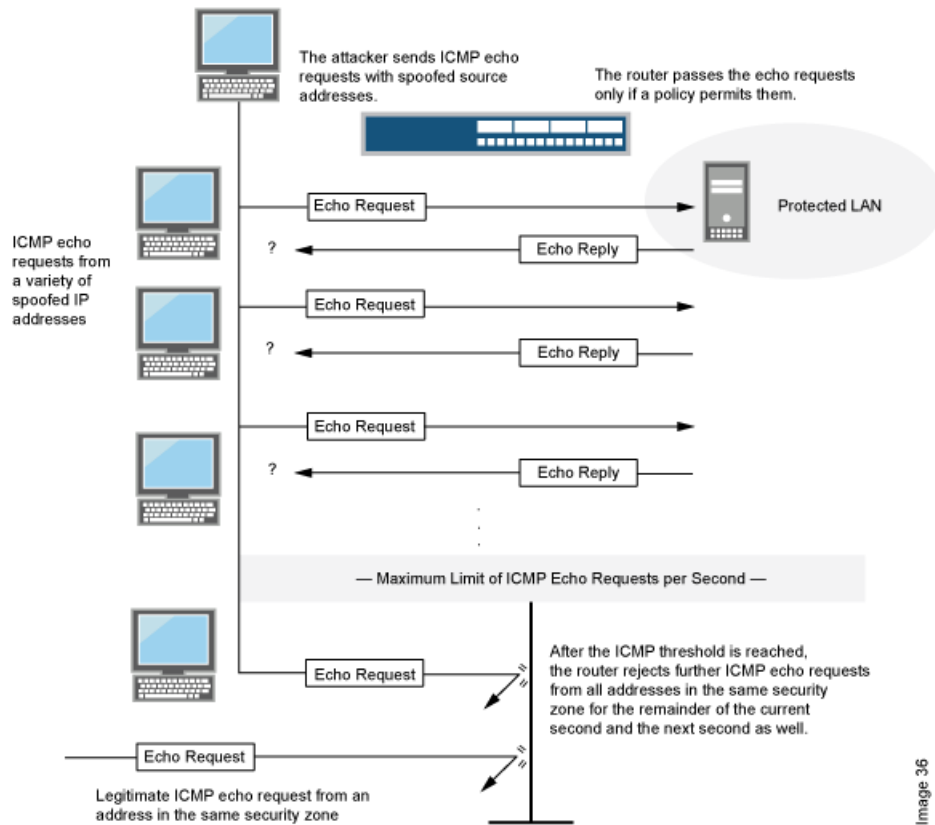


Image 36

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Enabling ICMP Flood Protection on page 1089](#)
- [DoS Attack Overview on page 1065](#)

### Example: Enabling ICMP Flood Protection

This example shows how to enable ICMP flood protection.

- [Requirements on page 1089](#)
- [Overview on page 1089](#)
- [Configuration on page 1090](#)
- [Verification on page 1090](#)

#### Requirements

No special configuration beyond device initialization is required before configuring this feature.

#### Overview

In this example, you enable ICMP flood protection. The value unit is ICMP packets per second, or pps. The default value is 1000 pps. You specify the zone where a flood might originate.

- Step-by-Step Procedure**
- Configuration**
- To enable ICMP flood protection:
1. Specify the ICMP flood threshold value.  

```
[edit]  
user@host# set security screen ids-option 1000-icmp-flood icmp flood threshold 1000
```
  2. Set the security zone for zone screen.  

```
[edit]  
user@host# set security zones security-zone zone screen 1000-icmp-flood
```
  3. If you are done configuring the device, commit the configuration.  

```
[edit]  
user@host# commit
```

#### Verification

To verify the configuration is working properly, enter the **show security screen ids-option 1000-icmp-flood** and **show security zones** commands in operational mode.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding ICMP Flood Attacks on page 1088](#)
- [DoS Attack Overview on page 1065](#)

## UDP Flood Attacks

- [Understanding UDP Flood Attacks on page 1090](#)
- [Example: Enabling UDP Flood Protection on page 1091](#)

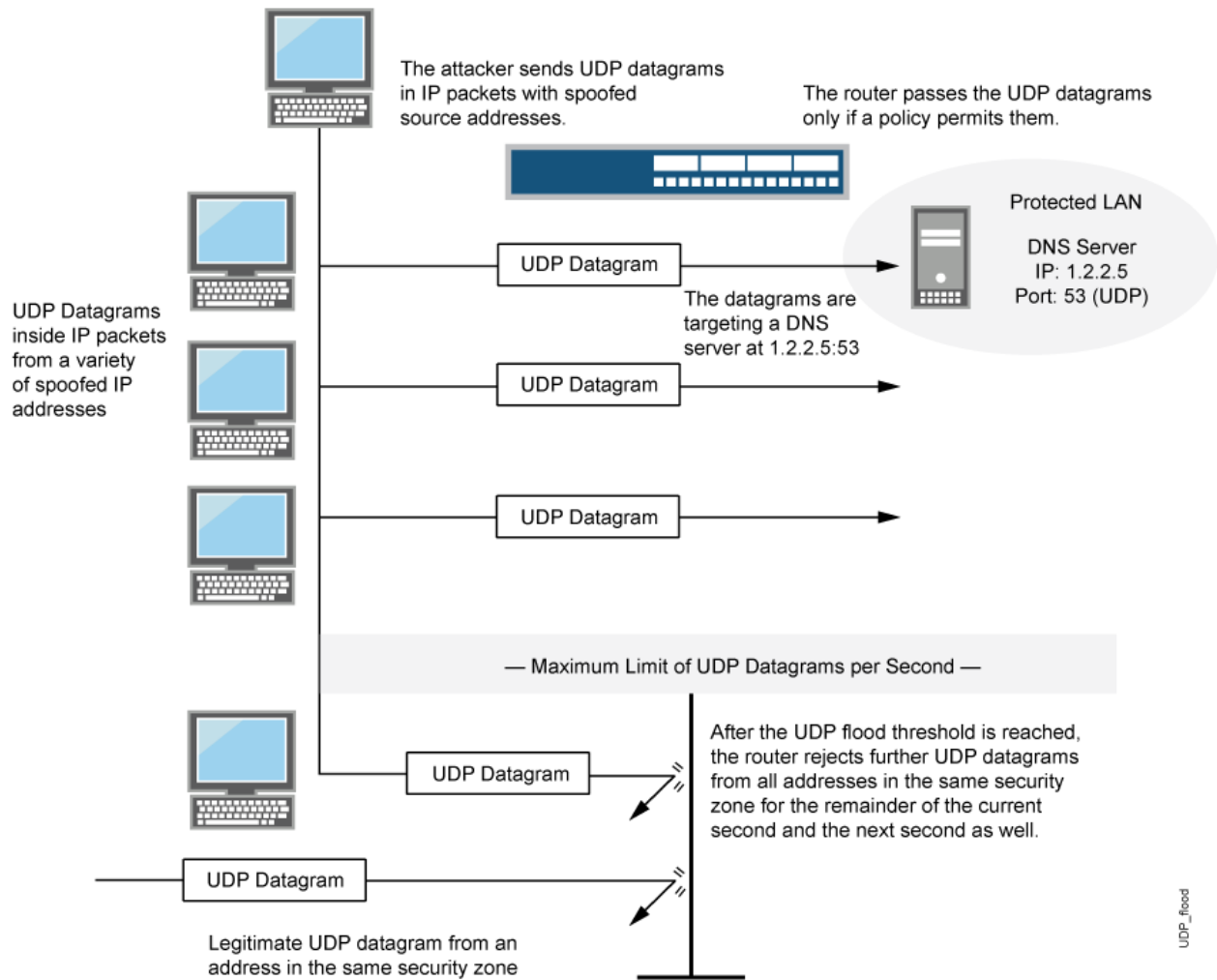
### Understanding UDP Flood Attacks

Similar to an ICMP flood, a UDP flood occurs when an attacker sends IP packets containing UDP datagrams with the purpose of slowing down the victim to the point that the victim can no longer handle valid connections.

After enabling the UDP flood protection feature, you can set a threshold that, once exceeded, invokes the UDP flood attack protection feature. (The default threshold value is 1000 packets per second, or pps.) If the number of UDP datagrams from one or more sources to a single destination and UDP port exceeds this threshold, Junos OS ignores further UDP datagrams to that destination and port for the remainder of that second plus the next second as well. See Figure 86 on page 1091.



Figure 86: UDP Flooding



- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Example: Enabling UDP Flood Protection on page 1091](#)
  - [DoS Attack Overview on page 1065](#)

### Example: Enabling UDP Flood Protection

This example shows how to enable UDP flood protection.

- [Requirements on page 1092](#)
- [Overview on page 1092](#)
- [Configuration on page 1092](#)
- [Verification on page 1092](#)

### Requirements

No special configuration beyond device initialization is required before configuring this feature.

### Overview

In this example, you enable UDP flood protection. The value unit is UDP packets per second, or pps. The default value is 1000 pps. You specify the zone where a flood might originate.

### Configuration

#### Step-by-Step Procedure

To enable UDP flood protection:

1. Specify the UDP flood threshold value.

```
[edit]
user@host# set security screen ids-option 1000-udp-flood udp flood threshold
1000
```

2. Set the security zone for external screen.

```
[edit]
user@host# set security zones security-zone external screen 1000-udp-flood
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security screen ids-option 1000-udp-flood** and **show security zones** commands in operational mode.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding UDP Flood Attacks](#) on page 1090
- [DoS Attack Overview](#) on page 1065

## Land Attacks

- [Understanding Land Attacks](#) on page 1092
- [Example: Protecting Against a Land Attack](#) on page 1093

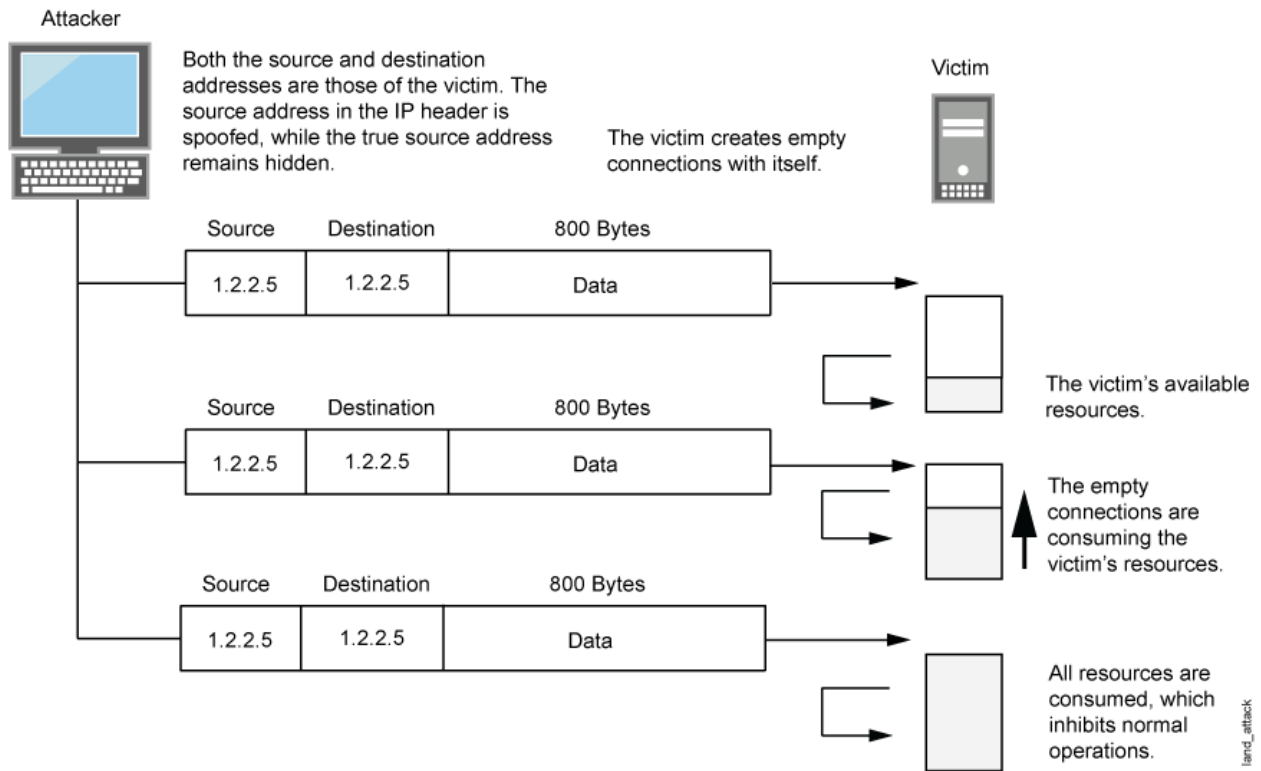
### Understanding Land Attacks

---

Combining a SYN attack with IP spoofing, a land attack occurs when an attacker sends spoofed SYN packets containing the IP address of the victim as both the destination and the source IP address.

The receiving system responds by sending the SYN-ACK packet to itself, creating an empty connection that lasts until the idle timeout value is reached. Flooding a system with such empty connections can overwhelm the system, causing a denial of service (DoS). See Figure 87 on page 1093.

Figure 87: Land Attack



When you enable the screen option to block land attacks, Junos OS combines elements of the SYN flood defense and IP spoofing protection to detect and block any attempts of this nature.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Protecting Against a Land Attack on page 1093](#)
- [DoS Attack Overview on page 1065](#)

#### Example: Protecting Against a Land Attack

This example shows how to protect against a land attack.

- [Requirements on page 1093](#)
- [Overview on page 1094](#)
- [Configuration on page 1094](#)
- [Verification on page 1094](#)

#### Requirements

No special configuration beyond device initialization is required before configuring this feature.

### Overview

This example shows how to enable protection against a land attack. In this example, you set the security screen object name as `land` and set the security zone as `zone`.

### Configuration

**Step-by-Step Procedure** To enable protection against a land attack:

1. Specify the screen object name.

```
[edit]
user@host# set security screen ids-option land tcp land
```

2. Set the security zone.

```
[edit]
user@host# set security zones security-zone zone screen land
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the `show security screen ids-option land` and `show security zones` commands in operational mode.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Land Attacks](#) on page 1092
- [DoS Attack Overview](#) on page 1065

---

## OS-Specific DoS Attacks

- [OS-Specific DoS Attacks Overview](#) on page 1094
- [Ping of Death Attacks](#) on page 1095
- [Teardrop Attacks](#) on page 1096
- [WinNuke Attacks](#) on page 1098

### OS-Specific DoS Attacks Overview

If an attacker not only identifies the IP address and responsive port numbers of an active host but also its operating system (OS), instead of resorting to brute-force attacks, the attacker can launch more elegant attacks that can produce one-packet or two-packet “kills.”

OS-specific denial-of-service (DoS) attacks, including ping of death attacks, teardrop attacks, and WinNuke attacks, can cripple a system with minimal effort. If Junos OS is protecting hosts susceptible to these attacks, you can configure Junos OS to detect these attacks and block them before they reach their target.

- Related Documentation**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Ping of Death Attacks on page 1095
  - DoS Attack Overview on page 1065

## Ping of Death Attacks

- Understanding Ping of Death Attacks on page 1095
- Example: Protecting Against a Ping of Death Attack on page 1096

### Understanding Ping of Death Attacks

OS-specific DoS attacks, such as ping of death attacks, can cripple a system with minimal effort.

The maximum allowable IP packet size is 65,535 bytes, including the packet header, which is typically 20 bytes. An ICMP echo request is an IP packet with a pseudo header, which is 8 bytes. Therefore, the maximum allowable size of the data area of an ICMP echo request is 65,507 bytes ( $65,535 - 20 - 8 = 65,507$ ).

However, many ping implementations allow the user to specify a packet size larger than 65,507 bytes. A grossly oversized ICMP packet can trigger a range of adverse system reactions such as denial of service (DoS), crashing, freezing, and rebooting.

When you enable the ping of death screen option, Junos OS detects and rejects such oversized and irregular packet sizes even when the attacker hides the total packet size by fragmenting it. See Figure 88 on page 1095.



**NOTE:** For information about IP specifications, see RFC 791, *Internet Protocol*. For information about ICMP specifications, see RFC 792, *Internet Control Message Protocol*. For information about ping of death attacks, see <http://www.insecure.org/splotts/ping-o-death.html>.

**Figure 88: Ping of Death**



The size of this packet is 65,538 bytes. It exceeds the size limit prescribed by RFC 791, *Internet Protocol*, which is 65,535 bytes. As the packet is transmitted, it becomes broken into numerous fragments. The reassembly process might cause the receiving system to crash.

- Related Documentation**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Example: Protecting Against a Ping of Death Attack on page 1096
  - DoS Attack Overview on page 1065

### Example: Protecting Against a Ping of Death Attack

---

This example shows how to protect against a ping-of-death attack.

- Requirements on page 1096
- Overview on page 1096
- Configuration on page 1096
- Verification on page 1096

#### **Requirements**

No special configuration beyond device initialization is required before configuring this feature.

#### **Overview**

In this example, you enable protection against a ping-of-death attack and specify the zone where the attack originates.

#### **Configuration**

#### **Step-by-Step Procedure**

To enable protection against a ping of death:

1. Specify the screen object name.  

```
[edit]  
user@host# set security screen ids-option ping-death icmp ping-death
```
2. Set the security zone for zone screen.  

```
[edit]  
user@host# set security zones security-zone zone screen ping-death
```
3. If you are done configuring the device, commit the configuration.  

```
[edit]  
user@host# commit
```

#### **Verification**

To verify the configuration is working properly, enter the **show security screen ids-option ping-death** and **show security zones** commands in operational mode.

#### **Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Ping of Death Attacks on page 1095
- DoS Attack Overview on page 1065

## Teardrop Attacks

- Understanding Teardrop Attacks on page 1096
- Example: Protecting Against a Teardrop Attack on page 1098

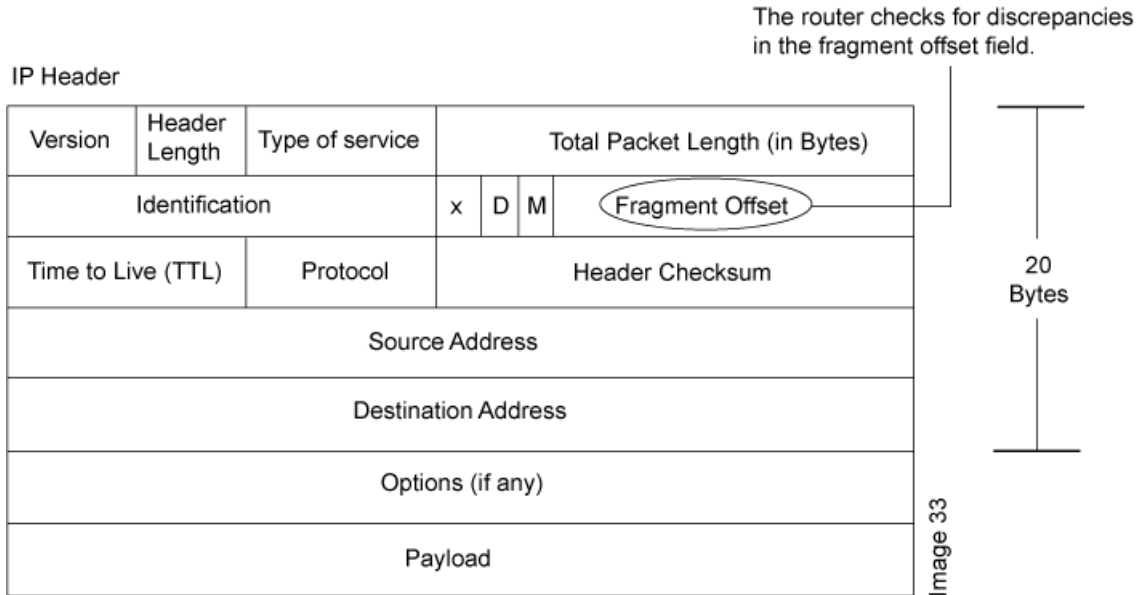
### Understanding Teardrop Attacks

---

OS-specific denial-of-service (DoS) attacks, such as teardrop attacks, can cripple a system with minimal effort.

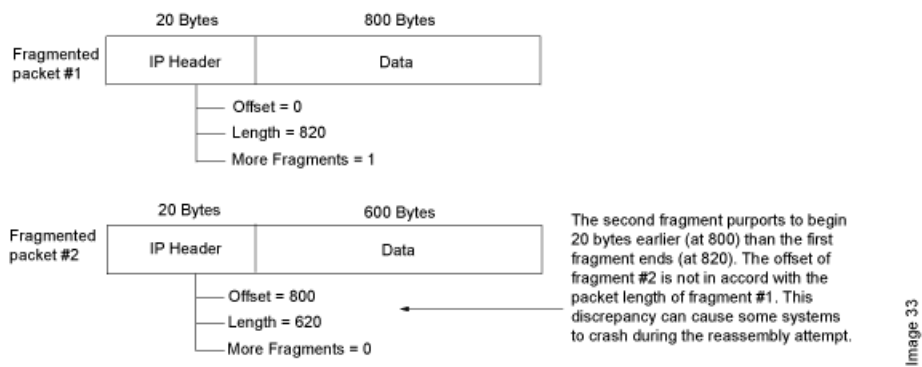
Teardrop attacks exploit the reassembly of fragmented IP packets. In the IP header, one of the fields is the fragment offset field, which indicates the starting position, or offset, of the data contained in a fragmented packet relative to the data of the original unfragmented packet. See Figure 89 on page 1097.

Figure 89: Teardrop Attacks



When the sum of the offset and size of one fragmented packet differ from that of the next fragmented packet, the packets overlap, and the server attempting to reassemble the packet can crash, especially if it is running an older OS that has this vulnerability. See Figure 90 on page 1097.

Figure 90: Fragment Discrepancy



After you enable the teardrop attack screen option, whenever Junos OS detects this discrepancy in a fragmented packet, it drops it.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Protecting Against a Teardrop Attack on page 1098](#)
- [DoS Attack Overview on page 1065](#)

---

### Example: Protecting Against a Teardrop Attack

---

This example shows how to protect against a teardrop attack.

- Requirements on page 1098
- Overview on page 1098
- Configuration on page 1098
- Verification on page 1098

#### Requirements

No special configuration beyond device initialization is required before configuring this feature.

#### Overview

In this example, you enable protection against a teardrop attack and also specify the zone where the attack originates.

#### Configuration

#### Step-by-Step Procedure

To enable protection against teardrop attack:

1. Specify the screen name.  

```
[edit]  
user@host# set security screen ids-option tear-drop ip tear-drop
```
2. Associate the screen with a security zone.  

```
[edit]  
user@host# set security zones security-zone zone screen tear-drop
```
3. If you are done configuring the device, commit the configuration.  

```
[edit]  
user@host# commit
```

#### Verification

To verify the configuration is working properly, enter the **show security screen ids-option tear-drop** and **show security zones** commands in operational mode.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Teardrop Attacks on page 1096
- DoS Attack Overview on page 1065

### WinNuke Attacks

- Understanding WinNuke Attacks on page 1098
- Example: Protecting Against a WinNuke Attack on page 1100

---

### Understanding WinNuke Attacks

---

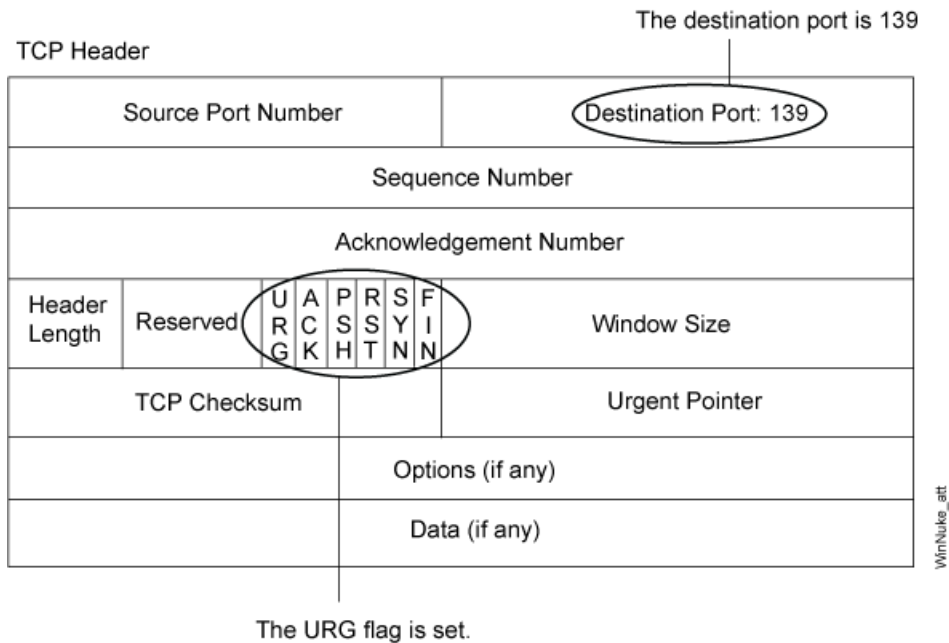
OS-specific denial-of-service (DoS) attacks, such as WinNuke attacks, can cripple a system with minimal effort.



WinNuke is a DoS attack targeting any computer on the Internet running Windows. The attacker sends a TCP segment—usually to NetBIOS port 139 with the urgent (URG) flag set—to a host with an established connection (see Figure 91 on page 1099). This introduces a NetBIOS fragment overlap, which causes many machines running Windows to crash. After the attacked machine is rebooted, the following message appears, indicating that an attack has occurred:

```
An exception OE has occurred at 0028:[address] in VxD MSTCP(01) +
000041AE. This was called from 0028:[address] in VxD NDIS(01) +
00008660. It may be possible to continue normally.
Press any key to attempt to continue.
Press CTRL+ALT+DEL to restart your computer. You will lose any unsaved information in
all applications.
Press any key to continue.
```

Figure 91: WinNuke Attack Indicators



If you enable the WinNuke attack defense screen option, Junos OS scans any incoming Microsoft NetBIOS session service (port 139) packets. If Junos OS observes that the URG flag is set in one of those packets, it unsets the URG flag, clears the URG pointer, forwards the modified packet, and makes an entry in the event log noting that it has blocked an attempted WinNuke attack.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Example: Protecting Against a WinNuke Attack on page 1100
- DoS Attack Overview on page 1065

### Example: Protecting Against a WinNuke Attack

---

This example shows how to protect against a WinNuke attack.

- Requirements on page 1100
- Overview on page 1100
- Configuration on page 1100
- Verification on page 1100

#### **Requirements**

No special configuration beyond device initialization is required before configuring this feature.

#### **Overview**

In this example, you enable protection against a WinNuke attack and specify the zone where the attack originates.

#### **Configuration**

#### **Step-by-Step Procedure**

To enable protection against WinNuke attack:

1. Specify the screen name.  

```
[edit]  
user@host# set security screen ids-option winnuke tcp winnuke
```
2. Associate the screen with a security zone.  

```
[edit]  
user@host# set security zones security-zone zone screen winnuke
```
3. If you are done configuring the device, commit the configuration.  

```
[edit]  
user@host# commit
```

#### **Verification**

To verify the configuration is working properly, enter the **show security screen ids-option winnuke** and **show security zones** commands in operational mode.

#### **Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding WinNuke Attacks on page 1098
- DoS Attack Overview on page 1065

## PART 10

# Application Identification

- Junos OS Application Identification on page 1103
- AppTrack Application Tracking on page 1129



# Junos OS Application Identification

- Understanding Junos OS Application Identification Services on page 1103
- Junos OS Application Identification Application Package on page 1104
- Junos OS Application Identification for Nested Applications on page 1109
- Junos OS Application Identification Custom Application Signature Definitions on page 1110
- Application System Cache on page 1121
- Memory and Session Limits on page 1125
- Heuristic Detection of Encrypted P2P Applications on page 1127
- Disabling Junos OS Application Identification (CLI Procedure) on page 1127

## Understanding Junos OS Application Identification Services

---

Juniper Networks provides predefined application signatures that detect Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) applications running on nonstandard ports. You can also create custom application and nested application signatures to identify applications that are not part of the predefined database. Identifying these applications provides the following benefits:

- Allows Intrusion Detection and Prevention (IDP) to apply appropriate attack objects to applications running on nonstandard ports.
- Improves performance by narrowing the scope of attack signatures for applications without decoders.
- Enables you to create detailed reports using AppTrack on the applications passing through the device.

The application signatures identify an application by matching patterns in the first few packets of a session. The application identification module matches patterns for both client-to-server and server-to-client sessions.

Application identification is enabled by default and is automatically turned on when you configure the default application in an IDP or an AppTrack policy. However, when you specify an application in the policy rule, application identification is disabled and attack objects are applied based on the specified application. This specific application configuration overwrites the automatic identification process.



**NOTE:** The Junos OS application identification application signature package update is a separately licensed subscription service. You must install the application identification application signature update license key on your device to download and install the signature database updates provided by Juniper Networks. When your license key expires, you can continue to use the locally stored application package contents.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Junos OS Administration Guide for Security Devices](#)
- [Example: Configuring IDP Applications and Services on page 731](#)
- [Understanding IDP Application Identification on page 795](#)
- [Understanding AppTrack on page 1129](#)
- [Understanding the Junos OS Application Identification Application Package on page 1104](#)
- [Understanding Junos OS Application Identification Custom Application Definitions on page 1111](#)
- [IDP Policies Overview on page 701](#)
- [Understanding IDP Service and Application Bindings by Attack Objects on page 796](#)

## Junos OS Application Identification Application Package

- [Understanding the Junos OS Application Identification Application Package on page 1104](#)
- [Example: Updating the Junos OS Application Identification Extracted Application Package Automatically on page 1106](#)
- [Updating the Junos OS Application Identification Extracted Application Package Manually \(CLI Procedure\) on page 1107](#)
- [Verifying the Junos OS Application Identification Extracted Application Package on page 1108](#)

### Understanding the Junos OS Application Identification Application Package

Juniper Networks regularly updates the predefined application identification application package database that is part of the IDP signature database and makes it available on the Juniper Networks website. This package includes a list of known application objects that can be used in Intrusion Detection and Prevention (IDP) and AppTrack to match traffic. It contains application objects such as ftp and DNS as well as nested applications such as Facebook, Kazaa, and many instant messenger programs. The application database is visible in the configuration, and you can create custom application signatures.

You need to download the application package before configuring application identification or AppTrack. You can perform the download manually or automatically, and the download command handles the download and installation of the application package. When you download the extracted package manually, you can watch the install

process, change the download URL, or import custom application or nested application signature files.



**NOTE:** Uninstalling the application package will not remove any custom application or nested application signatures that you have created. All predefined Juniper applications have the prefix “junos”, so make sure you do not use “junos” for your custom signature names.

If you do not have IDP enabled and will use application identification with AVT, you will run the following command: **request services application-identification download**. This command will extract and install the application portion of the IDP signature database to your configuration.

If you have IDP enabled and will use application identification, you will continue to run the IDP signature database download. To download the IDP signature database, run the following command: **request security idp security-package download**. The application package download can be performed manually or automatically.



**NOTE:** If you have an IDP-enabled device and will use application identification, we recommend that you only download the IDP signature database. This will avoid having two versions of the application database, which may become out of sync.



**NOTE:** The Junos OS application identification application signature package update is a separately licensed subscription service. You must install the application identification application signature update license key on your device to download and install the signature database updates provided by Juniper Networks. When your license key expires, you can continue to use the locally stored application package contents.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Junos OS Administration Guide for Security Devices](#)
- Understanding Junos OS Application Identification Custom Application Definitions on page 1111
- Understanding the IDP Signature Database on page 777.
- Understanding Junos OS Application Identification Services on page 1103
- Example: Updating the Junos OS Application Identification Extracted Application Package Automatically on page 1106

## Example: Updating the Junos OS Application Identification Extracted Application Package Automatically

The following example updates the predefined application signature package automatically.

- Requirements on page 1106
- Overview on page 1106
- Configuration on page 1106
- Verification on page 1107

### Requirements

Before you begin, configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.



**NOTE:** DNS must be set up as well as we will need to resolve the name of the update server.

### Overview

In this example, you want to download the current version of the application signature package periodically. The download should start at 11:59 PM on December 10. To maintain the most current information, you want to update the package automatically every two days, from your company's intranet site.



**NOTE:** This configuration works on higher end devices such as the SRX3400 device.

### Configuration

- [xref target has no title]

#### J-Web Step-by-Step Procedure

To set up the automatic download and periodic update with the J-Web interface:

1. Enter **Configure>Security>Application Signature** to display the Applications Signature page.
2. Click **Global Settings**.
3. Click the **Download Scheduler** tab, and modify the following fields:
  - URL: **https://acmegizmo.com/app-sig-updates/latest**
  - Enable Schedule Update: Select the check box.
  - Interval: **48**
4. Click **Reset Setting** to clear the existing start time, enter the new start time in MM-DD.hh:mm format, and click **OK**.



- Start Time: **12-10.23:59**
5. Click **Commit Options>Commit** to commit your changes.
  6. Click **Check Status** to monitor the progress of an active download or update, or to check the outcome of the latest update.

### Step-by-Step Procedure

To use the CLI to automatically update the Junos OS application signature package:

1. Specify the URL of the download server, and the time and interval for the download.

[edit]

```
user@host# set services application-identification download automatic interval 48
start-time 12-10.23:59 url https://acmegizmo.com/app-sig-updates/latest
```

2. If you are done configuring the device, commit the configuration.

[edit]

```
user@host# commit
```

### Verification

To verify that the application signature package is being updated properly, enter the **show services application-identification version** command. Review the version number and details for the latest update.

- [\[xref target has no title\]](#)

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding the Junos OS Application Identification Application Package on page 1104

## Updating the Junos OS Application Identification Extracted Application Package Manually (CLI Procedure)

Before you begin, configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

Juniper Networks regularly updates the predefined application identification application package database and makes it available on the Juniper Networks website. This package includes a list of known application objects that can be used in Intrusion Detection and Prevention (IDP) policies and AppTrack to match traffic.

The configuration instructions in this topic describes how to download the application identification application package and create a policy, and specify the new policy as the active policy. The download process will also install the application package.

1. To manually download and update the application package:

```
user@host> request services application-identification download
```

To download a specific version of the application package:

```
user@host> request services application-identification download version
version-number
```

To change the download URL for the application package from configuration mode:

```
[edit]
user@host#set services application-identification download url URL or File Path
```



**NOTE:** If you change the download URL and you want to keep that change, make sure you commit.

To uninstall the application package:

```
user@host>request services application-identification uninstall
```

2. To check the current version of the application package:

```
show services application-identification version
```

3. The application package will now be part of your configuration. From configuration mode in the CLI, enter the **show services application-identification** command to verify the configuration.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Junos OS CLI Reference](#)
- Understanding the Junos OS Application Identification Application Package on page 1104
- Example: Updating the Junos OS Application Identification Extracted Application Package Automatically on page 1106

## Verifying the Junos OS Application Identification Extracted Application Package

**Purpose** Juniper Networks regularly updates the predefined application identification application package database and makes it available on the Juniper Networks website, so it is important that you have the most recent version.

When you download the application package from the IDP signature database, you will see a status message after you enter the download command. For example, on successful download, you will see the following message (where *xxxx* is the package version number):

```
Application package xxxx is downloaded successfully
```

The syslog will also show the result of the download.

**Action** • To view the contents of the application package that is inserted into the configuration after successful download:

```
user@host> show services application-identification
```

The following output shows the first entry in the application package database, which is the predefined AIM application:

```
application junos:AIM {
  type AIM;
  index 61;
  port-mapping {
    port-range {
```

```

        tcp 5190;
    }
}
signature {
    port-range {
        tcp 0-65535;
    }
    client-to-server {
        dfa-pattern "(\\*\01[^\07]*\00.*|CONNECT
login\.oscar\.ao1\.com).*";
    }
    server-to-client {
        dfa-pattern "(\\*\01|HTTP/1\.[01] 200 Connection established\x0d
0a 0d 0a\x).*";
    }
    min-data 10;
    order 9;
}
}

```

- To check the version of the current application package from configuration mode (the version information will be the first line item):

```
user@host> show services application-identification
```

- To check the version from operational mode:

```
user@host> show services application-identification version
```

You will see the following output if package version 1608 is installed successfully:

```
Application package version: 1608
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding the Junos OS Application Identification Application Package on page 1104](#)
- [Example: Updating the Junos OS Application Identification Extracted Application Package Automatically on page 1106](#)

## Junos OS Application Identification for Nested Applications

- [Understanding Junos OS Application Identification for Nested Applications on page 1109](#)
- [Activating Junos OS Application Identification for Nested Applications \(CLI Procedure\) on page 1110](#)

### Understanding Junos OS Application Identification for Nested Applications

With the greater use of application protocol encapsulation, the need arises to support the identification of multiple different applications running on the same Layer 7 protocols. For example, applications such as Facebook and Yahoo Messenger can both run over HTTP, but there is a need to identify them as two different applications running on the same Layer 7 protocol. In order to do this, the current application identification layer is split into two layers: Layer 7 nested applications and Layer 7 protocols.

The included predefined application signatures have been created to detect the Layer 7 nested applications whereas the existing Layer 7 protocol signatures, such as FTP and HTTP, still function in the same manner. These predefined application signatures can be used in attack objects.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding the Junos OS Application Identification Application Package on page 1104
- Activating IDP Application Identification for Nested Applications (CLI Procedure) on page 800
- Understanding Junos OS Application Identification Services on page 1103

## Activating Junos OS Application Identification for Nested Applications (CLI Procedure)

Application identification for nested applications is turned on by default. You can manually turn it off by using the CLI.

To disable nested application identification:

1. `user@host# set services application-identification nested-application-settings no-nested-application`
2. If you want to reenable nested application identification, delete the configuration statement:  
`user@host# delete services application-identification nested-application-settings no-nested-application`
3. If you are finished configuring the device, commit the configuration.
4. To verify the configuration, enter the `show services application-identification nested-applications` command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Junos OS CLI Reference](#)
- Understanding IDP Application Identification for Nested Applications on page 800
- Understanding the Junos OS Application Identification Application Package on page 1104
- Understanding Junos OS Application Identification Services on page 1103

---

## Junos OS Application Identification Custom Application Signature Definitions

- Understanding Junos OS Application Identification Custom Application Definitions on page 1111
- Example: Configuring Junos OS Application Identification Custom Application Definitions on page 1114
- Example: Configuring Junos OS Application Identification Custom Nested Application Definitions on page 1117

## Understanding Junos OS Application Identification Custom Application Definitions

Application identification supports user-defined custom application signatures for applications and nested applications. With custom application signatures, you can create signatures that will detect applications that are not part of the predefined application package. When you perform an update or uninstall the application package, custom applications will not be modified or removed.

Both predefined and custom application signatures are located in the **[services application-identification application]** hierarchy. The predefined and custom applications for nested application signatures are located in the **[services application-identification nested-application]** hierarchy.

When you create custom application or nested application signatures:

- Make sure that your entries are unique to entries in the predefined application database. All predefined signatures provided by Juniper have the prefix “junos” in the application signature name, for example junos:ftp, junos:facebook, so do not use that prefix when naming your custom signatures.
- Make sure that the index entries are unique among all application and nested application custom signatures since custom application signatures and custom nested application signatures share the same index pool.

Once you download the application signature package, you can view signatures by running the **show services application-identification** command. You can use the predefined signatures as a base for creating your custom signatures; however, make sure your application name does not start with junos and that the index number of each signature is unique.

This topic contains the following sections:

- Custom Application Definitions on page 1111
- Custom Nested Application Definitions on page 1112

### Custom Application Definitions

Table 106 on page 1111 compares custom and predefined configuration parameters for applications. These differences ensure that custom application signatures are unique to the predefined signatures so they are not deleted when the predefined application package is updated or deleted.

**Table 106: Comparison of Custom Application Signature and Predefined Signature**

Predefined	Custom
Index range: 1 through 32,767	Index range: 32,768 through 65,534
Name prefix junos	Name prefix is user defined (junos is reserved for predefined signatures)
Order field unique for all applications and nested applications	Order field unique for all applications and nested applications

Table 107 on page 1112 lists and describes the attributes available for creating a custom application signature. The hierarchy level is **[edit services application-identification application *application-name*]**.

**Table 107: Custom Application Signature Attributes**

Attribute	Description
application-name	Name of the custom application signature. Must be a unique name with a maximum length of 32 characters. (Required)
disable	Do not match traffic for this application. Default is off.
index	A number that is a one-to-one mapping to the application name that is used to ensure that each signature is unique. The index range for predefined applications is 1 through 32,767. The index range for custom applications and custom nested applications is 32,768 through 65,534. (Required)
<b>Signature Attributes</b>	
signature	Defines the application signature attributes for pattern matching. (Required)
client-to-server	Defines the attributes for traffic in the client-to-server direction.  dfa-pattern: Maximum length is 1023. (Optional)  regex: Enter a regular expression that should be matched for client-to-server traffic.
disable	Toggle on means that a signature method is not used to identify this application. The default is off.
min-data	The minimum number of bytes or packets to apply to the dfa-pattern. Default is 10, range is 4 through 1024.
order	When multiple patterns are matched for the same session, the lowest order number takes the highest priority. Must be unique. (Required)
port-range	Default ranges: TCP/0 through 65,535; UDP/0 through 65,535. (Optional)
server-to-client	Defines the attributes for traffic in the server-to-client direction.  dfa-pattern: Maximum length is 1023. (Optional)  regex: Enter a regular expression that should be matched for server-to-client traffic.

### Custom Nested Application Definitions

Table 108 on page 1113 compares custom and predefined configuration parameters for nested applications. These differences ensure that custom nested application signatures are unique to the predefined signatures so they are not deleted when you update or delete the predefined application package.

**Table 108: Comparison of Custom Nested Application Signature and Predefined Signature Parameters**

Predefined	Custom
Index range: 1 through 32,767	Index range: 32,768 through 65,534
Name prefix: junos	Name prefix: not unique and must not be junos
Order field: unique for all applications and nested applications	Order field: unique for all applications and nested applications

Table 109 on page 1113 lists and describes the attributes available for creating a custom nested application signature. The hierarchy level is [edit services application-identification nested-application *nested-application-name*].

**Table 109: Custom Nested Application Signature Attributes**

Attribute	Description
nested-application-name	Name of the custom nested application signature. Must be a unique name with a maximum length of 32 characters. (Required)
index	A number that is a one-to-one mapping to the application name that is used to ensure that each signature is unique. The index range for predefined applications is 1 through 32,767. The index range for custom applications and custom nested applications is 32,768 through 65,534. (Required)
protocol	The protocol that will be monitored to identify nested applications. HTTP is supported.
<b>Signature Attributes</b>	
signature <i>name</i>	Name of the custom nested application signature. Must be a unique name with a maximum length of 32 characters. (Required)
chain-order	Signatures can contain multiple members. If chain-order is on, those members are read in order. The default for this option is no chain order. If a signature contains only one member, this option is ignored.
maximum-transactions	The maximum number of transactions that should occur before a match is made.
member <i>name</i>	Defines a member name for a custom nested application signature. Custom signatures can contain multiple members that define attributes for an application. (The member name range is m01 through m16.)
context	Defines a service-specific context, such as http-url.
direction	The connection direction of the packets to apply pattern matching. The options are any, client-to-server, or server-to-client.
pattern	Define the dfa pattern to match in the context.

Table 109: Custom Nested Application Signature Attributes (*continued*)

Attribute	Description
order	When multiple patterns are matched for the same session, the lowest order number takes the highest priority. Must be unique. (Required)

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Configuring Junos OS Application Identification Custom Application Definitions on page 1114](#)
- [Understanding the Junos OS Application Identification Application Package on page 1104](#)
- [Understanding Junos OS Application Identification Services on page 1103](#)

### Example: Configuring Junos OS Application Identification Custom Application Definitions

This example shows how to configure custom application signatures for Junos OS application identification.

- [Requirements on page 1114](#)
- [Overview on page 1114](#)
- [Configuration on page 1114](#)
- [Verification on page 1116](#)

#### Requirements

No special configuration beyond device initialization is required before configuring this feature.

#### Overview

Application identification supports custom application signatures to detect applications as they pass through the device. When you configure custom signatures, make sure that your signatures are unique.

In this example, you create an application signature named `my-app` with an associated index of 33000. This application operates over the HTTP protocol from port 6400, a port within the TCP port range of 0 through 65,535. You want to check the first two packets of all traffic in both directions for the dfa pattern of `\xff\x[\xfa-\xff].*` To ensure that all predefined application signatures have a higher priority than `my-app`, set the signature order to 2580.

#### Configuration

#### CLI Quick Configuration

To quickly configure custom application signatures, copy the following commands and paste them into the CLI:

```
[edit]
set services application-identification application my-app type HTTP index 33000
signature port-range tcp 0-65535
set services application-identification application my-app signature client-to-server
dfa-pattern \xff\x[\xfa-\xff].*
```



```
set services application-identification application my-app signature server-to-client
  dfa-pattern \xff\x[\xfa-\xff].*
set services application-identification application my-app signature min-data 2 order
  2580
```

### J-Web Quick Configuration

To configure the my-app application signature with the J-Web interface, use the following procedure:

1. Select **Configure>Security>Application Signature** to display the Applications Signature page.
2. Click **Add** to create a new custom application signature.
3. Enter the following values on the Add Application page, and click **OK**.
  - Application Type: **HTTP**
  - Index: **33000**
  - Signature Name: **my-app**
  - Min Data: **2**
  - Order: **2580**
  - Client to server: **\xff\x[\xfa-\xff].\***
  - Server to client: **\xff\x[\xfa-\xff].\***
  - TCP Port: **0-65535**
  - Port Range: Select the check box.
4. Select **Commit Options>Commit** to commit the configuration and return to the main configuration page.

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure custom application signatures:

1. Set the application signature name that you will use in your policy for your custom application.
 

```
[edit services application identification]
user@host# set application my-app
```
2. Set the application type.
 

```
[edit services application identification]
user@host# set application my-app type HTTP
```
3. Set the index number.
 

```
[edit services application identification]
user@host# set application my-app index 33000
```

4. Set the signature information by starting with the signature port range.
 

```
[edit services application identification]
user@host# set application my-app signature port-range tcp 0-65535.
```
5. Set the signature client-to-server dfa pattern.
 

```
[edit services application identification]
user@host# set application my-app signature client-to-server dfa-pattern
\xff\x[\xfa-\xff].*
```
6. Set the signature server-to-client dfa pattern.
 

```
[edit services application identification]
user@host# set application my-app signature server-to-client dfa-pattern
\xff\x[\xfa-\xff].*
```
7. Set the signature minimum data value.
 

```
[edit services application identification]
user@host# set application my-app signature min-data 2
```
8. Set the signature order.
 

```
[edit services application identification]
user@host# set application my-app signature order 2580
```

**Results** From configuration mode, confirm your configuration by entering the **show services application-identification** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show services application-identification

application my-app {
  type HTTP;
  index 33000;
  signature {
    port-range {
      tcp 0-65535;
    }
    client-to-server {
      dfa-pattern "\xff\x[\xfa-\xff].*";
    }
    server-to-client {
      dfa-pattern "\xff\x[\xfa-\xff].*";
    }
    min-data 2;
    order 2580;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying the Custom Application Definitions on page 1117

### Verifying the Custom Application Definitions

**Purpose** Display predefined and custom application signatures and settings that are configured on your device. Note that predefined application signature names use the prefix `junos`.

**Action** From configuration mode, enter the `show services application-identification` command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding Junos OS Application Identification Custom Application Definitions on page 1111](#)
  - [Understanding the Junos OS Application Identification Application Package on page 1104](#)
  - [Understanding Junos OS Application Identification Services on page 1103](#)
  - [Example: Configuring IDP Policies for Application Identification on page 798](#)

## Example: Configuring Junos OS Application Identification Custom Nested Application Definitions

This example shows how to configure a custom nested application signature for Junos OS application identification.

- [Requirements on page 1117](#)
- [Overview on page 1117](#)
- [Configuration on page 1119](#)
- [Verification on page 1121](#)

### Requirements

Before you begin, configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

### Overview

Application identification supports custom application signatures that detect applications nested within an HTTP application. When you configure custom application signatures, make sure that your specifications are unique.

In this example, you define a custom nested application signature named `my-social-website-app` (index 34000). It is nested within HTTP and interacts with particular social websites. You refer to this application as a `Social-Website-App` type.



**NOTE:** In logging data, a common convention is used that appends the application type to the protocol name. For example, the nested application type will be displayed in the log file as `HTTP:Social-Website-App`. This naming convention is used explicitly in this example by appending the nested application type to the signature name as well.

A single signature specification named `my-social-website-sig:Social-Website-App` defines the search method and match criteria for identifying `my-social-website-app`. Only the first three transactions of client-to-server traffic need to be checked to determine if this nested application is present. Based on criteria specified in the single signature member, `m01`, the `http-header-host` portion of the HTTP header will be checked for the pattern `"*(facebook\.com|fbcdn\.net)"`. To avoid potential conflicts with other application signature matches, the order is set to 3765, ensuring that other predefined and custom applications have a higher priority than `my-social-website-app`.

## Configuration

**CLI Quick Configuration** To quickly configure a custom nested application signature, copy the following commands and paste them into the CLI:

```
[edit]
set services application-identification nested-application my-social-website-app type
  Social-Website-App index 34000 signature my-social-website-sig:Social-Website-App
  member m01 context http-header-host pattern ".*(facebook\.com|fbcdn\.net)"
  direction client-to-server
set services application-identification nested-application my-social-website-app signature
  my-social-website-sig:Social-Website-App maximum-transactions 3 order 3765
set services application-identification nested-application my-social-website-app protocol
  HTTP
```

**J-Web Quick Configuration** To configure a custom nested application signature with the J-Web interface, use the following procedure.

1. From the J-Web interface, select **Configure>Security>Application Signature** to display the Applications Signature page.
2. Enter **HTTP** in the Search box of the upper pane to display the HTTP entry at the top of the upper pane.
3. Select the HTTP application signature to display its existing nested application signatures and to activate the buttons in the lower pane.
4. Click **Add** in the lower pane to create a new nested application signature.
5. Enter the following values on the Add Nested Application page, and click **OK**.
  - Application Type: **Social-Website-App**
  - Index: **34000**
  - Name: **my-social-website-app**
  - Protocol: **HTTP**
  - Signature Name: **my-social-website-sig:Social-Website-App**
  - Order: **3765**
  - Maximum Transactions: **3**
  - Member Name: **m01**
  - Direction: **client-to-server**
  - Context: **http-header-host**
  - Pattern: **".\*(facebook\.com|fbcdn\.net)"**
6. Select **Commit Options>Commit** to commit the configuration and return to the main configuration page.

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To create a custom nested application signature:

1. Set the application name and type.

```
[edit services application identification]
user@host# set nested-application my-social-website-app type Social-Website-App
```

2. Set the index number.

```
[edit services application-identification]
user@host# set nested-application my-social-website-app index 34000
```

3. Set the protocol.

```
[edit services application-identification]
user@host# set nested-application my-social-website-app protocol HTTP
```

4. Set the signature information.

```
[edit services application identification]
user@host# set nested-application my-social-website-app signature
my-social-website-sig:Social-Website-App
```

5. Create a member named m01 for the signature that defines the application attributes. (The member name range is m01 through m16.)

```
[edit services application identification]
user@host# set nested-application my-social-website-app signature
my-social-website-sig:Social-Website-App member m01
```

6. Set the context to be used for matching the application.

```
[edit services application identification]
user@host# set nested-application my-social-website-app signature
my-social-website-sig:Social-Website-App member m01 context http-header-host
```

7. Set the pattern to match.

```
[edit services application identification]
user@host# set nested-application my-social-website-app signature
my-social-website-sig:Social-Website-App member m01 dfa-pattern
".*(facebook\.com|fbcdn\.net)"
```

8. Set the direction in which to match traffic.

```
[edit services application identification]
user@host# set nested-application my-social-website-app signature
my-social-website-sig:Social-Website-App member m01 direction client-to-server
```

9. Set the maximum number of transactions to search for a match to 3.

```
[edit services application identification]
user@host# set nested-application my-social-website-app signature
my-social-website-sig:Social-Website-App maximum-transactions 3
```

10. Set the matching order for this signature to 3765.

```
[edit services application identification]
```

```
user@host# set nested-application my-social-website-app signature
my-social-website-sig:Social-Website-App order 3765
```

**Results** From configuration mode, confirm your configuration by entering the **show services application-identification** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show services application-identification

nested-application my-social-website-app {
  type Social-Website-App;
  index 34000;
  protocol HTTP;
  signature my-social-website-sig:Social-Website-App {
    member m01 {
      context http-header-host;
      dfa-pattern ".*(facebook\.com|fbcdn\.net)";
      direction client-to-server;
    }
    maximum-transactions 3;
    order 3765;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Custom Nested Application Definition on page 1121

#### *Verifying Custom Nested Application Definition*

**Purpose** To display the configuration of all predefined and custom application signatures and nested application signatures for this device.

**Action** From configuration mode, enter the **show services application-identification** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding Junos OS Application Identification Custom Application Definitions on page 1111
  - Understanding the Junos OS Application Identification Application Package on page 1104
  - Understanding Junos OS Application Identification Services on page 1103

## Application System Cache

- Understanding the Application System Cache on page 1122
- Deactivating Application System Cache Information for Application Identification (CLI Procedure) on page 1122

- Understanding Application System Cache Information for Nested Application Identification on page 1123
- Deactivating Application System Cache Information for Nested Application Identification (CLI Procedure) on page 1123
- Verifying Application System Cache Statistics on page 1124

## Understanding the Application System Cache

Application system cache (ASC) saves the mapping between an application type and the corresponding destination IP address, destination port, protocol type, and service.

Once an application is identified, its information is saved in the ASC so that only one pattern matching is required for an application running on a particular system, thereby expediting the identification process.

A mapping is saved in the ASC only if the matched signature contains both client-to-server and server-to-client patterns. This process protects the system from hackers who might send malicious packets through a legitimate server port so that it is interpreted as a different application.

By default, the ASC saves the mapping information for 3600 seconds. However, you can configure the cache timeout value by using the CLI.

To minimize the impact on performance, application system cache is refreshed only when Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) traffic triggers a cache lookup. Without a cache lookup, the entries in the ASC remain unchanged even after cache timeout.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Junos OS Application Identification Services on page 1103
- Verifying Application System Cache Statistics on page 1124

## Deactivating Application System Cache Information for Application Identification (CLI Procedure)

Application caching is turned on by default. You can manually turn this caching off using the CLI.

```
user@host# set services application-identification no-application-system-cache
```

When you use the **show** command in the CLI operation mode for the application system cache (ASC), application cache is listed as **off**. Note that if the cache contains data from the prior implementation, the cached data is also displayed.

```
user@host> show services application-identification application-system-cache
```

```
Application System Cache Configurations:
```

```
application-cache: off
nested-application-cache: off
cache-entry-timeout: 3600 seconds
pic: 2/0
```

Vsys-ID	IP address	Port	Protocol	Application
0	5.0.0.1	80	TCP	HTTP



0	7.0.0.1	80	TCP	HTTP:FACEBOOK
---	---------	----	-----	---------------

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Verifying Application System Cache Statistics on page 1124](#)
- [Understanding Junos OS Application Identification Services on page 1103](#)

## Understanding Application System Cache Information for Nested Application Identification

Nested application identification information is saved in the application system cache (ASC) to improve performance. The ASC is updated when a different application is identified. The only circumstances in which nested application information is not cached are the following:

- The application system cache is turned off for nested application identification.
- The matched application signatures have only client-to-server members.
- There is no valid server-to-client response seen for a transaction. This is done to prevent an attacker from sending invalid client-to-server requests to poison the ASC.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Junos OS Application Identification Services on page 1103](#)
- [Verifying Application System Cache Statistics on page 1124](#)

## Deactivating Application System Cache Information for Nested Application Identification (CLI Procedure)

Caching for nested applications is turned on by default. You can manually turn this caching off using the CLI.

```
user@host# set services application-identification nested-application-settings
no-application-system-cache
```

When you use the **show** command in the CLI operation mode for the application system cache (ASC), application cache and nested application cache are listed as **off**. Note that if the cache contains data from the prior implementation, the cached data is also displayed.

```
user@host> show services application-identification application-system-cache
```

```
Application System Cache Configurations:
```

```
application-cache: off
nested-application-cache: off
cache-entry-timeout: 3600 seconds
pic: 2/0
```

Vsys-ID	IP address	Port	Protocol	Application
0	5.0.0.1	80	TCP	HTTP
0	7.0.0.1	80	TCP	HTTP:FACEBOOK

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Activating IDP Application Identification for Nested Applications \(CLI Procedure\)](#) on page 800
  - [Verifying Application System Cache Statistics](#) on page 1124
  - [Understanding Junos OS Application Identification Services](#) on page 1103

## Verifying Application System Cache Statistics

**Purpose** Verify the application system cache (ASC) statistics.



**NOTE:** The application system cache will display the cache for application identification applications and nested applications.

**Action** From CLI operation mode, enter the **show services application-identification application-system-cache** command.

### Sample Output

```
user@host> show services application-identification application-system-cache
Application System Cache Configurations:
  application-cache: on
  nested-application-cache: on
  cache-entry-timeout: 3600 seconds
  pic: 2/0
Vsys-ID    IP address      Port    Protocol  Application
  0         5.0.0.1         80      TCP       HTTP
  0         7.0.0.1         80      TCP       HTTP:FACEBOOK
```

**Meaning** The output shows a summary of the ASC statistics information. Verify the following information:

- **Vsys-ID**—Displays the virtual system identification number.
- **IP address**—Displays the destination address.
- **Port**—Displays the destination port on the server.
- **Service**—Displays the name of the service or application identified on the destination port.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Junos OS CLI Reference](#)
  - [Understanding the IDP Application System Cache](#) on page 801
  - [Disabling Application Identification for an IDP Policy \(CLI Procedure\)](#) on page 799

## Memory and Session Limits

- Understanding Memory and Session Limit Settings for Junos OS Application Identification Services on page 1125
- Example: Setting Memory and Session Limits for Junos OS Application Identification Services on page 1126

### Understanding Memory and Session Limit Settings for Junos OS Application Identification Services

You can configure settings to limit the number of sessions running application identification and also limit memory usage for application identification.

- Memory limit for a session—You can configure the maximum amount of memory bytes that can be used to save packets for application identification for one TCP or UDP session. You can also configure a limit for global memory usage for application identification. Application identification is disabled for a session after the system reaches the specified memory limit for the session. However, application identification continues to match patterns. The matched application is saved to cache so that the next session can use it. This protects the system from attackers trying to bypass application identification by purposely sending large client-to-server packets.
- Number of sessions—You can configure the maximum number of sessions that can run application identification at the same time. Application identification is disabled after the system reaches the specified number of sessions. You limit the number of sessions so that you can prevent a denial-of-service (DoS) attack, which occurs when too many connection requests overwhelm and exhaust all the allocated resources on the system.

Table 110 on page 1125 shows the session capacity for a central point (CP) for SRX3400, SRX3600, SRX5600, and SRX5800 devices.

**Table 110: Maximum CP Sessions for Application Identification Services**

SRX Series Devices	Maximum Sessions	Central Point (CP)
SRX3400	2.25 million	Combo-mode CP
SRX3600	2.25 million	Combo-mode CP
SRX5600	10 million	Full CP
	2.25 million	Combo-mode CP
SRX5800	10 million	Full CP
	2.25 million	Combo-mode CP

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Junos OS Application Identification Services on page 1103

- Example: Setting Memory and Session Limits for IDP Application Identification Services on page 805

## Example: Setting Memory and Session Limits for Junos OS Application Identification Services

This example shows how to configure memory and session limits for Junos OS application identification services.

- Requirements on page 1126
- Overview on page 1126
- Configuration on page 1126
- Verification on page 1126

### Requirements

---

Before you begin:

- Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Download the application package.

### Overview

---

In this example, you configure the maximum number of sessions that can run application identification at the same time as 600. You also configure 5000 bytes as the maximum amount of memory that can be used for saving packets for application identification for one TCP session.

### Configuration

---

#### Step-by-Step Procedure

To configure memory and session limits for Junos OS application identification services:

1. Specify the session limit for application identification.  

```
[edit]  
user@host# set services application-identification max-sessions 600
```
2. Specify the memory limit for application identification.  

```
[edit]  
user@host# set services application-identification max-tcp-session-packet-memory 5000
```
3. If you are done configuring the device, commit the configuration.  

```
[edit]  
user@host# commit
```

### Verification

---

To verify the configuration is working properly, enter the **show services application-identification** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Junos OS CLI Reference](#)
  - Understanding Memory and Session Limit Settings for IDP Application Identification on page 804
  - Understanding Junos OS Application Identification Services on page 1103

---

## Heuristic Detection of Encrypted P2P Applications

---

P2P applications like BitTorrent and Skype contain encrypted data packets. The SRX Series devices cannot identify the encrypted data packets with the current application signatures, which are based on regular expression patterns. Heuristics are used to detect such traffic and to improve the detection rate.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Junos OS CLI Reference](#)
  - Understanding Junos OS Application Identification Services on page 1103

---

## Disabling Junos OS Application Identification (CLI Procedure)

---

Application identification is enabled by default. You can disable application identification with the CLI.

To disable application identification:

```
user@host# set services application-identification no-application-identification
```

If you want to reenable application identification, delete the configuration statement that specifies disabling of application identification:

```
user@host# delete services application-identification no-application-identification
```

If you are finished configuring the device, commit the configuration.

To verify the configuration, enter the **show services application-identification** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Junos OS CLI Reference](#)
  - Understanding Junos OS Application Identification Services on page 1103



# AppTrack Application Tracking

- Understanding AppTrack on page 1129
- AppTrack Usage on page 1130

## Understanding AppTrack

---

AppTrack, an application tracking tool, provides statistics for analyzing bandwidth usage of your network. When enabled, AppTrack collects byte, packet, and duration statistics for application flows in the specified zone. By default, when each session closes, AppTrack generates a message that provides the byte and packet counts and duration of the session, and sends it to the host device. The Security Threat Response Manager (STRM) retrieves the data and provides flow-based application visibility.

AppTrack messages are similar to session logs and use syslog or structured syslog formats. The message also includes an application field for the session. If AppTrack identifies a custom-defined application and returns an appropriate name, the custom application name is included in the log message. (If the application identification process fails or has not yet completed when an update message is triggered, the message specifies **none** in the application field.)

If you enable AppTrack for a zone and specify a **session-update-interval** time, whenever a packet is received, AppTrack checks whether the time since the start of the session or since the last update is greater than the update interval. If so, AppTrack updates the counts and sends an update message to the host. If a short-lived session starts and ends within the update interval, AppTrack generates a message only at session close.

When you want the initial update message to be sent earlier than the specified update interval, use the **first-update-interval**. The **first-update-interval** lets you enter a shorter interval for the first update only. Alternatively, you can generate the initial update message at session start by using the **first-update** option.



**NOTE:** If you specify both the **first-update** option and the **first-update-interval** option, AppTrack sends an update message when the session begins. In this case, the **first-update-interval** value is ignored, and a second message is sent when the next full update interval has elapsed.

The close message updates the statistics for the last time and provides an explanation for the session closure. The following codes are used:

**TCP RST**—RST received from either end.

**TCP FIN**—FIN received from either end.

**Response received**—Response received for a packet request (such as **icmp req-reply**).

**ICMP error**—ICMP error received (such as **dest unreachable**).

**Aged out**—Session aged out.

**ALG**—ALG closed the session.

**IDP**—IDP closed the session.

**Parent closed**—Parent session closed.

**CLI**—Session cleared by a CLI statement.

**Policy delete**—Policy marked for deletion.

---

## AppTrack Usage

- Example: Configuring AppTrack on page 1130
- Example: Verifying AppTrack Operation (CLI) on page 1133

### Example: Configuring AppTrack

This example shows how to configure the AppTrack tracking tool so you can analyze the bandwidth usage of your network.

- Requirements on page 1130
- Overview on page 1130
- Configuration on page 1131
- Verification on page 1132

---

#### Requirements

Before you configure AppTrack, it is important that you understand conceptual information about AppTrack and Junos OS application identification. See “Understanding AppTrack” on page 1129 and “Understanding Junos OS Application Identification Services” on page 1103.

---

#### Overview

Application identification is enabled by default and is automatically turned on when you configure the default application in either an IDP or an AppTrack policy. This example shows how to enable application tracking for the security zone named trust. This example also shows how to configure the remote syslog device to receive AppTrack messages. The source IP address that is used when exporting security logs is 5.0.0.254 and the security logs are sent to the host located at address 5.0.0.1. The first message is generated



1 minute after the session starts and update messages are sent every 4 minutes after that or until the session ends. A final message is sent at session end.

### Configuration

**CLI Quick Configuration** To quickly configure AppTrack, copy the following commands and paste them into the CLI:

```
[edit]
set security log format syslog
set security log source-address 5.0.0.254
set security log stream idpdata host 5.0.0.1
set security zones security-zone trust application-tracking
set security application-tracking session-update-interval 4
set security application-tracking first-update-interval 1
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure AppTrack:

1. Configure the remote syslog device to receive AppTrack messages.

```
[edit]
user@host# set security log format syslog
user@host# set security log source-address 5.0.0.254
user@host# set security log stream idpdata host 5.0.0.1
```

2. Enable AppTrack for the security zone.

```
[edit security]
user@host# set zones security-zone trust application-tracking
```

3. Generate update messages at the specified interval.

```
[edit security]
user@host# set application-tracking session-update-interval 4
```

4. Generate the first message at the specified interval after session start.

```
[edit security]
user@host# set application-tracking first-update-interval 1
```

Alternatively, to generate a message at session start and send update messages every 5 minutes after that, you could use the **first-update** option instead of the **first-update-interval** option.

```
[edit security]
user@host# set application-tracking first-update
```



**NOTE:** If you specify both the **first-update** option and the **first-update-interval** option, the **first-update-interval** value is ignored.

**Results** From configuration mode, confirm your configuration by entering the **show security** and **show security application-tracking** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
user@host# show security
...
  application-tracking {
    first-update-interval 1;
    session-update-interval 4;
  }
...
  log {
    format syslog;
    source-address 5.0.0.254;
    stream idpdata {
      host {
        5.0.0.1;
      }
    }
  }
}
user@host# show security application-tracking
...
  security-zone trust {
    application-tracking;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying AppTrack Operation on page 1132
- Verifying Security Flow Session Statistics on page 1132
- Verifying Application System Cache Statistics on page 1133
- Verifying the Status of Application Identification Counter Values on page 1133

### *Verifying AppTrack Operation*

**Purpose** View the AppTrack counters periodically to monitor tracking.

**Action** From operational mode, enter the **show application-tracking counters** command.

### *Verifying Security Flow Session Statistics*

**Purpose** Compare byte and packet counts in logged messages with the session statistics from the **show security flow session** command output.

**Action** From operational mode, enter the **show security flow session** command.

**Verifying Application System Cache Statistics**

**Purpose** Compare cache statistics such as IP address, port, protocol, and service for an application from the **show services application-identification application-system-cache** command output.

**Action** From operational mode, enter the **show services application-identification application-system-cache** command.

**Verifying the Status of Application Identification Counter Values**

**Purpose** Compare session statistics for application identification counter values from the **show services application-identification counter** command output.

**Action** From operational mode, enter the **show services application-identification counter** command.

**Related Documentation** • [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

**Example: Verifying AppTrack Operation (CLI)**

The following examples provide two ways to monitor AppTrack operation.

- View AppTrack counters periodically to monitor tracking.

```
user@host> show security application-tracking counters
```

AVT counters:	Value
Session create messages	1
Session close messages	1
Session volume updates	0
Failed messages	0

- Compare byte and packet counts in logged messages with the session statistics from the **show** command output.

```
user@host> show security flow session
```

```
Flow Sessions on FPC6 PIC0:
```

```
Session ID: 120000044, Policy name: policy-in-out/4, Timeout: 1796, Valid
In: 4.0.0.1/39075 --> 5.0.0.1/21;tcp, If: ge-0/0/0.0, Pkts: 22, Bytes:
1032
Out: 5.0.0.1/21 --> 4.0.0.1/39075;tcp, If: ge-0/0/1.0, Pkts: 24, Bytes:
1442
```

```
Valid sessions: 1
Pending sessions: 0
Invalidated sessions: 0
Sessions in other states: 0
Total sessions: 1
```

Byte and packet totals in the session statistics should approximate the counts logged by AppTrack but might not be exactly the same. AppTrack counts only incoming bytes

and packets. System-generated packets are not included in the total, and dropped packets are not deducted.

For command option descriptions and values, see the *Junos OS CLI Reference*.

For general information about monitoring events and managing system log files, see the *Junos OS Administration Guide for Security Devices*.

PART 11

# Chassis Cluster

- [Chassis Cluster on page 1137](#)



# Chassis Cluster

- Chassis Cluster Overview on page 1137
- Understanding Chassis Cluster Formation on page 1138
- Chassis Cluster Redundancy Groups on page 1139
- Chassis Cluster Redundant Ethernet Interfaces on page 1162
- Conditional Route Advertising in a Chassis Cluster on page 1175
- Chassis Cluster Control Plane on page 1181
- Chassis Cluster Data Plane on page 1197
- Consequences of Enabling Chassis Cluster on page 1210
- Building a Chassis Cluster on page 1222
- Chassis Cluster Upgrades on page 1250
- Disabling Chassis Cluster on page 1254
- Understanding Multicast Routing on a Chassis Cluster on page 1254
- Asymmetric Chassis Cluster Deployment on page 1255
- Active/Passive Chassis Cluster Deployment (J Series Devices) on page 1269
- Active/Passive Chassis Cluster Deployment (SRX Series Devices) on page 1283
- Active/Passive Chassis Cluster Deployment with an IPsec Tunnel on page 1312

## Chassis Cluster Overview

---

Chassis clustering provides network node redundancy by grouping a pair of the same kind of supported SRX Series devices or J Series devices into a cluster. The devices must be running the same version of Junos OS. The control ports on the respective nodes are connected to form a control plane that synchronizes configuration and kernel state to facilitate the high availability of interfaces and services. Similarly, the data plane on the respective nodes is connected over the fabric ports to form a unified data plane. The fabric link allows for the management of cross-node flow processing and for the management of session redundancy.

The control plane software operates in active or backup mode. When configured as a chassis cluster, the two nodes back up each other, with one node acting as the primary device and the other as the secondary device, ensuring stateful failover of processes and

services in the event of system or hardware failure. If the primary device fails, the secondary device takes over processing of traffic.

The data plane software operates in active/active mode. In a chassis cluster, session information is updated as traffic traverses either device, and this information is transmitted between the nodes over the fabric link to guarantee that established sessions are not dropped when a failover occurs. In active/active mode, it is possible for traffic to ingress the cluster on one node and egress from the other node.



**NOTE:** In Junos OS Release 10.4, SRX Series and J Series devices running IP version 6 (IPv6) can be deployed in active/active (failover) chassis cluster configurations in addition to the existing support of active/passive (failover) chassis cluster configurations. An interface can be configured with an IPv4 address, IPv6 address, or both. Address book entries can include any combination of IPv4 addresses, IPv6 addresses, and Domain Name System (DNS) names.

Chassis cluster functionality includes:

- Resilient system architecture, with a single active control plane for the entire cluster and multiple Packet Forwarding Engines. This architecture presents a single device view of the cluster.
- Synchronization of configuration and dynamic runtime states between nodes within a cluster.
- Monitoring of physical interfaces, and failover if the failure parameters cross a configured threshold.

The different states that a cluster can be in at any given instant are as follows: hold, primary, secondary-hold, secondary, ineligible, and disabled. A state transition can be triggered because of any event, such as interface monitoring, SPU monitoring, failures, and manual failovers.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Formation on page 1138](#)
- [Understanding Chassis Cluster Redundancy Groups on page 1139](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces on page 1162](#)

---

## Understanding Chassis Cluster Formation

To form a chassis cluster, a pair of the same kind of supported SRX Series devices or J Series devices are combined to act as a single system that enforces the same overall security. For SRX5600 and SRX5800 chassis clusters, the placement and type of Services Processing Cards (SPCs) must match in the two clusters. For SRX3400 and SRX3600 chassis clusters, the placement and type of SPCs, I/O cards (IOCs), and Network Processing Cards (NPCs) must match in the two devices.



For SRX Series branch devices (SRX100, SRX210, SRX220, SRX240, and SRX650) and J Series chassis clusters, although the devices must be the same kind, they can contain different Physical Interface Modules (PIMs).

When a device joins a cluster, it becomes a node of that cluster. With the exception of unique node settings and management IP addresses, nodes in a cluster share the same configuration.

You can deploy up to 15 chassis clusters in a Layer 2 domain. Clusters and nodes are identified in the following way:

- A cluster is identified by a *cluster ID* (**cluster-id**) specified as a number from 1 through 15.
- A cluster node is identified by a *node ID* (**node**) specified as a number from 0 to 1.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Chassis Cluster Overview on page 1137](#)
- [Understanding Control Link VLAN Tagging on a Chassis Cluster](#)
- [Understanding Chassis Cluster Redundancy Groups on page 1139](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces on page 1162](#)
- [Understanding the Chassis Cluster Control Plane on page 1182](#)
- [Understanding the Chassis Cluster Data Plane on page 1197](#)
- [Understanding What Happens When Chassis Cluster Is Enabled on page 1210](#)

## Chassis Cluster Redundancy Groups

- [Understanding Chassis Cluster Redundancy Groups on page 1139](#)
- [Chassis Cluster Redundancy Groups 0 Through 128 on page 1140](#)
- [Chassis Cluster Redundancy Group Interface Monitoring on page 1146](#)
- [Chassis Cluster Redundancy Group IP Address Monitoring on page 1148](#)
- [Understanding Chassis Cluster Monitoring of Global-Level Objects on page 1154](#)
- [Chassis Cluster Redundancy Group Failover on page 1156](#)

### Understanding Chassis Cluster Redundancy Groups

Chassis clustering provides high availability of interfaces and services through redundancy groups and primacy within groups.

A redundancy group is an abstract construct that includes and manages a collection of objects. A redundancy group contains objects on both nodes. A redundancy group is primary on one node and backup on the other at any time. When a redundancy group is said to be primary on a node, its objects on that node are active.

Redundancy groups are independent units of failover. Each redundancy group fails over from one node to the other independent of other redundancy groups. When a redundancy group fails over, all its objects fail over together.

Three things determine the primacy of a redundancy group: the priority configured for the node, the node ID (in case of tied priorities), and the order in which the node comes up. If a lower priority node comes up first, then it will assume the primacy for a redundancy group (and will stay as primary if preempt is not enabled).

A chassis cluster can include many redundancy groups, some of which might be primary on one node and some of which might be primary on the other. Alternatively, all redundancy groups can be primary on a single node. One redundancy group's primacy does not affect another redundancy group's primacy. You can create up to 128 redundancy groups.



**NOTE:** The maximum number of redundancy groups is equal to the number of redundant Ethernet interfaces that you configure.

---

You can configure redundancy groups to suit your deployment. You configure a redundancy group to be primary on one node and backup on the other node. You specify the node on which the group is primary by setting priorities for both nodes within a redundancy group configuration. The node with the higher priority takes precedence, and the redundancy group's objects on it are active.

If a redundancy group is configured so that both nodes have the same priority, the node with the lowest node ID number always takes precedence, and the redundancy group is primary on it. In a two-node cluster, node 0 always takes precedence in a priority tie.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Redundancy Group 0: Routing Engines](#) on page 1140
- [Understanding Chassis Cluster Redundancy Groups 1 Through 128](#) on page 1141
- [Understanding Chassis Cluster Redundancy Group Interface Monitoring](#) on page 1146
- [Understanding Chassis Cluster Redundancy Group IP Address Monitoring](#) on page 1148
- [Understanding Chassis Cluster Redundancy Group Failover](#) on page 1156
- [Understanding Chassis Cluster Formation](#) on page 1138

## Chassis Cluster Redundancy Groups 0 Through 128

- [Understanding Chassis Cluster Redundancy Group 0: Routing Engines](#) on page 1140
- [Understanding Chassis Cluster Redundancy Groups 1 Through 128](#) on page 1141
- [Example: Configuring Chassis Cluster Redundancy Groups](#) on page 1144

### [Understanding Chassis Cluster Redundancy Group 0: Routing Engines](#)

---

When you initialize a device in chassis cluster mode, the system creates a redundancy group referred to as redundancy group 0. Redundancy group 0 manages the primacy

and failover between the Routing Engines on each node of the cluster. As is the case for all redundancy groups, redundancy group 0 can be primary on only one node at a time. The node on which redundancy group 0 is primary determines which Routing Engine is active in the cluster. A node is considered the primary node of the cluster if its Routing Engine is the active one.

The redundancy group 0 configuration specifies the priority for each node. The following priority scheme determines redundancy group 0 primacy. Note that the three-second value is the interval if the default **heartbeat-threshold** and **heartbeat-interval** values are used.

- The node that comes up first (at least three seconds prior to the other node) is the primary node.
- If both nodes come up at the same time (or within three seconds of each other):
  - The node with the higher configured priority is the primary node.
  - If there is a tie (either because the same value was configured or because default settings were used), the node with the lower node ID (node 0) is the primary node.

The previous priority scheme applies to redundancy groups *x* (redundancy groups numbered 1 through 128) as well, provided preempt is not configured. (See “Understanding Chassis Cluster Redundancy Groups 1 Through 128” on page 1141.)

You cannot enable preemption for redundancy group 0. If you want to change the primary node for redundancy group 0, you must do a manual failover.



**CAUTION:** Be cautious and judicious in your use of redundancy group 0 manual failovers. A redundancy group 0 failover implies a Routing Engine failover, in which case all processes running on the primary node are killed and then spawned on the new master Routing Engine. This failover could result in loss of state, such as routing state, and degrade performance by introducing system churn.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Redundancy Groups 1 Through 128 on page 1141](#)
- [Example: Configuring Chassis Cluster Redundancy Groups on page 1144](#)
- [Understanding Chassis Cluster Redundancy Group Failover on page 1156](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces on page 1162](#)
- [Understanding Chassis Cluster Formation on page 1138](#)

#### [Understanding Chassis Cluster Redundancy Groups 1 Through 128](#)

You can configure one or more redundancy groups numbered 1 through 128, referred to as redundancy group *x*. The maximum number of redundancy groups is equal to the number of redundant Ethernet interfaces that you configure (see Table 112 on page 1163).

Each redundancy group *x* acts as an independent unit of failover and is primary on only one node at a time.

Each redundancy group *x* contains one or more redundant Ethernet interfaces. A redundant Ethernet interface is a pseudointerface that contains at minimum a pair of physical Gigabit Ethernet interfaces or a pair of Fast Ethernet interfaces. If a redundancy group is active on node 0, then the child links of all the associated redundant Ethernet interfaces on node 0 are active. If the redundancy group fails over to node 1, then the child links of all redundant Ethernet interfaces on node 1 become active.

The following priority scheme determines redundancy group *x* primacy, provided preempt is not configured. If preempt is configured, the node with the higher priority is the primary node. Note that the three-second value is the interval if the default **heartbeat-threshold** and **heartbeat-interval** values are used.

- The node that comes up first (at least three seconds prior to the other node) is the primary node.
- If both nodes come up at the same time (or within three seconds of each other):
  - The node with the higher configured priority is the primary node.
  - If there is a tie (either because the same value was configured or because default settings were used), the node with the lower node ID (node 0) is the primary node.

On SRX Series and J Series chassis clusters, you can configure multiple redundancy groups to load-share traffic across the cluster. For example, you can configure some redundancy groups *x* to be primary on one node and some redundancy groups *x* to be primary on the other node. You can also configure a redundancy group *x* in a one-to-one relationship with a single redundant Ethernet interface to control which interface traffic flows through.

The traffic for a redundancy group is processed on the node where the redundancy group is active. Because more than one redundancy group can be configured, it is possible that the traffic from some redundancy groups will be processed on one node while the traffic for other redundancy groups is processed on the other node (depending on where the redundancy group is active). Multiple redundancy groups make it possible for traffic to arrive over an ingress interface of one redundancy group and over an egress interface that belongs to another redundancy group. In this situation, the ingress and egress interfaces might not be active on the same node. When this happens, the traffic is forwarded over the fabric link to the appropriate node.

When you configure a redundancy group *x*, you must specify a priority for each node to determine the node on which the redundancy group *x* is primary. The node with the higher priority is selected as primary. The primacy of a redundancy group *x* can fail over from one node to the other. When a redundancy group *x* fails over to the other node, its redundant Ethernet interfaces on that node are active and their interfaces are passing traffic.

Table 111 on page 1143 gives an example of redundancy group *x* in an SRX Series chassis cluster and indicates the node on which the group is primary. It shows the redundant Ethernet interfaces and their interfaces configured for redundancy group *x*.



**NOTE:** Some devices have both Gigabit Ethernet ports and Fast Ethernet ports.

**Table 111: Example of Redundancy Groups in a Chassis Cluster**

Group	Primary	Priority	Objects	Interface (Node 0)	Interface (Node 1)
Redundancy group 0	Node 0	Node 0: 254	Routing Engine on node 0	—	—
		Node 1: 2	Routing Engine on node 1	—	—
Redundancy group 1	Node 0	Node 0: 254	Redundant Ethernet interface 0	<b>ge-1/0/0</b>	<b>ge-5/0/0</b>
		Node 1: 2	Redundant Ethernet interface 1	<b>ge-1/3/0</b>	<b>ge-5/3/0</b>
Redundancy group 2	Node 1	Node 0: 2	Redundant Ethernet interface 2	<b>ge-2/0/0</b>	<b>ge-6/0/0</b>
		Node 1: 254	Redundant Ethernet interface 3	<b>ge-2/3/0</b>	<b>ge-6/3/0</b>
Redundancy group 3	Node 0	Node 0: 254	Redundant Ethernet interface 4	<b>ge-3/0/0</b>	<b>ge-7/0/0</b>
		Node 1: 2	Redundant Ethernet interface 5	<b>ge-3/3/0</b>	<b>ge-7/3/0</b>

As the example for a chassis cluster in Table 111 on page 1143 shows:

- The Routing Engine on node 0 is active because redundancy group 0 is primary on node 0. (The Routing Engine on node 1 is passive, serving as backup.)
- Redundancy group 1 is primary on node 0. Interfaces **ge-1/0/0** and **ge-1/3/0** belonging to redundant Ethernet interface 0 and redundant Ethernet interface 1 are active and handling traffic.

- Redundancy group 2 is primary on node 1. Interfaces **ge-6/0/0** and **ge-6/3/0** belonging to redundant Ethernet interface 2 and redundant Ethernet interface 3 are active and handling traffic.
- Redundancy group 3 is primary on node 0. Interfaces **ge-3/0/0** and **ge-3/3/0** belonging to redundant Ethernet interface 4 and redundant Ethernet interface 5 are active and handling traffic.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Redundancy Group 0: Routing Engines on page 1140](#)
- [Example: Configuring Chassis Cluster Redundancy Groups on page 1144](#)
- [Understanding Chassis Cluster Redundancy Group Failover on page 1156](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces on page 1162](#)
- [Understanding Chassis Cluster Formation on page 1138](#)

### [Example: Configuring Chassis Cluster Redundancy Groups](#)

---

This example shows how to configure a chassis cluster redundancy group.

- [Requirements on page 1144](#)
- [Overview on page 1144](#)
- [Configuration on page 1145](#)
- [Verification on page 1146](#)

**Requirements**

Before you begin:

1. Set the chassis cluster node ID and cluster ID. See “Example: Setting the Chassis Cluster Node ID and Cluster ID” on page 1240.
2. Configure the chassis cluster management interface. See “Example: Configuring Chassis Cluster Management Interface” on page 1242.
3. Configure the chassis cluster fabric. See “Example: Configuring the Chassis Cluster Fabric” on page 1201.

**Overview**

A chassis cluster redundancy group is an abstract entity that includes and manages a collection of objects. Each redundancy group acts as an independent unit of failover and is primary on only one node at a time.

In this example, you create two chassis cluster redundancy groups, 0 and 1:

- 0—Node 0 is assigned a priority of 100, and node 1 is assigned a priority of 1.
- 1—Node 0 is assigned a priority of 100, and node 1 is assigned a priority of 1.

The preempt option is enabled, and the number of gratuitous ARP requests that an interface can send to notify other network devices of its presence after the redundancy group it belongs to has failed over is 4.

### Configuration

**CLI Quick Configuration** To quickly configure a chassis cluster redundancy group, copy the following commands and paste them into the CLI:

```
[edit]
set chassis cluster redundancy-group 0 node 0 priority 100
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 100
set chassis cluster redundancy-group 1 node 1 priority 1
set chassis cluster redundancy-group 1 preempt
set chassis cluster redundancy-group 1 gratuitous-arp-count 4
```

**Step-by-Step Procedure** To configure a chassis cluster redundancy group:

1. Specify a redundancy group's priority for primacy on each node of the cluster. The higher number takes precedence.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 0 node 0 priority 100
user@host# set chassis cluster redundancy-group 0 node 1 priority 1
user@host# set chassis cluster redundancy-group 1 node 0 priority 100
user@host# set chassis cluster redundancy-group 1 node 1 priority 1
```

2. Specify whether a node with a higher priority can initiate a failover to become primary for the redundancy group.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 preempt
```

3. Specify the number of gratuitous ARP requests that an interface can send to notify other network devices of its presence after the redundancy group it belongs to has failed over.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 gratuitous-arp-count 4
```

**Results** From configuration mode, confirm your configuration by entering the **show chassis cluster status redundancy-group** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
{primary:node0}[edit]
user@host# show chassis cluster
chassis {
  cluster {
    redundancy-group 0 {
      node 0 priority 100;
      node 1 priority 1;
    }
    redundancy-group 1 {
      node 0 priority 100;
      node 1 priority 1;
    }
  }
}
```

```

    preempt;
    gratuitous-arp-count 4;
  }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Chassis Cluster Redundancy Group Status on page 1146

### Verifying Chassis Cluster Redundancy Group Status

**Purpose** Verify the status of a chassis cluster redundancy group.

**Action** From operational mode, enter the **show chassis cluster status redundancy-group** command.

```

{primary:node0}
user@host>show chassis cluster status redundancy-group 1

Cluster ID: 1
Node          Priority      Status    Preempt  Manual failover

Redundancy group: 1 , Failover count: 1
node0         100          secondary no        no
node1         1            primary  yes       no

```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Chassis Cluster Redundancy Group 0: Routing Engines on page 1140
- Understanding Chassis Cluster Redundancy Groups 1 Through 128 on page 1141
- Understanding Chassis Cluster Redundancy Group Failover on page 1156
- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 1162
- Understanding Chassis Cluster Formation on page 1138

## Chassis Cluster Redundancy Group Interface Monitoring

- Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 1146
- Example: Configuring Chassis Cluster Interface Monitoring on page 1147

### Understanding Chassis Cluster Redundancy Group Interface Monitoring

For a redundancy group to automatically fail over to another node, its interfaces must be monitored. When you configure a redundancy group, you can specify a set of interfaces that the redundancy group is to monitor for status (or “health”) to determine whether the interface is up or down. A monitored interface can be a child interface of any of its redundant Ethernet interfaces. When you configure an interface for a redundancy group to monitor, you give it a weight.



Every redundancy group has a threshold tolerance value initially set to **255**. When an interface monitored by a redundancy group becomes unavailable, its weight is subtracted from the redundancy group's threshold. When a redundancy group's threshold reaches 0, it fails over to the other node. For example, if redundancy group 1 was primary on node 0, on the threshold-crossing event, redundancy group 1 becomes primary on node 1. In this case, all the child interfaces of redundancy group 1's redundant Ethernet interfaces begin handling traffic.

A redundancy group failover occurs because the cumulative weight of the redundancy group's monitored interfaces has brought its threshold value to 0. When the monitored interfaces of a redundancy group on both nodes reach their thresholds at the same time, the redundancy group is primary on the node with the lower node ID, in this case node 0.



**NOTE:** If you want to dampen the failovers occurring because of interface monitoring failures, use the `hold-down-interval` statement.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Redundancy Group 0: Routing Engines on page 1140](#)
- [Understanding Chassis Cluster Redundancy Groups 1 Through 128 on page 1141](#)
- [Example: Configuring Chassis Cluster Interface Monitoring on page 1147](#)
- [Understanding Chassis Cluster Redundancy Group IP Address Monitoring on page 1148](#)
- [Understanding Chassis Cluster Redundancy Group Failover on page 1156](#)
- [Understanding Chassis Cluster Monitoring of Global-Level Objects on page 1154](#)

#### Example: Configuring Chassis Cluster Interface Monitoring

This example shows how to specify that an interface be monitored by a specific redundancy group for automatic failover to another node. You assign a weight to the interface to be monitored.

- [Requirements on page 1147](#)
- [Overview on page 1147](#)
- [Configuration on page 1148](#)
- [Verification on page 1148](#)

##### **Requirements**

Before you begin, create a redundancy group. See “Example: Configuring Chassis Cluster Redundancy Groups” on page 1144.

##### **Overview**

You can configure your system to monitor the health of the interfaces belonging to a redundancy group. When you assign a weight to an interface to be monitored, the system monitors the interface for availability. If a physical interface fails, the weight is deducted from the corresponding redundancy group's threshold. Every redundancy group has a

threshold of 255. If the threshold hits 0, a failover is triggered, even if the redundancy group is in manual failover mode and the preempt option is not enabled.

In this example, you configure a weight of 255 to the ge-7/0/3 interface in the redundancy group named 1.

### Configuration

#### Step-by-Step Procedure

To configure chassis cluster interface monitoring:

1. Specify the interface to be monitored by a redundancy group.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-7/0/3
weight 255
```

2. If you are done configuring the device, commit the configuration.

```
{primary:node0}[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show chassis cluster interfaces** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Redundancy Group 0: Routing Engines on page 1140](#)
- [Understanding Chassis Cluster Redundancy Groups 1 Through 128 on page 1141](#)
- [Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 1146](#)
- [Understanding Chassis Cluster Redundancy Group IP Address Monitoring on page 1148](#)
- [Understanding Chassis Cluster Redundancy Group Failover on page 1156](#)
- [Understanding Chassis Cluster Monitoring of Global-Level Objects on page 1154](#)

## Chassis Cluster Redundancy Group IP Address Monitoring

- [Understanding Chassis Cluster Redundancy Group IP Address Monitoring on page 1148](#)
- [Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring on page 1151](#)

### Understanding Chassis Cluster Redundancy Group IP Address Monitoring

Redundancy group IP address monitoring checks end-to-end connectivity and allows a redundancy group to fail over because of the inability of a redundant Ethernet interface (known as a *reth*) to reach a configured IP address. Redundancy groups on both devices in a cluster can be configured to monitor specific IP addresses to determine whether an upstream device in the network is reachable. The redundancy group can be configured such that if the monitored IP address becomes unreachable, the redundancy group will fail over to its backup to maintain service. The primary difference between this monitoring feature and interface monitoring is that IP address monitoring allows for failover when the interface is still up but the network device it is connected to is not reachable for some

reason. It may be possible under those circumstances for the other node in the cluster to route traffic around the problem.



**NOTE:** If you want to dampen the failovers occurring because of IP address monitoring failures, use the `hold-down-interval` statement.

IP address monitoring configuration allows you to set not only the address to monitor and its failover weight but also a global IP address monitoring threshold and weight. Only after the IP address monitoring global-threshold is reached because of cumulative monitored address reachability failure will the IP address monitoring global-weight value be deducted from the redundant group's failover threshold. Thus, multiple addresses can be monitored simultaneously as well as monitored to reflect their importance to maintaining traffic flow. Also, the threshold value of an IP address that is unreachable and then becomes reachable again will be restored to the monitoring threshold. This will not, however, cause a failback unless the `preempt` option has been enabled.

When configured, the IP address monitoring failover value (global-weight) is considered along with interface monitoring—if set—and built-in failover monitoring, including SPU monitoring, cold-sync monitoring, and NPC monitoring (on supported platforms). The main IP addresses that should be monitored are router gateway addresses to ensure that valid traffic coming into the services gateway can be forwarded to the appropriate network router.

One Services Processing Unit (SPU) or Packet Forwarding Engine (PFE) per node is designated to send Internet Control Message Protocol (ICMP) ping packets for the monitored IP addresses on the cluster. The primary PFE sends ping packets using Address Resolution Protocol (ARP) requests resolved by the Routing Engine (RE). The source for these pings is the redundant Ethernet interface MAC and IP addresses. The secondary PFE resolves ARP requests for the monitored IP address itself. The source for these pings is the physical child MAC address and a secondary IP address configured on the redundant Ethernet interface. For the ping reply to be received on the secondary interface, the I/O card (IOC), central PFE processor, or Flex IOC adds both the physical child MAC address and the redundant Ethernet interface MAC address to its MAC table. The secondary PFE responds with the physical child MAC address to ARP requests sent to the secondary IP address configured on the redundant Ethernet interface.



**NOTE:** If the redundant Ethernet interface belongs to a VPN routing and forwarding (VRF) routing instance type, then IP address monitoring will not work.

The default interval to check the reachability of a monitored IP address is once per second. The interval can be adjusted using the `retry-interval` command. The default number of permitted consecutive failed ping attempts is 5. The number of allowed consecutive failed ping attempts can be adjusted using the `retry-count` command. After failing to reach a monitored IP address for the configured number of consecutive attempts, the IP address is determined to be unreachable and its failover value is deducted from the redundancy group's global-threshold.

Once the IP address is determined to be unreachable, its weight is deducted from the global-threshold. If the recalculated global-threshold value is not 0, the IP address is marked unreachable, but the global-weight is not deducted from the redundancy group's threshold. If the redundancy group IP monitoring global-threshold reaches 0 and there are unreachable IP addresses, the redundancy group will continuously fail over and fail back between the nodes until either an unreachable IP address becomes reachable or a configuration change removes unreachable IP addresses from monitoring. Note that both default and configured hold-down-interval failover dampening is still in effect.

Every redundancy group *x* has a threshold tolerance value initially set to 255. When an IP address monitored by redundancy group *x* becomes unavailable, its weight is subtracted from the redundancy group *x*'s threshold. When redundancy group *x*'s threshold reaches 0, it fails over to the other node. For example, if redundancy group 1 was primary on node 0, on the threshold-crossing event, redundancy group 1 becomes primary on node 1. In this case, all the child interfaces of redundancy group 1's redundant Ethernet interfaces begin handling traffic.

A redundancy group *x* failover occurs because the cumulative weight of the redundancy group *x*'s monitored IP addresses and other monitoring has brought its threshold value to 0. When the monitored IP addresses of redundancy group *x* on both nodes reach their thresholds at the same time, redundancy group *x* is primary on the node with the lower node ID, which is typically node 0.

Monitoring can be accomplished only if the IP address is reachable on a redundant Ethernet interface (known as a reth in CLI commands and interface listings), and IP addresses cannot be monitored over a tunnel. For an IP address to be monitored through a redundant Ethernet interface on a secondary cluster node, the interface must have a secondary IP address configured. IP address monitoring cannot be used on a chassis cluster running in transparent mode. The maximum number of monitoring IP addresses that can be configured per cluster is 64 for the SRX5000 line and 32 for the SRX1400 device and the SRX3000 line.



**NOTE:** Redundancy group IP address monitoring is not supported for IPv6 destinations in this release.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Redundancy Group 0: Routing Engines](#) on page 1140
- [Understanding Chassis Cluster Redundancy Groups 1 Through 128](#) on page 1141
- [Understanding Chassis Cluster Redundancy Group Interface Monitoring](#) on page 1146
- [Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring](#) on page 1151
- [Understanding Chassis Cluster Redundancy Group Failover](#) on page 1156
- [Understanding Chassis Cluster Monitoring of Global-Level Objects](#) on page 1154

### Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring

This example shows how to configure redundancy group IP address monitoring for a chassis cluster.

- Requirements on page 1151
- Overview on page 1151
- Configuration on page 1152
- Verification on page 1153

#### **Requirements**

Before you begin:

1. Set the chassis cluster node ID and cluster ID. See “Example: Setting the Chassis Cluster Node ID and Cluster ID” on page 1240.
2. Configure the chassis cluster management interface. See “Example: Configuring Chassis Cluster Management Interface” on page 1242.
3. Configure the chassis cluster fabric. See “Example: Configuring the Chassis Cluster Fabric” on page 1201.

#### **Overview**

You can configure redundancy groups to monitor upstream resources by pinging specific IP addresses that are reachable through redundant Ethernet interfaces on either node in a cluster. You can also configure global threshold, weight, retry interval, and retry count parameters for a redundancy group. When a monitored IP address becomes unreachable, the weight of that monitored IP address is deducted from the redundancy group IP address monitoring global threshold. When the global threshold reaches 0, the global weight is deducted from the redundancy group threshold. The retry interval determines the ping interval for each IP address monitored by the redundancy group. The pings are sent as soon as the configuration is committed. The retry count sets the number of allowed consecutive ping failures for each IP address monitored by the redundancy group.

In this example, you configure the following settings for redundancy group 1:

- IP address to monitor—10.1.1.10
- IP address monitoring global-weight—100
- IP address monitoring global-threshold—200



**NOTE:** The threshold applies cumulatively to all IP addresses monitored by the redundancy group.

- IP address retry-interval—3 seconds
- IP address retry-count—10
- Weight—150

- Redundant Ethernet interface—reth1.0
- Secondary IP address—10.1.1.101

### Configuration

**CLI Quick Configuration** To quickly configure redundancy group IP address monitoring, copy the following commands and paste them into the CLI:

```
{primary:node0}[edit]
user@host#
set chassis cluster redundancy-group 1 ip-monitoring global-weight 100
set chassis cluster redundancy-group 1 ip-monitoring global-threshold 200
set chassis cluster redundancy-group 1 ip-monitoring retry-interval 3
set chassis cluster redundancy-group 1 ip-monitoring retry-count 10
set chassis cluster redundancy-group 1 ip-monitoring family inet 10.1.1.10 weight 150
interface reth1.0 secondary-ip-address 10.1.1.101
```

**Step-by-Step Procedure** To configure redundancy group IP address monitoring:

1. Specify a global monitoring weight.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 ip-monitoring global-weight
100
```

2. Specify the global monitoring threshold.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 ip-monitoring global-threshold
200
```

3. Specify the retry interval.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 ip-monitoring retry-interval 3
```

4. Specify the retry count.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 ip-monitoring retry-count 10
```

5. Specify the IP address to be monitored, weight, redundant Ethernet interface, and secondary IP address.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 ip-monitoring family inet 10.1.1.10
weight 100 interface reth1.0 secondary-ip-address 10.1.1.101
```

**Results** From configuration mode, confirm your configuration by entering the **show chassis cluster redundancy-group 1** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
{primary:node0}[edit]
```

```

user@host# show chassis cluster redundancy-group 1
ip-monitoring {
  global-weight 100;
  global-threshold 200;
  family {
    inet {
      10.1.1.10 {
        weight 100;
        interface reth1.0 secondary-ip-address 10.1.1.101;
      }
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying the Status of Monitored IP Addresses for a Redundancy Group on page 1153

### Verifying the Status of Monitored IP Addresses for a Redundancy Group

**Purpose** Verify the status of monitored IP addresses for a redundancy group.

**Action** From operational mode, enter the **show chassis cluster ip-monitoring status** command. For information about a specific group, enter the **show chassis cluster ip-monitoring status redundancy-group** command.

```

{primary:node0}
user@host> show chassis cluster ip-monitoring status

```

node0:

-----

Redundancy group: 1

IP address	Status	Failure count	Reason
10.1.1.10	reachable	0	n/a
10.1.1.101	reachable	0	n/a

node1:

-----

Redundancy group: 1

IP address	Status	Failure count	Reason
10.1.1.10	reachable	0	n/a
10.1.1.101	reachable	0	n/a

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Chassis Cluster Redundancy Group 0: Routing Engines on page 1140
- Understanding Chassis Cluster Redundancy Groups 1 Through 128 on page 1141
- Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 1146

- Understanding Chassis Cluster Redundancy Group IP Address Monitoring on page 1148
- Understanding Chassis Cluster Redundancy Group Failover on page 1156
- Understanding Chassis Cluster Monitoring of Global-Level Objects on page 1154

## Understanding Chassis Cluster Monitoring of Global-Level Objects

There are various types of objects to monitor as you work with devices configured as chassis clusters, including global-level objects and objects that are specific to redundancy groups. This section describes the monitoring of global-level objects.

The SRX1400 device and the SRX3000 and SRX5000 lines have one or more Services Processing Units (SPUs) that run on a Services Processing Card (SPC). All flow-based services run on the SPU. Other SRX Series devices and all J Series devices have a flow-based forwarding process, *flowd*, which forwards packets through the device.

- Understanding SPU Monitoring on page 1154
- Understanding Flowd Monitoring on page 1154
- Understanding Cold-Sync Monitoring on page 1155

### Understanding SPU Monitoring

SPU monitoring tracks the health of the SPUs and of the central point (CP). The chassis manager on each SPC monitors the SPUs and the central point, and also maintains the heartbeat with the Routing Engine chassisd. In this hierarchical monitoring system, chassisd is the center for hardware failure detection. SPU monitoring is enabled by default.



**NOTE:** SPU monitoring is supported on the SRX1400 device and the SRX3000 and SRX5000 lines.

Persistent SPU and central point failure on a node is deemed a catastrophic Packet Forwarding Engine (PFE) failure. In this case, the node's PFE is disabled in the cluster by reducing the priorities of redundancy groups *x* to 0.

- A central point failure triggers failover to the secondary node. The failed node's PFE, which includes all SPCs and all I/O cards (IOCs), is automatically restarted. If the secondary central point has failed as well, the cluster is unable to come up because there is no primary device. Only the data plane (redundancy group *x*) is failed over.
- A single, failed SPU causes failover of redundancy group *x* to the secondary node. All IOCs and SPCs on the failed node are restarted and redundancy group *x* is failed over to the secondary node. Failover to the secondary node is automatic without the need for user intervention. When the failed (former) primary node has its failing component restored, failback is determined by the preempt configuration for the redundancy group *x*. The interval for dead SPU detection is 30 seconds.

### Understanding Flowd Monitoring

Flowd monitoring tracks the health of the flowd process. Flowd monitoring is enabled by default.





**NOTE:** Flowd monitoring is supported on SRX100, SRX210, and SRX220 devices. It is not supported on J Series devices.

Persistent flowd failure on a node is deemed a catastrophic Packet Forwarding Engine (PFE) failure. In this case, the node's PFE is disabled in the cluster by reducing the priorities of redundancy groups  $x$  to 0.

A failed flowd process causes failover of redundancy group  $x$  to the secondary node. Failover to the secondary node is automatic without the need for user intervention. When the failed (former) primary node has its failing component restored, failback is determined by the preempt configuration for the redundancy group  $x$ .

### Understanding Cold-Sync Monitoring

The process of synchronizing the data plane runtime objects (RTOs) on the startup of the SPUs or flowd is called *cold sync*. When all the RTOs are synchronized, the cold-sync process is complete, and the SPU or flowd on the node is ready to take over for the primary node, if needed. The process of monitoring the cold-sync state of all the SPUs or flowd on a node is called *cold-sync monitoring*. Keep in mind that when preempt is enabled, cold-sync monitoring prevents the node from taking over the mastership until the cold-sync process is completed for the SPUs or flowd on the node. Cold-sync monitoring is enabled by default.

When the node is rebooted, or when the SPUs or flowd come back up from failure, the priority for all the redundancy groups  $x$  is 0. When an SPU or flowd comes up, it tries to start the cold-sync process with its mirror SPU or flowd on the other node.

If this is the only node in the cluster, the priorities for all the redundancy groups  $x$  stay at 0 until a new node joins the cluster. Although the priority is at 0, the device can still receive and send traffic over its interfaces. A priority of 0 implies that it cannot fail over in case of a failure. When a new node joins the cluster, all the SPUs or flowd, as they come up, will start the cold-sync process with the mirror SPUs or flowd of the existing node.

When the SPU or flowd of a node that is already up detects the cold-sync request from the SPU or flowd of the peer node, it posts a message to the system indicating that the cold-sync process is complete. The SPUs or flowd of the newly joined node posts a similar message. However, they post this message only after all the RTOs are learned and cold-sync is complete. On receipt of completion messages from all the SPUs or flowd, the priority for redundancy groups  $x$  moves to the configured priority on each node if there are no other failures of monitored components, such as interfaces. This action ensures that the existing primary node for redundancy  $x$  groups always moves to the configured priority first. The node joining the cluster later moves to its configured priorities only after all its SPUs or flowd have completed their cold-sync process. This action in turn guarantees that the newly added node is ready with all the RTOs before it takes over mastership.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Chassis Cluster Redundancy Group 0: Routing Engines on page 1140

- Understanding Chassis Cluster Redundancy Groups 1 Through 128 on page 1141
- Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 1146
- Understanding Chassis Cluster Redundancy Group IP Address Monitoring on page 1148
- Understanding Chassis Cluster Redundancy Group Failover on page 1156
- Understanding Chassis Cluster Formation on page 1138

## Chassis Cluster Redundancy Group Failover

- Understanding Chassis Cluster Redundancy Group Failover on page 1156
- Understanding Chassis Cluster Redundancy Group Manual Failover on page 1157
- Initiating a Chassis Cluster Manual Redundancy Group Failover on page 1158
- Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers on page 1160
- Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 1161

### Understanding Chassis Cluster Redundancy Group Failover

Chassis cluster employs a number of highly efficient failover mechanisms that promote high availability to increase your system's overall reliability and productivity.

A redundancy group is a collection of objects that fail over as a group. Each redundancy group monitors a set of objects (physical interfaces), and each monitored object is assigned a weight. Each redundancy group has an initial threshold of **255**. When a monitored object fails, the weight of the object is subtracted from the threshold value of the redundancy group. When the threshold value reaches zero, the redundancy group fails over to the other node. As a result, all the objects associated with the redundancy group fail over as well. Graceful restart of the routing protocols enables the SRX Series device to minimize traffic disruption during a failover.

Because back-to-back redundancy group failovers that occur too quickly can cause a cluster to exhibit unpredictable behavior, a dampening time between failovers is needed. On a failover, the previous primary node moves to the secondary-hold state and stays there until the hold-down interval expires, after which it moves to the secondary state. If a failure is followed by a failure of the new primary node during the hold-down interval, the system fails over immediately.

The default dampening time is 300 seconds (5 minutes) for redundancy group 0 and is configurable to up to 1800 seconds with the **hold-down-interval** statement. For some configurations, such as ones with a large number of routes or logical interfaces, the default interval or the interval you set might not be sufficient. In such cases, the system automatically extends the dampening time in increments of 60 seconds until the system is ready for failover.

Redundancy groups *x* (redundancy groups numbered 1 through 128) have a default dampening time of 1 second, with a range of 0 through 1800 seconds.

The hold-down interval affects manual failovers, as well as automatic failovers associated with monitoring failures.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 1146](#)
- [Understanding Chassis Cluster Redundancy Group Manual Failover on page 1157](#)
- [Initiating a Chassis Cluster Manual Redundancy Group Failover on page 1158](#)
- [Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers on page 1160](#)
- [Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 1161](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces on page 1162](#)
- [Understanding Chassis Cluster Monitoring of Global-Level Objects on page 1154](#)

### [Understanding Chassis Cluster Redundancy Group Manual Failover](#)

You can initiate a redundancy group *x* failover manually. A manual failover applies until a failback event occurs.

For example, suppose that you manually do a redundancy group 1 failover from node 0 to node 1. Then an interface that redundancy group 1 is monitoring fails, dropping the threshold value of the new primary redundancy group to zero. This event is considered a failback event, and the system returns control to the original redundancy group.

You can also initiate a redundancy group 0 failover manually if you want to change the primary node for redundancy group 0. You cannot enable preemption for redundancy group 0.

When you do a manual failover for redundancy group 0, the node in the primary state transitions to the secondary-hold state. The node stays in the secondary-hold state for the default or configured time (a minimum of 300 seconds) and then transitions to the secondary state.

State transitions in cases where one node is in the secondary-hold state and the other node reboots, or the control link connection or fabric link connection is lost to that node, are described as follows:

- Reboot case—The node in the secondary-hold state transitions to the primary state; the other node goes dead (inactive).
- Control link failure case—The node in the secondary-hold state transitions to the ineligible state and then to a disabled state; the other node transitions to the primary state.
- Fabric link failure case—The node in the secondary-hold state transitions directly to the disabled state.

Keep in mind that during an in-service software upgrade (ISSU), the transitions described here cannot happen. Instead, the other (primary) node transitions directly to the secondary

state because Juniper releases earlier than 10.0 do not interpret the secondary-hold state. While you start an ISSU, if one of the nodes has one or more redundancy groups in the secondary-hold state, you must wait for them to move to the secondary state before you can do manual failovers to make all the redundancy groups be primary on one node.



**CAUTION:** Be cautious and judicious in your use of redundancy group 0 manual failovers. A redundancy group 0 failover implies a Routing Engine failover, in which case all processes running on the primary node are killed and then spawned on the new master Routing Engine. This failover could result in loss of state, such as routing state, and degrade performance by introducing system churn.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Redundancy Group Failover on page 1156](#)
- [Initiating a Chassis Cluster Manual Redundancy Group Failover on page 1158](#)
- [Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers on page 1160](#)
- [Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 1161](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces on page 1162](#)
- [Understanding Chassis Cluster Monitoring of Global-Level Objects on page 1154](#)
- [Upgrading Both Devices in a Chassis Cluster Using an ISSU on page 1251](#)

#### Initiating a Chassis Cluster Manual Redundancy Group Failover

You can initiate a failover manually with the **request** command. A manual failover bumps up the priority of the redundancy group for that member to 255.

Before you begin, complete the following tasks:

- [Example: Configuring Chassis Cluster Redundancy Groups on page 1144](#)
- [Example: Configuring Chassis Cluster Redundant Ethernet Interfaces on page 1164](#)
- [Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers on page 1160](#)



**CAUTION:** Be cautious and judicious in your use of redundancy group 0 manual failovers. A redundancy group 0 failover implies a Routing Engine (RE) failover, in which case all processes running on the primary node are killed and then spawned on the new master Routing Engine (RE). This failover could result in loss of state, such as routing state, and degrade performance by introducing system churn.

Use the **show** command to display the status of nodes in the cluster:

```
{primary:node0}
user@host> show chassis cluster status redundancy-group 0
Cluster ID: 9
Node                Priority      Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
  node0              254         primary   no       no
  node1              1          secondary no       no
```

Output to this command indicates that node 0 is primary.

Use the **request** command to trigger a failover and make node 1 primary:

```
{primary:node0}
user@host> request chassis cluster failover redundancy-group 0 node 1
```

```
-----
Initiated manual failover for redundancy group 0
```

Use the **show** command to display the new status of nodes in the cluster:

```
{secondary-hold:node0}
user@host> show chassis cluster status redundancy-group 0
Cluster ID: 9
Node                Priority      Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 2
  node0              254         secondary-hold no       yes
  node1              255         primary   no       yes
```

Output to this command shows that node 1 is now primary and node 0 is in the secondary-hold state. After 5 minutes, node 0 will transition to the secondary state.

You can reset the failover for redundancy groups by using the **request** command. This change is propagated across the cluster.

```
{secondary-hold:node0}
user@host> request chassis cluster failover reset redundancy-group 0 node 0
node0:
```

```
-----
No reset required for redundancy group 0.
```

```
node1:
```

```
-----
Successfully reset manual failover for redundancy group 0
```

You cannot trigger a back-to-back failover until the 5-minute interval expires.

```
{secondary-hold:node0}
user@host> request chassis cluster failover redundancy-group 0 node 0
node0:
```

```
-----
Manual failover is not permitted as redundancy-group 0 on node0 is in
secondary-hold state.
```

Use the **show** command to display the new status of nodes in the cluster:

```
{secondary-hold:node0}
user@host> show chassis cluster status redundancy-group 0
```

```

Cluster ID: 9
Node          Priority      Status      Preempt  Manual failover

Redundancy group: 0 , Failover count: 2
  node0        254         secondary-hold no        no
  node1        1           primary     no        no

```

Output to this command shows that a back-to-back failover has not occurred for either node.

After doing a manual failover, you must issue the **reset failover** command before requesting another failover.

When the primary node fails and comes back up, election of the primary node is done based on regular criteria (priority and preempt).

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Redundancy Group Manual Failover on page 1157](#)
- [Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers on page 1160](#)
- [Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 1161](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces on page 1162](#)
- [Understanding Chassis Cluster Monitoring of Global-Level Objects on page 1154](#)

#### Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers

This example shows how to configure the dampening time between back-to-back redundancy group failovers for a chassis cluster. Back-to-back redundancy group failovers that occur too quickly can cause a chassis cluster to exhibit unpredictable behavior.

##### Requirements

Before you begin:

- Understand redundancy group failover. See “Understanding Chassis Cluster Redundancy Group Failover” on page 1156.
- Understand redundancy group manual failover. See “Understanding Chassis Cluster Redundancy Group Manual Failover” on page 1157.

##### Overview

The dampening time is the minimum interval allowed between back-to-back failovers for a redundancy group. This interval affects manual failovers and automatic failovers caused by interface monitoring failures.

In this example, you set the minimum interval allowed between back-to-back failovers to 420 seconds for redundancy group 0.

**Configuration****Step-by-Step Procedure**

To configure the dampening time between back-to-back redundancy group failovers:

1. Set the dampening time for the redundancy group.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 0 hold-down-interval 420
```

2. If you are done configuring the device, commit the configuration.

```
{primary:node0}[edit]
user@host# commit
```

**Verification**

To verify the configuration is working properly, enter the **show configuration chassis cluster** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Redundancy Group Manual Failover on page 1157](#)
- [Initiating a Chassis Cluster Manual Redundancy Group Failover on page 1158](#)
- [Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 1161](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces on page 1162](#)
- [Understanding Chassis Cluster Monitoring of Global-Level Objects on page 1154](#)

**Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover**

Chassis clustering supports SNMP traps, which are triggered whenever there is a redundancy group failover.

The trap message can help you troubleshoot failovers. It contains the following information:

- The cluster ID and node ID
- The reason for the failover
- The redundancy group that is involved in the failover
- The redundancy group's previous state and current state

These are the different states that a cluster can be in at any given instant: hold, primary, secondary-hold, secondary, ineligible, and disabled. Traps are generated for the following state transitions (only a transition from a hold state does not trigger a trap):

- primary <-> secondary
- primary -> secondary-hold
- secondary-hold -> secondary

- secondary → ineligible
- ineligible → disabled
- ineligible → primary
- secondary → disabled

A transition can be triggered because of any event, such as interface monitoring, SPU monitoring, failures, and manual failovers.

The trap is forwarded over the control link if the outgoing interface is on a node different from the node on the Routing Engine that generates the trap.

You can specify that a trace log be generated by setting the **traceoptions flag snmp** statement.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Redundancy Group Manual Failover on page 1157](#)
- [Initiating a Chassis Cluster Manual Redundancy Group Failover on page 1158](#)
- [Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers on page 1160](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces on page 1162](#)
- [Understanding Chassis Cluster Monitoring of Global-Level Objects on page 1154](#)

---

## Chassis Cluster Redundant Ethernet Interfaces

- [Understanding Chassis Cluster Redundant Ethernet Interfaces on page 1162](#)
- [Example: Configuring Chassis Cluster Redundant Ethernet Interfaces on page 1164](#)
- [Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 1169](#)

### Understanding Chassis Cluster Redundant Ethernet Interfaces

A redundant Ethernet interface is a pseudointerface that includes at minimum one physical interface from each node of the cluster.

A redundant Ethernet interface must contain, at minimum, a pair of Fast Ethernet interfaces or a pair of Gigabit Ethernet interfaces that are referred to as child interfaces of the redundant Ethernet interface (the redundant parent). If two or more child interfaces from each node are assigned to the redundant Ethernet interface, a redundant Ethernet interface link aggregation group must be formed. A single redundant Ethernet interface might include a Fast Ethernet interface from node 0 and a Fast Ethernet interface from node 1 or a Gigabit Ethernet interface from node 0 and a Gigabit Ethernet interface from node 1. Although a redundant Ethernet interface's interfaces must be the same kind—either Fast Ethernet or Gigabit Ethernet—they do not need to be in the same slots on each node.

On SRX1400, SRX3400, SRX3600, SRX5600, and SRX5800 devices, 10-Gigabit Ethernet (xe) interfaces can be redundant Ethernet (reth) interfaces.





**NOTE:** A redundant Ethernet interface is referred to as a `reth` in configuration commands.

The maximum number of redundant Ethernet interfaces that you can configure varies, depending on the device type you are using, as shown in Table 112 on page 1163. Note that the number of redundant Ethernet interfaces configured determines the number of redundancy groups that can be configured.

**Table 112: Maximum Number of Redundant Ethernet Interfaces Allowed**

Device	Maximum Number of reth Interfaces
SRX100	8
SRX210	8
SRX220	8
SRX240	24
SRX650	68
SRX1400	128
SRX3400	128
SRX3600	128
SRX5600	128
SRX5800	128
J2320	128
J2350	128
J4350	128
J6350	128

A redundant Ethernet interface's child interface is associated with the redundant Ethernet interface as part of the child interface configuration. The redundant Ethernet interface child interface inherits most of its configuration from its parent.



**NOTE:** You can enable promiscuous mode on redundant Ethernet interfaces. When promiscuous mode is enabled on a Layer 3 Ethernet interface, all packets received on the interface are sent to the central point or Services Processing Unit (SPU), regardless of the destination MAC address of the packet. If you enable promiscuous mode on a redundant Ethernet interface, promiscuous mode is then enabled on any child physical interfaces.

To enable promiscuous mode on a redundant Ethernet interface, use the **promiscuous-mode** statement at the [edit interfaces] hierarchy.

A redundant Ethernet interface inherits its failover properties from the redundancy group *x* that it belongs to. A redundant Ethernet interface remains active as long as its primary child interface is available or active. For example, if **reth0** is associated with redundancy group 1 and redundancy group 1 is active on node 0, then **reth0** is up as long as the node 0 child of **reth0** is up.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Configuring Chassis Cluster Redundant Ethernet Interfaces on page 1164](#)
- [Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 1169](#)
- [Understanding Conditional Route Advertising in a Chassis Cluster on page 1176](#)
- [Understanding Chassis Cluster Formation on page 1138](#)

### Example: Configuring Chassis Cluster Redundant Ethernet Interfaces

This example shows how to configure chassis cluster redundant Ethernet interfaces. A redundant Ethernet interface is a pseudointerface that contains two or more physical interfaces, with at least one from each node of the cluster.

- [Requirements on page 1164](#)
- [Overview on page 1165](#)
- [Configuration on page 1165](#)
- [Verification on page 1168](#)

#### Requirements

Before you begin:

- [Example: Setting the Chassis Cluster Node ID and Cluster ID on page 1240](#)
- [Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 1245](#)
- [Example: Configuring the Chassis Cluster Fabric on page 1201](#)
- [Example: Configuring Chassis Cluster Redundancy Groups on page 1144](#)

## Overview

After physical interfaces have been assigned to the redundant Ethernet interface, you set the configuration that pertains to them at the level of the redundant Ethernet interface, and each of the child interfaces inherits the configuration.

A redundant Ethernet interface is referred to as a reth in configuration commands.



**NOTE:** You can enable promiscuous mode on redundant Ethernet interfaces. When promiscuous mode is enabled on a Layer 3 Ethernet interface, all packets received on the interface are sent to the central point or Services Processing Unit regardless of the destination MAC address of the packet. If you enable promiscuous mode on a redundant Ethernet interface, promiscuous mode is then enabled on any child physical interfaces.

To enable promiscuous mode on a redundant Ethernet interface, use the `promiscuous-mode` statement at the `[edit interfaces]` hierarchy.

## Configuration

### CLI Quick Configuration

To quickly configure redundant Ethernet interfaces for IPv4, copy the following commands and paste them into the CLI:

```
{primary:node0}[edit]
user@host# set interfaces ge-0/0/0 gigether-options redundant-parent reth1
set interfaces ge-7/0/0 gigether-options redundant-parent reth1
set interfaces fe-1/0/0 fast-ether-options redundant-parent reth2
set interfaces fe-8/0/0 fast-ether-options redundant-parent reth2
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet mtu 1500
set interfaces reth1 unit 0 family inet address 10.1.1.3/24
set security zones security-zone Trust interfaces reth1.0
```

To quickly configure redundant Ethernet interfaces for IPv6, copy the following commands and paste them into the CLI:

```
{primary:node0}[edit]
user@host# set interfaces ge-0/0/0 gigether-options redundant-parent reth1
set interfaces ge-7/0/0 gigether-options redundant-parent reth1
set interfaces fe-1/0/0 fast-ether-options redundant-parent reth2
set interfaces fe-8/0/0 fast-ether-options redundant-parent reth2
set interfaces reth2 redundant-ether-options redundancy-group 1
set interfaces reth2 unit 0 family inet6 mtu 1500
set interfaces reth2 unit 0 family inet6 address 2010:2010:201::2/64
set security zones security-zone Trust interfaces reth2.0
```

### Step-by-Step Procedure

To configure redundant Ethernet interfaces for IPv4:

1. Bind redundant child physical interfaces to reth1.

```
{primary:node0}[edit]
user@host# set interfaces ge-0/0/0 gigether-options redundant-parent reth1
user@host# set interfaces ge-7/0/0 gigether-options redundant-parent reth1
```

2. Bind redundant child physical interfaces to reth2.
 

```
{primary:node0}[edit]
user@host# set interfaces fe-1/0/0 fast-ether-options redundant-parent reth2
user@host# set interfaces fe-8/0/0 fast-ether-options redundant-parent reth2
```
3. Add reth1 to redundancy group 1.
 

```
{primary:node0}[edit]
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
```
4. Set the MTU size.
 

```
{primary:node0}[edit]
user@host# set interfaces reth1 unit 0 family inet mtu 1500
```
5. Assign an IP address to reth1.
 

```
{primary:node0}[edit]
user@host# set interfaces reth1 unit 0 family inet address 10.1.1.3/24
```
6. Associate reth1.0 to the trust security zone.
 

```
{primary:node0}[edit]
user@host# set security zones security-zone Trust interfaces reth1.0
```

#### Step-by-Step Procedure

To configure redundant Ethernet interfaces for IPv6:

1. Bind redundant child physical interfaces to reth1.
 

```
{primary:node0}[edit]
user@host# set interfaces ge-0/0/0 gigeother-options redundant-parent reth1
user@host# set interfaces ge-7/0/0 gigeother-options redundant-parent reth1
```
2. Bind redundant child physical interfaces to reth2.
 

```
{primary:node0}[edit]
user@host# set interfaces fe-1/0/0 fast-ether-options redundant-parent reth2
user@host# set interfaces fe-8/0/0 fast-ether-options redundant-parent reth2
```
3. Add reth2 to redundancy group 1.
 

```
{primary:node0}[edit]
user@host# set interfaces reth2 redundant-ether-options redundancy-group 1
```
4. Set the MTU size.
 

```
{primary:node0}[edit]
user@host# set interfaces reth2 unit 0 family inet6 mtu 1500
```
5. Assign an IP address to reth2.
 

```
{primary:node0}[edit]
user@host# set interfaces reth2 unit 0 family inet6 address 2010:2010:201::2/64
```
6. Associate reth2.0 to the trust security zone.
 

```
{primary:node0}[edit]
user@host# set security zones security-zone Trust interfaces reth2.0
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces reth0** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
{primary:node0}[edit]
user@host# show interfaces
interfaces {
  ...
  fe-1/0/0 {
    fastether-options {
      redundant-parent reth2;
    }
  }
  fe-8/0/0 {
    fastether-options {
      redundant-parent reth2;
    }
  }
  ge-0/0/0 {
    gigether-options {
      redundant-parent reth1;
    }
  }
  ge-7/0/0 {
    gigether-options {
      redundant-parent reth1;
    }
  }
  reth1 {
    redundant-ether-options {
      redundancy-group 1;
    }
    unit 0 {
      family inet {
        mtu 1500;
        address 10.1.1.3/24;
      }
    }
  }
  reth2 {
    redundant-ether-options {
      redundancy-group 1;
    }
    unit 0 {
      family inet6 {
        mtu 1500;
        address 2010:2010:201::2/64;
      }
    }
  }
  ...
}
```

```
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Chassis Cluster Redundant Ethernet Interfaces on page 1168
- Verifying Chassis Cluster Control Links on page 1168

#### *Verifying Chassis Cluster Redundant Ethernet Interfaces*

**Purpose** Verify the configuration of the chassis cluster redundant Ethernet interfaces.

**Action** From operational mode, enter the **show interfaces | match reth1** command:

```
{primary:node0}
user@host> show interfaces | match reth1
ge-0/0/0.0          up    down aenet  --> reth1.0
ge-7/0/0.0          up    down aenet  --> reth0.0
reth1               up    down
reth1.0             up    down inet    10.1.1.3/24

```

#### *Verifying Chassis Cluster Control Links*

**Purpose** Verify information about the control interface in a chassis cluster configuration.

**Action** From operational mode, enter the **show chassis cluster interfaces** command:

```
{primary:node0}
user@host> show chassis cluster interfaces
Control link 0 name: em0

Redundant-ethernet Information:
  Name      Status  Redundancy-group
  reth1     Up      1

Interface Monitoring:
  Interface  Weight  Status  Redundancy-group
  fe-1/0/0   200     Up      1
  fe-8/0/0   200     Up      1
  ge-0/0/0   200     Up      1
  ge-7/0/0   200     Up      1

```



**NOTE:** On SRX1400, SRX3400, SRX3600, SRX5600, and SRX5800 devices, eight-queue configurations are not reflected on the chassis cluster interface.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 1162
- Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 1169

- Understanding Conditional Route Advertising in a Chassis Cluster on page 1176
- Understanding Chassis Cluster Formation on page 1138

## Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups

- Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 1169
- Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 1171
- Example: Configuring Chassis Cluster Minimum Links on page 1174

### Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups

Support for Ethernet link aggregation groups (LAGs) based on IEEE 802.3ad makes it possible to aggregate physical interfaces on a standalone device. LAGs on standalone devices provide increased interface bandwidth and link availability. Aggregation of links in a chassis cluster allows a redundant Ethernet interface to add more than two physical child interfaces thereby creating a redundant Ethernet interface LAG. A redundant Ethernet interface LAG can have up to eight links per redundant Ethernet interface per node (for a total of 16 links per redundant Ethernet interface).

The aggregated links in a redundant Ethernet interface LAG provide the same bandwidth and redundancy benefits of a LAG on a standalone device with the added advantage of chassis cluster redundancy. A redundant Ethernet interface LAG has two types of simultaneous redundancy. The aggregated links within the redundant Ethernet interface on each node are redundant; if one link in the primary aggregate fails, its traffic load is taken up by the remaining links. If enough child links on the primary node fail, the redundant Ethernet interface LAG can be configured so that all traffic on the entire redundant Ethernet interface fails over to the aggregate link on the other node.

Aggregated Ethernet interfaces, known as local LAGs, are also supported on either node of a chassis cluster but cannot be added to redundant Ethernet interfaces. Local LAGs are indicated in the system interfaces list using an `ae-` prefix. Likewise any child interface of an existing local LAG cannot be added to a redundant Ethernet interface and vice versa. Note that it is necessary for the switch (or switches) used to connect the nodes in the cluster to have a LAG link configured and 802.3ad enabled for each LAG on both nodes so that the aggregate links are recognized as such and correctly pass traffic. The total maximum number of combined individual node LAG interfaces (`ae`) and redundant Ethernet (`reth`) interfaces per cluster is 128.



**NOTE:** The redundant Ethernet interface LAG child links from each node in the chassis cluster must be connected to a different LAG at the peer devices. If a single peer switch is used to terminate the redundant Ethernet interface LAG, two separate LAGs must be used in the switch.

Links from different PICs or IOCs and using different cable types (for example, copper and fiber-optic) can be added to the same redundant Ethernet interface LAG but the

speed of the interfaces must be the same and all interfaces must be in full duplex mode. We recommend, however, that for purposes of reducing traffic processing overhead, interfaces from the same PIC or IOC be used whenever feasible. Regardless, all interfaces configured in a redundant Ethernet interface LAG share the same virtual MAC address.



**NOTE:** SRX Series and J Series devices interface-monitoring feature now allows monitoring of redundant Ethernet/aggregated Ethernet interfaces.

Redundant Ethernet interface configuration also includes a minimum-links setting that allows you to set a minimum number of physical child links on the primary node in a given redundant Ethernet interface that must be working for the interface to be up. The default minimum-links value is 1. Note that the minimum-links setting only monitors child links on the primary node. Redundant Ethernet interfaces do not use physical interfaces on the backup node for either ingress or egress traffic.

Note the following support details:

- Quality of service (QoS) is supported in a redundant Ethernet interface LAG. Guaranteed bandwidth is, however, duplicated across all links. If a link is lost, there is a corresponding loss of guaranteed bandwidth.
- Layer 2 transparent mode and Layer 2 security features are supported in redundant Ethernet interface LAGs.
- Link Aggregation Control Protocol (LACP) is supported in chassis cluster deployments, where aggregated Ethernet interfaces and redundant Ethernet interfaces are supported simultaneously.
- Chassis cluster management, control, and fabric interfaces cannot be configured as redundant Ethernet interface LAGs or added to a redundant Ethernet interface LAG.
- Network processor bundling can coexist with redundant Ethernet interface LAGs on the same cluster. However, assigning an interface simultaneously to a redundant Ethernet interface LAG and a network processor bundle is not supported.
- Single flow throughput is limited to the speed of a single physical link regardless of the speed of the aggregate interface.



**NOTE:** For more information about Ethernet interface link aggregation and LACP, see the “Aggregated Ethernet” chapter of the *Junos OS Interfaces Configuration Guide for Security Devices*.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces on page 1162](#)
- [Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 1171](#)
- [Example: Configuring Chassis Cluster Minimum Links on page 1174](#)



- Understanding Conditional Route Advertising in a Chassis Cluster on page 1176
- Understanding Chassis Cluster Formation on page 1138

### Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups

This example shows how to configure a redundant Ethernet interface link aggregation group for a chassis cluster. Chassis cluster configuration supports more than one child interface per node in a redundant Ethernet interface. When at least two physical child interface links from each node are included in a redundant Ethernet interface configuration, the interfaces are combined within the redundant Ethernet interface to form a redundant Ethernet interface link aggregation group.

#### Requirements

Before you begin:

- Understand chassis cluster redundant Ethernet interfaces. See “Understanding Chassis Cluster Redundant Ethernet Interfaces” on page 1162.
- Configure chassis cluster redundant interfaces. See “Example: Configuring Chassis Cluster Redundant Ethernet Interfaces” on page 1164.
- Understand chassis cluster redundant Ethernet interface link aggregation groups. See “Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups” on page 1169.

#### Overview



**NOTE:** For aggregation to take place, the switch used to connect the nodes in the cluster must enable IEEE 802.3ad link aggregation for the redundant Ethernet interface physical child links on each node. Because most switches support IEEE 802.3ad and are also LACP capable, we recommend that you enable LACP on SRX Series devices. In cases where LACP is not available on the switch, you should not enable LACP on SRX Series devices.

In this example, you assign six Ethernet interfaces to reth1 to form the Ethernet interface link aggregation group:

- ge-1/0/1—reth1
- ge-1/0/2—reth1
- ge-1/0/3—reth1
- ge-12/0/1—reth1
- ge-12/0/2—reth1
- ge-12/0/3—reth1



**NOTE:** A maximum of eight physical interfaces per node in a cluster, for a total of 16 child interfaces, can be assigned to a single redundant Ethernet interface when a redundant Ethernet interface LAG is being configured.

### Configuration

**CLI Quick Configuration** To quickly configure a redundant Ethernet interface link aggregation group, copy the following commands and paste them into the CLI:

```
{primary:node0}[edit]
set interfaces ge-1/0/1 gigether-options redundant-parent reth1
set interfaces ge-1/0/2 gigether-options redundant-parent reth1
set interfaces ge-1/0/3 gigether-options redundant-parent reth1
set interfaces ge-12/0/1 gigether-options redundant-parent reth1
set interfaces ge-12/0/2 gigether-options redundant-parent reth1
set interfaces ge-12/0/3 gigether-options redundant-parent reth1
```

**Step-by-Step Procedure** To configure a redundant Ethernet interface link aggregation group:

1. Assign Ethernet interfaces to reth1.

```
{primary:node0}[edit]
user@host# set interfaces ge-1/0/1 gigether-options redundant-parent reth1
user@host# set interfaces ge-1/0/2 gigether-options redundant-parent reth1
user@host# set interfaces ge-1/0/3 gigether-options redundant-parent reth1
user@host# set interfaces ge-12/0/1 gigether-options redundant-parent reth1
user@host# set interfaces ge-12/0/2 gigether-options redundant-parent reth1
user@host# set interfaces ge-12/0/3 gigether-options redundant-parent reth1
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces reth1** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
user@host# show interfaces reth1
...
ge-1/0/1 {
  gigether-options {
    redundant-parent reth1;
  }
}
ge-1/0/2 {
  gigether-options {
    redundant-parent reth1;
  }
}
ge-1/0/3 {
  gigether-options {
    redundant-parent reth1;
  }
}
```

```

}
ge-12/0/1 {
  gigger-options {
    redundant-parent reth1;
  }
}
ge-12/0/2 {
  gigger-options {
    redundant-parent reth1;
  }
}
ge-12/0/3 {
  gigger-options {
    redundant-parent reth1;
  }
}
...

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

#### Verifying the Redundant Ethernet Interface LAG Configuration

**Purpose** Verify the redundant Ethernet interface LAG configuration.

**Action** From operational mode, enter the **show interfaces terse | match reth** command.

```

{primary:node0}
user@host> show interfaces terse | match reth
ge-1/0/1.0          up    down aenet  --> reth1.0
ge-1/0/2.0          up    down aenet  --> reth1.0
ge-1/0/3.0          up    down aenet  --> reth1.0
ge-12/0/1.0         up    down aenet  --> reth1.0
ge-12/0/2.0         up    down aenet  --> reth1.0
ge-12/0/3.0         up    down aenet  --> reth1.0
reth0               up    down
reth0.0             up    down inet    10.100.37.214/24
reth1               up    down
reth1.0             up    down inet

```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Example: Configuring Chassis Cluster Minimum Links on page 1174
- Understanding Conditional Route Advertising in a Chassis Cluster on page 1176
- Understanding Chassis Cluster Formation on page 1138

### Example: Configuring Chassis Cluster Minimum Links

This example shows how to specify a minimum number of physical links assigned to a redundant Ethernet interface on the primary node that must be working for the interface to be up.

- Requirements on page 1174
- Overview on page 1174
- Configuration on page 1174
- Verification on page 1175

#### Requirements

Before you begin:

- Understand redundant Ethernet interfaces. See “Understanding Chassis Cluster Redundant Ethernet Interfaces” on page 1162.
- Configure redundant Ethernet interfaces. See “Example: Configuring Chassis Cluster Redundant Ethernet Interfaces” on page 1164.
- Understand redundant Ethernet interface link aggregation groups. See “Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups” on page 1171.

#### Overview

When a redundant Ethernet interface has more than two child links, you can set a minimum number of physical links assigned to the interface on the primary node that must be working for the interface to be up. When the number of physical links on the primary node falls below the minimum-links value, the interface will be down even if some links are still working.

In this example, you specify that three child links on the primary node and bound to reth1 (minimum-links value) be working to prevent the interface from going down. For example, in a redundant Ethernet interface LAG configuration in which six interfaces are assigned to reth1, setting the minimum-links value to 3 means that all reth1 child links on the primary node must be working to prevent the interface’s status from changing to down.



**NOTE:** Although it is possible to set a minimum-links value for a redundant Ethernet interface with only two child interfaces (one on each node), we do not recommend it.

#### Configuration

##### Step-by-Step Procedure

To specify the minimum number of links:

1. Specify the minimum number of links for the redundant Ethernet interface.
 

```
{primary:node0}[edit]
user@host# set interfaces reth1 redundant-ether-options minimum-links 3
```
2. If you are done configuring the device, commit the configuration.

```
{primary:node0}[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show interface reth1** command.

```
{primary:node0}[edit]
```

```
user@host> show interfaces reth1
```

```
Physical interface: reth1, Enabled, Physical link is Down
  Interface index: 129, SNMP ifIndex: 548
  Link-level type: Ethernet, MTU: 1514, Speed: Unspecified, BPDU Error: None,
  MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Disabled, Minimum links needed: 3, Minimum bandwidth needed: 0
  Device flags   : Present Running
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x0
  Current address: 00:10:db:ff:10:01, Hardware address: 00:10:db:ff:10:01
  Last flapped   : 2010-09-15 15:54:53 UTC (1w0d 22:07 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
```

```
Logical interface reth1.0 (Index 68) (SNMP ifIndex 550)
  Flags: Hardware-Down Device-Down SNMP-Traps 0x0 Encapsulation: ENET2
  Statistics          Packets      pps          Bytes          bps
  Bundle:
    Input :             0           0             0             0
    Output:             0           0             0             0
  Security: Zone: untrust
  Allowed host-inbound traffic : bootp bfd bgp dns dvmrp igmp ldp msdp nhrp
  ospf pgm pim rip router-discovery rsvp sap vrrp dhcp finger ftp tftp
  ident-reset http https ike netconf ping reverse-telnet reverse-ssh rlogin
  rpm rsh snmp snmp-trap ssh telnet traceroute xnm-clear-text xnm-ssl lsping
  ntp sip
  Protocol inet, MTU: 1500
  Flags: Sendbcast-pkt-to-re
```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces on page 1162](#)
- [Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 1169](#)
- [Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 1171](#)
- [Understanding Conditional Route Advertising in a Chassis Cluster on page 1176](#)
- [Understanding Chassis Cluster Formation on page 1138](#)

## Conditional Route Advertising in a Chassis Cluster

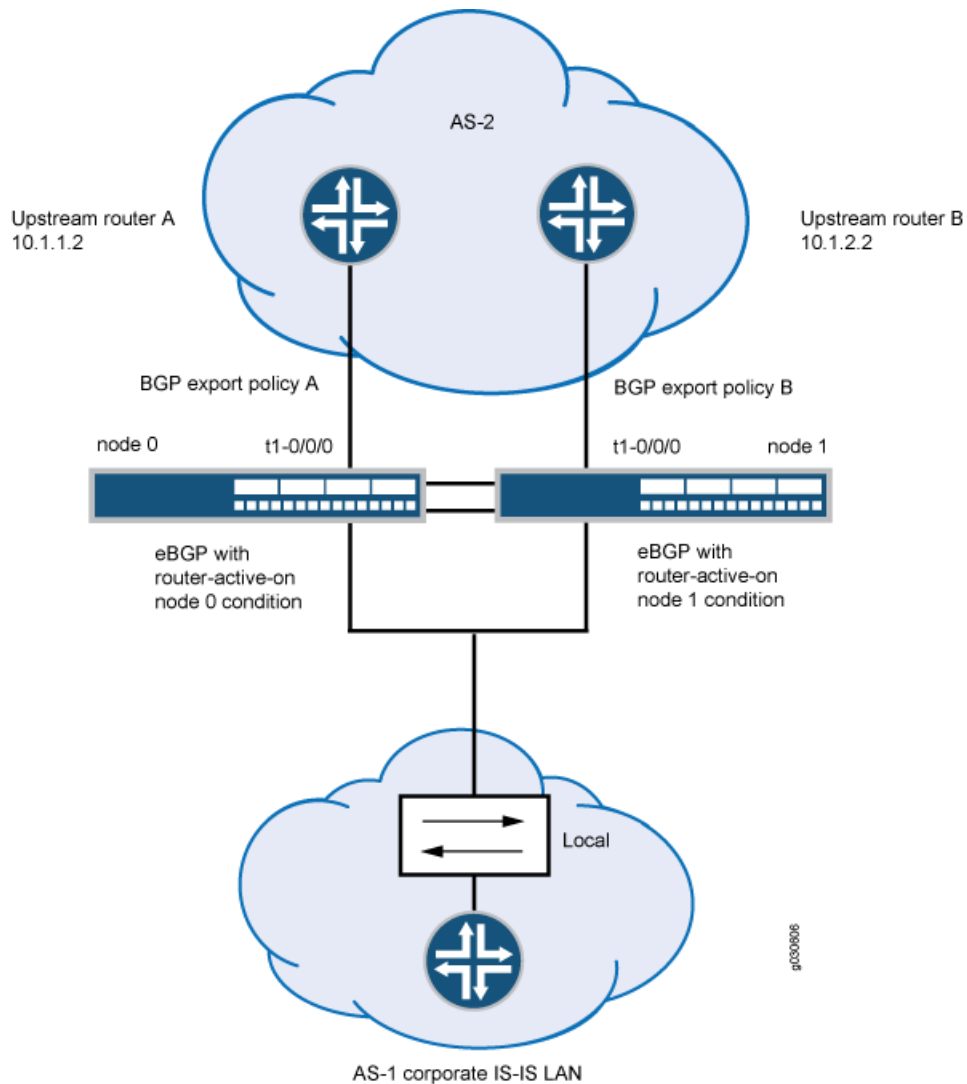
- [Understanding Conditional Route Advertising in a Chassis Cluster on page 1176](#)
- [Example: Configuring Conditional Route Advertising in a Chassis Cluster on page 1178](#)

## Understanding Conditional Route Advertising in a Chassis Cluster

Route advertisement over redundant Ethernet interfaces in a chassis cluster is complicated by the fact that the active node in the cluster can change dynamically. Conditional route advertisement enables you to advertise routes in such a way that incoming traffic from the core network is attracted to the Border Gateway Protocol (BGP) interface that exists on the same node as the currently active redundant Ethernet interface. In this way, traffic is processed by the active node and does not traverse the fabric interface between nodes. You do this by manipulating the BGP attribute at the time routes are advertised by BGP.

The goal of conditional route advertisement in a chassis cluster is to ensure that incoming traffic from the upstream network arrives on the node that is on the currently active redundant Ethernet interface. To understand how this works, keep in mind that in a chassis cluster, each node has its own set of interfaces. Figure 92 on page 1177 shows a typical scenario, with a redundant Ethernet interface connecting the corporate LAN, through a chassis cluster, to an external network segment.

Figure 92: Conditional Route Advertising



**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces on page 1162](#)
- [Example: Configuring Conditional Route Advertising in a Chassis Cluster on page 1178](#)
- [Understanding Chassis Cluster Formation on page 1138](#)

## Example: Configuring Conditional Route Advertising in a Chassis Cluster

This example shows how to configure conditional route advertising in a chassis cluster to ensure that incoming traffic from the upstream network arrives on the node that is on the currently active redundant Ethernet interface..

- Requirements on page 1178
- Overview on page 1178
- Configuration on page 1180

### Requirements

---

Before you begin, understand conditional route advertising in a chassis cluster. See “Understanding Conditional Route Advertising in a Chassis Cluster” on page 1176.

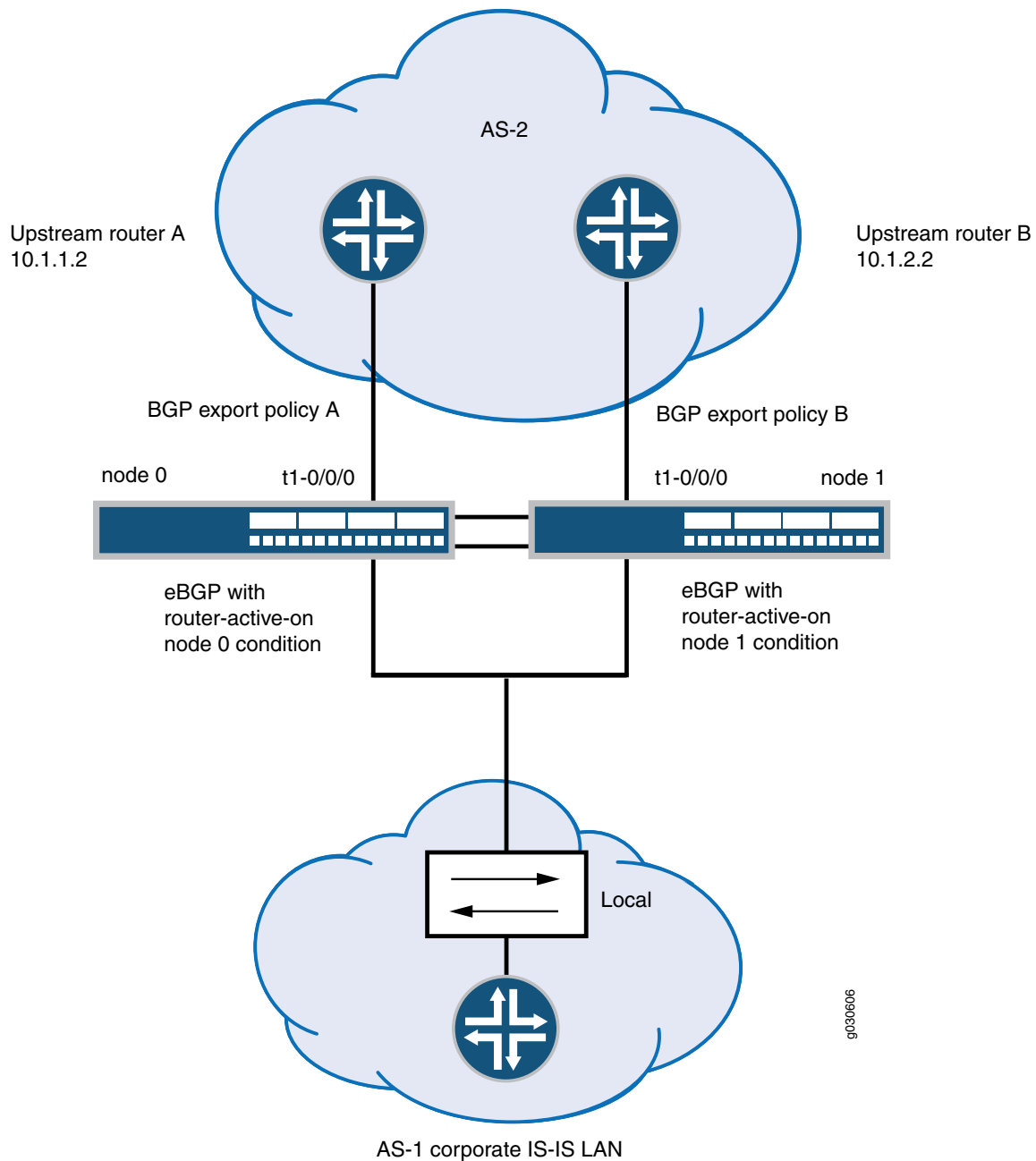
### Overview

---

As illustrated in Figure 93 on page 1179, routing prefixes learned from the redundant Ethernet interface through the IGP are advertised toward the network core using BGP. Two BGP sessions are maintained, one from interface t1-1/0/0 and one from t1-1/0/1 for BGP multihoming. All routing prefixes are advertised on both sessions. Thus, for a route advertised by BGP, learned over a redundant Ethernet interface, if the active redundant Ethernet interface is on the same node as the BGP session, you advertise the route with a “good” BGP attribute.



Figure 93: Conditional Route Advertising



To achieve this behavior, you apply a policy to BGP before exporting routes. An additional term in the policy match condition determines the current active redundant Ethernet interface child interface of the next hop before making the routing decision. When the active status of a child redundant Ethernet interface changes, BGP reevaluates the export policy for all routes affected.

The condition statement in this configuration works as follows. The command states that any routes evaluated against this condition will pass only if:

- The routes have a redundant Ethernet interface as their next-hop interface.
- The current child interface of the redundant Ethernet interface is active at node 0 (as specified by the `route-active-on node0` keyword).

```
{primary:node0}[edit]
user@host# set policy-options condition reth-nh-active-on-0 route-active-on node0
```

Note that a route might have multiple equal-cost next hops, and those next hops might be redundant Ethernet interfaces, regular interfaces, or a combination of both. The route still satisfies the requirement that it has a redundant Ethernet interface as its next hop.

If you use the BGP export policy set for node 0 in the previous example command, only OSPF routes that satisfy the following requirements will be advertised through the session:

- The OSPF routes have a redundant Ethernet interface as their next hop.
- The current child interface of the redundant Ethernet interface is currently active at node 0.

You must also create and apply a separate policy statement for the other BGP session by using this same process.

In addition to the BGP MED attribute, you can define additional BGP attributes, such as origin-code, as-path, and community.

### Configuration

#### CLI Quick Configuration

To quickly configure conditional route advertising, copy the following commands and paste them into the CLI:

```
{primary:node0}[edit]
set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0 from protocol ospf
set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0 from condition reth-nh-active-on-0
set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0 then metric 10
set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0 then accept
set policy-options condition reth-nh-active-on-0 route-active-on node0
```

#### Step-by-Step Procedure

To configure conditional route advertising:

1. Create the policies.

```
{primary:node0}[edit]
user@host# set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0
from protocol ospf
{primary:node0}[edit]
user@host# set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0
from condition reth-nh-active-on-0
{primary:node0}[edit]
user@host# set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0
then metric 10
{primary:node0}[edit]
user@host# set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0
then accept
{primary:node0}[edit]
```

```
user@host# set policy-options condition reth-nh-active-on-0 route-active-on node0
```

**Results** From configuration mode, confirm your configuration by entering the **show policy-options** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
{primary:node0}[edit]
user@host# show policy-options
policy-statement reth-nh-active-on-0 {
  term ospf-on-0 {
    from {
      protocol ospf;
      condition reth-nh-active-on-0;
    }
    then {
      metric 10;
      accept;
    }
  }
}
condition reth-nh-active-on-0 route-active-on node0;
```

If you are done configuring the device, enter **commit** from configuration mode.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces on page 1162](#)
- [Understanding Chassis Cluster Formation on page 1138](#)

## Chassis Cluster Control Plane

- [Understanding the Chassis Cluster Control Plane on page 1182](#)
- [Understanding Chassis Cluster Control Links on page 1183](#)
- [Example: Configuring Chassis Cluster Control Ports on page 1184](#)
- [Example: Configuring Chassis Cluster Control Ports for Dual Control Links on page 1186](#)
- [Understanding Chassis Cluster Dual Control Links on page 1188](#)
- [Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 1189](#)
- [Upgrading the Second Routing Engine When Using Chassis Cluster Dual Control Links on SRX5600 and SRX5800 Devices on page 1191](#)
- [Understanding Chassis Cluster Control Link Heartbeats on page 1192](#)
- [Understanding Chassis Cluster Control Link Failure and Recovery on page 1193](#)
- [Example: Configuring Chassis Cluster Control Link Recovery on page 1195](#)
- [Verifying Chassis Cluster Control Plane Statistics on page 1196](#)
- [Clearing Chassis Cluster Control Plane Statistics on page 1197](#)

## Understanding the Chassis Cluster Control Plane

The control plane software, which operates in active or backup mode, is an integral part of Junos OS that is active on the primary node of a cluster. It achieves redundancy by communicating state, configuration, and other information to the inactive Routing Engine on the secondary node. If the master Routing Engine fails, the secondary one is ready to assume control.

The control plane software:

- Runs on the Routing Engine and oversees the entire chassis cluster system, including interfaces on both nodes
- Manages system and data plane resources, including the Packet Forwarding Engine (PFE) on each node
- Synchronizes the configuration over the control link
- Establishes and maintains sessions, including authentication, authorization, and accounting (AAA) functions
- Manages application-specific signaling protocols
- Establishes and maintains management sessions, such as Telnet connections
- Handles asymmetric routing
- Manages routing state, Address Resolution Protocol (ARP) processing, and Dynamic Host Configuration Protocol (DHCP) processing

Information from the control plane software follows two paths:

- On the primary node (where the Routing Engine is active), control information flows from the Routing Engine to the local Packet Forwarding Engine.
- Control information flows across the control link to the secondary node's Routing Engine and Packet Forwarding Engine.

The control plane software running on the master Routing Engine maintains state for the entire cluster, and only processes running on its node can update state information. The master Routing Engine synchronizes state for the secondary node and also processes all host traffic.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Control Links on page 1183](#)
- [Example: Configuring Chassis Cluster Control Ports on page 1184](#)
- [Understanding Chassis Cluster Dual Control Links on page 1188](#)
- [Understanding Chassis Cluster Control Link Heartbeats on page 1192](#)
- [Understanding Chassis Cluster Control Link Failure and Recovery on page 1193](#)
- [Understanding Chassis Cluster Formation on page 1138](#)

## Understanding Chassis Cluster Control Links

The control link relies on a proprietary protocol to transmit session state, configuration, and liveness signals across the nodes.

On SRX5600 and SRX5800 devices, by default, all control ports are disabled. Each SPC in a device has two control ports, and each device can have multiple SPCs plugged into it. To set up the control link in a chassis cluster with SRX5600 or SRX5800 devices, you connect and configure the control ports that you will use on each device (**fpcn** and **fpcn**) and then initialize the device in cluster mode.

For SRX3400 and SRX3600 devices, there are dedicated chassis cluster (HA) control ports on the switch fabric board. No control link configuration is needed for SRX3400 and SRX3600 devices.

For SRX1400 devices, dedicated control ports on the SYSIO (port 10 and port 11) are available. When the devices are not in cluster mode, these ports can be used as revenue ports. No control link configuration is needed for SRX1400 devices.

For SRX650 and SRX240 devices, the control link uses the **ge-0/0/1** interface.

For SRX100, SRX210, and SRX220 devices, the control link uses the **fe-0/0/7** interface.

In a J Series chassis cluster, the control link is a physical connection between the **ge-0/0/3** ports on each device, with both transformed into **fxp1**.

For details about port and interface usage for management, control, and fabric links, see Table 114 on page 1213 and Table 115 on page 1219.

To set up the control link on J Series devices, you connect the control interfaces on the two devices back-to-back. When you initialize a device in cluster mode, Junos OS renames the control interface to **fxp1** and uses that interface for the cluster control link. To enable the control link to transmit data, the system provides each **fxp1** control link interface with an internal IP address.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Configuring Chassis Cluster Control Ports on page 1184](#)
- [Understanding Chassis Cluster Dual Control Links on page 1188](#)
- [Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 1189](#)
- [Understanding Chassis Cluster Control Link Heartbeats on page 1192](#)
- [Understanding Chassis Cluster Control Link Failure and Recovery on page 1193](#)
- [Understanding the Chassis Cluster Control Plane on page 1182](#)

## Example: Configuring Chassis Cluster Control Ports

This example shows how to configure chassis cluster control ports on SRX5600 and SRX5800 devices. You need to configure the control ports that you will use on each device to set up the control link.

- Requirements on page 1184
- Overview on page 1184
- Configuration on page 1184
- Verification on page 1185

### Requirements

---

Before you begin:

- Understand chassis cluster control links. See “Understanding Chassis Cluster Control Links” on page 1183.
- Physically connect the control ports on the devices. See “Connecting SRX Series Hardware to Create a Chassis Cluster” on page 1223.

### Overview

---

By default, all control ports on SRX5600 and SRX5800 devices are disabled. After connecting the control ports, establishing the chassis cluster, and configuring the control ports, the control link is set up.

This example configures control ports with the following FPCs and ports as the control link:

- FPC 4, port 0
- FPC 10, port 0

### Configuration

---

#### CLI Quick Configuration

To quickly configure control ports for use as the control link for the chassis cluster, copy the following commands and paste them into the CLI:

```
{primary:node0}[edit]
set chassis cluster control-ports fpc 4 port 0
set chassis cluster control-ports fpc 10 port 0
```

#### Step-by-Step Procedure

To configure control ports for use as the control link for the chassis cluster:

1. Specify the control ports.

```
{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 4 port 0
{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 10 port 0
```

**Results** From configuration mode, confirm your configuration by entering the **show chassis cluster** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
{primary:node0}[edit]
user@host# show chassis cluster
...
control-ports {
  fpc 4 port 0;
  fpc 10 port 0;
}
...
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying the Chassis Cluster Status on page 1185

#### *Verifying the Chassis Cluster Status*

**Purpose** Verify the chassis cluster status.

**Action** From operational mode, enter the **show chassis cluster status** command.

```
{primary:node0}
user@host> show chassis cluster status
Cluster ID: 1
Node                Priority    Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
node0                100       primary  no       no
node1                 1         secondary no       no

Redundancy group: 1 , Failover count: 1
node0                 0         primary  no       no
node1                 0         secondary no       no
```

**Meaning** Use the **show chassis cluster status** command to confirm that the devices in the chassis cluster are communicating with each other. The chassis cluster is functioning properly, as one device is the primary node and the other is the secondary node.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Chassis Cluster Dual Control Links on page 1188
- Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 1189
- Understanding Chassis Cluster Control Link Heartbeats on page 1192
- Understanding Chassis Cluster Control Link Failure and Recovery on page 1193

- Understanding the Chassis Cluster Control Plane on page 1182

## Example: Configuring Chassis Cluster Control Ports for Dual Control Links

This example shows how to configure chassis cluster control ports for use as dual control links on SRX5600 and SRX5800 devices. You need to configure the control ports that you will use on each device to set up the control links.

- Requirements on page 1186
- Overview on page 1186
- Configuration on page 1186
- Verification on page 1187

### Requirements

---

Before you begin:

- Understand chassis cluster control links. See “Understanding Chassis Cluster Control Links” on page 1183.
- Physically connect the control ports on the devices. See “Connecting SRX Series Hardware to Create a Chassis Cluster” on page 1223.

### Overview

---

By default, all control ports on SRX5600 and SRX5800 devices are disabled. After connecting the control ports, establishing the chassis cluster, and configuring the control ports, the control links are set up.

This example configures control ports with the following FPCs and ports as the dual control links:

- FPC 4, port 0
- FPC 10, port 0
- FPC 6, port 1
- FPC 12, port 1

### Configuration

---

#### CLI Quick Configuration

To quickly configure control ports for use as dual control links for the chassis cluster, copy the following commands and paste them into the CLI:

```
{primary:node0}[edit]  
set chassis cluster control-ports fpc 4 port 0  
set chassis cluster control-ports fpc 10 port 0  
set chassis cluster control-ports fpc 6 port 1  
set chassis cluster control-ports fpc 12 port 1
```

#### Step-by-Step Procedure

To configure control ports for use as dual control links for the chassis cluster:

1. Specify the control ports.



```
{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 4 port 0
{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 10 port 0
{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 6 port 1
{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 12 port 1
```

**Results** From configuration mode, confirm your configuration by entering the **show chassis cluster** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
{primary:node0}[edit]
user@host# show chassis cluster
...
control-ports {
  fpc 4 port 0;
  fpc 6 port 1;
  fpc 10 port 0;
  fpc 12 port 1;
}
...
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying the Chassis Cluster Status on page 1187

#### *Verifying the Chassis Cluster Status*

**Purpose** Verify the chassis cluster status.

**Action** From operational mode, enter the **show chassis cluster status** command.

```
{primary:node0}
user@host> show chassis cluster status
Cluster ID: 1
Node                Priority    Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
  node0              100       primary  no       no
  node1               1         secondary no       no

Redundancy group: 1 , Failover count: 1
  node0               0         primary  no       no
  node1               0         secondary no       no
```

**Meaning** Use the **show chassis cluster status** command to confirm that the devices in the chassis cluster are communicating with each other. The chassis cluster is functioning properly, as one device is the primary node and the other is the secondary node.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Dual Control Links on page 1188](#)
- [Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 1189](#)
- [Understanding Chassis Cluster Control Link Heartbeats on page 1192](#)
- [Understanding Chassis Cluster Control Link Failure and Recovery on page 1193](#)
- [Understanding the Chassis Cluster Control Plane on page 1182](#)

## Understanding Chassis Cluster Dual Control Links

Dual control links, where two pairs of control link interfaces are connected between each device in a cluster, are supported for the SRX5000 and SRX3000 lines. Having two control links helps to avoid a possible single point of failure.

For the SRX5000 line, this functionality requires a second Routing Engine, as well as a second Switch Control Board (SCB) to house the Routing Engine, to be installed on each device in the cluster. The purpose of the second Routing Engine is only to initialize the switch on the SCB.

For the SRX3000 line, this functionality requires an SRX Clustering Module (SCM) to be installed on each device in the cluster. Although the SCM fits in the Routing Engine slot, it is not a Routing Engine. SRX3000 line devices do not support a second Routing Engine. The purpose of the SCM is to initialize the second control link.

For SRX1400 devices, ports 10 and 11 on the SYSIO can be configured as dual control links. When the devices are not in cluster mode, these ports can be used as revenue ports.



**NOTE:** For the SRX5000 line, the second Routing Engine must be running Junos OS Release 10.0 or later. For the SRX3000 line, the cluster must be running Junos OS Release 10.2 or later (the SCM is not supported in earlier releases and might be incorrectly recognized). For SRX1400 Services Gateways, the cluster must be running Junos OS Release 11.1 or later.

The second Routing Engine, to be installed on SRX5000 line devices only, does not provide backup functionality. It does not need to be upgraded, even when there is a software upgrade of the master Routing Engine on the same node. Note the following conditions:

- You cannot run the CLI or enter configuration mode on the second Routing Engine.
- You do not need to set the chassis ID and cluster ID on the second Routing Engine.

- You need only a console connection to the second Routing Engine. (A console connection is not needed unless you want to check that the second Routing Engine booted up or to upgrade a software image.)
- You cannot log in to the second Routing Engine from the master Routing Engine.



**NOTE:** As long as the first Routing Engine is installed (even if it is rebooting or failing), the second Routing Engine cannot take over the chassis mastership; that is, it cannot control all the hardware on the chassis. The second Routing Engine can only become the master when the master Routing Engine is not present.

#### Related Documentation

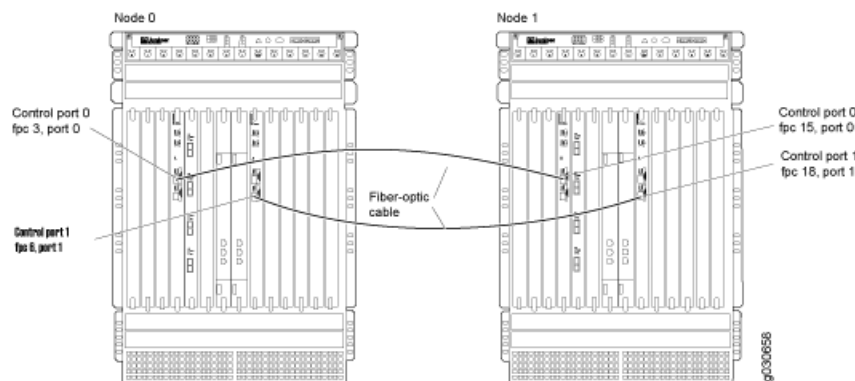
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Control Links on page 1183](#)
- [Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 1189](#)
- [Upgrading the Second Routing Engine When Using Chassis Cluster Dual Control Links on SRX5600 and SRX5800 Devices on page 1191](#)
- [Understanding Chassis Cluster Control Link Heartbeats on page 1192](#)
- [Understanding Chassis Cluster Control Link Failure and Recovery on page 1193](#)
- [Understanding the Chassis Cluster Control Plane on page 1182](#)

## Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster

For the SRX5000 and SRX3000 lines, you can connect two control links between the two devices, effectively reducing the chance of control link failure.

For devices in the SRX5000 line, connect two pairs of the same type of Ethernet ports. For each device, you can use ports on the same Services Processing Card (SPC), but we recommend that they be on two different SPCs to provide high availability. Figure 94 on page 1189 shows a pair of SRX5800 devices with dual control links connected. In this example, control port 0 and control port 1 are connected on different SPCs.

**Figure 94: Connecting Dual Control Links (SRX5800 Devices)**



For SRX1400 Services Gateways and devices in the SRX3000 line, connect two pairs of the same type of Ethernet ports. For each device, use both available built-in ports. Figure 95 on page 1190 shows a pair of SRX3400 devices and Figure 96 on page 1190 shows a pair of SRX1400 devices with dual control links connected.

Figure 95: Connecting Dual Control Links (SRX3400 Devices)

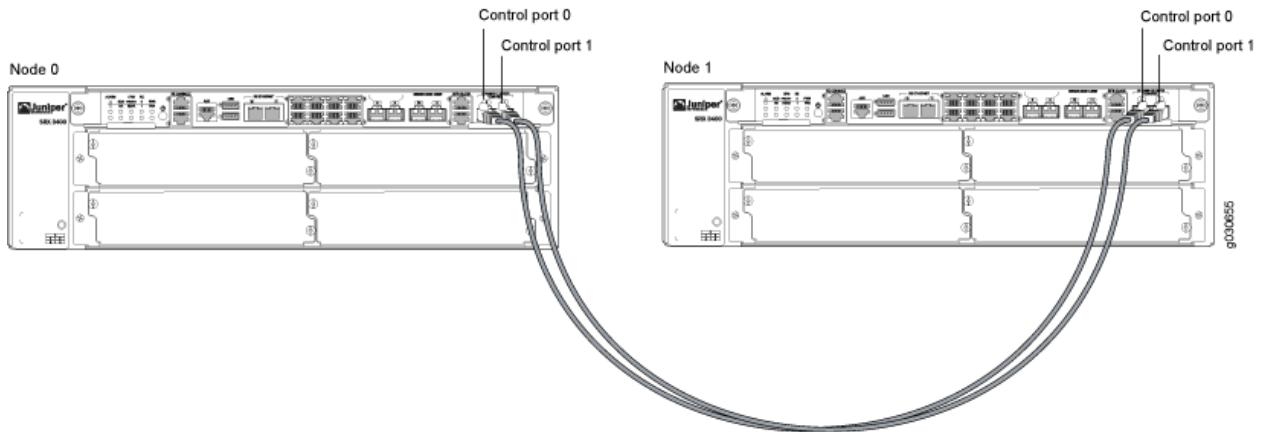
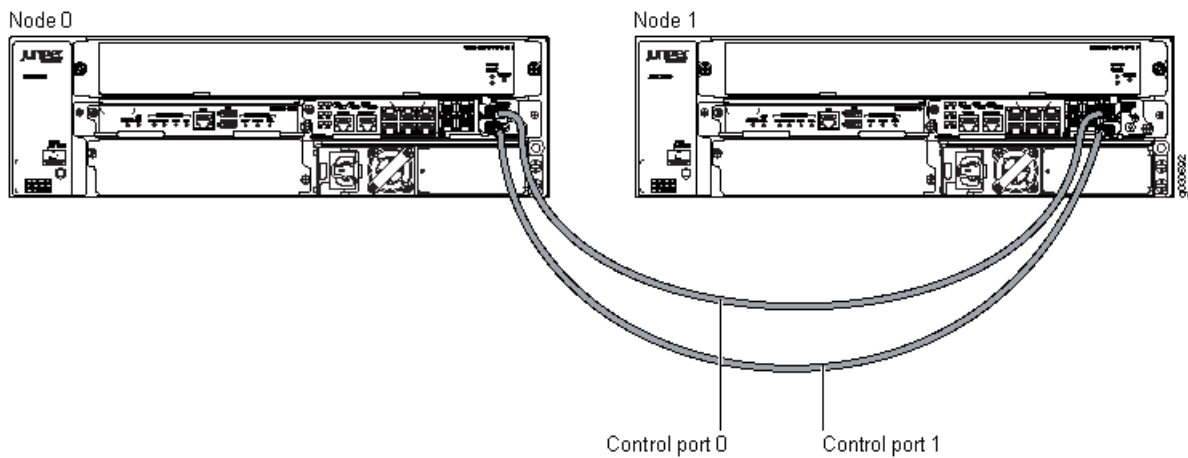


Figure 96: Connecting Dual Control Links (SRX1400 Devices)



**NOTE:** For devices in both the SRX5000 and SRX3000 lines, you must connect control port 0 on one node to control port 0 on the other node and, likewise, control port 1 to control port 1. If you connect control port 0 to control port 1, the nodes cannot receive heartbeat packets across the control links.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Control Links on page 1183](#)
- [Understanding Chassis Cluster Dual Control Links on page 1188](#)
- [Upgrading the Second Routing Engine When Using Chassis Cluster Dual Control Links on SRX5600 and SRX5800 Devices on page 1191](#)

- Understanding Chassis Cluster Control Link Heartbeats on page 1192
- Understanding Chassis Cluster Control Link Failure and Recovery on page 1193
- Understanding the Chassis Cluster Control Plane on page 1182

## Upgrading the Second Routing Engine When Using Chassis Cluster Dual Control Links on SRX5600 and SRX5800 Devices

For SRX5600 and SRX5800 devices, a second Routing Engine is required for each device in a cluster if you are using dual control links. The second Routing Engine does not provide backup functionality; its purpose is only to initialize the switch on the Switch Control Board (SCB). The second Routing Engine must be running Junos OS Release 10.0 or later.

Because you cannot run the CLI or enter configuration mode on the second Routing Engine, you cannot upgrade the Junos OS image with the usual upgrade commands. Instead, use the master Routing Engine (RE0) to create a bootable USB storage device, which you can then use to install a software image on the second Routing Engine (RE1).

To upgrade the software image on the second Routing Engine (RE1):

1. Use FTP to copy the installation media into the `/var/tmp` directory of the master Routing Engine (RE0).
2. Insert a USB storage device into the USB port on the master Routing Engine (RE0).
3. In the UNIX shell, navigate to the `/var/tmp` directory:

```
start shell
cd /var/tmp
```

4. Log in as root or superuser:

```
su [enter]
password: [enter SU password]
```

5. Issue the following command:

```
dd if=installMedia of=/dev/externalDrive bs=64
```

where

- **externalDrive**—Refers to the removable media name. For example, the removable media name on an SRX5000 line device is `da0` for both Routing Engines.
- **installMedia**—Refers to the installation media downloaded into the `/var/tmp` directory. For example, `install-media-srx5000-10.1R1-domestic.tgz`.

The following code example can be used to write the image that you copied to the master Routing Engine (RE0) in step 1 onto the USB storage device:

```
dd if=install-media-srx5000-10.1R1-domestic.tgz of=/dev/da0 bs=64k
```

6. Log out as root or superuser:

```
exit
```

7. After the software image is written to the USB storage device, remove the device and insert it into the USB port on the second Routing Engine (RE1).
8. Move the console connection from the master Routing Engine (RE0) to the second Routing Engine (RE1), if you do not already have a connection.
9. Reboot the second Routing Engine (RE1). Issue the following command:

```
# reboot
```

- When the following system output appears, press **y**:

```
WARNING: The installation will erase the contents of your disks.  
Do you wish to continue (y/n)?
```

- When the following system output appears, remove the USB storage device and press **Enter**:

```
Eject the installation media and hit [Enter] to reboot?
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Control Links on page 1183](#)
- [Understanding Chassis Cluster Dual Control Links on page 1188](#)
- [Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 1189](#)
- [Understanding Chassis Cluster Control Link Heartbeats on page 1192](#)
- [Understanding Chassis Cluster Control Link Failure and Recovery on page 1193](#)
- [Understanding the Chassis Cluster Control Plane on page 1182](#)

## Understanding Chassis Cluster Control Link Heartbeats

Junos OS transmits heartbeat signals over the control link at a configured interval. The system uses heartbeat transmissions to determine the “health” of the control link. If the number of missed heartbeats has reached the configured threshold, the system assesses whether a failure condition exists.

You specify the heartbeat threshold and heartbeat interval when you configure the chassis cluster.

The system monitors the control link's status by default.

For dual control links, which are supported on the SRX1400 Services Gateways and SRX5000 and SRX3000 lines, the Juniper Services Redundancy Protocol process (jsrpd) sends and receives the control heartbeat messages on both control links. As long as heartbeats are received on one of the control links, Junos OS considers the other node to be alive.

The product of the heartbeat-threshold option and the heartbeat-interval option defines the wait time before failover is triggered. The default values of these options produce a wait time of 3 seconds. A heartbeat-threshold of 5 and a heartbeat-interval of 1000

milliseconds would yield a wait time of 5 seconds. Setting the heartbeat-threshold to 4 and the heartbeat-interval to 1250 milliseconds would also yield a wait time of 5 seconds.

In a chassis cluster environment, as the number of logical interfaces is scaled upward, the time before a failover is triggered needs to be increased accordingly. At maximum capacity on an SRX5600 or an SRX5800 device, we recommend that you increase the configured time before failover to at least 5 seconds. In a large chassis cluster configuration on an SRX3400 or SRX3600 device, we recommend increasing the wait to 8 seconds.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Control Links on page 1183](#)
- [Understanding Chassis Cluster Dual Control Links on page 1188](#)
- [Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 1189](#)
- [Understanding Chassis Cluster Control Link Failure and Recovery on page 1193](#)
- [Understanding the Chassis Cluster Control Plane on page 1182](#)

### Understanding Chassis Cluster Control Link Failure and Recovery

If the control link fails, Junos OS disables the secondary node to prevent the possibility of each node becoming primary for all redundancy groups, including redundancy group 0.

A control link failure is described as not receiving heartbeats over the control link; however, heartbeats are still received over the fabric link.

In the event of a legitimate control link failure, redundancy group 0 remains primary on the node on which it is currently primary, inactive redundancy groups x on the primary node become active, and the secondary node enters a disabled state.



**NOTE:** When the secondary node is disabled, you can still log in to the management port and run diagnostics.

To determine if a legitimate control link failure has occurred, the system relies on redundant liveliness signals sent across the control link and the data link.

The system periodically transmits probes over the fabric data link and heartbeat signals over the control link. Probes and heartbeat signals share a common sequence number that maps them to a unique time event. The software identifies a legitimate control link failure if the following two conditions exist:

- The threshold number of heartbeats were lost.
- At least one probe with a sequence number corresponding to that of a missing heartbeat signal was received on the data link.

When a legitimate control link failure occurs, the following conditions apply:

- Redundancy group 0 remains primary on the node on which it is presently primary (and thus its Routing Engine remains active), and all redundancy groups x on the node become primary.

If the system cannot determine which Routing Engine is primary, the node with the higher priority value for redundancy group 0 is primary and its Routing Engine is active. (You configure the priority for each node when you configure the **redundancy-group** statement for redundancy group 0.)

- The system disables the secondary node.

To recover a device from the disabled mode, you must reboot the device. When you reboot the disabled node, the node synchronizes its dynamic state with the primary node.



**NOTE:** If you make any changes to the configuration while the secondary node is disabled, execute the **commit** command to synchronize the configuration after you reboot the node. If you did not make configuration changes, the configuration file remains synchronized with that of the primary node.

You cannot enable preemption for redundancy group 0. If you want to change the primary node for redundancy group 0, you must do a manual failover.

When you use dual control links (supported on the SRX1400 Services Gateways and SRX5000 and SRX3000 lines), note the following conditions:

- Host inbound or outbound traffic can be impacted for up to 3 seconds during a control link failure. For example, consider a case where redundancy group 0 is primary on node 0 and there is a Telnet session to the Routing Engine through a network interface port on node 1. If the currently active control link fails, the Telnet session will lose packets for 3 seconds, until this failure is detected.
- A control link failure that occurs while the commit process is running across two nodes might lead to commit failure. In this situation, run the **commit** command again after 3 seconds.



**NOTE:** For SRX5000 and SRX3000 lines, dual control links require a second Routing Engine on each node of the chassis cluster.

You can specify that control link recovery be done automatically by the system by setting the **control-link-recovery** statement. In this case, once the system determines that the control link is healthy, it issues an automatic reboot on the disabled node. When the disabled node reboots, the node joins the cluster again.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Chassis Cluster Control Links on page 1183
- Understanding Chassis Cluster Dual Control Links on page 1188



- Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 1189
- Understanding Chassis Cluster Control Link Heartbeats on page 1192
- Example: Configuring Chassis Cluster Control Link Recovery on page 1195
- Verifying Chassis Cluster Control Plane Statistics on page 1196

## Example: Configuring Chassis Cluster Control Link Recovery

This example shows how to enable control link recovery, which allows the system to automatically take over after the control link recovers from a failure.

- Requirements on page 1195
- Overview on page 1195
- Configuration on page 1195
- Verification on page 1196

### Requirements

Before you begin:

- Understand chassis cluster control links. See “Understanding Chassis Cluster Control Links” on page 1183.
- Understand chassis cluster dual control links. See “Understanding Chassis Cluster Dual Control Links” on page 1188.
- Connect dual control links in a chassis cluster. See “Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster” on page 1189.

### Overview

You can enable the system to perform control link recovery automatically. After the control link recovers, the system takes the following actions:

- It checks whether it receives at least 30 consecutive heartbeats on the control link or, in the case of dual control links (SRX1400 Services Gateways and SRX5000 and SRX3000 lines only), on either control link. This is to ensure that the control link is not flapping and is healthy.
- After it determines that the control link is healthy, the system issues an automatic reboot on the node that was disabled when the control link failed. When the disabled node reboots, it can rejoin the cluster. There is no need for any manual intervention.

In this example, you enable chassis cluster control link recovery.

### Configuration

#### Step-by-Step Procedure

To enable chassis cluster control-link-recovery:

1. Enable control link recovery.

```
{primary:node0}[edit]
user@host# set chassis cluster control-link-recovery
```

- If you are done configuring the device, commit the configuration.

```
{primary:node0}[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show configuration chassis cluster** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Chassis Cluster Control Links on page 1183
- Understanding Chassis Cluster Dual Control Links on page 1188
- Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 1189
- Understanding Chassis Cluster Control Link Heartbeats on page 1192
- Understanding Chassis Cluster Control Link Failure and Recovery on page 1193
- Verifying Chassis Cluster Control Plane Statistics on page 1196

## Verifying Chassis Cluster Control Plane Statistics

**Purpose** Display chassis cluster control-plane statistics.

**Action** From the CLI, enter the **show chassis cluster control-plane statistics** command:

```
{primary:node1}
user@host> show chassis cluster control-plane statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 124
    Heartbeat packets received: 125
Fabric link statistics:
  Probes sent: 124
  Probes received: 125

{primary:node1}
user@host> show chassis cluster control-plane statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258698
    Heartbeat packets received: 258693
  Control link 1:
    Heartbeat packets sent: 258698
    Heartbeat packets received: 258693
Fabric link statistics:
  Probes sent: 258690
  Probes received: 258690
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 1189

- Example: Configuring Chassis Cluster Control Link Recovery on page 1195
- Clearing Chassis Cluster Control Plane Statistics on page 1197

## Clearing Chassis Cluster Control Plane Statistics

To clear displayed chassis cluster control plane statistics, enter the **clear chassis cluster control-plane statistics** command from the CLI:

```
{primary:node1}
user@host> clear chassis cluster control-plane statistics
```

```
Cleared control-plane statistics
```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Example: Configuring Chassis Cluster Control Link Recovery on page 1195
- Verifying Chassis Cluster Control Plane Statistics on page 1196

## Chassis Cluster Data Plane

- Understanding the Chassis Cluster Data Plane on page 1197
- Understanding Chassis Cluster Fabric Links on page 1199
- Understanding Chassis Cluster Dual Fabric Links on page 1200
- Example: Configuring the Chassis Cluster Fabric on page 1201
- Example: Configuring the Chassis Cluster Dual Fabric Links with Matching Slots and Ports on page 1204
- Example: Configuring Chassis Cluster Dual Fabric Links with Different Slots and Ports on page 1206
- Verifying Chassis Cluster Data Plane Interfaces on page 1208
- Verifying Chassis Cluster Data Plane Statistics on page 1209
- Clearing Chassis Cluster Data Plane Statistics on page 1210

## Understanding the Chassis Cluster Data Plane

The data plane software, which operates in active/active mode, manages flow processing and session state redundancy and processes transit traffic. All packets belonging to a particular session are processed on the same node to ensure that the same security treatment is applied to them. The system identifies the node on which a session is active and forwards its packets to that node for processing. (After a packet is processed, the Packet Forwarding Engine transmits the packet to the node on which its egress interface exists if that node is not the local one.)

To provide for session (or flow) redundancy, the data plane software synchronizes its state by sending special payload packets called runtime objects (RTOs) from one node to the other across the fabric data link. By transmitting information about a session between the nodes, RTOs ensure the consistency and stability of sessions if a failover were to occur, and thus they enable the system to continue to process traffic belonging

to existing sessions. To ensure that session information is always synchronized between the two nodes, the data plane software gives RTOs transmission priority over transit traffic.

- Understanding Session RTOs on page 1198
- Understanding Data Forwarding on page 1198
- Understanding Fabric Data Link Failure and Recovery on page 1199

### Understanding Session RTOs

---

The data plane software creates RTOs for UDP and TCP sessions and tracks state changes. It also synchronizes traffic for IPv4 pass-through protocols such as Generic Routing Encapsulation (GRE) and IPsec.

RTOs for synchronizing a session include:

- Session creation RTOs on the first packet
- Session deletion and ageout RTOs
- Change-related RTOs, including:
  - TCP state changes
  - Timeout synchronization request and response messages
  - RTOs for creating and deleting temporary openings in the firewall (pinholes) and child session pinholes

### Understanding Data Forwarding

---

For Junos OS, flow processing occurs on a single node on which the session for that flow was established and is active. This approach ensures that the same security measures are applied to all packets belonging to a session.

A chassis cluster can receive traffic on an interface on one node and send it out to an interface on the other node. (In active/active mode, the ingress interface for traffic might exist on one node and its egress interface on the other.)

This traversal is required in the following situations:

- When packets are processed on one node, but need to be forwarded out an egress interface on the other node
- When packets arrive on an interface on one node, but must be processed on the other node

If the ingress and egress interfaces for a packet are on one node, but the packet must be processed on the other node because its session was established there, it must traverse the data link twice. This can be the case for some complex media sessions, such as voice-over-IP (VoIP) sessions.

## Understanding Fabric Data Link Failure and Recovery



**NOTE:** Intrusion Detection and Prevention (IDP) services do not support failover. For this reason, IDP services are not applied for sessions that were present prior to the failover. IDP services are applied for new sessions created on the new primary node.

The fabric data link is vital to the chassis cluster. If the link is unavailable, traffic forwarding and RTO synchronization are affected, which can result in loss of traffic and unpredictable system behavior.

To eliminate this possibility, Junos OS detects fabric faults and disables one node of the cluster. It determines that a fabric fault has occurred if a fabric probe is not received but the fabric interface is active.

To recover from this state, you must reboot the disabled node. When you reboot it, the node synchronizes its state and RTOs with the primary node.



**NOTE:** If you make any changes to the configuration while the secondary node is disabled, execute the `commit` command to synchronize the configuration after you reboot the node. If you did not make configuration changes, the configuration file remains synchronized with that of the primary node.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Chassis Cluster Dual Fabric Links on page 1200](#)
- [Example: Configuring the Chassis Cluster Fabric on page 1201](#)
- [Verifying Chassis Cluster Data Plane Interfaces on page 1208](#)
- [Verifying Chassis Cluster Data Plane Statistics on page 1209](#)
- [Clearing Chassis Cluster Data Plane Statistics on page 1210](#)
- [Understanding Chassis Cluster Formation on page 1138](#)

## Understanding Chassis Cluster Fabric Links

The data link is referred to as the fabric interface. It is used by the cluster's Packet Forwarding Engines to transmit transit traffic and to synchronize the data plane software's dynamic runtime state. When the system creates the fabric interface, the software assigns it an internally derived IP address to be used for packet transmission.

The fabric is a physical connection between two nodes of a cluster and is formed by connecting a pair of Ethernet interfaces back-to-back (one from each node).

Unlike for the control link, whose interfaces are determined by the system, you specify the physical interfaces to be used for the fabric data link in the configuration.

For SRX Series chassis clusters, the fabric link can be any pair of Ethernet interfaces spanning the cluster; for J Series chassis clusters, the fabric link can be any pair of Gigabit Ethernet interface.



**NOTE:** For SRX Series chassis clusters made up of SRX100, SRX210, SRX220, SRX240, or SRX650 devices, SFP interfaces on Mini-PIMs cannot be used as the fabric link.

Table 113 on page 1200 shows the fabric interface types that are supported for SRX Series devices.

**Table 113: Supported Fabric Interface Types for SRX Series Devices**

SRX5000 line	SRX3000 line	SRX1400	SRX650	SRX240	SRX220	SRX210	SRX100
Fast Ethernet	Fast Ethernet		Fast Ethernet	Fast Ethernet		Fast Ethernet	Fast Ethernet
Gigabit Ethernet	Gigabit Ethernet	Gigabit Ethernet	Gigabit Ethernet	Gigabit Ethernet	Gigabit Ethernet	Gigabit Ethernet	
10-Gigabit Ethernet	10-Gigabit Ethernet	10-Gigabit Ethernet					

For details about port and interface usage for management, control, and fabric links, see Table 114 on page 1213 and Table 115 on page 1219.

The fabric data link does not support fragmentation. To accommodate this state, jumbo frame support is enabled by default on the link with an MTU size of 8940 bytes. To ensure that traffic that transits the data link does not exceed this size, we recommend that no other interfaces exceed the fabric data link's MTU size.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding the Chassis Cluster Data Plane on page 1197](#)
- [Understanding Chassis Cluster Dual Fabric Links on page 1200](#)
- [Example: Configuring the Chassis Cluster Fabric on page 1201](#)
- [Verifying Chassis Cluster Data Plane Interfaces on page 1208](#)
- [Verifying Chassis Cluster Data Plane Statistics on page 1209](#)
- [Clearing Chassis Cluster Data Plane Statistics on page 1210](#)
- [Understanding Chassis Cluster Formation on page 1138](#)

## Understanding Chassis Cluster Dual Fabric Links

You can connect two fabric links between each device in a cluster, which provides a redundant fabric link between the members of a cluster. Having two fabric links helps to avoid a possible single point of failure.

When you use dual fabric links, the RTOs and probes are sent on one link and the fabric-forwarded and flow-forwarded packets are sent on the other link. If one fabric link fails, the other fabric link handles the RTOs and probes, as well as the data forwarding. The system selects the physical interface with the lowest slot, PIC, or port number on each node for the RTOs and probes.

For all SRX Series and J Series devices, you can connect two fabric links between the two devices, effectively reducing the chance of control link failure.

In most SRX Series devices in a chassis cluster, you can configure any pair of Gigabit Ethernet interfaces or any pair of 10-Gigabit interfaces to serve as the fabric between nodes. Interfaces on SRX210 devices are Fast Ethernet or Gigabit Ethernet (the paired interfaces must be of a similar type) and all interfaces on SRX100 devices are Fast Ethernet interfaces.

In a J Series chassis cluster, you can configure any pair of Gigabit Ethernet interfaces to serve as the fabric between nodes.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding the Chassis Cluster Data Plane on page 1197](#)
- [Understanding Chassis Cluster Fabric Links on page 1199](#)
- [Example: Configuring the Chassis Cluster Fabric on page 1201](#)
- [Verifying Chassis Cluster Data Plane Interfaces on page 1208](#)
- [Verifying Chassis Cluster Data Plane Statistics on page 1209](#)
- [Clearing Chassis Cluster Data Plane Statistics on page 1210](#)
- [Understanding Chassis Cluster Formation on page 1138](#)

### Example: Configuring the Chassis Cluster Fabric

This example shows how to configure the chassis cluster fabric. The fabric is the back-to-back data connection between the nodes in a cluster. Traffic on one node that needs to be processed on the other node or to exit through an interface on the other node passes over the fabric. Session state information also passes over the fabric.

- [Requirements on page 1201](#)
- [Overview on page 1201](#)
- [Configuration on page 1202](#)
- [Verification on page 1203](#)

#### Requirements

Before you begin, set the chassis cluster ID and chassis cluster node ID. See “Example: Setting the Chassis Cluster Node ID and Cluster ID” on page 1240.

#### Overview

In most SRX Series devices in a chassis cluster, you can configure any pair of Gigabit Ethernet interfaces or any pair of 10-Gigabit interfaces to serve as the fabric between

nodes. Interfaces on SRX210 devices are Fast Ethernet or Gigabit Ethernet (the paired interfaces must be of a similar type) and all interfaces on SRX100 devices are Fast Ethernet interfaces.

In a J Series chassis cluster, you can configure any pair of Gigabit Ethernet interfaces to serve as the fabric between nodes.

You cannot configure filters, policies, or services on the fabric interface. Fragmentation is not supported on the fabric link. The MTU size is 8980 bytes. We recommend that no interface in the cluster exceed this MTU size. Jumbo frame support on the member links is enabled by default.

This example illustrates how to configure the fabric link.

Only the same type of interfaces can be configured as fabric children, and you must configure an equal number of child links for **fab0** and **fab1**.



**NOTE:** If you are connecting each of the fabric links through a switch, you must enable the jumbo frame feature on the corresponding switch ports. If both of the fabric links are connected through the same switch, the RTO-and-probes pair must be in one virtual LAN (VLAN) and the data pair must be in another VLAN. Here too, the jumbo frame feature must be enabled on the corresponding switch ports.

### Configuration

**CLI Quick Configuration** To quickly configure the chassis cluster fabric, copy the following commands and paste them into the CLI:

```
{primary:node0}[edit]
set interfaces fab0 fabric-options member-interfaces ge-0/0/1
set interfaces fab1 fabric-options member-interfaces ge-7/0/1
```

**Step-by-Step Procedure** To configure the chassis cluster fabric:

1. Specify the fabric interfaces.

```
{primary:node0}[edit]
user@host# set interfaces fab0 fabric-options member-interfaces ge-0/0/1
{primary:node0}[edit]
user@host# set interfaces fab1 fabric-options member-interfaces ge-7/0/1
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
{primary:node0}[edit]
user@host# show interfaces
```



```

...
fab0 {
  fabric-options {
    member-interfaces {
      ge-0/0/1;
    }
  }
}
fab1 {
  fabric-options {
    member-interfaces {
      ge-7/0/1;
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform the following task:

- Verifying the Chassis Cluster Fabric on page 1203

#### *Verifying the Chassis Cluster Fabric*

**Purpose** Verify the chassis cluster fabric.

**Action** From operational mode, enter the **show interfaces terse | match fab** command.

```
{primary:node0}
```

```

user@host> show interfaces terse | match fab
ge-0/0/1.0          up    up    aenet  --> fab0.0
ge-7/0/1.0          up    up    aenet  --> fab1.0
fab0                 up    up
fab0.0              up    up    inet   30.17.0.200/24
fab1                 up    up
fab1.0              up    up    inet   30.18.0.200/24

```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding the Chassis Cluster Data Plane on page 1197
- Understanding Chassis Cluster Fabric Links on page 1199
- Understanding Chassis Cluster Dual Fabric Links on page 1200
- Verifying Chassis Cluster Data Plane Interfaces on page 1208
- Verifying Chassis Cluster Data Plane Statistics on page 1209
- Clearing Chassis Cluster Data Plane Statistics on page 1210
- Understanding Chassis Cluster Formation on page 1138

## Example: Configuring the Chassis Cluster Dual Fabric Links with Matching Slots and Ports

This example shows how to configure the chassis cluster fabric with dual fabric links with matching slots and ports. The fabric is the back-to-back data connection between the nodes in a cluster. Traffic on one node that needs to be processed on the other node or to exit through an interface on the other node passes over the fabric. Session state information also passes over the fabric.

- Requirements on page 1204
- Overview on page 1204
- Configuration on page 1205
- Verification on page 1206

### Requirements

Before you begin, set the chassis cluster ID and chassis cluster node ID. See “Example: Setting the Chassis Cluster Node ID and Cluster ID” on page 1240.

### Overview

In most SRX Series devices in a chassis cluster, you can configure any pair of Gigabit Ethernet interfaces or any pair of 10-Gigabit interfaces to serve as the fabric between nodes. Interfaces on SRX210 devices are Fast Ethernet or Gigabit Ethernet (the paired interfaces must be of a similar type) and all interfaces on SRX100 devices are Fast Ethernet interfaces.

In a J Series chassis cluster, you can configure any pair of Gigabit Ethernet interfaces to serve as the fabric between nodes.

You cannot configure filters, policies, or services on the fabric interface. Fragmentation is not supported on the fabric link. The MTU size is 8980 bytes. We recommend that no interface in the cluster exceed this MTU size. Jumbo frame support on the member links is enabled by default.

This example illustrates how to configure the fabric link with dual fabric links with matching slots and ports on each node.

A typical configuration is where the dual fabric links are formed with matching slots/ports on each node. That is, **ge-3/0/0** on node 0 and **ge-10/0/0** on node 1 match, as do **ge-0/0/0** on node 0 and **ge-7/0/0** on node 1 (the FPC slot offset is 7).

Only the same type of interfaces can be configured as fabric children, and you must configure an equal number of child links for **fab0** and **fab1**.



**NOTE:** If you are connecting each of the fabric links through a switch, you must enable the jumbo frame feature on the corresponding switch ports. If both of the fabric links are connected through the same switch, the RTO-and-probes pair must be in one virtual LAN (VLAN) and the data pair must be in another VLAN. Here too, the jumbo frame feature must be enabled on the corresponding switch ports.

## Configuration

**CLI Quick Configuration** To quickly configure the chassis cluster fabric with dual fabric links with matching slots and ports on each node, copy the following commands and paste them into the CLI:

```
{primary:node0}[edit]
set interfaces fab0 fabric-options member-interfaces ge-0/0/0
set interfaces fab0 fabric-options member-interfaces ge-3/0/0
set interfaces fab1 fabric-options member-interfaces ge-7/0/0
set interfaces fab1 fabric-options member-interfaces ge-10/0/0
```

**Step-by-Step Procedure** To configure the chassis cluster fabric with dual fabric links with matching slots and ports on each node:

1. Specify the fabric interfaces.

```
{primary:node0}[edit]
user@host# set interfaces fab0 fabric-options member-interfaces ge-0/0/0
user@host# set interfaces fab0 fabric-options member-interfaces ge-3/0/0
user@host# set interfaces fab1 fabric-options member-interfaces ge-7/0/0
user@host# set interfaces fab1 fabric-options member-interfaces ge-10/0/0
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
{primary:node0}[edit]
user@host# show interfaces
...
fab0 {
  fabric-options {
    member-interfaces {
      ge-0/0/0;
      ge-3/0/0;
    }
  }
}
fab1 {
  fabric-options {
    member-interfaces {
      ge-7/0/0;
      ge-10/0/0;
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform the following task:

- Verifying the Chassis Cluster Fabric on page 1206

### *Verifying the Chassis Cluster Fabric*

**Purpose** Verify the chassis cluster fabric.

**Action** From operational mode, enter the **show interfaces terse | match fab** command.

```
{primary:node0}
```

```
user@host> show interfaces terse | match fab
ge-0/0/0.0          up    up    aenet  --> fab0.0
ge-3/0/0.0          up    up    aenet  --> fab0.0
ge-7/0/0.0          up    up    aenet  --> fab1.0
ge-10/0/0.0         up    up    aenet  --> fab1.0
fab0                 up    up
fab0.0               up    up    inet   30.17.0.200/24
fab1                 up    up
fab1.0               up    up    inet   30.18.0.200/24
```

## Example: Configuring Chassis Cluster Dual Fabric Links with Different Slots and Ports

This example shows how to configure the chassis cluster fabric with dual fabric links with different slots and ports. The fabric is the back-to-back data connection between the nodes in a cluster. Traffic on one node that needs to be processed on the other node or to exit through an interface on the other node passes over the fabric. Session state information also passes over the fabric.

- Requirements on page 1206
- Overview on page 1206
- Configuration on page 1207
- Verification on page 1208

### Requirements

Before you begin, set the chassis cluster ID and chassis cluster node ID. See “Example: Setting the Chassis Cluster Node ID and Cluster ID” on page 1240.

### Overview

In most SRX Series devices in a chassis cluster, you can configure any pair of Gigabit Ethernet interfaces or any pair of 10-Gigabit interfaces to serve as the fabric between nodes. Interfaces on SRX210 devices are Fast Ethernet or Gigabit Ethernet (the paired interfaces must be of a similar type) and all interfaces on SRX100 devices are Fast Ethernet interfaces.

In a J Series chassis cluster, you can configure any pair of Gigabit Ethernet interfaces to serve as the fabric between nodes.

You cannot configure filters, policies, or services on the fabric interface. Fragmentation is not supported on the fabric link. The MTU size is 8980 bytes. We recommend that no

interface in the cluster exceed this MTU size. Jumbo frame support on the member links is enabled by default.

This example illustrates how to configure the fabric link with dual fabric links with different slots and ports on each node.

Make sure you physically connect the RTO-and-probes link to the RTO-and-probes link on the other node. Likewise, make sure you physically connect the data link to the data link on the other node.

That is, physically connect the following two pairs:

- The node 0 RTO-and-probes link **ge-2/1/9** to the node 1 RTO-and-probes link **ge-11/0/0**
- The node 0 data link **ge-2/2/5** to the node 1 data link **ge-11/3/0**

Only the same type of interfaces can be configured as fabric children, and you must configure an equal number of child links for **fab0** and **fab1**.



**NOTE:** If you are connecting each of the fabric links through a switch, you must enable the jumbo frame feature on the corresponding switch ports. If both of the fabric links are connected through the same switch, the RTO-and-probes pair must be in one virtual LAN (VLAN) and the data pair must be in another VLAN. Here too, the jumbo frame feature must be enabled on the corresponding switch ports.

### Configuration

#### CLI Quick Configuration

To quickly configure the chassis cluster fabric with dual fabric links with matching slots and ports on each node, copy the following commands and paste them into the CLI:

```
{primary:node0}[edit]
set interfaces fab0 fabric-options member-interfaces ge-2/1/9
set interfaces fab0 fabric-options member-interfaces ge-2/2/5
set interfaces fab1 fabric-options member-interfaces ge-11/0/0
set interfaces fab1 fabric-options member-interfaces ge-11/3/0
```

#### Step-by-Step Procedure

To configure the chassis cluster fabric with dual fabric links with different slots and ports on each node:

1. Specify the fabric interfaces.

```
{primary:node0}[edit]
user@host# set interfaces fab0 fabric-options member-interfaces ge-2/1/9
user@host# set interfaces fab0 fabric-options member-interfaces ge-2/2/5
user@host# set interfaces fab1 fabric-options member-interfaces ge-11/0/0
user@host# set interfaces fab1 fabric-options member-interfaces ge-11/3/0
```

#### Results

From configuration mode, confirm your configuration by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
{primary:node0}[edit]
user@host# show interfaces
...
fab0 {
  fabric-options {
    member-interfaces {
      ge-2/1/9;
      ge-2/2/5;
    }
  }
}
fab1 {
  fabric-options {
    member-interfaces {
      ge-11/0/0;
      ge-11/3/0;
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform the following task:

- Verifying the Chassis Cluster Fabric on page 1208

#### *Verifying the Chassis Cluster Fabric*

**Purpose** Verify the chassis cluster fabric.

**Action** From operational mode, enter the **show interfaces terse | match fab** command.

```
{primary:node0}
user@host> show interfaces terse | match fab
ge-2/1/9.0          up    up    aenet  --> fab0.0
ge-2/2/5.0          up    up    aenet  --> fab0.0
ge-11/0/0.0         up    up    aenet  --> fab1.0
ge-11/3/0.0         up    up    aenet  --> fab1.0
fab0                 up    up
fab0.0               up    up    inet   30.17.0.200/24
fab1                 up    up
fab1.0               up    up    inet   30.18.0.200/24
```

### Verifying Chassis Cluster Data Plane Interfaces

**Purpose** Display chassis cluster data plane interface status.

**Action** From the CLI, enter the **show chassis cluster data-plane interfaces** command:

```
{primary:node1}
```

```

user@host> show chassis cluster data-plane interfaces
fab0:
  Name           Status
  ge-2/1/9       up
  ge-2/2/5       up
fab1:
  Name           Status
  ge-8/1/9       up
  ge-8/2/5       up

```

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Example: Configuring the Chassis Cluster Fabric on page 1201
  - Verifying Chassis Cluster Data Plane Statistics on page 1209
  - Clearing Chassis Cluster Data Plane Statistics on page 1210

## Verifying Chassis Cluster Data Plane Statistics

**Purpose** Display chassis cluster data plane statistics.

**Action** From the CLI, enter the `show chassis cluster data-plane statistics` command:

```

{primary:node1}
user@host> show chassis cluster data-plane statistics

Services Synchronized:
  Service name           RT0s sent  RT0s received
  Translation context    0          0
  Incoming NAT           0          0
  Resource manager       0          0
  Session create         0          0
  Session close          0          0
  Session change         0          0
  Gate create            0          0
  Session ageout refresh requests 0          0
  Session ageout refresh replies 0          0
  IPSec VPN              0          0
  Firewall user authentication 0          0
  MGCP ALG               0          0
  H323 ALG               0          0
  SIP ALG                0          0
  SCCP ALG               0          0
  PPTP ALG              0          0
  RTSP ALG              0          0

```

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Example: Configuring the Chassis Cluster Fabric on page 1201
  - Verifying Chassis Cluster Data Plane Interfaces on page 1208
  - Clearing Chassis Cluster Data Plane Statistics on page 1210

## Clearing Chassis Cluster Data Plane Statistics

To clear displayed chassis cluster data plane statistics, enter the **clear chassis cluster data-plane statistics** command from the CLI:

```
{primary:node1}
user@host> clear chassis cluster data-plane statistics
```

```
Cleared data-plane statistics
```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Configuring the Chassis Cluster Fabric on page 1201](#)
- [Verifying Chassis Cluster Data Plane Interfaces on page 1208](#)
- [Verifying Chassis Cluster Data Plane Statistics on page 1209](#)

## Consequences of Enabling Chassis Cluster

- [Understanding What Happens When Chassis Cluster Is Enabled on page 1210](#)
- [Node Interfaces on Active SRX Series Chassis Clusters on page 1211](#)
- [Node Interfaces on Active J Series Chassis Clusters on page 1219](#)
- [Management Interface on an Active Chassis Cluster on page 1221](#)
- [Fabric Interface on an Active Chassis Cluster on page 1222](#)
- [Control Interface on an Active Chassis Cluster on page 1222](#)

## Understanding What Happens When Chassis Cluster Is Enabled

After wiring the two devices together as described in “Connecting SRX Series Hardware to Create a Chassis Cluster” on page 1223 or “Connecting J Series Hardware to Create a Chassis Cluster” on page 1238, you use CLI *operational mode* commands to enable chassis clustering by assigning a cluster ID and node ID on each chassis in the cluster. The cluster ID is the same on both nodes.

To do this, you connect to the console port on the primary device, give it a node ID, and identify the cluster it will belong to, and then reboot the system. You then connect the console port to the other device, give it a node ID, and assign it the same cluster ID you gave to the first node, and then reboot the system. In both instances, you can cause the system to boot automatically by including the **reboot** parameter in the CLI command line. (For further explanation of primary and secondary nodes, see “Understanding Chassis Cluster Redundancy Groups” on page 1139.)



**CAUTION:** The factory default configuration for SRX100, SRX210, and SRX220 devices automatically enables Layer 2 Ethernet switching. Because Layer 2 Ethernet switching is not supported in chassis cluster mode, if you use the factory default configuration for these devices, you must delete the Ethernet switching configuration before you enable chassis clustering. See “Disabling



Switching on SRX100, SRX210, and SRX220 Devices Before Enabling Chassis Clustering” on page 1234.



**NOTE:** On SRX240 and SRX650 devices, Layer 2 Ethernet switching is supported in chassis cluster mode from Junos OS Release 11.1 onward.



**CAUTION:** After fabric interfaces have been configured on a chassis cluster, removing the fabric configuration on either node will cause the redundancy group 0 (RG0) secondary node to move to a disabled state. (Resetting a device to the factory default configuration removes the fabric configuration and thereby causes the RG0 secondary node to move to a disabled state.) After the fabric configuration is committed, do not reset either device to the factory default configuration.

Figure 97 on page 1216 shows how the FPC slots are numbered on two nodes in an SRX5000 line chassis cluster. Other figures show slot numbering on both nodes in other SRX Series chassis clusters. Figure 105 on page 1221 shows how the PIM slots are numbered on two nodes in a J Series chassis cluster.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Node Interfaces on Active SRX Series Chassis Clusters on page 1211](#)
- [Node Interfaces on Active J Series Chassis Clusters on page 1219](#)
- [Management Interface on an Active Chassis Cluster on page 1221](#)
- [Fabric Interface on an Active Chassis Cluster on page 1222](#)
- [Control Interface on an Active Chassis Cluster on page 1222](#)
- [Disabling Chassis Cluster on page 1254](#)

## Node Interfaces on Active SRX Series Chassis Clusters

Normally, on SRX Series devices, the built-in interfaces are numbered as follows:

<b>For Most SRX Series Devices</b>	ge-0/0/0	ge-0/0/1	ge-0/0/2	ge-0/0/3	...
<b>For SRX210 Devices</b>	ge-0/0/0	ge-0/0/1	fe-0/0/2	fe-0/0/3	...
<b>For SRX100 Devices</b>	fe-0/0/0	fe-0/0/1	fe-0/0/2	fe-0/0/3	...



**CAUTION:** Layer 2 switching must not be enabled on an SRX Series device when chassis clustering is enabled. If you have enabled Layer 2 switching, make sure you disable it before enabling chassis clustering.

The factory default configuration for SRX100, SRX210, and SRX220 devices automatically enables Layer 2 Ethernet switching. Because Layer 2 Ethernet switching is not supported in chassis cluster mode, if you use the factory default configuration for these devices, you must delete the Ethernet switching configuration before you enable chassis clustering. See “Disabling Switching on SRX100, SRX210, and SRX220 Devices Before Enabling Chassis Clustering” on page 1234.



**NOTE:** On SRX240 and SRX650 devices, Layer 2 Ethernet switching is supported in chassis cluster mode from Junos OS Release 11.1 onward.

For chassis clustering, all SRX Series devices have a built-in management interface named **fxp0**. For most SRX Series devices, the **fxp0** interface is a dedicated port. For SRX100, SRX210, and SRX220 devices, after you enable chassis clustering and reboot the system, the built-in interface named **fe-0/0/6** is repurposed as the management interface and is automatically renamed **fxp0**.

For the SRX5000 line, control interfaces are configured on SPCs. For the SRX3000 line and the SRX1400, SRX650, and SRX240 devices, control interfaces are dedicated Gigabit Ethernet ports. For SRX100, SRX210, and SRX220 devices, after you enable chassis clustering and reboot the system, the built-in interface named **fe-0/0/7** is repurposed as the control interface and is automatically renamed **fxp1**.

After the devices are connected as a cluster, the slot numbering on one device changes and thus the interface numbering will change. The slot number for each slot in both nodes is determined using the following formula:

**cluster slot number = (node ID \* maximum slots per node) + local slot number**

In chassis cluster mode, the interfaces on the secondary node are renumbered internally. For example, the management interface port on the front panel of each SRX210 device is still labeled **fe-0/0/6**, but internally, the node 1 port is referred to as **fe-2/0/6**.

In chassis cluster mode, all FPC related configuration is performed under **edit chassis node node-id fpc** hierarchy. In non-cluster mode, the FPC related configuration is performed under **edit chassis fpc** hierarchy.

Table 114 on page 1213 shows the slot numbering, as well as the physical port and logical interface numbering, for both of the SRX Series devices that become node 0 and node 1 of the chassis cluster after the cluster is formed.

Table 114: SRX Series Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming

Model	Chassis	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
5800	Node 0	12 (FPC slots)	0 – 11	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				fxp0	em0	fab0
	Node 1		12 – 23	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				fxp0	em0	fab1
5600	Node 0	6 (FPC slots)	0 – 5	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				fxp0	em0	fab0
	Node 1		6 – 11	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				fxp0	em0	fab1
3600	Node 0	13 (CFM slots)	0 – 12	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab0
	Node 1		13 – 25	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab1

Table 114: SRX Series Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming (*continued*)

Model	Chassis	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
3400	Node 0	8 (CFM slots)	0 – 7	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab0
	Node 1		8 – 15	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab1
1400	Node 0	4 (FPC slots)	0 – 3	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab0
	Node 1		4 – 7	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab1
650	Node 0	9 (PIM slots)	0 – 8	ge-0/0/0	ge-0/0/1	Any Ethernet port
				fxp0	fxp1	fab0
	Node 1		9 – 17	ge-9/0/0	ge-9/0/1	Any Ethernet port
				fxp0	fxp1	fab1
240	Node 0	5 (PIM slots)	0 – 4	ge-0/0/0	ge-0/0/1	Any Ethernet port
				fxp0	fxp1	fab0
	Node 1		5 – 9	ge-5/0/0	ge-5/0/1	Any Ethernet port
				fxp0	fxp1	fab1

**Table 114: SRX Series Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming (continued)**

Model	Chassis	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
220	Node 0	3 (PIM slots)	0 – 2	ge-0/0/6	ge-0/0/7	Any Ethernet port
				fxp0	fxp1	fab0
	Node 1		3 – 5	ge-3/0/6	ge-3/0/7	Any Ethernet port
				fxp0	fxp1	fab1
210	Node 0	2 (PIM slots)	0 and 1	fe-0/0/6	fe-0/0/7	Any Ethernet port
				fxp0	fxp1	fab0
	Node 1		2 and 3	fe-2/0/6	fe-2/0/7	Any Ethernet port
				fxp0	fxp1	fab1
100	Node 0	1 (PIM slot)	0	fe-0/0/6	fe-0/0/7	Any Ethernet port
				fxp0	fxp1	fab0
	Node 1		1	fe-1/0/6	fe-1/0/7	Any Ethernet port
				fxp0	fxp1	fab1

Information about chassis cluster slot numbering is also provided in Figure 97 on page 1216, Figure 98 on page 1217, Figure 99 on page 1217, Figure 100 on page 1217, Figure 101 on page 1218, Figure 102 on page 1218, Figure 103 on page 1218, and Figure 104 on page 1218.

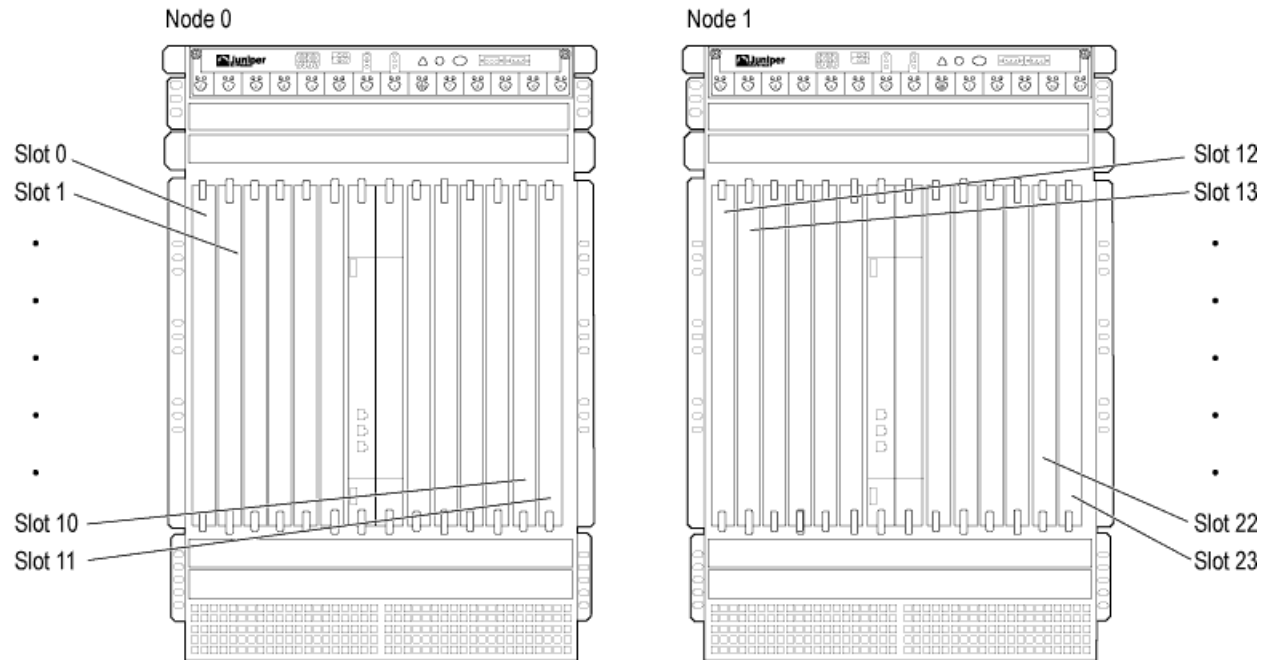


**NOTE:** See the appropriate *Services Gateway Hardware Guide* for details about SRX Series devices. The *Junos OS Interfaces Configuration Guide for Security Devices* provides a full discussion of the interface naming convention.

After you enable chassis clustering, the two chassis joined together cease to exist as individuals and now represent a single system. As a single system, the cluster now has twice as many slots. (See Figure 97 on page 1216, Figure 98 on page 1217, Figure 99 on

page 1217, Figure 100 on page 1217, Figure 101 on page 1218, Figure 102 on page 1218, Figure 103 on page 1218, and Figure 104 on page 1218.)

**Figure 97: FPC Slot Numbering in an SRX Series Chassis Cluster (SRX5800 Devices)**



**NOTE:** SRX5600 and SRX5800 devices have Flex I/O Cards (Flex IOCs) that have two slots to accept the following port modules:

- SRX-IOC-4XGE-XFP 4-Port XFP
- SRX-IOC-16GE-TX 16-Port RJ-45
- SRX-IOC-16GE-SFP 16-Port SFP

You can use these port modules to add from 4 to 16 Ethernet ports to your SRX Series device. Port numbering for these modules is

*slot/port module/port*

where *slot* is the number of the slot in the device in which the Flex IOC is installed; *port module* is 0 for the upper slot in the Flex IOC or 1 for the lower slot when the card is vertical, as in an SRX5800 device; and *port* is the number of the port on the port module. When the card is horizontal, as in an SRX5600 device, *port module* is 0 for the left-hand slot or 1 for the right-hand slot.

See the *Services Gateway Hardware Guide* for your specific SRX Series model.

Figure 98: Slot Numbering in an SRX Series Chassis Cluster (SRX3400 Devices)

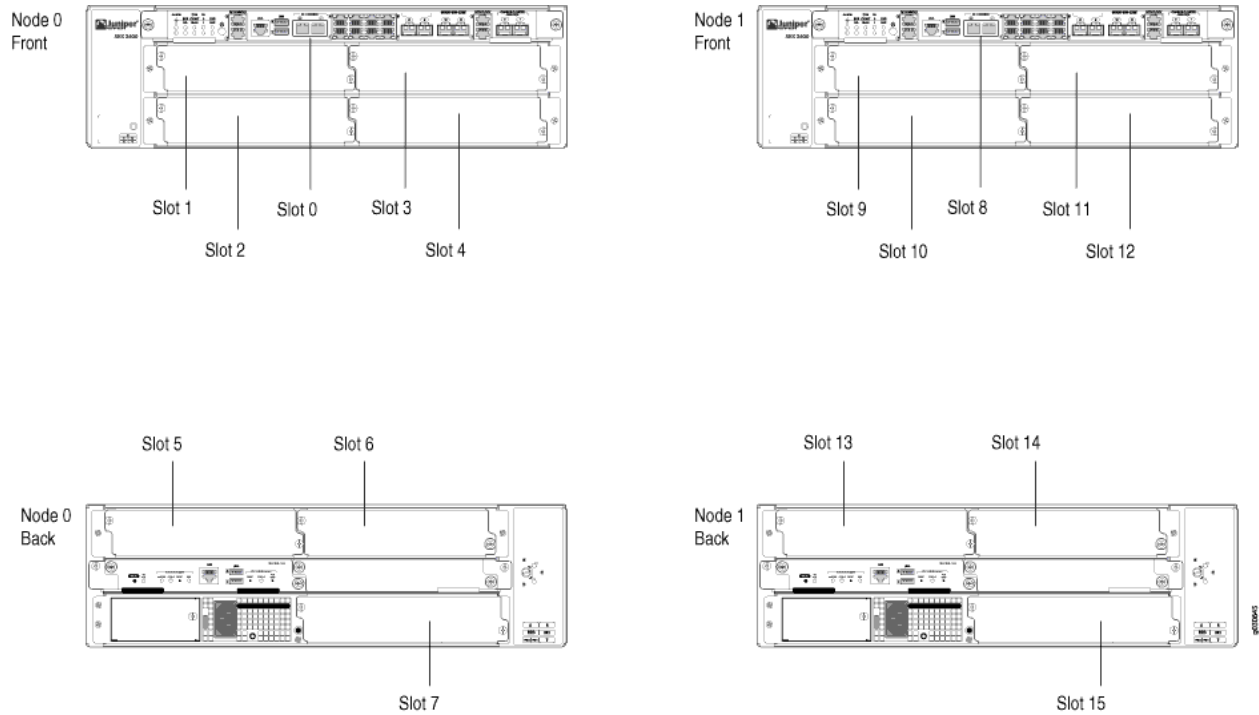


Figure 99: Slot Numbering in an SRX Series Chassis Cluster (SRX1400 Devices)

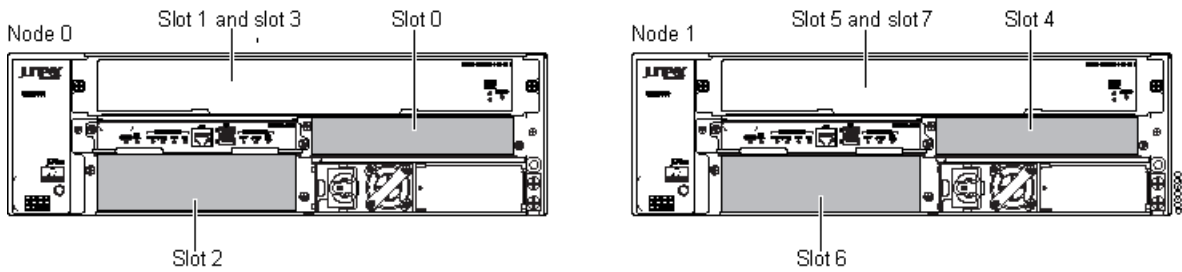


Figure 100: Slot Numbering in an SRX Series Chassis Cluster (SRX650 Devices)

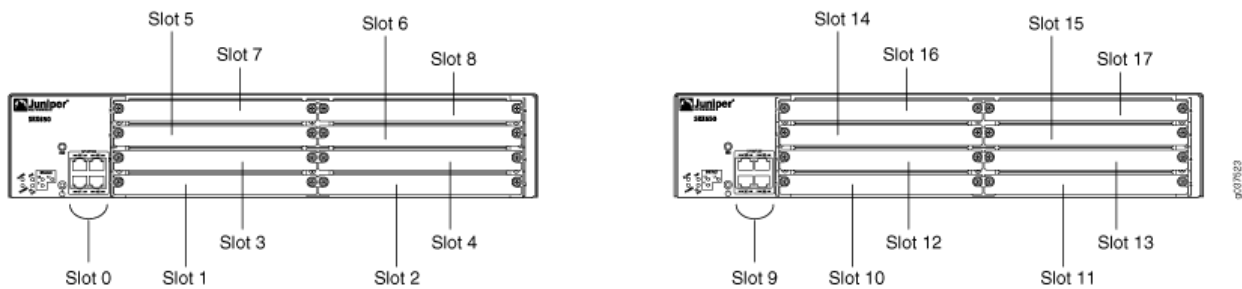


Figure 101: Slot Numbering in an SRX Series Chassis Cluster (SRX240 Devices)

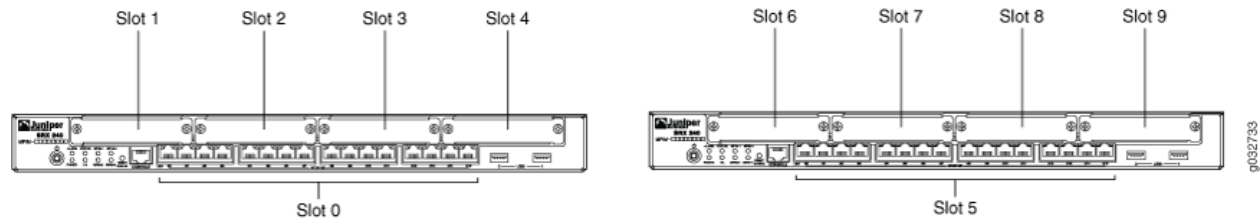


Figure 102: Slot Numbering in an SRX Series Chassis Cluster (SRX220 Devices)

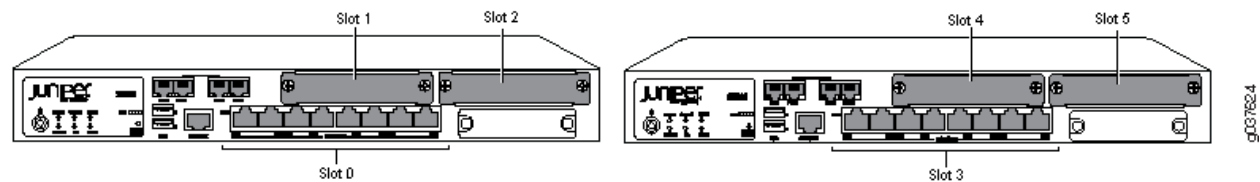


Figure 103: Slot Numbering in an SRX Series Chassis Cluster (SRX210 Devices)

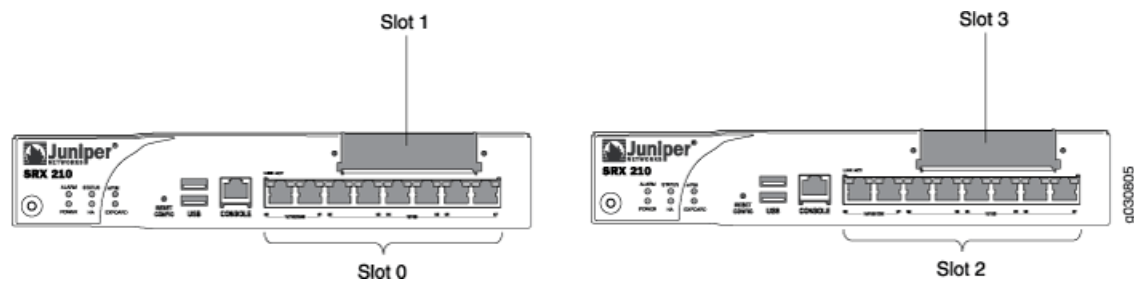
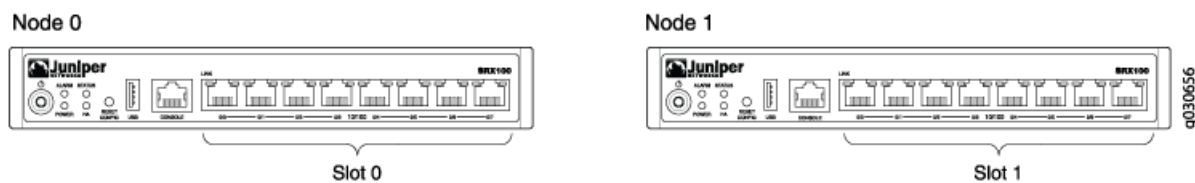


Figure 104: Slot Numbering in an SRX Series Chassis Cluster (SRX100 Devices)



**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding What Happens When Chassis Cluster Is Enabled on page 1210](#)
- [Disabling Switching on SRX100, SRX210, and SRX220 Devices Before Enabling Chassis Clustering on page 1234](#)
- [Node Interfaces on Active J Series Chassis Clusters on page 1219](#)
- [Management Interface on an Active Chassis Cluster on page 1221](#)
- [Fabric Interface on an Active Chassis Cluster on page 1222](#)
- [Control Interface on an Active Chassis Cluster on page 1222](#)



## Node Interfaces on Active J Series Chassis Clusters

Normally, on J Series devices, the built-in interfaces are numbered as follows:

ge-0/0/0                      ge-0/0/1                      ge-0/0/2                      ge-0/0/3                      ...



**CAUTION:** Layer 2 switching must not be enabled on J Series device when chassis clustering is enabled. If you have enabled Layer 2 switching, make sure you disable it before enabling chassis clustering.

After you enable chassis clustering and reboot the system, the built-in interface named ge-0/0/2 is repurposed as the management interface and is automatically renamed **fxp0**. Likewise, the built-in interface named ge-0/0/3 is repurposed as the control interface and is automatically renamed **fxp1**.

After the devices are connected as a cluster, the slot numbering and thus the interface numbering will change for one device. The cluster determines the slot number for each slot in both nodes using the following formula:

**cluster slot number = (node ID \* maximum slots per node) + local slot number**

In chassis cluster mode, the interfaces on the secondary node are renumbered internally. For example, the management interface port on the front panel of each J2320 device is still labeled ge-0/0/2, but internally, the node 1 port is referred to as ge-4/0/2.

Table 115 on page 1219 shows the slot numbering, as well as the port and interface numbering, for both of the J Series devices that become node 0 and node 1 of the cluster after the cluster is formed.

**Table 115: J Series Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming**

Model	Chassis	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
J2320	Node 0	4 (PIM slots); <i>includes one preset slot</i>	0 – 3	ge-0/0/2	ge-0/0/3	Any Gigabit Ethernet port
				fxp0	fxp1	fab0
	Node 1		4 – 7	ge-4/0/2	ge-4/0/3	Any Gigabit Ethernet port
				fxp0	fxp1	fab1

Table 115: J Series Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming (*continued*)

Model	Chassis	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
J2350	Node 0	6 (PIM slots); <i>includes one preset slot</i>	0 – 5	ge-0/0/2	ge-0/0/3	Any Gigabit Ethernet port
				fxp0	fxp1	fab0
	Node 1		6 – 11	ge-6/0/2	ge-6/0/3	Any Gigabit Ethernet port
				fxp0	fxp1	fab1
J4350 and J6350	Node 0	7 (PIM slots); <i>includes one preset slot</i>	0 – 6	ge-0/0/2	ge-0/0/3	Any Gigabit Ethernet port
				fxp0	fxp1	fab0
	Node 1		7 – 13	ge-7/0/2	ge-7/0/3	Any Gigabit Ethernet port
				fxp0	fxp1	fab1

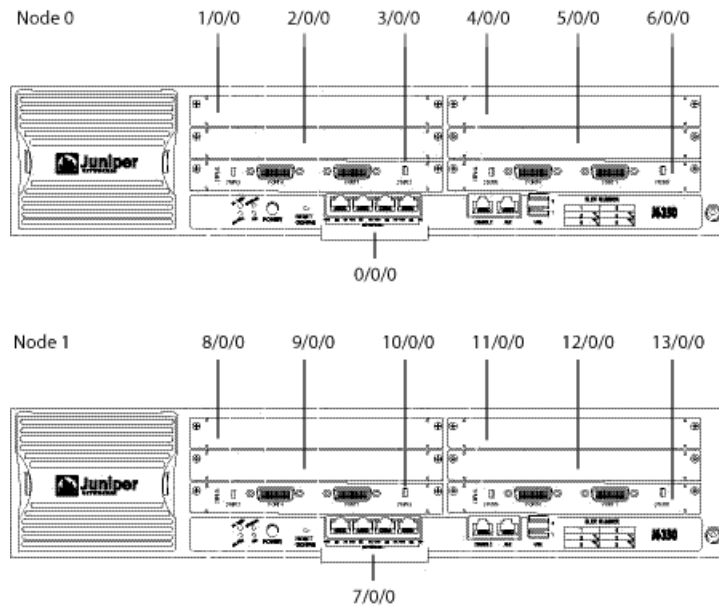
Information about chassis cluster slot numbering is also provided in Figure 105 on page 1221.



**NOTE:** See the *J Series Services Routers Hardware Guide* for details about J Series devices. The *Junos OS Interfaces Configuration Guide for Security Devices* provides a full discussion of the interface naming convention.

After you enable chassis clustering, the two chassis joined together cease to exist as individuals and now represent a single system. As a single system, the cluster now has twice as many PIM slots. (See Figure 105 on page 1221.)

Figure 105: PIM Slot Numbering in a J Series Chassis Cluster (J6350 Devices)



**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding What Happens When Chassis Cluster Is Enabled on page 1210
- Node Interfaces on Active SRX Series Chassis Clusters on page 1211
- Management Interface on an Active Chassis Cluster on page 1221
- Fabric Interface on an Active Chassis Cluster on page 1222
- Control Interface on an Active Chassis Cluster on page 1222

### Management Interface on an Active Chassis Cluster

The **fxp0** interfaces function like standard management interfaces on SRX Series and J Series devices and allow network access to each node in the cluster. You must, however, first connect to each node through the console port and assign a unique IP address to each **fxp0** interface.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding What Happens When Chassis Cluster Is Enabled on page 1210
- Node Interfaces on Active SRX Series Chassis Clusters on page 1211
- Node Interfaces on Active J Series Chassis Clusters on page 1219
- Fabric Interface on an Active Chassis Cluster on page 1222
- Control Interface on an Active Chassis Cluster on page 1222

## Fabric Interface on an Active Chassis Cluster

The fabric is the data link between the nodes and is used to forward traffic between the chassis. Traffic arriving on a node that needs to be processed on the other is forwarded over the fabric data link. Similarly, traffic processed on a node that needs to exit through an interface on the other node is forwarded over the fabric. The fabric also provides for synchronization of session state objects created by operations such as authentication, Network Address Translation (NAT), Application Layer Gateways (ALGs), and IP Security (IPsec) sessions. For SRX Series chassis clusters, the fabric link can be any pair of Ethernet interfaces spanning the cluster; for J Series chassis clusters, the fabric link can be any pair of Gigabit Ethernet interfaces.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding What Happens When Chassis Cluster Is Enabled](#) on page 1210
- [Node Interfaces on Active SRX Series Chassis Clusters](#) on page 1211
- [Node Interfaces on Active J Series Chassis Clusters](#) on page 1219
- [Management Interface on an Active Chassis Cluster](#) on page 1221
- [Control Interface on an Active Chassis Cluster](#) on page 1222

## Control Interface on an Active Chassis Cluster

The control interfaces provide the control link between the two nodes in the cluster and are used for routing updates and for control plane signal traffic, such as heartbeat and threshold information that triggers node failover. The control link is also used to synchronize the configuration between the nodes. When you submit configuration statements to the cluster, the configuration is automatically synchronized over the control link.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding What Happens When Chassis Cluster Is Enabled](#) on page 1210
- [Node Interfaces on Active SRX Series Chassis Clusters](#) on page 1211
- [Node Interfaces on Active J Series Chassis Clusters](#) on page 1219
- [Management Interface on an Active Chassis Cluster](#) on page 1221
- [Fabric Interface on an Active Chassis Cluster](#) on page 1222

## Building a Chassis Cluster

---

- [Connecting SRX Series Hardware to Create a Chassis Cluster](#) on page 1223
- [Layer 2 Ethernet Switching Capability in Chassis Cluster Mode](#) on page 1227
- [Example: Configuring Switch Fabric Interfaces to Enable Switching in Chassis Cluster Mode \(CLI\)](#) on page 1228
- [Example: Configuring IRB and VLAN with Members Across Two Nodes \(CLI\)](#) on page 1229

- Example: Configuring Aggregated Ethernet Device with LAG and LACP (CLI) on page 1232
- Disabling Switching on SRX100, SRX210, and SRX220 Devices Before Enabling Chassis Clustering on page 1234
- SRX Series Chassis Cluster Configuration Overview on page 1235
- Connecting J Series Hardware to Create a Chassis Cluster on page 1238
- J Series Chassis Cluster Configuration Overview on page 1239
- Example: Setting the Chassis Cluster Node ID and Cluster ID on page 1240
- Example: Configuring Chassis Cluster Management Interface on page 1242
- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 1245
- Verifying a Chassis Cluster Configuration on page 1246
- Verifying Chassis Cluster Statistics on page 1247
- Clearing Chassis Cluster Statistics on page 1248
- Verifying Chassis Cluster Failover Status on page 1249
- Clearing Chassis Cluster Failover Status on page 1250

## Connecting SRX Series Hardware to Create a Chassis Cluster

An SRX Series chassis cluster is created by physically connecting two identical cluster-supported SRX Series devices together using a pair of the same type of Ethernet connections. The connection is made for both a control link and a fabric (data) link between the two devices.



**NOTE:** You can connect two control links (SRX1400 Services Gateways and SRX5000 and SRX3000 lines only) and two fabric links between the two devices in the cluster to reduce the chance of control link and fabric link failure. See “Understanding Chassis Cluster Dual Control Links” on page 1188 and “Understanding Chassis Cluster Dual Fabric Links” on page 1200.

Control links in a chassis cluster are made using specific ports.

You must use the following ports to form the control link on the branch SRX Series devices:

- For SRX100 devices, connect the fe-0/0/7 on node 0 to the fe-1/0/7 on node 1.
- For SRX210 devices, connect the fe-0/0/7 on node 0 to the fe-2/0/7 on node 1.
- For SRX220 devices, connect the ge-0/0/7 on node 0 to the ge-3/0/7 on node 1.
- For SRX240 devices, connect the ge-0/0/1 on node 0 to the ge-5/0/1 on node 1.
- For SRX650 devices, connect the ge-0/0/1 on node 0 to the ge-9/0/1 on node 1.

For a device from the SRX3000 line, the connection that serves as the control link must be between the built-in control ports on each device.

SRX5000 line devices do not have built-in ports, so the control link for these gateways must be the control ports on their Services Processing Cards (SPCs) with a slot numbering offset of 6 for SRX5600 devices and 12 for SRX5800 devices.

When you connect a single control link on SRX3000 or SRX5000 line devices, the control link ports are a one-to-one mapping with the Routing Engine slot. If your Routing Engine is in slot 0, you must use control port 0 to link the Routing Engines.



**NOTE:** For dual control links on SRX3000 line devices, the Routing Engine must be in slot 0 and the SRX Clustering Module (SCM) in slot 1. The opposite configuration (SCM in slot 0 and Routing Engine in slot 1) is not supported.

For dual control links on SRX1400 devices, both control (ports 10 and 11) links that are present in the SYSIO can be configured as dual control ports for redundancy.

The fabric link connection for the SRX100 must be a pair of Fast Ethernet interfaces and for the SRX210 must be a pair of either Fast Ethernet or Gigabit Ethernet interfaces. The fabric link connection must be any pair of either Gigabit Ethernet or 10-Gigabit Ethernet interfaces on all other SRX Series devices.

Figure 106 on page 1224, Figure 107 on page 1225, Figure 108 on page 1225, Figure 109 on page 1225, Figure 110 on page 1225, Figure 111 on page 1226, Figure 112 on page 1226, and Figure 113 on page 1226 show pairs of SRX Series devices with the fabric links and control links connected.

**Figure 106: Connecting SRX Series Devices in a Cluster (SRX5800 Devices)**

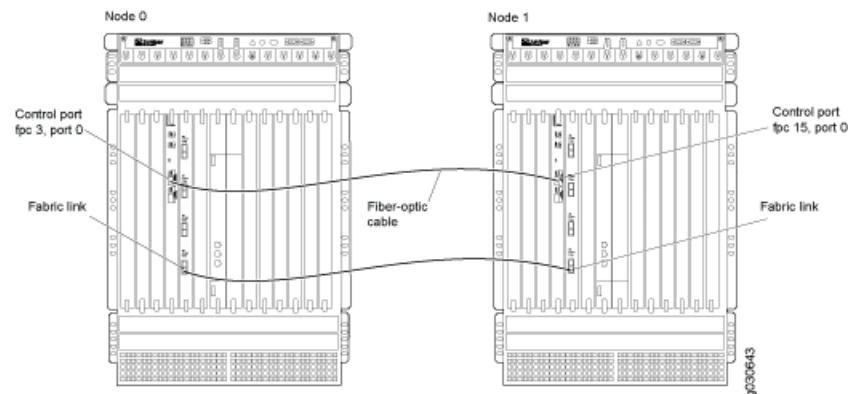


Figure 107: Connecting SRX Series Devices in a Cluster (SRX3400 Devices)

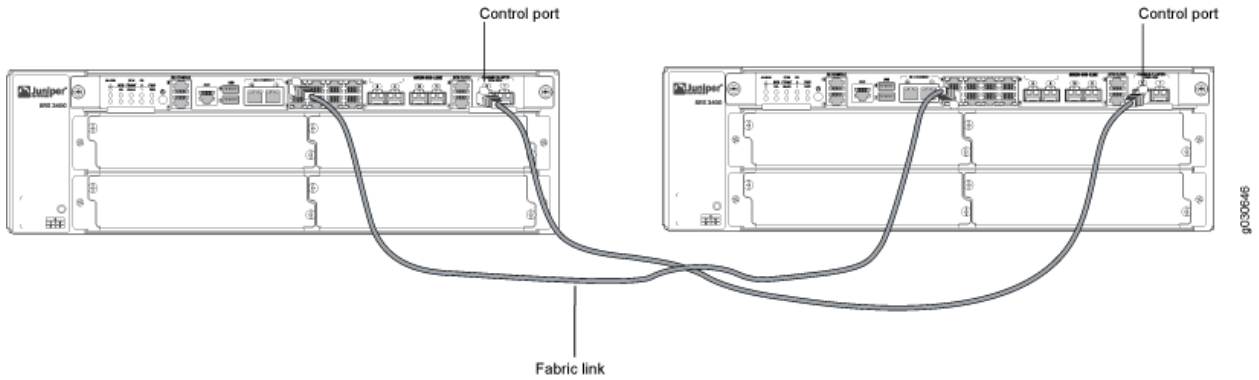


Figure 108: Connecting SRX Series Devices in a Cluster (SRX1400 Devices)

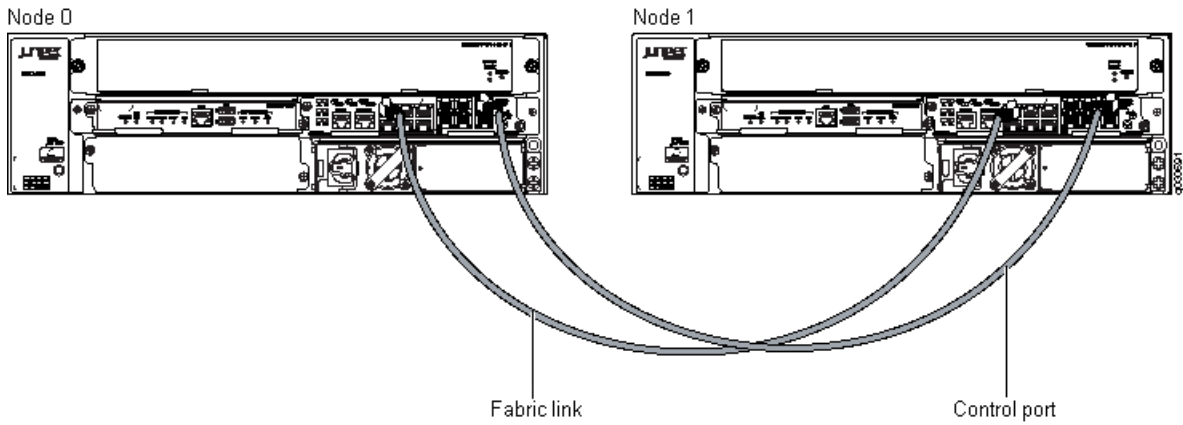


Figure 109: Connecting SRX Series Devices in a Cluster (SRX650 Devices)

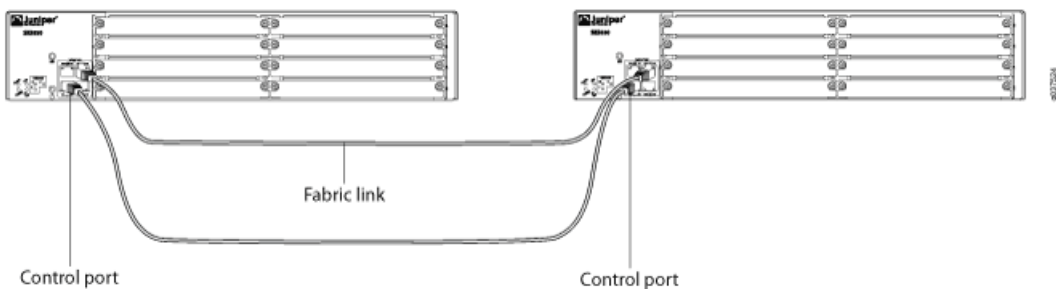


Figure 110: Connecting SRX Series Devices in a Cluster (SRX240 Devices)

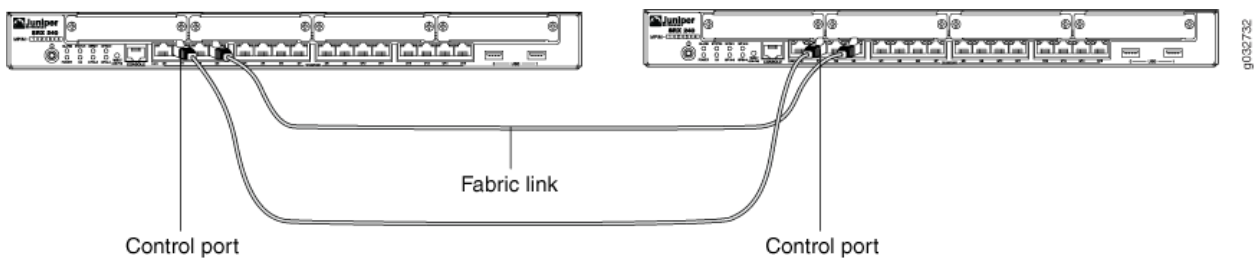


Figure 111: Connecting SRX Series Devices in a Cluster (SRX220 Devices)

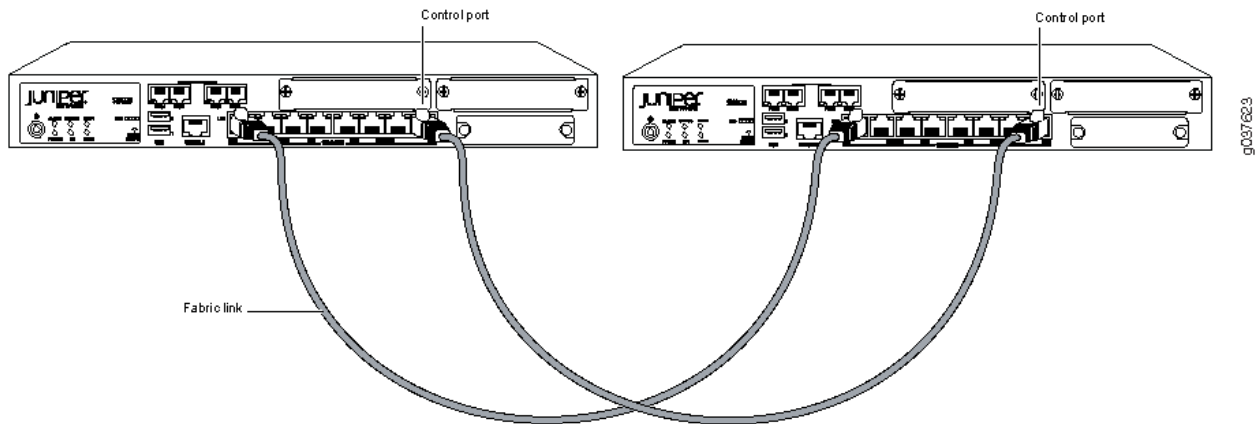


Figure 112: Connecting SRX Series Devices in a Cluster (SRX210 Devices)

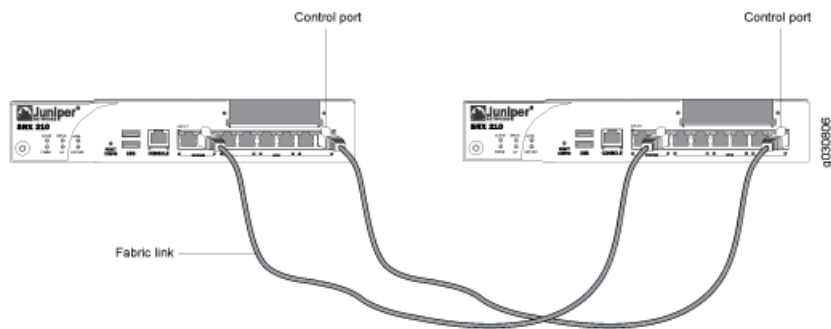
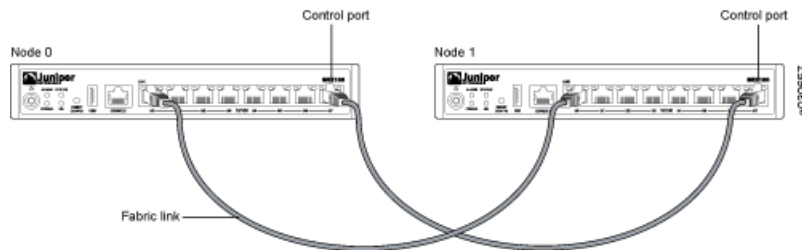


Figure 113: Connecting SRX Series Devices in a Cluster (SRX100 Devices)



**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding What Happens When Chassis Cluster Is Enabled on page 1210](#)
- [SRX Series Chassis Cluster Configuration Overview on page 1235](#)
- [Disabling Switching on SRX100, SRX210, and SRX220 Devices Before Enabling Chassis Clustering on page 1234](#)
- [Example: Setting the Chassis Cluster Node ID and Cluster ID on page 1240](#)
- [Example: Configuring Chassis Cluster Management Interface on page 1242](#)
- [Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 1245](#)



## Layer 2 Ethernet Switching Capability in Chassis Cluster Mode

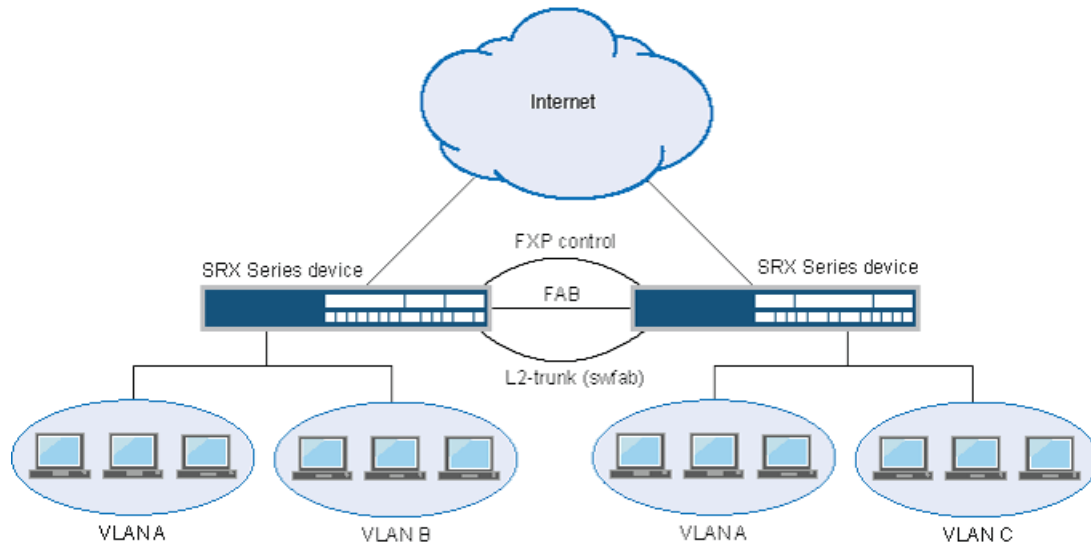
- Understanding Layer 2 Ethernet Switching Capability in Chassis Cluster on SRX240 and SRX650 Devices on page 1227

### Understanding Layer 2 Ethernet Switching Capability in Chassis Cluster on SRX240 and SRX650 Devices

Ethernet ports support various Layer 2 features such as Spanning Tree Protocols (xSTP), DOT1X, Link Aggregation (LAG), Internet Group Membership Protocol (IGMP), GARP, VLAN Registration Protocol (GVRP), Link Layer Discovery Protocol (LLDP), and snooping. The enhanced feature extends Layer 2 switching capability to devices in a chassis cluster. This feature allows users to use Ethernet switching features on both nodes of a chassis cluster. The Ethernet ports on either of the nodes can be configured for family Ethernet switching. Users can configure a Layer 2 VLAN domain with member ports from both the nodes and the Layer 2 switching protocols on both the devices.

Figure 114 on page 1227 shows the Layer 2 switching across chassis cluster nodes:

Figure 114: Layer 2 Ethernet Switching Across Chassis Cluster Nodes



To ensure that Layer 2 switching works seamlessly across chassis cluster nodes, a dedicated physical link connecting the nodes is required. This type of link is called a *switching fabric interface (swfab)*. Its purpose is to carry Layer 2 traffic between the nodes.



**NOTE:** Configuring a LAG with members across nodes is not supported.



**WARNING:** If swfab interface is not configured on both the nodes and if you try to configure Ethernet switching-related features on the nodes, behavior of the nodes might be unpredictable.

### ***Understanding Chassis Cluster Failover and New Primary Election***

When chassis cluster failover occurs, a new primary node is elected and the Ethernet Switching Daemon (ESWD) runs in a different node. During failover, chassis control subsystem is restarted. Also during failover, the traffic outage occurs until the PICs are up and the VLAN entries are reprogrammed. After fail over, all Layer 2 protocols reconverge because Layer 2 protocols states are not maintained in the secondary node.



**NOTE:** The Q-in-Q feature in chassis cluster mode is not supported because of chip limitation for swfab interface configuration in Broadcom chipsets.

#### **Related Documentation**

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Example: Configuring Switch Fabric Interfaces to Enable Switching in Chassis Cluster Mode (CLI) on page 1228
- Example: Configuring IRB and VLAN with Members Across Two Nodes (CLI) on page 1229
- Example: Configuring Aggregated Ethernet Device with LAG and LACP (CLI) on page 1232

### **Example: Configuring Switch Fabric Interfaces to Enable Switching in Chassis Cluster Mode (CLI)**

This example shows how to configure swfab to enable switching in chassis cluster mode.

- Requirements on page 1228
- Overview on page 1228
- Configuration on page 1229
- Verification on page 1229

#### **Requirements**

---

The physical link used as the switch fabric members must be directly connected. Switching supported ports must be used for swfab interfaces. For SRX650, the swfab member ports must belong to the same GPIM. Members spanning across multiple GPIMs are not supported.

Before you begin, read through the following example to understand the configuration of chassis cluster fabric:

- Example: Configuring the Chassis Cluster Fabric on page 1201

#### **Overview**

---

New pseudointerfaces swfab0 and swfab1 will be created for Layer 2 fabric functionality. Users need to configure dedicated Ethernet ports on each side of the node to be associated with the swfab interface.

## Configuration

### Step-by-Step Procedure

To configure swfab interfaces:

1. Configure swfab0 and swfab1 to associate switch fabric interfaces to enable switching across the nodes. Note that swfab0 corresponds to node 0 and swfab1 corresponds to node 1.

```
{primary:node0} [edit]
user@host# set interfaces swfab0 fabric-options member-interfaces ge-0/0/6
user@host# set interfaces swfab0 fabric-options member-interfaces ge-0/0/7
user@host# set interfaces swfab1 fabric-options member-interfaces ge-5/0/6
user@host# set interfaces swfab1 fabric-options member-interfaces ge-5/0/7
```

2. If you are done configuring the device, commit the configuration.

```
{primary:node0} [edit]
user@host# commit
```

## Verification

**Purpose** Verify that the user will be allowed to configure multiple ports as members of swfab ports.

**Action** From configuration mode, enter the **show interfaces swfab0** command to view the configured interfaces for each port.

```
user@host# show interfaces swfab0
fabric-options{
  member-interfaces {
    ge-0/0/6;
    ge-0/0/7;
  }
}
```

From the configuration mode, enter the **show chassis cluster ethernet-switching interfaces** command to view the appropriate member interfaces.

```
user@host# show chassis cluster ethernet-switching interfaces
swfab0:
  Name                Status
  ge-0/0/6             up
  ge-0/0/7             up
swfab1:
  Name                Status
  ge-5/0/6             up
  ge-5/0/7             up
```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

## Example: Configuring IRB and VLAN with Members Across Two Nodes (CLI)

- Requirements on page 1230
- Overview on page 1230

- Configuration on page 1230
- Verification on page 1231

### Requirements

No special configuration beyond device initialization is required before configuring this feature.

### Overview

This example shows configuration of IRB and configuration of VLAN with members across node 0 and node 1.

### Configuration

#### Step-by-Step Procedure

To configure VLAN, follow the steps from 1 to 4 and then commit the configuration. To configure IRB, follow the steps from 1 to 8.

1. Configure Ethernet switching on the node0 interface.
 

```
{primary:node0} [edit]
user@host# set interfaces ge-2/0/0 unit 0 family ethernet-switching
```
2. Configure Ethernet switching on the node1 interface.
 

```
{primary:node0} [edit]
user@host# set interfaces ge-11/0/0 unit 0 family ethernet-switching
```
3. Create VLAN vlan10 with vlan-id 10.
 

```
{primary:node0} [edit]
user@host# set vlans vlan10 vlan-id 10
```
4. Add ports from both nodes to the VLAN.
 

```
{primary:node0} [edit]
user@host# set vlans vlan10 interface ge-2/0/0
user@host# set vlans vlan10 interface ge-11/0/0
```
5. Assign an IP address to the VLAN.
 

```
{primary:node0} [edit]
user@host# set interfaces vlan unit 10 family inet address 45.45.45.1/24
```
6. Associate Layer 3 VLAN interface to vlan10.
 

```
{primary:node0} [edit]
user@host# set vlans vlan10 l3-interface vlan.10
```
7. Check the configuration by entering the **show vlans** and **show interfaces** commands.
 

```
user@host# show vlans
vlan10 {
  vlan-id 10;
  interface {
    ge-2/0/0.0;
    ge-11/0/0.0;
  }
}
```

```

    13-interface vlan.10;
}

user@host# show interfaces
ge-2/0/0 {
  unit 0 {
    family ethernet-switching;
  }
}
ge-11/0/0 {
  unit 0 {
    family ethernet-switching;
  }
}
vlan {
  unit 10 {
    family inet {
      address 45.45.45.1/24;
    }
  }
}
}

```

- If you are done configuring the device, commit the configuration.

```

[edit]
user@host# commit

```

## Verification

**Purpose** To verify that the configurations of VLAN and IRB are working properly.

**Action** From configuration mode, enter the **show interfaces terse ge-2/0/0** command to view the node 0 interface.

```

user@host# run show interfaces terse ge-2/0/0
Interface      Admin Link Proto  Local      Remote
ge-2/0/0       up    up
ge-2/0/0.0     up    up  eth-switch

```

From configuration mode, enter the **show interfaces terse ge-11/0/0** command to view the node 1 interface.

```

user@host# run show interfaces terse ge-11/0/0
Interface      Admin Link Proto  Local      Remote
ge-11/0/0     up    up
ge-11/0/0.0   up    up  eth-switch

```

From configuration mode, enter the **show vlans** command to view the VLAN interface.

```

user@host# run show vlans
Name      Tag  Interfaces
default   1    None
vlan10    10   ge-2/0/0.0*, ge-11/0/0.0*

```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

## Example: Configuring Aggregated Ethernet Device with LAG and LACP (CLI)

- Requirements on page 1232
- Overview on page 1232
- Configuration on page 1232
- Verification on page 1233

### Requirements

No special configuration beyond device initialization is required before configuring this feature.

### Overview

This example shows the configuration of aggregated Ethernet (ae) devices with LAG and LACP.

### Configuration

#### Step-by-Step Procedure

To configure LAG:

1. Configure the number of ae devices with LAG interface that you need to create.
 

```
[edit]
user@host# set chassis aggregated-devices ethernet device-count 5
```
2. Add a port to the ae device with LAG.
 

```
[edit]
user@host# set interfaces ge-2/0/1 gigheter-options 802.3ad ae0
user@host# set interfaces ge-2/0/2 gigheter-options 802.3ad ae0
```
3. Configure LACP for the ae device with LAG.
 

```
[edit]
user@host# set interfaces ae0 aggregated-ether-options lacp active
```
4. Configure family Ethernet switching for the ae device with LAG.
 

```
[edit]
user@host# set interfaces ae0 unit 0 family ethernet-switching
```
5. Configure VLAN.
 

```
[edit]
user@host# set vlans vlan20 vlan-id 20
```
6. Add the ae interface to the VLAN.
 

```
[edit]
user@host# set vlans vlan20 interface ae0
```
7. Check the configuration by entering the **show vlans** and **show interfaces** commands
 

```
user@host# show vlans
vlan20 {
  vlan-id 20;
  interface {
    ae0.0;
```

```

    }
}

user@host# show interfaces
ge-2/0/1 {
  gigether-options {
    802.3ad ae0;
  }
}
ge-2/0/2 {
  gigether-options {
    802.3ad ae0;
  }
}
ae0 {
  aggregated-ether-options {
    lACP {
      active;
    }
  }
  unit 0 {
    family ethernet-switching;
  }
}

```

- If you are done configuring the device, commit the configuration.

```

[edit]
user@host# commit

```



**NOTE:** Likewise, you can configure other devices with LAG and LACP.

## Verification

**Purpose** Verify that you can configure ae devices with LAG and LACP.

**Action** From configuration mode, enter the **show lACP interfaces** to view the LACP interfaces.

```

user@host# run show lACP interfaces
Aggregated interface: ae0
LACP state:
  Role   Exp  Def  Dist  Col  Syn  Aggr  Timeout  Activity
ge-2/0/1 Actor No   No   Yes  Yes  Yes  Fast    Active
ge-2/0/1 Partner No   No   Yes  Yes  Yes  Fast    Active
ge-2/0/2 Actor No   No   Yes  Yes  Yes  Fast    Active
ge-2/0/2 Partner No   No   Yes  Yes  Yes  Fast    Active
LACP protocol:
  Receive State  Transmit State  Mux State
ge-2/0/1        Current    Fast periodic  Collecting distributing
ge-2/0/2        Current    Fast periodic  Collecting distributing

```

From configuration mode, enter the **show vlans** command to view the VLAN interfaces.

```

user@host# run show vlans
Name      Tag    Interfaces
default   1      None
vlan20    20     ae0.0

```

From configuration mode, enter the **show interfaces (interface name)** command to view the status of the ge-2/0/1 and ge-2/0/2 interfaces.

```
user@host# run show interfaces ge-2/0/1 terse
Interface           Admin Link Proto  Local           Remote
ge-2/0/1            up    up
ge-2/0/1.0          up    up  aenet  --> ae0.0

user@host# run show interfaces ge-2/0/2 terse
Interface           Admin Link Proto  Local           Remote
ge-2/0/2            up    up
ge-2/0/2.0          up    up  aenet  --> ae0.0
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

## Disabling Switching on SRX100, SRX210, and SRX220 Devices Before Enabling Chassis Clustering

The factory default configuration for SRX100, SRX210, and SRX220 devices automatically enables Layer 2 Ethernet switching. Because Layer 2 Ethernet switching is not supported in chassis cluster mode, if you use the factory default configuration for these devices, you must delete the Ethernet switching configuration before you enable chassis clustering.



**NOTE:** Before enabling chassis cluster, any configuration on the control link and management link must be removed.



**NOTE:** On SRX240 and SRX650 devices, Layer 2 Ethernet switching is supported in chassis cluster mode from Junos OS Release 11.1 onward.



**CAUTION:** Enabling chassis clustering while Ethernet switching is enabled is not a supported configuration. Doing so might result in undesirable behavior from the devices, leading to possible network instability.

Specifically, the factory default configuration includes virtual LAN (VLAN) configuration, and the chassis cluster control link is Ethernet switching enabled. To use the control link, the Ethernet switching family must be disabled on the interface.

The following procedure shows how to configure chassis clustering on SRX100, SRX210, and SRX220 devices from the factory default configuration. Follow the procedure before starting the chassis cluster configuration for each of the two devices to be clustered.

We recommend that you do these steps through a console port connection, but if you do not, you will have to connect through the console after you complete the steps. For information about how to connect through the console, see the “Connecting and Configuring the Device” section in the appropriate *SRX Series Services Gateway Getting Started Guide*.



1. Enter configuration mode.
2. Enter the following commands:
 

```
user@host# set system root-authentication plain-text-password
```

This setting is required if a root user password was not set.

```
user@host# delete vlans
user@host# delete interfaces vlan
user@host# delete interfaces interface-range interfaces-trust
user@host# delete security zones security-zone trust interfaces
user@host# commit
```

Note that once you commit this configuration, the management interfaces will be lost and need to be re-created.

Likewise, if you are not using the factory default configuration on these devices (SRX100, SRX210, and SRX220), but you enable Ethernet switching on the interfaces, be sure to disable Ethernet switching before you enable chassis clustering.



**NOTE:** The default configuration for other SRX Series devices and all J Series devices does not automatically enable Ethernet switching. However, if you have enabled Ethernet switching, be sure to disable it before enabling clustering on these devices too.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding What Happens When Chassis Cluster Is Enabled on page 1210](#)
- [Node Interfaces on Active SRX Series Chassis Clusters on page 1211](#)
- [Node Interfaces on Active J Series Chassis Clusters on page 1219](#)
- [Management Interface on an Active Chassis Cluster on page 1221](#)
- [Control Interface on an Active Chassis Cluster on page 1222](#)

## SRX Series Chassis Cluster Configuration Overview

This section provides an overview of the basic steps to create an SRX Series chassis cluster.

For the basic steps to set up a J Series chassis cluster, see “J Series Chassis Cluster Configuration Overview” on page 1239.

Before you begin, connect the SRX Series devices using the instructions in “Connecting SRX Series Hardware to Create a Chassis Cluster” on page 1223



**NOTE:** For SRX5000 line chassis clusters, the placement and type of SPCs must match in the two devices. For SRX3000 line chassis clusters, the placement and type of SPCs, IOCs, and NPCs must match in the two devices. For SRX100, SRX210, SRX220, SRX240, and SRX650 chassis clusters, the placement and type of GPIMs, XGPIMs, XPIMs, and Mini-PIMs (as applicable) must match in the two devices.

To create an SRX Series chassis cluster:

1. Physically connect a pair of the same kind of supported SRX Series devices together:

- a. Create the fabric link between two nodes in a cluster by connecting any pair of Ethernet interfaces. For most SRX Series devices, the only requirement is that both interfaces be Gigabit Ethernet interfaces (or 10-Gigabit Ethernet interfaces). For SRX220 devices, connect a pair of Gigabit Ethernet interfaces. For SRX210 devices, both interfaces must be of a similar type (that is, a pair of Fast Ethernet interfaces or a pair of Gigabit Ethernet interfaces). For SRX100 devices, connect a pair of Fast Ethernet interfaces. Figure 106 on page 1224 shows nodes connected using built-in I/O ports for the fabric link.

When using dual fabric link functionality, connect the two pairs of Ethernet interfaces that you will use on each device. See “Understanding Chassis Cluster Dual Fabric Links” on page 1200.

- b. Connect the control ports that you will use on each device (for example, **fpc3** and **fpc15**, as shown in Figure 106 on page 1224). For SRX3600, SRX3400, SRX650, and SRX240 devices, the control ports are dedicated Gigabit Ethernet ports. For SRX210 and SRX100 devices, the control port is the highest numbered port (**fe-0/0/7**).

When using dual control link functionality (SRX5000 and SRX3000 lines only), connect the two pairs of control ports that you will use on each device (for example, **fpc3** and **fpc15** for the first pair and **fpc6** and **fpc18** for the second, as shown in Figure 94 on page 1189). See “Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster” on page 1189.

For SRX5600 and SRX5800 devices, control ports should be on corresponding slots in the two devices, with the following slot numbering offsets:

Device	Offset
SRX5800	12 (for example, <b>fpc3</b> and <b>fpc15</b> )
SRX5600	6 (for example, <b>fpc3</b> and <b>fpc9</b> )

2. Connect the first device to be initialized in the cluster to the console port. This is the node that forms the cluster.

For connection instructions, see the appropriate *Services Gateway Getting Started Guide*.

3. Configure the control ports (SRX5000 line only). See “Example: Configuring Chassis Cluster Control Ports” on page 1184.
4. Use CLI operational mode commands to enable clustering:
  - a. Identify the cluster by giving it the cluster ID.
  - b. Identify the node by giving it its own node ID and then reboot the system.

See “Example: Setting the Chassis Cluster Node ID and Cluster ID” on page 1240.
5. Connect to the console port on the other device and use CLI operational mode commands to enable clustering:
  - a. Identify the cluster that the device is joining by setting the same cluster ID you set on the first node.
  - b. Identify the node by giving it its own node ID and then reboot the system.
6. Configure the management interfaces on the cluster. See “Example: Configuring Chassis Cluster Management Interface” on page 1242.
7. Configure the cluster with the CLI. See:
  - a. Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 1245
  - b. Example: Configuring the Chassis Cluster Fabric on page 1201
  - c. Example: Configuring Chassis Cluster Redundancy Groups on page 1144
  - d. Example: Configuring Chassis Cluster Redundant Ethernet Interfaces on page 1164
  - e. Example: Configuring Chassis Cluster Interface Monitoring on page 1147
8. Initiate manual failover. See “Initiating a Chassis Cluster Manual Redundancy Group Failover” on page 1158.
9. Configure conditional route advertisement over redundant Ethernet interfaces. See “Understanding Conditional Route Advertising in a Chassis Cluster” on page 1176.
10. Verify the configuration. See “Verifying a Chassis Cluster Configuration” on page 1246.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding What Happens When Chassis Cluster Is Enabled on page 1210](#)
- [Connecting SRX Series Hardware to Create a Chassis Cluster on page 1223](#)
- [Example: Setting the Chassis Cluster Node ID and Cluster ID on page 1240](#)
- [Example: Configuring Chassis Cluster Management Interface on page 1242](#)
- [Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 1245](#)
- [Verifying a Chassis Cluster Configuration on page 1246](#)

## Connecting J Series Hardware to Create a Chassis Cluster

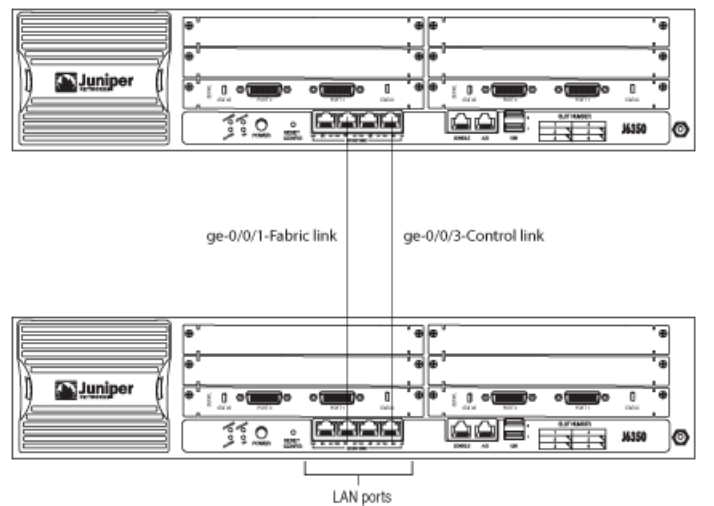
To create a J Series chassis cluster, you must physically connect a pair of the same kind of supported J Series devices back-to-back over a pair of Gigabit Ethernet connections. The connection that serves as the control link must be the built-in interface **ge-0/0/3**. The fabric link connection can be a combination of any pair of Gigabit Ethernet interfaces on the devices.



**NOTE:** You can connect two fabric links between the two devices in the cluster to reduce the chance of fabric link failure. See “Understanding Chassis Cluster Dual Fabric Links” on page 1200.

Figure 115 on page 1238 shows two J Series devices connected using the built-in interfaces for both the fabric and control links.

Figure 115: Connecting J Series Devices in a Cluster (J6350 Devices)



**NOTE:** When chassis clustering is enabled on a J Series router, two interface ports are used to link the two devices: the ge-0/0/3 interface (fxp1 port) is used for the control interface and one port is used for the fabric link (using either one of the built-in interfaces (ge-0/0/0 or ge-0/0/1) or one of the ports of a uPIM). Also, the ge-0/0/2 interface (fxp0 port) is used for the management link. This means that three of the four onboard Gigabit Ethernet ports are in use; if additional ports are required for transit traffic, then a PIM or uPIM is required.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding What Happens When Chassis Cluster Is Enabled](#) on page 1210
- [J Series Chassis Cluster Configuration Overview](#) on page 1239

- Example: Setting the Chassis Cluster Node ID and Cluster ID on page 1240
- Example: Configuring Chassis Cluster Management Interface on page 1242
- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 1245
- Verifying a Chassis Cluster Configuration on page 1246

## J Series Chassis Cluster Configuration Overview

This section provides an overview of the basic steps to create a J Series chassis cluster.

For the basic steps to set up an SRX Series chassis cluster, see “SRX Series Chassis Cluster Configuration Overview” on page 1235.

Before you begin, connect the J Series devices using the instructions in “Connecting J Series Hardware to Create a Chassis Cluster” on page 1238



**NOTE:** For J Series chassis clusters, the two nodes in a cluster must be identical models, but can have any combination of PIMs installed.

To create a J Series chassis cluster:

1. Physically connect a pair of the same kind of supported J Series devices together:
  - a. Create the fabric link between two nodes in a cluster by connecting any pair of Gigabit Ethernet interfaces, either the built-in interfaces or interfaces on the PIMs. The only requirement is that both interfaces be Gigabit Ethernet interfaces. Figure 115 on page 1238 shows nodes connected using the built-in **ge-0/0/1** interface for the fabric link.
 

When using dual fabric link functionality, connect the two pairs of Gigabit Ethernet interfaces that you will use on each device. For more information, see “Understanding Chassis Cluster Dual Fabric Links” on page 1200.
  - b. Connect the **ge-0/0/3** interfaces together to create the control link.
2. Connect the first device to be initialized in the cluster to the console port. This is the node that forms the cluster.
 

For connection instructions, see the *J Series Services Routers Hardware Guide*.
3. Use CLI operational mode commands to enable clustering:
  - a. Identify the cluster by giving it a cluster ID.
  - b. Identify the node by giving it its own node ID and then reboot the system.
 

See “Example: Setting the Chassis Cluster Node ID and Cluster ID” on page 1240.
4. Connect to the console port on the other device and use CLI operational mode commands to enable clustering:

- a. Identify the cluster that the device is joining by setting the same cluster ID you set on the first node.
  - b. Identify the node by giving it its own node ID and then reboot the system.
5. Configure the management interfaces on the cluster. See “Example: Configuring Chassis Cluster Management Interface” on page 1242.
  6. Configure the cluster with the CLI. See:
    - a. Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 1245
    - b. Example: Configuring the Chassis Cluster Fabric on page 1201
    - c. Example: Configuring Chassis Cluster Redundancy Groups on page 1144
    - d. Example: Configuring Chassis Cluster Redundant Ethernet Interfaces on page 1164
    - e. Example: Configuring Chassis Cluster Interface Monitoring on page 1147
  7. Initiate manual failover. See “Initiating a Chassis Cluster Manual Redundancy Group Failover” on page 1158.
  8. Configure conditional route advertisement over redundant Ethernet interfaces. See “Understanding Conditional Route Advertising in a Chassis Cluster” on page 1176.
  9. Verify the configuration. See “Verifying a Chassis Cluster Configuration” on page 1246.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding What Happens When Chassis Cluster Is Enabled on page 1210](#)
- [Connecting J Series Hardware to Create a Chassis Cluster on page 1238](#)
- [Example: Setting the Chassis Cluster Node ID and Cluster ID on page 1240](#)
- [Example: Configuring Chassis Cluster Management Interface on page 1242](#)
- [Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 1245](#)
- [Verifying a Chassis Cluster Configuration on page 1246](#)

### Example: Setting the Chassis Cluster Node ID and Cluster ID

This example shows how to set the chassis cluster node ID and chassis cluster ID, which you must configure after connecting two devices together. A chassis cluster ID identifies the cluster to which the devices belong, and a chassis cluster node ID identifies a unique node within the cluster.

- [Requirements on page 1241](#)
- [Overview on page 1241](#)
- [Configuration on page 1241](#)
- [Verification on page 1241](#)

## Requirements

Before you begin:

- Disable switching on the SRX100, SRX210, and SRX220 devices. See “Disabling Switching on SRX100, SRX210, and SRX220 Devices Before Enabling Chassis Clustering” on page 1234.
- Ensure that you can connect to each device through the console port.

## Overview

The system uses the chassis cluster ID and chassis cluster node ID to apply the correct configuration for each node (for example, when you use the **apply-groups** command to configure the chassis cluster management interface). The chassis cluster ID and node ID statements are written to the EPROM, and the statements take effect when the system is rebooted.

In this example, you configure a chassis cluster ID of 1. You also configure a chassis cluster node ID of 0 for the first node, which allows redundancy groups to be primary on this node when priority settings for both nodes are the same, and a chassis cluster node ID of 1 for the other node.

## Configuration

### Step-by-Step Procedure

To specify the chassis cluster node ID and cluster ID:

1. Connect to the first device through the console port.
 

```
user@host> set chassis cluster cluster-id 1 node 0 reboot
Successfully enabled chassis cluster. Going to reboot now.
```
2. Connect to the second device through the console port.
 

```
user@host> set chassis cluster cluster-id 1 node 1 reboot
Successfully enabled chassis cluster. Going to reboot now.
```

## Verification

To verify the configuration is working properly, perform this task:

### Verifying Chassis Cluster Status

**Purpose** Verify the status of a chassis cluster.

**Action** From operational mode, enter the **show chassis cluster status** command.

```
{primary:node0}[edit]
user@host> show chassis cluster status
```

```
Cluster ID: 1
Node          Priority      Status      Preempt  Manual failover
-----
Redundancy group: 0 , Failover count: 1
node0         100          primary     no       no
node1         1            secondary  no       no
```

```

Redundancy group: 1 , Failover count: 1
node0                0          primary    no        no
node1                0          secondary  no        no

```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding What Happens When Chassis Cluster Is Enabled on page 1210](#)
- [SRX Series Chassis Cluster Configuration Overview on page 1235](#)
- [J Series Chassis Cluster Configuration Overview on page 1239](#)
- [Example: Configuring Chassis Cluster Management Interface on page 1242](#)
- [Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 1245](#)

## Example: Configuring Chassis Cluster Management Interface

This example shows how to provide network management access to a chassis cluster.

### Requirements

Before you begin, set the chassis cluster node ID and cluster ID. See “Example: Setting the Chassis Cluster Node ID and Cluster ID” on page 1240.

### Overview

You must assign a unique IP address to each node in the cluster to provide network management access. This configuration is not replicated across the two nodes.

For most SRX Series chassis clusters, the fxp0 interface is a dedicated port. For SRX100, SRX210, and J Series chassis clusters, the fxp0 interface is repurposed from a built-in interface. In a J Series chassis cluster, you configure management access to the cluster by defining a unique hostname for each node and assigning a unique IP address to the fxp0 interface on each node. The fxp0 interface is created when the system reboots the devices after you designate one node as the primary device and the other as the secondary device.



**NOTE:** If you try to access the nodes in a cluster over the network before you configure the fxp0 interface, you will lose access to the cluster.

In this example, you configure the following information for IPv4:

- Node 0 name—node0-router
- IP address assigned to node 0—10.1.1.1/24
- Node 1 name—node1-router
- IP address assigned to node 1—10.1.1.2/24

In this example, you configure the following information for IPv6:



- Node 0 name—node0-router
- IP address assigned to node 0—2010:2010:201::2/64
- Node 1 name—node1-router
- IP address assigned to node 1—2010:2010:201::3/64

### Configuration

#### CLI Quick Configuration

To quickly configure a chassis cluster management interface for IPv4, copy the following commands and paste them into the CLI:

```
{primary:node0}[edit]
user@host#
set groups node0 system host-name node0-router
set groups node0 interfaces fxp0 unit 0 family inet address 10.1.1.1/24
set groups node1 system host-name node1-router
set groups node1 interfaces fxp0 unit 0 family inet address 10.1.1.2/24
```

To quickly configure a chassis cluster management interface for IPv6, copy the following commands and paste them into the CLI:

```
{primary:node0}[edit]
user@host#
set groups node0 system host-name node0-router
set groups node0 interfaces fxp0 unit 0 family inet6 address 2010:2010:201::2/64
set groups node1 system host-name node1-router
set groups node1 interfaces fxp0 unit 0 family inet6 address 2010:2010:201::3/64
```

#### Step-by-Step Procedure

To configure a chassis cluster management interface for IPv4:

1. Configure the name of node 0 and assign an IP address.

```
{primary:node0}[edit]
user@host# set groups node0 system host-name node0-router
user@host# set groups node0 interfaces fxp0 unit 0 family inet address 10.1.1.1/24
```

2. Configure the name of node 1 and assign an IP address.

```
{primary:node0}[edit]
set groups node1 system host-name node1-router
set groups node1 interfaces fxp0 unit 0 family inet address 10.1.1.2/24
```

3. If you are done configuring the device, commit the configuration.

```
{primary:node0}[edit]
user@host# commit
```

#### Step-by-Step Procedure

To configure a chassis cluster management interface for IPv6:

1. Configure the name of node 0 and assign an IP address.

```
{primary:node0}[edit]
user@host# set groups node0 system host-name node0-router
user@host# set groups node0 interfaces fxp0 unit 0 family inet6 address
2010:2010:201::2/64
```

2. Configure the name of node 1 and assign an IP address.

```
{primary:node0}[edit]
user@host# set groups node1 system host-name node1-router
user@host# set groups node1 interfaces fxp0 unit 0 family inet6 address
2010:2010:201::3/64
```

3. If you are done configuring the device, commit the configuration.

```
{primary:node0}[edit]
user@host# commit
```

**Results** From configuration mode, confirm your configuration by entering the **show groups** and **show apply-groups** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
{primary:node0}[edit]
user@host# show groups
node0 {
  system {
    host-name node0-router;
  }
  interfaces {
    fxp0 {
      unit 0 {
        family inet {
          address 10.1.1.1/24;
        }
      }
    }
  }
}
node1 {
  system {
    host-name node1-router;
  }
  interfaces {
    fxp0 {
      unit 0 {
        family inet {
          address 10.1.1.2/24;
        }
      }
    }
  }
}

{primary:node0}[edit]
user@host# show apply-groups
## Last changed: 2010-09-16 11:08:29 UTC
apply-groups "${node}";
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

---

To confirm that the configuration is working properly, perform this task:

- Verifying the Chassis Cluster Management Interface Configuration on page 1245

#### **Verifying the Chassis Cluster Management Interface Configuration**

**Purpose** Verify the chassis cluster management interface configuration.

**Action** To verify the configuration is working properly, enter the **show config** command.

#### **Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding What Happens When Chassis Cluster Is Enabled on page 1210
- SRX Series Chassis Cluster Configuration Overview on page 1235
- J Series Chassis Cluster Configuration Overview on page 1239
- Example: Setting the Chassis Cluster Node ID and Cluster ID on page 1240
- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 1245
- Verifying a Chassis Cluster Configuration on page 1246

### Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster

This example shows how to specify the number of redundant Ethernet interfaces for a chassis cluster. You must configure the redundant Ethernet interfaces count so that the redundant Ethernet interfaces that you configure are recognized.

- Requirements on page 1245
- Overview on page 1245
- Configuration on page 1246
- Verification on page 1246

#### Requirements

---

Before you begin, set the chassis cluster ID and chassis cluster node ID. See “Example: Setting the Chassis Cluster Node ID and Cluster ID” on page 1240.

#### Overview

---

Before you configure redundant Ethernet interfaces for a chassis cluster, you must specify the number of redundant Ethernet interfaces for the chassis cluster.

In this example, you set the number of redundant Ethernet interfaces for a chassis cluster to 2.

---

## Configuration

### Step-by-Step Procedure

To set the number of redundant Ethernet interfaces for a chassis cluster:

- Specify the number of redundant Ethernet interfaces:
 

```
{primary:node0}[edit]
user@host# set chassis cluster reth-count 2
```
- If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

## Verification

To verify the configuration is working properly, enter the **show configuration chassis cluster** command.

---

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding What Happens When Chassis Cluster Is Enabled on page 1210](#)
- [SRX Series Chassis Cluster Configuration Overview on page 1235](#)
- [J Series Chassis Cluster Configuration Overview on page 1239](#)
- [Example: Setting the Chassis Cluster Node ID and Cluster ID on page 1240](#)
- [Example: Configuring Chassis Cluster Management Interface on page 1242](#)
- [Verifying a Chassis Cluster Configuration on page 1246](#)

## Verifying a Chassis Cluster Configuration

**Purpose** Display chassis cluster verification options.

**Action** From the CLI, enter the **show chassis cluster ?** command:

```
{primary:node1}
user@host> show chassis cluster ?
Possible completions:
  interfaces          Display chassis-cluster interfaces
  statistics          Display chassis-cluster traffic statistics
  status              Display chassis-cluster status
```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Setting the Chassis Cluster Node ID and Cluster ID on page 1240](#)
- [Example: Configuring Chassis Cluster Management Interface on page 1242](#)
- [Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 1245](#)
- [Verifying Chassis Cluster Statistics on page 1247](#)
- [Clearing Chassis Cluster Statistics on page 1248](#)

- Verifying Chassis Cluster Failover Status on page 1249

## Verifying Chassis Cluster Statistics

**Purpose** Display information about chassis cluster services and interfaces.

**Action** From the CLI, enter the **show chassis cluster statistics** command:

```
{primary:node1}
user@host> show chassis cluster statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 798
    Heartbeat packets received: 784
Fabric link statistics:
  Probes sent: 793
  Probes received: 0
Services Synchronized:
  Service name           RTOs sent   RTOs received
  Translation context    0           0
  Incoming NAT           0           0
  Resource manager       0           0
  Session create         0           0
  Session close          0           0
  Session change         0           0
  Gate create            0           0
  Session ageout refresh requests 0           0
  Session ageout refresh replies 0           0
  IPSec VPN              0           0
  Firewall user authentication 0           0
  MGCP ALG               0           0
  H323 ALG               0           0
  SIP ALG                0           0
  SCCP ALG               0           0
  PPTP ALG               0           0
  RTSP ALG               0           0
```

```
{primary:node1}
user@host> show chassis cluster statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
  Control link 1:
    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
Fabric link statistics:
  Probes sent: 258681
  Probes received: 258681
Services Synchronized:
  Service name           RTOs sent   RTOs received
  Translation context    0           0
  Incoming NAT           0           0
  Resource manager       0           0
  Session create         1           0
  Session close          1           0
  Session change         0           0
```

```

Gate create                                0          0
Session ageout refresh requests           0          0
Session ageout refresh replies           0          0
IPSec VPN                                 0          0
Firewall user authentication              0          0
MGCP ALG                                  0          0
H323 ALG                                  0          0
SIP ALG                                   0          0
SCCP ALG                                  0          0
PPTP ALG                                  0          0
RPC ALG                                    0          0
RTSP ALG                                  0          0
RAS ALG                                    0          0
MAC address learning                      0          0
GPRS GTP                                  0          0

```

```

{primary:node1}
user@host> show chassis cluster statistics

```

```

Control link statistics:
Control link 0:
  Heartbeat packets sent: 82371
  Heartbeat packets received: 82321
Control link 1:
  Heartbeat packets sent: 0
  Heartbeat packets received: 0

```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Example: Setting the Chassis Cluster Node ID and Cluster ID on page 1240
- Example: Configuring Chassis Cluster Management Interface on page 1242
- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 1245
- Verifying a Chassis Cluster Configuration on page 1246
- Clearing Chassis Cluster Statistics on page 1248
- Verifying Chassis Cluster Failover Status on page 1249

## Clearing Chassis Cluster Statistics

To clear displayed information about chassis cluster services and interfaces, enter the **clear chassis cluster statistics** command from the CLI:

```

{primary:node1}
user@host> clear chassis cluster statistics

```

```

Cleared control-plane statistics
Cleared data-plane statistics

```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Example: Setting the Chassis Cluster Node ID and Cluster ID on page 1240
- Example: Configuring Chassis Cluster Management Interface on page 1242

- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 1245
- Verifying Chassis Cluster Statistics on page 1247
- Verifying Chassis Cluster Failover Status on page 1249

## Verifying Chassis Cluster Failover Status

**Purpose** Display the failover status of a chassis cluster.

**Action** From the CLI, enter the **show chassis cluster status** command:

```
{primary:node1}
user@host> show chassis cluster status
Cluster ID: 3
Node name           Priority  Status  Preempt  Manual failover

Redundancy-group: 0, Failover count: 1
node0                254     primary no       no
node1                2       secondary no      no

Redundancy-group: 1, Failover count: 1
node0                254     primary no       no
node1                1       secondary no      no
```

```
{primary:node1}
user@host> show chassis cluster status
Cluster ID: 15
Node                Priority  Status  Preempt  Manual failover

Redundancy group: 0 , Failover count: 5
node0               200     primary no       no
node1               0       lost    n/a      n/a

Redundancy group: 1 , Failover count: 41
node0               101     primary no       no
node1               0       lost    n/a      n/a
```

```
{primary:node1}
user@host> show chassis cluster status
Cluster ID: 15
Node                Priority  Status  Preempt  Manual failover

Redundancy group: 0 , Failover count: 5
node0               200     primary no       no
node1               0       unavailable n/a     n/a

Redundancy group: 1 , Failover count: 41
node0               101     primary no       no
node1               0       unavailable n/a     n/a
```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Example: Setting the Chassis Cluster Node ID and Cluster ID on page 1240
- Example: Configuring Chassis Cluster Management Interface on page 1242

- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 1245
- Verifying a Chassis Cluster Configuration on page 1246
- Verifying Chassis Cluster Statistics on page 1247
- Clearing Chassis Cluster Failover Status on page 1250

## Clearing Chassis Cluster Failover Status

To clear the failover status of a chassis cluster, enter the **clear chassis cluster failover-count** command from the CLI:

```
{primary:node1}
user@host> clear chassis cluster failover-count
Cleared failover-count for all redundancy-groups
```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Example: Setting the Chassis Cluster Node ID and Cluster ID on page 1240
- Example: Configuring Chassis Cluster Management Interface on page 1242
- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 1245
- Verifying a Chassis Cluster Configuration on page 1246
- Verifying Chassis Cluster Statistics on page 1247
- Verifying Chassis Cluster Failover Status on page 1249

## Chassis Cluster Upgrades

---

- Upgrading Each Device in a Chassis Cluster Separately on page 1250
- Upgrading Both Devices in a Chassis Cluster Using a Low-Impact ISSU on page 1251

### Upgrading Each Device in a Chassis Cluster Separately

Devices in a chassis cluster can be upgraded separately one at a time; some models allow one device after the other to be upgraded using failover and an in-service software upgrade (ISSU) to reduce the operational impact of the upgrade.

To upgrade each device in a chassis cluster separately:



**NOTE:** During this type of chassis cluster upgrade, a service disruption of about 3 to 5 minutes occurs.

1. Load the new image file on node 0.
2. Perform the image upgrade without rebooting the node by entering:

```
user@host> request system software add image_name
```



3. Load the new image file on node 1.
4. Repeat Step 2.
5. Reboot both nodes simultaneously.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Upgrading Both Devices in a Chassis Cluster Using a Low-Impact ISSU on page 1251](#)
- [Disabling Chassis Cluster on page 1254](#)
- [Verifying a Chassis Cluster Configuration on page 1246](#)
- [Understanding Chassis Cluster Formation on page 1138](#)

### Upgrading Both Devices in a Chassis Cluster Using a Low-Impact ISSU

- [Upgrading Both Devices in a Chassis Cluster Using an ISSU on page 1251](#)
- [Rolling Back Devices in a Chassis Cluster After an ISSU on page 1252](#)
- [Guarding Against Service Failure in a Chassis Cluster ISSU on page 1252](#)
- [Enabling an Automatic Chassis Cluster Node Failback After an ISSU on page 1253](#)
- [Troubleshooting Chassis Cluster ISSU Failures on page 1253](#)
- [Deciphering Mismatched Control Link Statistics During a Chassis Cluster ISSU on page 1253](#)

### Upgrading Both Devices in a Chassis Cluster Using an ISSU

For some platforms, devices in a chassis cluster can be upgraded without a service disruption using an in-service software upgrade (ISSU). The chassis cluster ISSU feature allows both devices in a cluster to be upgraded from supported Junos OS versions with a traffic impact similar to that of redundancy group failovers.



**NOTE:** ISSU does not support software downgrades.



**NOTE:** If you upgrade from a Junos OS version that supports only IPv4 to a version that supports both IPv4 and IPv6, the IPv4 traffic will continue to work during the upgrade process. If you upgrade from a Junos OS version that supports both IPv4 and IPv6 to a version that supports both IPv4 and IPv6, both the IPv4 and IPv6 traffic will continue to work during the upgrade process. Junos OS Release 10.2 and later releases support flow-based processing for IPv6 traffic. For more information, see “Enabling Flow-Based Processing for IPv6 Traffic” in the *Junos OS Security Configuration Guide*.

Before you begin, note the following:

- The ISSUs are available only for Junos OS Release 9.6 and later.

- Before starting an ISSU, you should fail over all redundancy groups so that they are all active on only one device. See “Initiating a Chassis Cluster Manual Redundancy Group Failover” on page 1158
- We recommend that routing protocols graceful restart be enabled prior to starting an ISSU.

Once all redundancy groups are active on one device, the upgrade is initiated by using a request command:

1. Fail over all redundancy groups to one device.
2. Start the ISSU by entering the following command:

```
user@host> request system software in-service-upgrade image_name reboot
```

If **reboot** is not included in the command, you need to manually reboot each device as the ISSU completes the software image update.

3. Wait for both devices to complete the upgrade, then verify that all policies, zones, redundancy groups, and other RTOs return to their correct states. Also verify that both devices in the cluster are running the new Junos OS build.



**NOTE:** During the upgrade, both devices might experience redundancy group failovers, but traffic is not disrupted. Each device validates the package and checks version compatibility before doing the upgrade. If the system finds that the new package is not version compatible with the currently installed version, the device refuses the upgrade or prompts you to take corrective action. Sometimes a single feature is not compatible, in which case the upgrade software prompts you to either abort the upgrade or turn off the feature before doing the upgrade.

---

This feature is available only through the command-line interface. See the “request system software in-service-upgrade” section of the *Junos OS CLI Reference*.

### Rolling Back Devices in a Chassis Cluster After an ISSU

If the ISSU fails to complete and only one device in the cluster has been upgraded, you can roll back to the previous configuration on that device alone by using the following commands on the upgraded device:

- **request chassis cluster in-service-upgrade abort**
- **request system software rollback**
- **request system reboot**

### Guarding Against Service Failure in a Chassis Cluster ISSU

The ISSU command has one option: **no-old-master-upgrade**. This option leaves the current master device in a nonupgraded state, which is a precaution against service failure. The **no-old-master-upgrade** option allows routing control to be quickly returned to the old master device if the newly upgraded device does not operate correctly.

Use of the **no-old-master-upgrade** option requires that you run a standard upgrade on the old master device after the ISSU is completed on the backup device.

If you use the **no-old-master-upgrade** option, when the backup device completes its upgrade and you are confident that the new build is operating as expected, then upgrade the old master as follows:

1. Run **request system software add *image\_name***.
2. Run **request chassis cluster in-service-upgrade abort** to stop the ISSU process.
3. Run **request system reboot**.

### Enabling an Automatic Chassis Cluster Node Failback After an ISSU

If you want redundancy groups to automatically return to node 0 as the primary after the ISSU is complete, you must set the redundancy group priority such that node 0 is primary and enable the preempt option. Note that this method works for all redundancy groups except redundancy group 0. You must manually fail over redundancy group 0. To set the redundancy group priority and enable the preempt option, see “Example: Configuring Chassis Cluster Redundancy Groups” on page 1144. To manually fail over a redundancy group, see “Initiating a Chassis Cluster Manual Redundancy Group Failover” on page 1158.



**NOTE:** To upgrade node 0 and make it available in the chassis cluster, manually reboot node 0. Node 0 does not reboot automatically.

### Troubleshooting Chassis Cluster ISSU Failures

Certain circumstances might cause an ISSU attempt to fail. This section explains two of them.

- If you attempt to upgrade a device pair running a Junos OS image earlier than Release 9.6, the ISSU will fail without changing anything about either device in the cluster. Devices running Junos OS Releases earlier than 9.6 must be upgraded separately using individual device upgrade procedures.
- If the secondary device experiences a power-off condition before it boots up using the new image specified when the ISSU is initiated, when power is restored the newly upgraded device will still be waiting to end the ISSU. To end the ISSU on the secondary device, run **request chassis cluster in-service-upgrade abort** followed by **reboot** to abort the ISSU on that device.

### Deciphering Mismatched Control Link Statistics During a Chassis Cluster ISSU

When using dual control links (supported on the SRX5000 and SRX3000 lines only), mismatched control link statistics might be reported with the **show chassis cluster statistics** and **show chassis cluster control-plane statistics** commands while you run an ISSU with nodes on devices running different releases. (ISSUs are available in Junos OS Release 9.6 and later and dual control links are available in Junos OS Release 10.0 and later.) For example, assume that one node on a device is running Junos OS Release 9.6 and another

node on a device is running Junos OS Release 10.0. In this example, a mismatch might occur because the latter device is sending heartbeats on both control links, but the other device is receiving heartbeats only on one control link.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Upgrading Each Device in a Chassis Cluster Separately on page 1250](#)
- [Disabling Chassis Cluster on page 1254](#)

---

## Disabling Chassis Cluster

To disable chassis cluster, enter the following command:

```
{primary:node1}
user@host> set chassis cluster disable reboot
Successfully disabled chassis cluster. Going to reboot now.
```

After the system reboots, the chassis cluster is disabled.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Upgrading Each Device in a Chassis Cluster Separately on page 1250](#)
- [Upgrading Both Devices in a Chassis Cluster Using a Low-Impact ISSU on page 1251](#)
- [Understanding What Happens When Chassis Cluster Is Enabled on page 1210](#)
- [Understanding Chassis Cluster Formation on page 1138](#)

---

## Understanding Multicast Routing on a Chassis Cluster

Multicast routing support across nodes in a chassis cluster allows multicast protocols, such as Protocol Independent Multicast (PIM) versions 1 and 2, Internet Group Management Protocol (IGMP), Session Announcement Protocol (SAP), and Distance Vector Multicast Routing Protocol (DVMRP), to send traffic across interfaces in the cluster. Note, however, that the multicast protocols should not be enabled on the chassis management interface (**fxp0**) or on the fabric interfaces (**fab0** and **fab1**). Multicast sessions will be synched across the cluster and will be maintained during redundant group failovers. During failover, as with other types of traffic, there might be some multicast packet loss.

Multicast data forwarding in a chassis cluster uses the incoming interface to determine whether or not the session remains active. Packets will be forwarded to the peer node if a leaf session's outgoing interface is on the peer instead of on the incoming interface's node. Multicast routing on a chassis cluster supports tunnels for both incoming and outgoing interfaces.

Multicast configuration on a chassis cluster is the same as multicast configuration on a standalone device (see the "Configuring a Multicast Network" chapter of the *Junos OS Routing Protocols and Policies Configuration Guide for Security Devices*).

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Chassis Cluster Overview on page 1137](#)
  - [Understanding Chassis Cluster Formation on page 1138](#)

## Asymmetric Chassis Cluster Deployment

---

- [Understanding Asymmetric Routing Chassis Cluster Deployment on page 1255](#)
- [Example: Configuring an Asymmetric Chassis Cluster Pair on page 1257](#)

### Understanding Asymmetric Routing Chassis Cluster Deployment

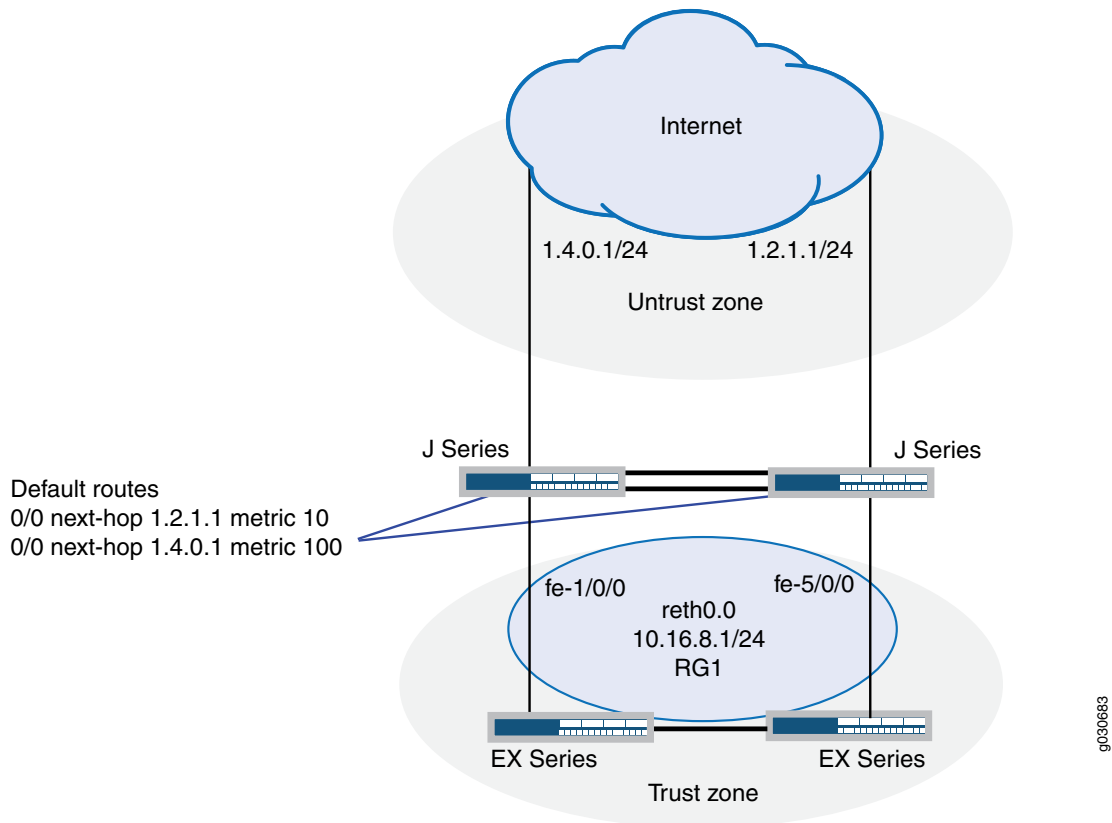
In this case, chassis cluster makes use of its asymmetric routing capability (see Figure 116 on page 1256). Traffic received by a node is matched against that node's session table. The result of this lookup determines whether or not that node should process the packet or forward it to the other node over the fabric link. Sessions are anchored on the egress node for the first packet that created the session. If traffic is received on the node in which the session is not anchored, those packets are forwarded over the fabric link to the node where the session is anchored.



**NOTE:** The anchor node for the session can change if there are changes in routing during the session.

---

Figure 116: Asymmetric Routing Chassis Cluster Scenario (J Series Devices)



In this scenario, two Internet connections are used, with one being preferred. The connection to the trust zone is done by using a redundant Ethernet interface to provide LAN redundancy for the devices in the trust zone. This scenario describes two failover cases in which sessions originate in the trust zone with a destination of the Internet (untrust zone).

- Understanding Failures in the Trust Zone Redundant Ethernet Interface on page 1256
- Understanding Failures in the Untrust Zone Interfaces on page 1257

#### Understanding Failures in the Trust Zone Redundant Ethernet Interface

Under normal operating conditions, traffic flows from the trust zone interface **fe-1/0/0**, belonging to **reth0.0**, to the Internet. Because the primary Internet connection is on node 0, sessions are both created in node 0 and synced to node 1. However, sessions are only active on node 0.

A failure in interface **fe-1/0/0** triggers a failover of the redundancy group, causing interface **fe-5/0/0** in node 1 to become active. After the failover, traffic arrives at node 1. After session lookup, the traffic is sent to node 0 because the session is active on this node. Node 0 then processes the traffic and forwards it to the Internet. The return traffic follows a similar process. The traffic arrives at node 0 and gets processed for security purposes—for example, antispam scanning, antivirus scanning, and application of security

policies—on node 0 because the session is anchored to node 0. The packet is then sent to node 1 through the fabric interface for egress processing and eventual transmission out of node 1 through interface **fe-5/0/0**.

### Understanding Failures in the Untrust Zone Interfaces

In this case, sessions are migrated from node to node. Under normal operating conditions, traffic is processed by only node 0. A failure of interface **ge-0/0/0** on node 0 causes a change in the routing table, so that it now points to interface **ge-7/0/0** in node 1. After the failure, sessions in node 0 become inactive, and the passive sessions in node 1 become active. Traffic arriving from the trust zone is still received on interface **fe-1/0/0**, but is forwarded to node 1 for processing. After traffic is processed in node 1, it is forwarded to the Internet through interface **ge-7/0/0**.

In this chassis cluster configuration, redundancy group 1 is used to control the redundant Ethernet interface connected to the trust zone. As configured in this scenario, redundancy group 1 fails over only if interface **fe-1/0/0** or **fe-5/0/0** fails, but not if the interfaces connected to the Internet fail. Optionally, the configuration could be modified to permit redundancy group 1 to monitor all interfaces connected to the Internet and fail over if an Internet link were to fail. So, for example, the configuration can allow redundancy group 1 to monitor **ge-0/0/0** and make **fe-5/0/0** active for **reth0** if the **ge-0/0/0** Internet link fails. (This option is not described in the following configuration examples.)

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Configuring an Asymmetric Chassis Cluster Pair on page 1257](#)
- [Example: Configuring an Asymmetric Chassis Cluster Pair \(J-Web\)](#)
- [Understanding What Happens When Chassis Cluster Is Enabled on page 1210](#)
- [Understanding Chassis Cluster Formation on page 1138](#)

### Example: Configuring an Asymmetric Chassis Cluster Pair

This example shows how to configure a chassis cluster pair of J Series devices to allow asymmetric routing. Configuring asymmetric routing for a chassis cluster allows traffic received on either device to be processed seamlessly.

- [Requirements on page 1258](#)
- [Overview on page 1258](#)
- [Configuration on page 1261](#)
- [Verification on page 1265](#)

## Requirements

---

Before you begin:

1. Physically connect a pair of J Series devices together, ensuring that they are the same models. This example uses a pair of J2320 Services Router devices.
  - a. To create the fabric link, connect a Gigabit Ethernet interface on one device to another Gigabit Ethernet interface on the other device. See “Connecting J Series Hardware to Create a Chassis Cluster” on page 1238.
  - b. To create the control link, connect the ge-0/0/3 Gigabit Ethernet interfaces of the two devices. See “Connecting J Series Hardware to Create a Chassis Cluster” on page 1238.
2. Connect to one of the devices using the console port. (This is the node that forms the cluster.)
  - a. Set the cluster ID and node number.

```
user@host> set chassis cluster cluster-id 1 node 0 reboot
```
3. Connect to the other device using the console port.
  - a. Set the cluster ID and node number.

```
user@host> set chassis cluster cluster-id 1 node 1 reboot
```

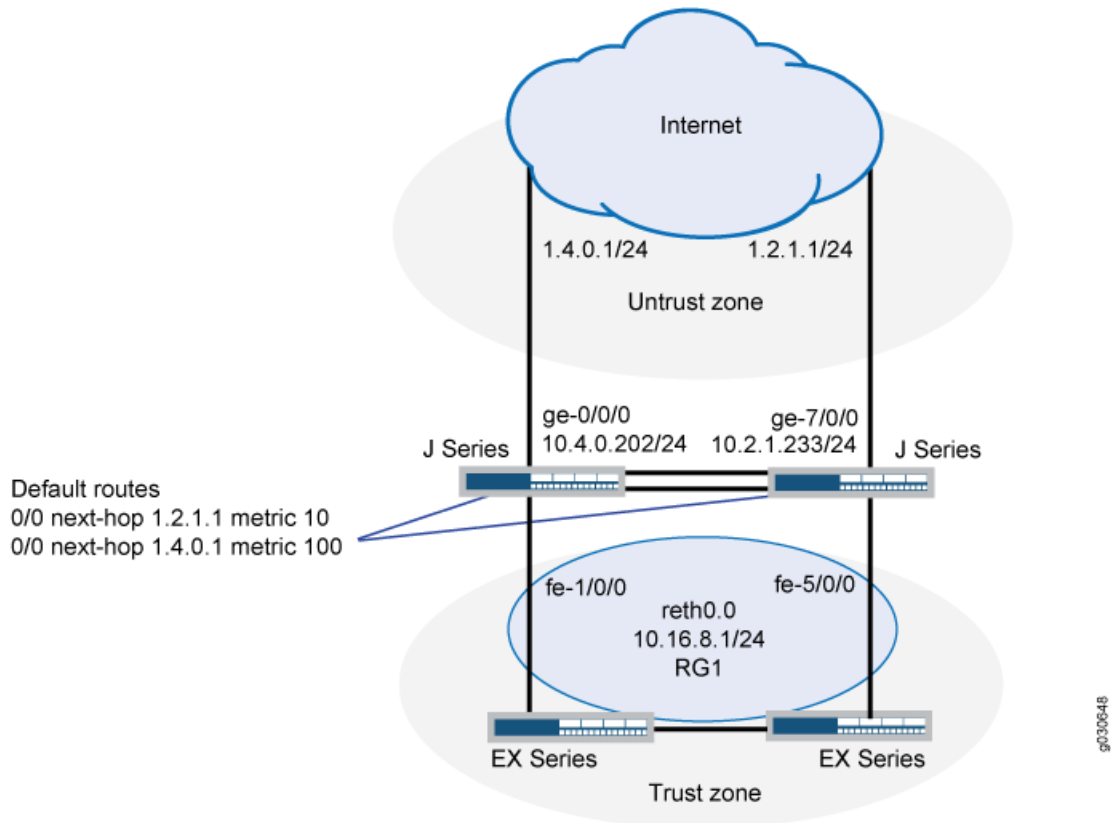
## Overview

---

In this example, a chassis cluster provides asymmetric routing. As illustrated in Figure 117 on page 1259, two Internet connections are used, with one being preferred. The connection to the trust zone is provided by a redundant Ethernet interface to provide LAN redundancy for the devices in the trust zone.



Figure 117: Asymmetric Routing Chassis Cluster Topology



In this example, you configure group (applying the configuration with the `apply-groups` command) and chassis cluster information. Then you configure security zones and security policies. See Table 116 on page 1259 through Table 119 on page 1261.

Table 116: Group and Chassis Cluster Configuration Parameters

Feature	Name	Configuration Parameters
Groups	node0	<ul style="list-style-type: none"> <li>• Hostname: jseries-1</li> <li>• Interface: fxp0 <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 192.168.100.50/24</li> </ul> </li> </ul>
	node1	<ul style="list-style-type: none"> <li>• Hostname: jseries-2</li> <li>• Interface: fxp0 <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 192.168.100.51/24</li> </ul> </li> </ul>

Table 117: Chassis Cluster Configuration Parameters

Feature	Name	Configuration Parameters
Fabric links	fab0	Interface: ge-0/0/1

Table 117: Chassis Cluster Configuration Parameters (*continued*)

Feature	Name	Configuration Parameters
	fab1	Interface: ge-7/0/1
Heartbeat interval	–	1000
Heartbeat threshold	–	3
Redundancy group	1	<ul style="list-style-type: none"> <li>• Priority: <ul style="list-style-type: none"> <li>• Node 0: 100</li> <li>• Node 1: 1</li> </ul> </li> </ul>
		Interface monitoring <ul style="list-style-type: none"> <li>• fe-1/0/0</li> <li>• fe-5/0/0</li> </ul>
Number of redundant Ethernet interfaces	–	1
Interfaces	ge-0/0/0	<ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 1.4.0.202/24</li> </ul>
	ge-7/0/0	<ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 1.2.1.233/24</li> </ul>
	fe-1/0/0	<ul style="list-style-type: none"> <li>•</li> </ul> Redundant parent: reth0
	fe-5/0/0	<ul style="list-style-type: none"> <li>•</li> </ul> Redundant parent: reth0
	reth0	<ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 10.16.8.1/24</li> </ul>

Table 118: Security Zone Configuration Parameters

Name	Configuration Parameters
trust	The reth0.0 interface is bound to this zone.
untrust	The ge-0/0/0.0 and ge-4/0/0.0 interfaces are bound to this zone.

Table 119: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters
This security policy permits traffic from the trust zone to the untrust zone.	ANY	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address any</li> <li>destination-address any</li> <li>application any</li> </ul> </li> <li>Action: permit</li> </ul>

### Configuration

#### CLI Quick Configuration

To quickly configure an asymmetric chassis cluster pair, copy the following commands and paste them into the CLI:

```
{primary:node0}[edit]
set groups node0 system host-name jseries-1
set groups node0 interfaces fxp0 unit 0 family inet address 192.168.100.50/24
set groups node1 system host-name jseries-2
set groups node1 interfaces fxp0 unit 0 family inet address 192.168.100.51/24
set apply-groups "${node}"
set interfaces fab0 fabric-options member-interfaces ge-0/0/1
set interfaces fab1 fabric-options member-interfaces ge-7/0/1
set chassis cluster reth-count 1
set chassis cluster heartbeat-interval 1000
set chassis cluster heartbeat-threshold 3
set chassis cluster redundancy-group 1 node 0 priority 100
set chassis cluster redundancy-group 1 node 1 priority 1
set chassis cluster redundancy-group 1 interface-monitor fe-1/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor fe-5/0/0 weight 255
set interfaces ge-0/0/0 unit 0 family inet address 1.4.0.202/24
set interfaces fe-1/0/0 fastether-options redundant-parent reth0
set interfaces ge-7/0/0 unit 0 family inet address 1.2.1.233/24
set interfaces fe-5/0/0 fastether-options redundant-parent reth0
set interfaces reth0 unit 0 family inet address 10.16.8.1/24
set routing-options static route 0.0.0.0/0 qualified-next-hop 1.4.0.1 metric 10
set routing-options static route 0.0.0.0/0 qualified-next-hop 1.2.1.1 metric 100
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone untrust interfaces ge-7/0/0.0
set security zones security-zone trust interfaces reth0.0
set security policies from-zone trust to-zone untrust policy ANY match source-address
any
set security policies from-zone trust to-zone untrust policy ANY match destination-address
any
set security policies from-zone trust to-zone untrust policy ANY match application any
set security policies from-zone trust to-zone untrust policy ANY then permit
```

#### Step-by-Step Procedure

To configure an asymmetric chassis cluster pair:

1. Configure the management interface.

```
{primary:node0}[edit]
user@host# set groups node0 system host-name jseries-1
user@host# set groups node0 interfaces fxp0 unit 0 family inet address
192.168.100.50/24
```

```

user@host# set groups node1 system host-name jseries-2
user@host# set groups node1 interfaces fxp0 unit 0 family inet address
192.168.100.51/24
user@host# set apply-groups "${node}"

```

2. Configure the fabric interface.

```

{primary:node0}[edit]
user@host# set interfaces fab0 fabric-options member-interfaces ge-0/0/1
user@host# set interfaces fab1 fabric-options member-interfaces ge-7/0/1

```

3. Configure the number of redundant Ethernet interfaces.

```

{primary:node0}[edit]
user@host# set chassis cluster reth-count 1

```

4. Configure the redundancy groups.

```

{primary:node0}[edit]
user@host# set chassis cluster heartbeat-interval 1000
user@host# set chassis cluster heartbeat-threshold 3
user@host# set chassis cluster node 0
user@host# set chassis cluster node 1
user@host# set chassis cluster redundancy-group 1 node 0 priority 100
user@host# set chassis cluster redundancy-group 1 node 1 priority 1
user@host# set chassis cluster redundancy-group 1 interface-monitor fe-1/0/0
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor fe-5/0/0
weight 255

```

5. Configure the redundant Ethernet interfaces.

```

{primary:node0}[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 1.4.0.202/24
user@host# set interfaces fe-1/0/0 fastether-options redundant-parent reth0
user@host# set interfaces ge-7/0/0 unit 0 family inet address 1.2.1.233/24
user@host# set interfaces fe-5/0/0 fastether-options redundant-parent reth0
user@host# set interfaces reth0 unit 0 family inet address 10.16.8.1/24

```

6. Configure the static routes (one to each ISP, with preferred route through ge-0/0/0).

```

{primary:node0}[edit]
user@host# set routing-options static route 0.0.0.0/0 qualified-next-hop 1.4.0.1
metric 10
user@host# set routing-options static route 0.0.0.0/0 qualified-next-hop 1.2.1.1
metric 100

```

7. Configure the security zones.

```

{primary:node0}[edit]
user@host# set security zones security-zone untrust interfaces ge-0/0/0.0
user@host# set security zones security-zone untrust interfaces ge-7/0/0.0
user@host# set security zones security-zone trust interfaces reth0.0

```

8. Configure the security policies.

```

{primary:node0}[edit]
user@host# set security policies from-zone trust to-zone untrust policy ANY match
source-address any
user@host# set security policies from-zone trust to-zone untrust policy ANY match
destination-address any

```

```

user@host# set security policies from-zone trust to-zone untrust policy ANY match
application any
user@host# set security policies from-zone trust to-zone untrust policy ANY then
permit

```

**Results** From operational mode, confirm your configuration by entering the **show configuration** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```

user@host> show configuration
version x.xx.x;
groups {
  node0 {
    system {
      host-name jseries-1;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 192.168.100.50/24;
          }
        }
      }
    }
  }
  node1 {
    system {
      host-name jseries-2;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 192.168.100.51/24;
          }
        }
      }
    }
  }
}
apply-groups "${node}";
chassis {
  cluster {
    reth-count 1;
    heartbeat-interval 1000;
    heartbeat-threshold 3;
    redundancy-group 1 {
      node 0 priority 100;
      node 1 priority 1;
    }
    interface-monitor {
      fe-1/0/0 weight 255;
    }
  }
}

```

```
        fe-5/0/0 weight 255;
    }
}
}
interfaces {
  fe-1/0/0 {
    fastether-options {
      redundant-parent reth0;
    }
  }
  fe-5/0/0 {
    fastether-options {
      redundant-parent reth0;
    }
  }
  ge-0/0/0 {
    unit 0 {
      family inet {
        address 1.4.0.202/24;
      }
    }
  }
  ge-7/0/0 {
    unit 0 {
      family inet {
        address 1.2.1.233/24;
      }
    }
  }
  fab0 {
    fabric-options {
      member-interfaces {
        ge-0/0/1;
      }
    }
  }
  fab1 {
    fabric-options {
      member-interfaces {
        ge-7/0/1;
      }
    }
  }
  reth0 {
    fastether-options {
      redundancy-group 1;
    }
    unit 0 {
      family inet {
        address 10.16.8.1/24;
      }
    }
  }
}
...

```

```

routing-options {
  static {
    route 0.0.0.0/0 {
      next-hop 1.4.0.1;
      metric 10;
    }
  }
}
routing-options {
  static {
    route 0.0.0.0/0 {
      next-hop 1.2.1.1;
      metric 100;
    }
  }
}
security {
  zones {
    security-zone untrust {
      interfaces {
        ge-0/0/0.0;
        ge-7/0/0.0;
      }
    }
    security-zone trust {
      interfaces {
        reth0.0;
      }
    }
  }
  policies {
    from-zone trust to-zone untrust {
      policy ANY {
        match {
          source-address any;
          destination-address any;
          application any;
        }
        then {
          permit;
        }
      }
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Chassis Cluster Status on page 1266
- Verifying Chassis Cluster Interfaces on page 1266
- Verifying Chassis Cluster Statistics on page 1266

- Verifying Chassis Cluster Control Plane Statistics on page 1267
- Verifying Chassis Cluster Data Plane Statistics on page 1267
- Verifying Chassis Cluster Redundancy Group Status on page 1268
- Troubleshooting with Logs on page 1268

#### *Verifying Chassis Cluster Status*

**Purpose** Verify the chassis cluster status, failover status, and redundancy group information.

**Action** From operational mode, enter the **show chassis cluster status** command.

```
{primary:node0}
user@host> show chassis cluster status
Cluster ID: 1
Node                Priority    Status    Preempt  Manual failover

Redundancy group: 1 , Failover count: 1
  node0              100       primary  no       no
  node1              1         secondary no       no
```

#### *Verifying Chassis Cluster Interfaces*

**Purpose** Verify information about chassis cluster interfaces.

**Action** From operational mode, enter the **show chassis cluster interfaces** command.

```
{primary:node0}
user@host> show chassis cluster interfaces
Control link name: fxp1

Redundant-ethernet Information:
  Name      Status    Redundancy-group
  reth0     Up        1

Interface Monitoring:
  Interface  Weight    Status    Redundancy-group
  fe-1/0/0   255      Up        1
  fe-5/0/0   255      Up        1
```

#### *Verifying Chassis Cluster Statistics*

**Purpose** Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitored interfaces in the cluster.

**Action** From operational mode, enter the **show chassis cluster statistics** command.

```
{primary:node0}
user@host> show chassis cluster statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 228
    Heartbeat packets received: 2370
    Heartbeat packets errors: 0
Fabric link statistics:
  Probes sent: 2272
```



```

    Probes received: 597
    Probe errors: 0
    Services Synchronized:
    Service name                RTOs sent   RTOs received
    Translation context         0           0
    Incoming NAT                0           0
    Resource manager            6           0
    Session create              160         0
    Session close               147         0
    Session change              0           0
    Gate create                 0           0
    Session ageout refresh requests 0           0
    Session ageout refresh replies 0           0
    IPSec VPN                   0           0
    Firewall user authentication 0           0
    MGCP ALG                    0           0
    H323 ALG                    0           0
    SIP ALG                     0           0
    SCCP ALG                    0           0
    PPTP ALG                    0           0
    RPC ALG                     0           0
    RTSP ALG                    0           0
    RAS ALG                     0           0
    MAC address learning        0           0
    GPRS GTP                    0           0

```

#### *Verifying Chassis Cluster Control Plane Statistics*

**Purpose** Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).

**Action** From operational mode, enter the `show chassis cluster control-plane statistics` command.

```

{primary:node0}
user@host> show chassis cluster control-plane statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
    Heartbeat packets errors: 0
Fabric link statistics:
  Probes sent: 258681
  Probes received: 258681
  Probe errors: 0

```

#### *Verifying Chassis Cluster Data Plane Statistics*

**Purpose** Verify information about the number of RTOs sent and received for services.

**Action** From operational mode, enter the `show chassis cluster data-plane statistics` command.

```

{primary:node0}
user@host> show chassis cluster data-plane statistics

Services Synchronized:
  Service name                RTOs sent   RTOs received
  Translation context         0           0
  Incoming NAT                0           0

```

Resource manager	6	0
Session create	160	0
Session close	147	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0
Session ageout refresh replies	0	0
IPSec VPN	0	0
Firewall user authentication	0	0
MGCP ALG	0	0
H323 ALG	0	0
SIP ALG	0	0
SCCP ALG	0	0
PPTP ALG	0	0
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

### Verifying Chassis Cluster Redundancy Group Status

**Purpose** Verify the state and priority of both nodes in a cluster and information about whether the primary node has been preempted or whether there has been a manual failover.

**Action** From operational mode, enter the **chassis cluster status redundancy-group** command.

```
{primary:node0}
user@host> show chassis cluster status redundancy-group 1
Cluster ID: 1
  Node           Priority  Status  Preempt  Manual failover

Redundancy-Group: 1, Failover count: 1
node0           100     primary no       no
node1           1       secondary no       no
```

### Troubleshooting with Logs

**Purpose** Use these logs to identify any chassis cluster issues. You should run these logs on both nodes.

**Action** From operational mode, enter these **show** commands.

```
user@host> show log jsrpd
user@host> show log chassisd
user@host> show log messages
user@host> show log dcd
user@host> show traceoptions
```

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Asymmetric Routing Chassis Cluster Deployment on page 1255
- Example: Configuring an Asymmetric Chassis Cluster Pair (J-Web)
- Understanding What Happens When Chassis Cluster Is Enabled on page 1210
- Understanding Chassis Cluster Formation on page 1138

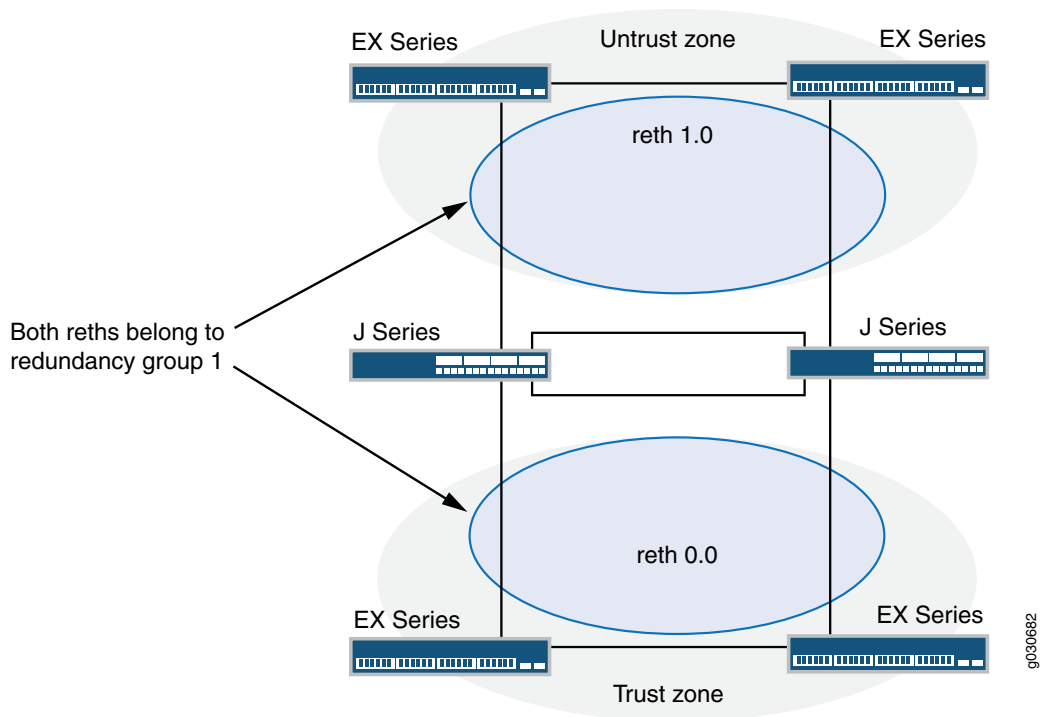
## Active/Passive Chassis Cluster Deployment (J Series Devices)

- Understanding Active/Passive Chassis Cluster Deployment on page 1269
- Example: Configuring an Active/Passive Chassis Cluster Pair (CLI) on page 1270
- Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web) on page 1281

### Understanding Active/Passive Chassis Cluster Deployment

In this case, a single device in the cluster is used to route all traffic while the other device is used only in the event of a failure (see Figure 118 on page 1269). When a failure occurs, the backup device becomes master and controls all forwarding.

Figure 118: Active/Passive Chassis Cluster Scenario (J Series Devices)



An active/passive chassis cluster can be achieved by using redundant Ethernet interfaces (reths) that are all assigned to the same redundancy group. If any of the interfaces in an active group in a node fails, the group is declared inactive and all the interfaces in the group fail over to the other node.

This configuration minimizes the traffic over the fabric link because only one node in the cluster forwards traffic at any given time.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Example: Configuring an Active/Passive Chassis Cluster Pair (CLI) on page 1270
- Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web) on page 1281
- Understanding What Happens When Chassis Cluster Is Enabled on page 1210

- Understanding Chassis Cluster Formation on page 1138

## Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)

This example shows how to configure active/passive chassis clustering for J Series devices.

- Requirements on page 1270
- Overview on page 1270
- Configuration on page 1273
- Verification on page 1278

### Requirements

---

Before you begin:

1. Physically connect a pair of J Series devices together, ensuring that they are the same models. This example uses a pair of J2320 Services Router devices.
  - a. To create the fabric link, connect a Gigabit Ethernet interface on one device to another Gigabit Ethernet interface on the other device. See “Connecting J Series Hardware to Create a Chassis Cluster” on page 1238.
  - b. To create the control link, connect the ge-0/0/3 Gigabit Ethernet interfaces of the two devices. See “Connecting J Series Hardware to Create a Chassis Cluster” on page 1238.
2. Connect to one of the devices using the console port. (This is the node that forms the cluster.)
  - a. Set the cluster ID and node number.

```
user@host> set chassis cluster cluster-id 1 node 0 reboot
```

3. Connect to the other device using the console port.
  - a. Set the cluster ID and node number.

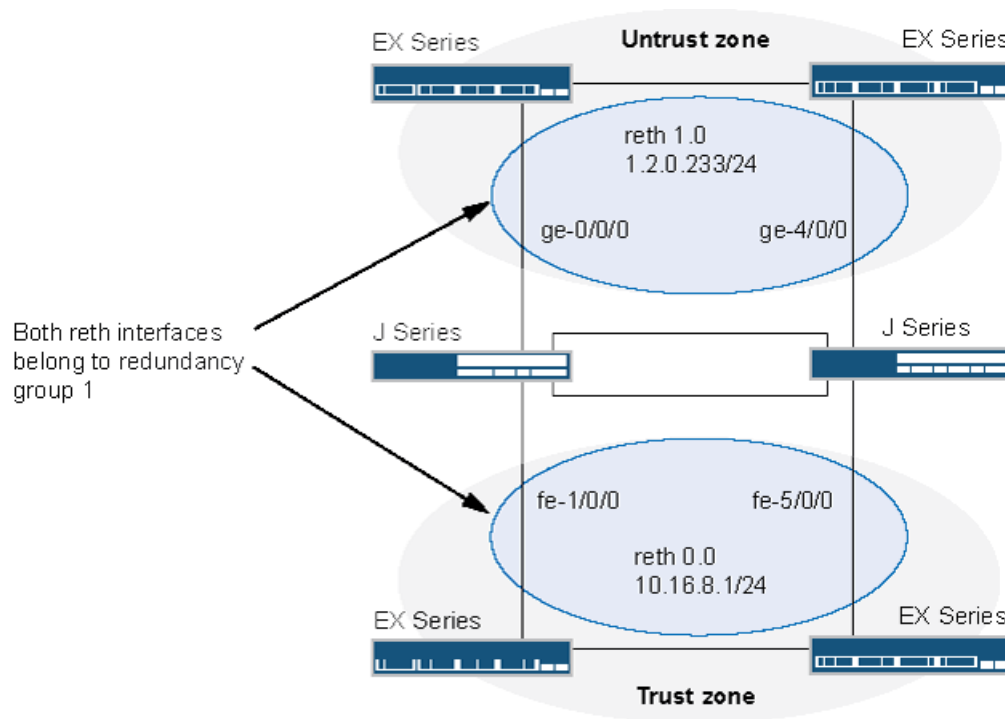
```
user@host> set chassis cluster cluster-id 1 node 1 reboot
```

### Overview

---

In this example, a single device in the cluster is used to route all traffic, and the other device is used only in the event of a failure. (See Figure 119 on page 1271.) When a failure occurs, the backup device becomes master and controls all forwarding.

Figure 119: Active/Passive Chassis Cluster Topology (J Series Devices)



You can create an active/passive chassis cluster by configuring redundant Ethernet interfaces (reths) that are all assigned to the same redundancy group. This configuration minimizes the traffic over the fabric link because only one node in the cluster forwards traffic at any given time.

In this example, you configure group (applying the configuration with the **apply-groups** command) and chassis cluster information. Then you configure security zones and security policies. See Table 120 on page 1271 through Table 123 on page 1273.

Table 120: Group and Chassis Cluster Configuration Parameters

Feature	Name	Configuration Parameters
Groups	node0	<ul style="list-style-type: none"> <li>• Hostname: J2320-A</li> <li>• Interface: fxp0 <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 192.168.3.110/24</li> </ul> </li> </ul>
	node1	<ul style="list-style-type: none"> <li>• Hostname: J2320-B</li> <li>• Interface: fxp0 <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 192.168.3.111/24</li> </ul> </li> </ul>

Table 121: Chassis Cluster Configuration Parameters

Feature	Name	Configuration Parameters
Fabric links	fab0	Interface: ge-0/0/1
	fab1	Interface: ge-4/0/1
Heartbeat interval	–	1000
Heartbeat threshold	–	3
Redundancy group	0	<ul style="list-style-type: none"> <li>Priority: <ul style="list-style-type: none"> <li>Node 0: 254</li> <li>Node 1: 1</li> </ul> </li> </ul>
	1	<ul style="list-style-type: none"> <li>Priority: <ul style="list-style-type: none"> <li>Node 0: 254</li> <li>Node 1: 1</li> </ul> </li> </ul>
		Interface monitoring <ul style="list-style-type: none"> <li>fe-1/0/0</li> <li>fe-5/0/0</li> <li>ge-0/0/0</li> <li>ge-4/0/0</li> </ul>
Number of redundant Ethernet interfaces	–	2
Interfaces	ge-0/0/0	Redundant parent: reth1
	ge-4/0/0	Redundant parent: reth1
	fe-1/0/0	Redundant parent: reth0
	fe-5/0/0	Redundant parent: reth0
	reth0	Redundancy group: 1
		<ul style="list-style-type: none"> <li>Unit 0</li> <li>10.16.8.1/24</li> </ul>
	reth1	Redundancy group: 1
		<ul style="list-style-type: none"> <li>Unit 0</li> <li>11.2.0.233/24</li> </ul>

Table 122: Security Zone Configuration Parameters

Name	Configuration Parameters
trust	The reth1.0 interface is bound to this zone.
untrust	The reth0.0 interface is bound to this zone.

Table 123: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters
This security policy permits traffic from the trust zone to the untrust zone.	ANY	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address any</li> <li>destination-address any</li> <li>application any</li> </ul> </li> <li>Action: permit</li> </ul>

### Configuration

#### CLI Quick Configuration

To quickly configure a chassis cluster on a J2320 Services Router, copy the following commands and paste them into the CLI.

```
[edit]
set groups node0 system host-name J2320-A
set groups node0 interfaces fxp0 unit 0 family inet address 192.168.3.110/24
set groups node1 system host-name J2320-B
set groups node1 interfaces fxp0 unit 0 family inet address 192.168.3.111/24
set apply-groups "${node}"
set interfaces fab0 fabric-options member-interfaces ge-0/0/1
set interfaces fab1 fabric-options member-interfaces ge-4/0/1
set chassis cluster heartbeat-interval 1000
set chassis cluster heartbeat-threshold 3
set chassis cluster redundancy-group 0 node 0 priority 100
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 100
set chassis cluster redundancy-group 1 node 1 priority 1
set chassis cluster redundancy-group 1 interface-monitor fe-1/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor fe-5/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-4/0/0 weight 255
set chassis cluster reth-count 2
set interfaces ge-0/0/0 gigether-options redundant-parent reth1
set interfaces ge-4/0/0 gigether-options redundant-parent reth1
set interfaces fe-1/0/0 fastether-options redundant-parent reth0
set interfaces fe-5/0/0 fastether-options redundant-parent reth0
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 10.16.8.1/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 1.2.0.233/24
set security zones security-zone untrust interfaces reth1.0
set security zones security-zone trust interfaces reth0.0
set security policies from-zone trust to-zone untrust policy ANY match source-address
any
```

```

set security policies from-zone trust to-zone untrust policy ANY match destination-address
any
set security policies from-zone trust to-zone untrust policy ANY match application any
set security policies from-zone trust to-zone untrust policy ANY then permit

```

### Step-by-Step Procedure

To configure an active/passive chassis cluster pair with J2320 Services Router devices:

1. Configure the management interface.

```

{primary:node0}[edit]
user@host# set groups node0 system host-name J2320-A
user@host# set groups node0 interfaces fxp0 unit 0 family inet address
192.168.3.110/24
user@host# set groups node1 system host-name J2320-B
user@host# set groups node1 interfaces fxp0 unit 0 family inet address
192.168.3.111/24
user@host# set apply-groups “${node}”

```

2. Configure the fabric interface.

```

{primary:node0}[edit]
user@host# set interfaces fab0 fabric-options member-interfaces ge-0/0/1
user@host# set interfaces fab1 fabric-options member-interfaces ge-4/0/1

```

3. Configure heartbeat settings.

```

{primary:node0}[edit]
user@host# set chassis cluster heartbeat-interval 1000
user@host# set chassis cluster heartbeat-threshold 3

```

4. Configure redundancy groups.

```

{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 0 node 0 priority 100
user@host# set chassis cluster redundancy-group 0 node 1 priority 1
user@host# set chassis cluster redundancy-group 1 node 0 priority 100
user@host# set chassis cluster redundancy-group 1 node 1 priority 1
user@host# set chassis cluster redundancy-group 1 interface-monitor fe-1/0/0
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor fe-5/0/0
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/0
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-4/0/0
weight 255

```

5. Configure redundant Ethernet interfaces.

```

{primary:node0}[edit]
user@host# set chassis cluster reth-count 2
user@host# set interfaces ge-0/0/0 gigether-options redundant-parent reth1
user@host# set interfaces ge-4/0/0 gigether-options redundant-parent reth1
user@host# set interfaces fe-1/0/0 fastether-options redundant-parent reth0
user@host# set interfaces fe-5/0/0 fastether-options redundant-parent reth0
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
user@host# set interfaces reth0 unit 0 family inet address 10.16.8.1/24
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
user@host# set interfaces reth1 unit 0 family inet address 1.2.0.233/24

```



6. Configure security zones.

```
{primary:node0}[edit]
user@host# set security zones security-zone untrust interfaces reth1.0
user@host# set security zones security-zone trust interfaces reth0.0
```

7. Configure security policies.

```
{primary:node0}[edit]
user@host# set security policies from-zone trust to-zone untrust policy ANY match
source-address any
user@host# set security policies from-zone trust to-zone untrust policy ANY match
destination-address any
user@host# set security policies from-zone trust to-zone untrust policy ANY match
application any
user@host# set security policies from-zone trust to-zone untrust policy ANY then
permit
```

**Results** From configuration mode, confirm your configuration by entering the **show configuration** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
user@host> show configuration
version x.xx.x;
groups {
  node0 {
    system {
      host-name J2320-A;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 192.168.3.110/24;
          }
        }
      }
    }
  }
  node1 {
    system {
      host-name J2320-B;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 192.168.3.111/24;
          }
        }
      }
    }
  }
}
```

```
}
apply-groups "${node}";
chassis {
  cluster {
    reth-count 2;
    heartbeat-interval 1000;
    heartbeat-threshold 3;
    redundancy-group 0 {
      node 0 priority 100;
      node 1 priority 1;
    }
    redundancy-group 1 {
      node 0 priority 100;
      node 1 priority 1;
      interface-monitor {
        fe-1/0/0 weight 255;
        fe-5/0/0 weight 255;
        ge-0/0/0 weight 255;
        ge-4/0/0 weight 255;
      }
    }
  }
}
}
interfaces {
  fe-1/0/0 {
    fastether-options {
      redundant-parent reth0;
    }
  }
  fe-5/0/0 {
    fastether-options {
      redundant-parent reth0;
    }
  }
  ge-0/0/0 {
    fastether-options {
      redundant-parent reth1;
    }
  }
  ge-4/0/0 {
    fastether-options {
      redundant-parent reth1;
    }
  }
  fab0 {
    fabric-options {
      member-interfaces {
        ge-0/0/1;
      }
    }
  }
  fab1 {
    fabric-options {
      member-interfaces {
        ge-4/0/1;
      }
    }
  }
}
```

```

    }
  }
  reth0 {
    redundant-ether-options {
      redundancy-group 1;
    }
    unit 0 {
      family inet {
        address 10.16.8.1/24;
      }
    }
  }
  reth1 {
    redundant-ether-options {
      redundancy-group 1;
    }
    unit 0 {
      family inet {
        address 1.2.0.233/24;
      }
    }
  }
}
...
security {
  zones {
    security-zone untrust {
      interfaces {
        reth1.0;
      }
    }
    security-zone trust {
      interfaces {
        reth0.0;
      }
    }
  }
  policies {
    from-zone trust to-zone untrust {
      policy ANY {
        match {
          source-address any;
          destination-address any;
          application any;
        }
        then {
          permit;
        }
      }
    }
  }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Chassis Cluster Status on page 1278
- Verifying Chassis Cluster Interfaces on page 1278
- Verifying Chassis Cluster Statistics on page 1279
- Verifying Chassis Cluster Control Plane Statistics on page 1279
- Verifying Chassis Cluster Data Plane Statistics on page 1280
- Verifying Chassis Cluster Redundancy Group Status on page 1280
- Troubleshooting with Logs on page 1280

### Verifying Chassis Cluster Status

**Purpose** Verify the chassis cluster status, failover status, and redundancy group information.

**Action** From operational mode, enter the **show chassis cluster status** command.

```
{primary:node0}
user@host> show chassis cluster status
Cluster ID: 1
Node                Priority    Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
  node0              100       primary  no       no
  node1              1         secondary no       no

Redundancy group: 1 , Failover count: 1
  node0              100       primary  no       no
  node1              1         secondary no       no
```

### Verifying Chassis Cluster Interfaces

**Purpose** Verify information about chassis cluster interfaces.

**Action** From operational mode, enter the **show chassis cluster interfaces** command.

```
{primary:node0}
user@host> show chassis cluster interfaces
Control link name: fxp1

Redundant-ethernet Information:
  Name      Status    Redundancy-group
  reth0     Up        1
  reth1     Up        1

Interface Monitoring:
  Interface  Weight    Status    Redundancy-group
  fe-1/0/0   255      Up        1
  fe-5/0/0   255      Up        1
  ge-0/0/0   255      Up        1
  ge-4/0/0   255      Up        1
```

**Verifying Chassis Cluster Statistics**

**Purpose** Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitored interfaces in the cluster.

**Action** From operational mode, enter the **show chassis cluster statistics** command.

```
{primary:node0}
user@host> show chassis cluster statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 2276
    Heartbeat packets received: 2280
    Heartbeat packets errors: 0
Fabric link statistics:
  Probes sent: 2272
  Probes received: 597
  Probe errors: 0
Services Synchronized:
  Service name                RTOs sent   RTOs received
  Translation context          0           0
  Incoming NAT                 0           0
  Resource manager             6           0
  Session create               161         0
  Session close                148         0
  Session change               0           0
  Gate create                  0           0
  Session ageout refresh requests 0           0
  Session ageout refresh replies 0           0
  IPSec VPN                    0           0
  Firewall user authentication 0           0
  MGCP ALG                     0           0
  H323 ALG                     0           0
  SIP ALG                      0           0
  SCCP ALG                     0           0
  PPTP ALG                     0           0
  RPC ALG                      0           0
  RTSP ALG                     0           0
  RAS ALG                      0           0
  MAC address learning         0           0
  GPRS GTP                     0           0
```

**Verifying Chassis Cluster Control Plane Statistics**

**Purpose** Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).

**Action** From operational mode, enter the **show chassis cluster control-plane statistics** command.

```
{primary:node0}
user@host> show chassis cluster control-plane statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
```

```

    Heartbeat packets errors: 0
Fabric link statistics:
  Probes sent: 258681
  Probes received: 258681
  Probe errors: 0

```

### *Verifying Chassis Cluster Data Plane Statistics*

**Purpose** Verify information about the number of RTOs sent and received for services.

**Action** From operational mode, enter the **show chassis cluster data-plane statistics** command.

```

{primary:node0}
user@host> show chassis cluster data-plane statistics

Services Synchronized:
  Service name                RTOs sent  RTOs received
  Translation context          0           0
  Incoming NAT                 0           0
  Resource manager             6           0
  Session create               161         0
  Session close                148         0
  Session change               0           0
  Gate create                  0           0
  Session ageout refresh requests 0           0
  Session ageout refresh replies 0           0
  IPSec VPN                   0           0
  Firewall user authentication 0           0
  MGCP ALG                    0           0
  H323 ALG                    0           0
  SIP ALG                     0           0
  SCCP ALG                    0           0
  PPTP ALG                    0           0
  RPC ALG                     0           0
  RTSP ALG                    0           0
  RAS ALG                     0           0
  MAC address learning        0           0
  GPRS GTP                    0           0

```

### *Verifying Chassis Cluster Redundancy Group Status*

**Purpose** Verify the state and priority of both nodes in a cluster and information about whether the primary node has been preempted or whether there has been a manual failover.

**Action** From operational mode, enter the **chassis cluster status redundancy-group** command.

```

{primary:node0}
user@host> show chassis cluster status redundancy-group 1
Cluster ID: 1
  Node                Priority  Status  Preempt  Manual failover

Redundancy-Group: 1, Failover count: 1
  node0                100     primary no        no
  node1                 1       secondary no        no

```

### *Troubleshooting with Logs*

**Purpose** Use these logs to identify any chassis cluster issues. You should run these logs on both nodes.

**Action** From operational mode, enter these **show** commands.

```
user@host> show log jsrpd
user@host> show log chassisd
user@host> show log messages
user@host> show log dcd
user@host> show traceoptions
```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Active/Passive Chassis Cluster Deployment on page 1269
- Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web) on page 1281
- Understanding What Happens When Chassis Cluster Is Enabled on page 1210
- Understanding Chassis Cluster Formation on page 1138

### Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web)

1. Enable clustering. See Step 1 in “Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)” on page 1270.
2. Configure the management interface. See Step 2 in “Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)” on page 1270.
3. Configure the fabric interface. See Step 3 in “Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)” on page 1270.
4. Configure the redundancy groups.
  - Select **Configure>System Properties>Chassis Cluster**.
  - Enter the following information, and then click **Apply**:
    - Redundant ether-Interface Count: **2**
    - Heartbeat Interval: **1000**
    - Heartbeat Threshold: **3**
    - Nodes: **0**
    - Group Number: **0**
    - Priorities: **100**
  - Enter the following information, and then click **Apply**:
    - Nodes: **0**
    - Group Number: **1**
    - Priorities: **1**
  - Enter the following information, and then click **Apply**:
    - Nodes: **1**

Group Number: **0**

Priorities: **100**

- Enter the following information, and then click **Apply**:

Nodes: **1**

Group Number: **1**

Priorities: **1**

Interface Monitor—Interface: **fe-1/0/0**

Interface Monitor—Weight: **255**

Interface Monitor—Interface: **fe-5/0/0**

Interface Monitor—Weight: **255**

Interface Monitor—Interface: **ge-0/0/0**

Interface Monitor—Weight: **255**

Interface Monitor—Interface: **ge-4/0/0**

Interface Monitor—Weight: **255**

5. Configure the redundant Ethernet interfaces.

- Select **Configure>System Properties>Chassis Cluster**.
- Select **ge-0/0/0**.
- Enter **reth1** in the Redundant Parent box.
- Click **Apply**.
- Select **ge-4/0/0**.
- Enter **reth1** in the Redundant Parent box.
- Click **Apply**.
- Select **fe-1/0/0**.
- Enter **reth0** in the Redundant Parent box.
- Click **Apply**.
- Select **fe-5/0/0**.
- Enter **reth0** in the Redundant Parent box.



- Click **Apply**.
  - See Step 5 in “Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)” on page 1270 for the last four configuration settings.
6. Configure the security zones. See Step 6 in “Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)” on page 1270.
  7. Configure the security policies. See Step 7 in “Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)” on page 1270.
  8. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Active/Passive Chassis Cluster Deployment on page 1269](#)
- [Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\) on page 1270](#)
- [Understanding What Happens When Chassis Cluster Is Enabled on page 1210](#)
- [Understanding Chassis Cluster Formation on page 1138](#)

## Active/Passive Chassis Cluster Deployment (SRX Series Devices)

---

- [Example: Configuring an SRX Series Services Gateway for the Branch as a Chassis Cluster on page 1283](#)
- [Example: Configuring an SRX Series Services Gateway for the High-End as a Chassis Cluster on page 1297](#)

### Example: Configuring an SRX Series Services Gateway for the Branch as a Chassis Cluster

This example shows how to set up chassis clustering on an SRX Series for the branch device.

- [Requirements on page 1283](#)
- [Overview on page 1284](#)
- [Configuration on page 1286](#)
- [Verification on page 1293](#)

#### Requirements

---

Before you begin:

- **Disable switching.** Layer 2 Ethernet switching is not supported in chassis cluster mode. See “Disabling Switching on SRX100, SRX210, and SRX220 Devices Before Enabling Chassis Clustering” on page 1234.
- **Physically connect the two devices and ensure that they are the same models.** For example, on the SRX210 Services Gateway, connect **fe-0/0/7** on node 0 to **fe-0/0/7** on node 1.

- Set the two devices to cluster mode and reboot the devices. You must enter the following operational mode commands on both devices, for example:

- On node 0:

```
user@host> set chassis cluster cluster-id 1 node 0 reboot
```

- On node 1:

```
user@host> set chassis cluster cluster-id 1 node 1 reboot
```

The cluster-id is the same on both devices, but the node ID must be different because one device is node 0 and the other device is node 1. The range for the cluster-id is 0 through 15 and setting it to 0 is equivalent to disabling cluster mode.

- After clustering occurs for the devices, continuing with the SRX210 Services Gateway example, the **fe-0/0/7** interface on node 1 changes to **fe-2/0/7**. After the reboot, the following interfaces are assigned and repurposed to form a cluster:
  - **fe-0/0/6** becomes **fxp0** and is used for individual management of the chassis cluster.
  - **fe-0/0/7** becomes **fxp1** is used as the control link within the chassis cluster.
  - The other interfaces are also renamed on the secondary device. For example, the **ge-0/0/0** interface is renamed **ge-2/0/0** on node 1 on the secondary device.

See “Node Interfaces on Active SRX Series Chassis Clusters” on page 1211 for complete mapping of the SRX Series devices.



**NOTE:** The ports used for the control link, **fe-0/0/7**, must be connected with a cable. A switch cannot be used to connect the control link. You must also decide which port to use as the third link to connect the devices and use as the fabric link between the devices. This port can be any available Gigabit Ethernet or Fast Ethernet port other than **fe-0/0/6** and **fe-0/0/7**.

From this point forward, configuration of the cluster is synchronized between the node members and the two separate devices function as one device.

## Overview

This example shows how to set up chassis clustering on an SRX Series for the branch device. The following services gateways for the branch are supported:

- SRX100
- SRX210
- SRX220
- SRX240
- SRX650

Depending on the device used, node 1 rennumbers its interfaces by adding the total number of system FPCs to the original FPC number of the interface. See Table 124 on page 1285 for interface renumbering on the SRX Series device.

**Table 124: SRX Series Services Gateways Interface Renumbering**

SRX Series Services Gateway	Control Link Name	Renumbering Constant	Node 0 Interface Name	Node 1 Interface Name
SRX100	fe-0/0/7	1	fe-0/0/0	fe-1/0/0
SRX210	fe-0/0/7	2	ge-0/0/0	ge-2/0/0
SRX220	ge-0/0/7	3	ge-0/0/0	ge-3/0/0
SRX240	ge-0/0/1	5	ge-0/0/0	ge-5/0/0
SRX650	ge-0/0/1	9	ge-0/0/0	ge-9/0/0

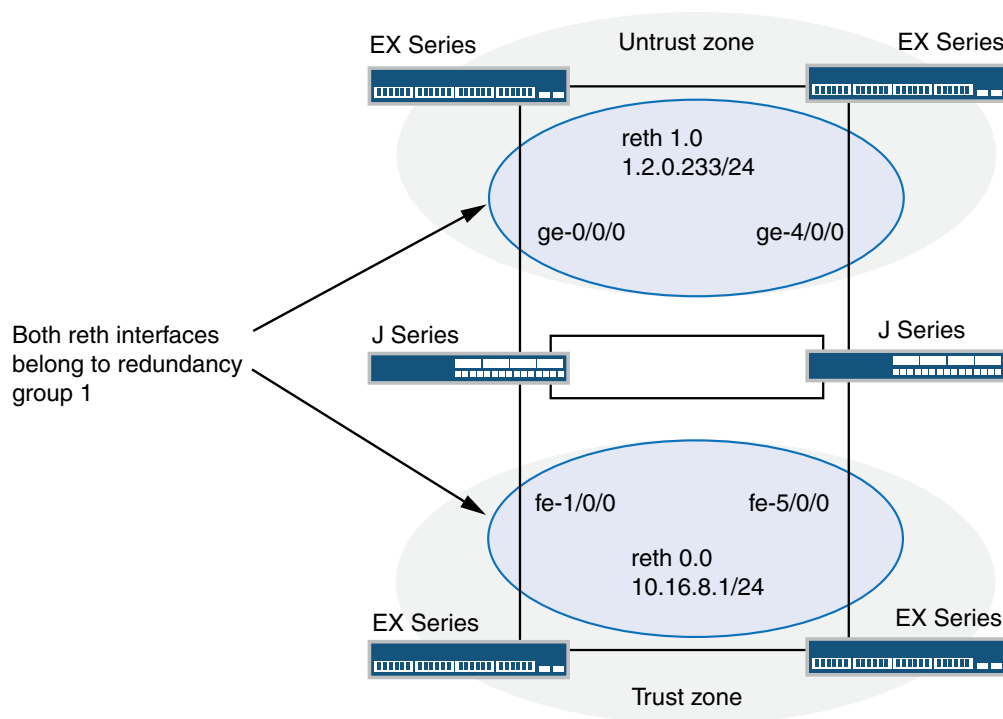
After clustering is enabled, the system creates fxp0, fxp1, and fab interfaces. Depending on the device, the fxp0 and fxp1 interfaces that are mapped to a physical interface are not user defined. However, the fab interface is user defined. see Table 125 on page 1285 for mapping of the fxp0 and fxp1 interfaces on the SRX Series devices.

**Table 125: SRX Series Services Gateways fxp0 and fxp1 Interfaces Mapping**

SRX Series Services Gateway	fxp0 Interface	fxp1 Interface	fab Interface
SRX100	fe-0/0/6	fe-0/0/7	user defined
SRX210	ge-0/0/0	fe-0/0/7	user defined
SRX220	fe-0/0/6	fe-0/0/7	user defined
SRX240	ge-0/0/0	ge-0/0/1	user defined
SRX650	ge-0/0/0	ge-0/0/1	user defined

Figure 120 on page 1286 shows the topology used in this example.

Figure 120: SRX Series for the Branch Topology Example



### Configuration

#### CLI Quick Configuration

To quickly configure a chassis cluster on an SRX210 Services Gateway, copy the following commands and paste them into the CLI:

```
On {primary:node0}
```

```
[edit]
```

```
set groups node0 system host-name srx-node0
```

```
set groups node0 interfaces fxp0 unit 0 family inet address 192.16.35.46/24
```

```
set groups node1 system host-name srx-node1
```

```
set groups node1 interfaces fxp0 unit 0 family inet address 192.16.35.47/24
```

```
set groups node0 system backup-router <backup next-hop from fxp0> destination  
<management network/mask>
```

```
set groups node1 system backup-router <backup next-hop from fxp0> destination  
<management network/mask>
```

```
set apply-groups "${node}"
```

```
set interfaces fab0 fabric-options member-interfaces ge-0/0/1
```

```
set interfaces fab1 fabric-options member-interfaces ge-2/0/1

set chassis cluster redundancy-group 0 node 0 priority 100

set chassis cluster redundancy-group 0 node 1 priority 1

set chassis cluster redundancy-group 1 node 0 priority 100

set chassis cluster redundancy-group 1 node 1 priority 1

set chassis cluster redundancy-group 1 interface-monitor fe-0/0/3 weight 255

set chassis cluster redundancy-group 1 interface-monitor fe-0/0/2 weight 255

set chassis cluster redundancy-group 1 interface-monitor fe-2/0/3 weight 255

set chassis cluster redundancy-group 1 interface-monitor fe-2/0/2 weight 255

set chassis cluster reth-count 2

set interfaces fe-0/0/2 fastether-options redundant-parent reth1

set interfaces fe-2/0/2 fastether-options redundant-parent reth1

set interfaces reth1 redundant-ether-options redundancy-group 1

set interfaces reth1 unit 0 family inet address 1.2.0.233/24

set interfaces fe-0/0/3 fastether-options redundant-parent reth0

set interfaces fe-2/0/3 fastether-options redundant-parent reth0

set interfaces reth0 redundant-ether-options redundancy-group 1

set interfaces reth0 unit 0 family inet address 10.16.8.1/24

set security zones security-zone Untrust interfaces reth1.0

set security zones security-zone Trust interfaces reth0.0
```

If you are configuring an SRX Series for the branch device other than the SRX210 device, see Table 126 on page 1288 for command and interface settings for your device and substitute these commands into your CLI.

Table 126: SRX Series Services Gateways for the Branch Interface Settings

Command	SRX100	SRX210	SRX220	SRX240	SRX650
set interfaces fab0 fabric-options member-interfaces	fe-0/0/1	ge-0/0/1	ge-0/0/0 to ge-0/0/5	ge-0/0/2	ge-0/0/2
set interfaces fab1 fabric-options member-interfaces	fe-1/0/1	ge-2/0/1	ge-3/0/0 to ge-3/0/5	ge-5/0/2	ge-9/0/2
set chassis cluster redundancy-group 1 interface-monitor	fe-0/0/0 weight 255	fe-0/0/3 weight 255	ge-0/0/0 weight 255	ge-0/0/5 weight 255	ge-1/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor	fe-0/0/2 weight 255	fe-0/0/2 weight 255	ge-3/0/0 weight 255	ge-5/0/5 weight 255	ge-10/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor	fe-1/0/0 weight 255	fe-2/0/3 weight 255	ge-0/0/1 weight 255	ge-0/0/6 weight 255	ge-1/0/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor	fe-1/0/2 weight 255	fe-2/0/2 weight 255	ge-3/0/1 weight 255	ge-5/0/6 weight 255	ge-10/0/1 weight 255
set interfaces fastether-options redundant-parent reth1	fe-0/0/2	fe-0/0/2	ge-0/0/2	ge-0/0/5	ge-1/0/0
	fastether-options redundant-parent reth1	fastether-options redundant-parent reth1	fastether-options redundant-parent reth0	gigether-options redundant-parent reth1	gigether-options redundant-parent reth1
set interfaces fastether-options redundant-parent reth1	fe-1/0/2	fe-2/0/2	ge-0/0/3	ge-5/0/5	ge-10/0/0
	fastether-options redundant-parent reth1	fastether-options redundant-parent reth1	fastether-options redundant-parent reth1	gigether-options redundant-parent reth1	gigether-options redundant-parent reth1
set interfaces fastether-options redundant-parent reth0	fe-0/0/0	fe-0/0/3	ge-3/0/2	ge-0/0/6	ge-1/0/1
	fastether-options redundant-parent reth0	fastether-options redundant-parent reth0	fastether-options redundant-parent reth0	gigether-options redundant-parent reth0	gigether-options redundant-parent reth0
set interfaces fastether-options redundant-parent reth0	fe-1/0/0	fe-2/0/3	ge-3/0/3	ge-5/0/6	ge-10/0/1
	fastether-options redundant-parent reth0	fastether-options redundant-parent reth0	fastether-options redundant-parent reth1	gigether-options redundant-parent reth0	gigether-options redundant-parent reth0

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a chassis cluster on an SRX Series for the branch device:



**NOTE:** Perform Steps 1 through 5 on the primary device (node 0). They are automatically copied over to the secondary device (node 1) when you execute a **commit** command. The configurations are synchronized because the control link and fab link interfaces are activated. To verify the configurations, use the **show interface terse** command and review the output.

1. Set up hostnames and management IP addresses for each device using configuration groups. These configurations are specific to each device and are unique to its specific node.

```
user@host# set groups node0 system host-name srx-node0
```

```
user@host# set groups node0 interfaces fxp0 unit 0 family inet address
192.16.35.46/24
```

```
user@host# set groups node1 system host-name srx-node1
```

```
user@host# set groups node1 interfaces fxp0 unit 0 family inet address
192.16.35.47/24
```

Set the default route and backup router for each node.

```
set groups node0 system backup-router <backup next-hop from fxp0> destination
<management network/mask>
```

```
set groups node1 system backup-router <backup next-hop from fxp0> destination
<management network/mask>
```

Set the **apply-group** command so that the individual configurations for each node set by the previous commands are applied only to that node.

```
user@host# set apply-groups "${node}"
```

2. Define the interfaces used for the fab connection (data plane links for RTO sync) by using physical ports **ge-0/0/1** from each node. These interfaces must be connected back-to-back, or through a Layer 2 infrastructure.

```
user@host# set interfaces fab0 fabric-options member-interfaces ge-0/0/1
```

```
user@host# set interfaces fab1 fabric-options member-interfaces ge-2/0/1
```

3. Set up redundancy group 0 for the Routing Engine failover properties, and set up redundancy group 1 (all interfaces are in one redundancy group in this example) to define the failover properties for the redundant Ethernet interfaces.

```
user@host# set chassis cluster redundancy-group 0 node 0 priority 100
```

```
user@host# set chassis cluster redundancy-group 0 node 1 priority 1
```

```
user@host# set chassis cluster redundancy-group 1 node 0 priority 100
```

```
user@host# set chassis cluster redundancy-group 1 node 1 priority 1
```

4. Set up interface monitoring to monitor the health of the interfaces and trigger redundancy group failover.



**NOTE:** We do not recommend Interface monitoring for redundancy group 0 because it causes the control plane to switch from one node to another node in case interface flap occurs.

```
user@host# set chassis cluster redundancy-group 1 interface-monitor fe-0/0/3  
weight 255
```

```
user@host# set chassis cluster redundancy-group 1 interface-monitor fe-0/0/2  
weight 255
```

```
user@host# set chassis cluster redundancy-group 1 interface-monitor fe-2/0/3  
weight 255
```

```
user@host# set chassis cluster redundancy-group 1 interface-monitor fe-2/0/2  
weight 255
```



**NOTE:** Interface failover only occurs after the weight reaches 0.

5. Set up the redundant Ethernet (reth) interfaces and assign the redundant interface to a zone.

```
user@host# set chassis cluster reth-count 2
```

```
user@host# set interfaces fe-0/0/2 fastether-options redundant-parent reth1
```

```
user@host# set interfaces fe-2/0/2 fastether-options redundant-parent reth1
```

```
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
```

```
user@host# set interfaces reth1 unit 0 family inet address 1.2.0.233/24
```

```
user@host# set interfaces fe-0/0/3 fastether-options redundant-parent reth0
```

```
user@host# set interfaces fe-2/0/3 fastether-options redundant-parent reth0
```

```
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
```



```
user@host# set interfaces reth0 unit 0 family inet address 10.16.8.1/24
```

```
user@host# set security zones security-zone Untrust interfaces reth1.0
```

```
user@host# set security zones security-zone Trust interfaces reth0.0
```

**Results** From operational mode, confirm your configuration by entering the **show configuration** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
> show configuration
version x.xx.x;
groups {
  node0 {
    system {
      host-name SRX210-1;
      backup-router 10.100.22.1 destination 66.129.243.0/24;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 192.16.35.46/24;
          }
        }
      }
    }
  }
  node1 {
    system {
      host-name SRX210-2;
      backup-router 10.100.21.1 destination 66.129.243.0/24;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 192.16.35.47/24;
          }
        }
      }
    }
  }
}
apply-groups "${node}";
chassis {
  cluster {
    reth-count 2;
    redundancy-group 0 {
      node 0 priority 100;
      node 1 priority 1;
    }
    redundancy-group 1 {
      node 0 priority 100;
      node 1 priority 1;
    }
  }
}
```

```
        interface-monitor {
            fe-0/0/3 weight 255;
            fe-0/0/2 weight 255;
            fe-2/0/2 weight 255;
            fe-2/0/3 weight 255;
        }
    }
}
interfaces {
    fe-0/0/2 {
        fastether-options {
            redundant-parent reth1;
        }
        unit 0 {
            family inet {
                address 2.2.2.2/30;
            }
        }
    }
    fe-0/0/3 {
        fastether-options {
            redundant-parent reth0;
        }
    }
    fe-2/0/2 {
        fastether-options {
            redundant-parent reth1;
        }
    }
    fe-2/0/3 {
        fastether-options {
            redundant-parent reth0;
        }
    }
    fab0 {
        fabric-options {
            member-interfaces {
                ge-0/0/1;
            }
        }
    }
    fab1 {
        fabric-options {
            member-interfaces {
                ge-2/0/1;
            }
        }
    }
    reth0 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
                address 10.16.8.1/24;
            }
        }
    }
    reth1 {
        redundant-ether-options {
```

```

        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 1.2.0.233/24;
        }
    }
}
...
security {
    zones {
        security-zone Untrust {
            interfaces {
                reth1.0;
            }
        }
        security-zone Trust {
            interfaces {
                reth0.0;
            }
        }
    }
    policies {
        from-zone Trust to-zone Untrust {
            policy 1 {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
    }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Chassis Cluster Status on page 1293
- Verifying Chassis Cluster Interfaces on page 1294
- Verifying Chassis Cluster Statistics on page 1294
- Verifying Chassis Cluster Control Plane Statistics on page 1295
- Verifying Chassis Cluster Data Plane Statistics on page 1295
- Verifying Chassis Cluster Redundancy Group Status on page 1296
- Troubleshooting with Logs on page 1296

#### *Verifying Chassis Cluster Status*

**Purpose** Verify the chassis cluster status, failover status, and redundancy group information.

**Action** From operational mode, enter the **show chassis cluster status** command.

```
{primary:node0}
user@host# show chassis cluster status
Cluster ID: 1
Node                Priority    Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
  node0              100       primary  no       no
  node1              1         secondary no       no

Redundancy group: 1 , Failover count: 1
  node0              0         primary  no       no
  node1              0         secondary no       no
```

### *Verifying Chassis Cluster Interfaces*

**Purpose** Verify information about chassis cluster interfaces.

**Action** From operational mode, enter the **show chassis cluster interfaces** command.

```
{primary:node0}
user@host> show chassis cluster interfaces
Control link name: fxp1

Redundant-ethernet Information:
  Name      Status    Redundancy-group
  reth0     Up        1
  reth1     Up        1

Interface Monitoring:
  Interface  Weight    Status    Redundancy-group
  fe-2/0/3   255      Up        1
  fe-2/0/2   255      Up        1
  fe-0/0/2   255      Up        1
  fe-0/0/3   255      Up        1
```

### *Verifying Chassis Cluster Statistics*

**Purpose** Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitored interfaces in the cluster.

**Action** From operational mode, enter the **show chassis cluster statistics** command.

```
{primary:node0}
user@host> show chassis cluster statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 2276
    Heartbeat packets received: 2280
    Heartbeat packets errors: 0
Fabric link statistics:
  Probes sent: 2272
  Probes received: 597
  Probe errors: 0
Services Synchronized:
  Service name                RTOs sent    RTOs received
  Translation context          0            0
```

Incoming NAT	0	0
Resource manager	6	0
Session create	161	0
Session close	148	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0
Session ageout refresh replies	0	0
IPSec VPN	0	0
Firewall user authentication	0	0
MGCP ALG	0	0
H323 ALG	0	0
SIP ALG	0	0
SCCP ALG	0	0
PPTP ALG	0	0
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

### *Verifying Chassis Cluster Control Plane Statistics*

**Purpose** Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).

**Action** From operational mode, enter the **show chassis cluster control-plane statistics** command.

```
{primary:node0}
user@host> show chassis cluster control-plane statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 2294
    Heartbeat packets received: 2298
    Heartbeat packets errors: 0
Fabric link statistics:
  Probes sent: 2290
  Probes received: 615
  Probe errors: 0
```

### *Verifying Chassis Cluster Data Plane Statistics*

**Purpose** Verify information about the number of RTOs sent and received for services.

**Action** From operational mode, enter the **show chassis cluster data-plane statistics** command.

```
{primary:node0}
user@host> show chassis cluster data-plane statistics

Services Synchronized:
  Service name           RTOs sent  RTOs received
  Translation context    0          0
  Incoming NAT           0          0
  Resource manager       6          0
  Session create         161        0
  Session close          148        0
  Session change         0          0
  Gate create            0          0
```

Session ageout refresh requests	0	0
Session ageout refresh replies	0	0
IPSec VPN	0	0
Firewall user authentication	0	0
MGCP ALG	0	0
H323 ALG	0	0
SIP ALG	0	0
SCCP ALG	0	0
PPTP ALG	0	0
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

### *Verifying Chassis Cluster Redundancy Group Status*

**Purpose** Verify the state and priority of both nodes in a cluster and information about whether the primary node has been preempted or whether there has been a manual failover.

**Action** From operational mode, enter the **chassis cluster status redundancy-group** command.

```
{primary:node0}
user@host> show chassis cluster status redundancy-group 1
Cluster ID: 1
  Node           Priority  Status  Preempt  Manual failover

Redundancy group: 1, Failover count: 1
  node0          100     primary no       no
  node1          50     secondary no       no
```

### *Troubleshooting with Logs*

**Purpose** Use these logs to identify any chassis cluster issues. You should run these logs on both nodes.

**Action** From operational mode, enter these **show log** commands.

```
user@host> show log jsrpd
user@host> show log chassisd
user@host> show log messages
user@host> show log dcd
user@host> show traceoptions
```

- Related Documentation**
- Disabling Switching on SRX100, SRX210, and SRX220 Devices Before Enabling Chassis Clustering on page 1234
  - Understanding Chassis Cluster Redundancy Groups on page 1139.
  - Node Interfaces on Active SRX Series Chassis Clusters on page 1211
  - Example: Configuring an SRX Series Services Gateway for the High-End as a Chassis Cluster on page 1297

## Example: Configuring an SRX Series Services Gateway for the High-End as a Chassis Cluster

This example shows how to set up basic active/passive chassis clustering on a high-end SRX Series device.

- Requirements on page 1297
- Overview on page 1297
- Configuration on page 1299
- Verification on page 1309

### Requirements

---

Before you begin:

- You need two SRX5800 Services Gateways with identical hardware configurations, one MX240 edge router, and one EX8208 Ethernet Switch.
- Physically connect the two devices (back-to-back for the fabric and control ports) and ensure that they are the same models. You can configure both the fabric and control ports on the SRX5000 line. For the SRX1400 devices and the SRX3000 line, you can configure the fabric ports only.
- Set the two devices to cluster mode and reboot the devices. You must enter the following operational mode commands on both devices, for example:

- On node 0:

```
user@host> set chassis cluster cluster-id 1 node 0 reboot
```

- On node 1:

```
user@host> set chassis cluster cluster-id 1 node 1 reboot
```

The cluster-id is the same on both devices, but the node ID must be different because one device is node 0 and the other device is node 1. The range for the cluster-id is 0 through 15 and setting it to 0 is equivalent to disabling cluster mode.

If you have multiple SRX Series clusters on a single L3 broadcast domain, then you must assign different cluster IDs to each cluster, or else there will be a MAC address conflict.

From this point forward, configuration of the cluster is synchronized between the node members and the two separate devices function as one device.

### Overview

---

This example shows how to set up basic active/passive chassis clustering on a high-end SRX Series device. The basic active/passive example is the most common type of chassis cluster. The following high-end SRX Series devices are supported:

- SRX1400
- SRX3400
- SRX3600

- SRX5600
- SRX5800

The basic active/passive chassis cluster consists of two devices:

- One device actively provides routing, firewall, NAT, VPN, and security services, along with maintaining control of the chassis cluster.
- The other device passively maintains its state for cluster failover capabilities should the active device become inactive.



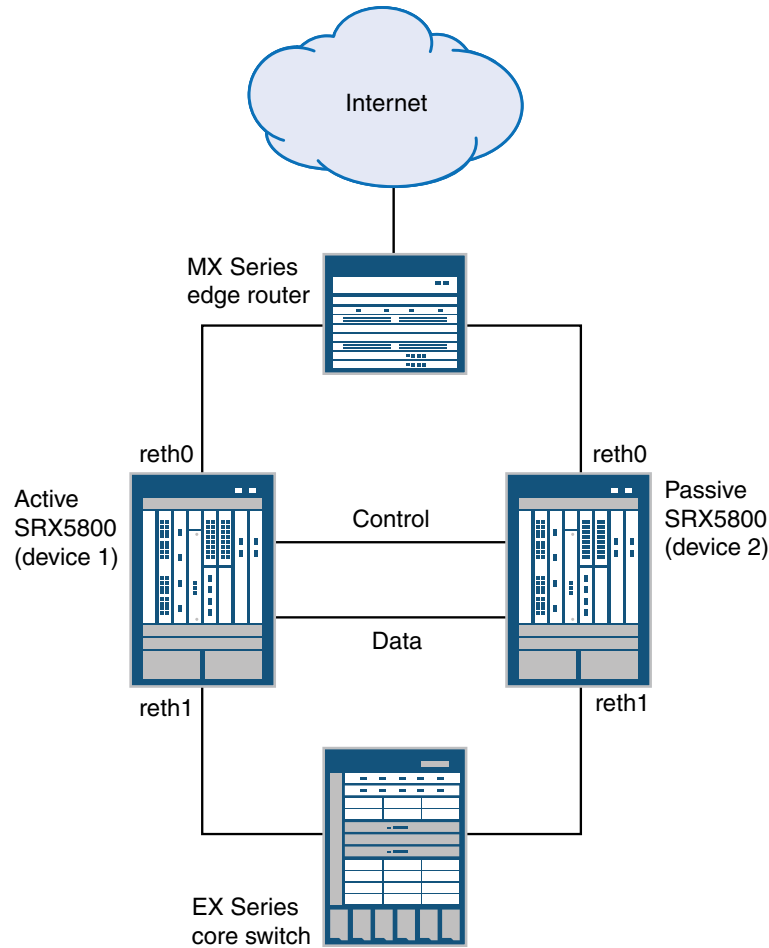
**NOTE:** This active/passive mode example for the SRX5800 Services Gateway does not describe in detail miscellaneous configurations such as how to configure NAT, security policies, or VPNs. They are essentially the same as they would be for standalone configurations. See “NAT Overview” on page 1335, “Security Policies Overview” on page 145, and “VPN Overview” on page 451. However, if you are performing proxy ARP in chassis cluster configurations, you must apply the proxy ARP configurations to the reth interfaces rather than the member interfaces because the RETH interfaces hold the logical configurations. See “Configuring Proxy ARP (CLI Procedure)” on page 1427. You can also configure separate logical interface configurations using VLANs and trunked interfaces in the SRX5800 Services Gateway. These configurations are similar to the standalone implementations using VLANs and trunked interfaces.

---

Figure 121 on page 1299 shows the topology used in this example.



Figure 121: Basic Active/Passive Chassis Clustering on a High-End SRX Series Device Topology Example



### Configuration

**CLI Quick Configuration** To quickly configure a chassis cluster on an SRX5800 Services Gateway, copy the following commands and paste them into the CLI:

On {primary:node0}

[edit]

```
set chassis cluster control-ports fpc 1 port 0
```

```
set chassis cluster control-ports fpc 13 port 0
```

```
set interfaces fab0 fabric-options member-interfaces ge-11/3/0
```

```
set interfaces fab1 fabric-options member-interfaces ge-23/3/0
```

```
set groups node0 system host-name SRX5800-1
```

```
set groups node0 interfaces fxp0 unit 0 family inet address 10.3.5.1/24

set groups node0 system backup-router 10.3.5.254 destination 0.0.0.0/0

set groups node1 system host-name SRX5800-2

set groups node1 interfaces fxp0 unit 0 family inet address 10.3.5.2/24

set groups node1 system backup-router 10.3.5.254 destination 0.0.0.0/0

set apply-groups "${node}"

set chassis cluster reth-count 2

set chassis cluster redundancy-group 0 node 0 priority 129

set chassis cluster redundancy-group 0 node 1 priority 128

set chassis cluster redundancy-group 1 node 0 priority 129

set chassis cluster redundancy-group 1 node 1 priority 128

set interfaces xe-6/0/0 gigether-options redundant-parent reth0

set interfaces xe-6/1/0 gigether-options redundant-parent reth1

set interfaces xe-18/0/0 gigether-options redundant-parent reth0

set interfaces xe-18/1/0 gigether-options redundant-parent reth1

set interfaces reth0 redundant-ether-options redundancy-group 1

set interfaces reth0 unit 0 family inet address 1.1.1.1/24

set interfaces reth1 redundant-ether-options redundancy-group 1

set interfaces reth1 unit 0 family inet address 2.2.2.1/24

set chassis cluster redundancy-group 1 interface-monitor xe-6/0/0 weight 255

set chassis cluster redundancy-group 1 interface-monitor xe-6/1/0 weight 255

set chassis cluster redundancy-group 1 interface-monitor xe-18/0/0 weight 255

set chassis cluster redundancy-group 1 interface-monitor xe-18/1/0 weight 255

set chassis cluster control-link-recovery
```

```
set security zones security-zone untrust interfaces reth0.0
```

```
set security zones security-zone trust interfaces reth1.0
```

```
set routing-options static route 0.0.0.0/0 next-hop 1.1.1.254
```

```
set routing-options static route 2.0.0.0/8 next-hop 2.2.2.254
```

To quickly configure an EX8208 Core Switch, copy the following commands and paste them into the CLI:

```
On {primary:node0}
```

```
[edit]
```

```
set interfaces xe-1/0/0 unit 0 family ethernet-switching port-mode access vlan members SRX5800
```

```
set interfaces xe-2/0/0 unit 0 family ethernet-switching port-mode access vlan members SRX5800
```

```
set interfaces vlan unit 50 family inet address 2.2.2.254/24
```

```
set vlans SRX5800 vlan-id 50
```

```
set vlans SRX5800 l3-interface vlan.50
```

```
set routing-options static route 0.0.0.0/0 next-hop 2.2.2.1/24
```

To quickly configure an MX240 edge router, copy the following commands and paste them into the CLI:

```
On {primary:node0}
```

```
[edit]
```

```
set interfaces xe-1/0/0 encapsulation ethernet-bridge unit 0 family bridge
```

```
set interfaces xe-2/0/0 encapsulation ethernet-bridge unit 0 family bridge
```

```
set interfaces irb unit 0 family inet address 1.1.1.254/24
```

```
set routing-options static route 2.0.0.0/8 next-hop 1.1.1.1
```

```
set routing-options static route 0.0.0.0/0 next-hop (upstream router)
```

```
set bridge-domains SRX5800 vlan-id X (could be set to "none")
```

```
set bridge-domains SRX5800 domain-type bridge routing-interface irb.0
```

```
set bridge-domains SRX5800 domain-type bridge interface xe-1/0/0
```

```
set bridge-domains SRX5800 domain-type bridge interface xe-2/0/0
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a chassis cluster on a high-end SRX Series device:



**NOTE:** Perform the following steps on the primary device (node 0). They are automatically copied over to the secondary device (node 1) when you execute a **commit** command.

1. Configure control ports for the clusters. Select FPC 1/13 because the control plane should always be on the lowest SPC/SPU in the cluster (for this example, it is slot 0). For maximum reliability, place the control ports on a separate SPC from the control plane (for this example, use SPC in slot 1).



**NOTE:** For the rest of this example, all commands are applied on the control plane regardless of which member is active.

```
user@host# set chassis cluster control-ports fpc 1 port 0
```

```
user@host# set chassis cluster control-ports fpc 13 port 0
```

2. Configure the fabric (data) ports of the cluster that are used to pass RTOs in active/passive mode. For this example, use one of the 1-Gigabit Ethernet ports because running out of bandwidth using active/passive mode is not an issue. Define two fabric interfaces, one on each chassis, to connect together.

```
user@host# set interfaces fab0 fabric-options member-interfaces ge-11/3/0
```

```
user@host# set interfaces fab1 fabric-options member-interfaces ge-23/3/0
```

3. Because the SRX5800 Services Gateway chassis cluster configuration is contained within a single common configuration, to assign some elements of the configuration to a specific member only, you must use the Junos OS node-specific configuration method called groups. The **set apply-groups \${node}** command uses the node variable to define how the groups are applied to the nodes; each node recognizes its number and accepts the configuration accordingly. You must also configure out-of-band management on the fxp0 interface of the SRX5800 Services Gateway using separate IP addresses for the individual control planes of the cluster.

```
user@host# set groups node0 system host-name SRX5800-1
```

```
user@host# set groups node0 interfaces fxp0 unit 0 family inet address 10.3.5.1/24
```

```
user@host# set groups node0 system backup-router 10.3.5.254 destination 0.0.0.0/0
```

```
user@host# set groups node1 system host-name SRX5800-2
```

```
user@host# set groups node1 interfaces fxp0 unit 0 family inet address 10.3.5.2/24
```

```
user@host# set groups node1 system backup-router 10.3.5.254 destination 0.0.0.0/0
```

```
user@host# set apply-groups "${node}"
```

4. Configure redundancy groups for chassis clustering. Each node has interfaces in a redundancy group where interfaces are active in active redundancy groups (multiple active interfaces can exist in one redundancy group). Redundancy group 0 controls the control plane and redundancy group 1+ controls the data plane and includes the data plane ports. For this active/passive mode example, only one chassis cluster member is active at a time so you need to define redundancy groups 0 and 1 only. Besides redundancy groups, you must also define:
  - Redundant Ethernet groups—Configure how many redundant Ethernet interfaces (member links) will be active on the device so that the system can allocate the appropriate resources for it.
  - Priority for control plane and data plane—Define which device has priority (for chassis cluster, high priority is preferred) for the control plane, and which device is preferred to be active for the data plane.



**NOTE:** In active/passive or active/active mode, the control plane (redundancy group 0) can be active on a chassis different from the data plane (redundancy group 1+ and groups) chassis. However, for this example we recommend having both the control and data plane active on the same chassis member. When traffic passes through the fabric link to go to another member node, latency is introduced (z line mode traffic).

```
user@host# set chassis cluster reth-count 2
```

```
user@host# set chassis cluster redundancy-group 0 node 0 priority 129
```

```
user@host# set chassis cluster redundancy-group 0 node 1 priority 128
```

```
user@host# set chassis cluster redundancy-group 1 node 0 priority 129
```

```
user@host# set chassis cluster redundancy-group 1 node 1 priority 128
```

5. Configure the data interfaces on the platform so that in the event of a data plane failover, the other chassis cluster member can take over the connection seamlessly. Seamless transition to a new active node will occur with data plane failover. In case of control plane failover, all the daemons are restarted on the new node thus enabling a graceful restart to avoid losing neighborship with peers (ospf, bgp). This promotes a seamless transition to the new node without any packet loss.

You must define the following items:

- Define the membership information of the member interfaces to the reth interface.
- Define which redundancy group the reth interface is a member of. For this active/passive example, it is always 1.
- Define reth interface information such as the IP address of the interface.

```
user@host# set interfaces xe-6/0/0 gigether-options redundant-parent reth0
```

```
user@host# set interfaces xe-6/1/0 gigether-options redundant-parent reth1
```

```
user@host# set interfaces xe-18/0/0 gigether-options redundant-parent reth0
```

```
user@host# set interfaces xe-18/1/0 gigether-options redundant-parent reth1
```

```
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
```

```
user@host# set interfaces reth0 unit 0 family inet address 1.1.1.1/24
```

```
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
```

```
user@host# set interfaces reth1 unit 0 family inet address 2.2.2.1/24
```

6. Configure the chassis cluster behavior in case of a failure. For the SRX5800 Services Gateway, the failover threshold is set at 255. You can alter the weights to determine the impact on the chassis failover. You must also configure control link recovery. The recovery automatically causes the secondary node to reboot should the control link fail, and then come back online. Enter these commands on node 0.

```
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-6/0/0  
weight 255
```

```
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-6/1/0  
weight 255
```

```
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-18/0/0  
weight 255
```

```
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-18/1/0  
weight 255
```

```
user@host# set chassis cluster control-link-recovery
```

This step completes the chassis cluster configuration part of the active/passive mode example for the SRX5800 Services Gateway. The rest of this procedure describes how to configure the zone, virtual router, routing, EX8208 Core Switch, and MX240 Edge Router to complete the deployment scenario.

7. Configure and connect the reth interfaces to the appropriate zones and virtual routers. For this example, leave the reth0 and reth1 interfaces in the default virtual router inet.0, which does not require any additional configuration.

```
user@host# set security zones security-zone untrust interfaces reth0.0
```

```
user@host# set security zones security-zone trust interfaces reth1.0
```

8. For this active/passive mode example, because of the simple network architecture, use static routes to define how to route to the other network devices.

```
user@host# set routing-options static route 0.0.0.0/0 next-hop 1.1.1.254
```

```
user@host# set routing-options static route 2.0.0.0/8 next-hop 2.2.2.254
```

9. For the EX8208 Ethernet Switch, the following commands provide only an outline of the applicable configuration as it pertains to this active/passive mode example for the SRX5800 Services Gateway; most notably the VLANs, routing, and interface configuration.

```
user@host# set interfaces xe-1/0/0 unit 0 family ethernet-switching port-mode
access vlan members SRX5800
```

```
user@host# set interfaces xe-2/0/0 unit 0 family ethernet-switching port-mode
access vlan members SRX5800
```

```
user@host# set interfaces vlan unit 50 family inet address 2.2.2.254/24
```

```
user@host# set vlans SRX5800 vlan-id 50
```

```
user@host# set vlans SRX5800 l3-interface vlan.50
```

```
user@host# set routing-options static route 0.0.0.0/0 next-hop 2.2.2.1/24
```

10. For the MX240 edge router, the following commands provide only an outline of the applicable configuration as it pertains to this active/passive mode example for the SRX5800 Services Gateway; most notably you must use an IRB interface within a virtual switch instance on the switch.

```
user@host# set interfaces xe-1/0/0 encapsulation ethernet-bridge unit 0 family
bridge
```

```
user@host# set interfaces xe-2/0/0 encapsulation ethernet-bridge unit 0 family
bridge
```

```
user@host# set interfaces irb unit 0 family inet address 1.1.1.254/24
```

```
user@host# set routing-options static route 2.0.0.0/8 next-hop 1.1.1.1
```

```
user@host# set routing-options static route 0.0.0.0/0 next-hop (upstream router)
```

```
user@host# set bridge-domains SRX5800 vlan-id X (could be set to "none")
```

```
user@host# set bridge-domains SRX5800 domain-type bridge routing-interface
  irb.0
```

```
user@host# set bridge-domains SRX5800 domain-type bridge interface xe-1/0/0
```

```
user@host# set bridge-domains SRX5800 domain-type bridge interface xe-2/0/0
```

**Results** From operational mode, confirm your configuration by entering the **show configuration** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
> show configuration
version x.xx.x;
groups {
  node0 {
    system {
      host-name SRX58001;
      backup-router 10.3.5.254 destination 0.0.0.0/0;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 10.3.5.1/24;
          }
        }
      }
    }
  }
  node1 {
    system {
      host-name SRX58002;
      backup-router 10.3.5.254 destination 0.0.0.0/0;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 10.3.5.2/24;
          }
        }
      }
    }
  }
}
apply-groups "${node}";
system {
  root-authentication {
    encrypted-password "$1$zTMjraKG$qU8rjxoHzC6Y/WDmYpR9r.";
  }
  name-server {
    4.2.2.2;
  }
  services {
    ssh {
      root-login allow;
    }
  }
}
```



```

        netconf {
            ssh;
        }
        web-management {
            http {
                interface fxp0.0;
            }
        }
    }
}
chassis {
    cluster {
        control-link-recovery;
        reth-count 2;
        control-ports {
            fpc 1 port 0;
            fpc 13 port 0;
        }
        redundancy-group 0 {
            node 0 priority 129;
            node 1 priority 128;
        }
        redundancy-group 1 {
            node 0 priority 129;
            node 1 priority 128;
            interface-monitor {
                xe-6/0/0 weight 255;
                xe-6/1/0 weight 255;
                xe-18/0/0 weight 255;
                xe-18/1/0 weight 255;
            }
        }
    }
}
}
interfaces {
    xe-6/0/0 {
        gigeother-options {
            redundant-parent reth0;
        }
    }
    xe-6/1/0 {
        gigeother-options {
            redundant-parent reth1;
        }
    }
    xe-18/0/0 {
        gigeother-options {
            redundant-parent reth0;
        }
    }
    xe-18/1/0 {
        gigeother-options {
            redundant-parent reth1;
        }
    }
}
fab0 {
    fabric-options {
        member-interfaces {
            ge-11/3/0;
        }
    }
}
}

```

```
}
fab1 {
  fabric-options {
    member-interfaces {
      ge-23/3/0;
    }
  }
}
reth0 {
  redundant-ether-options {
    redundancy-group 1;
  }
  unit 0 {
    family inet {
      address 1.1.1.1/24;
    }
  }
}
reth1 {
  redundant-ether-options {
    redundancy-group 1;
  }
  unit 0 {
    family inet {
      address 2.2.2.1/24;
    }
  }
}
}
routing-options {
  static {
    route 0.0.0.0/0 {
      next-hop 1.1.1.254;
    }
    route 2.0.0.0/8 {
      next-hop 2.2.2.254;
    }
  }
}
security {
  zones {
    security-zone trust {
      host-inbound-traffic {
        system-services {
          all;
        }
      }
      interfaces {
        reth0.0;
      }
    }
    security-zone untrust {
      interfaces {
        reth1.0;
      }
    }
  }
  policies {
    from-zone trust to-zone untrust {
      policy 1 {
        match {
```

```

        source-address any;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
}
default-policy {
    deny-all;
}
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Chassis Cluster Status on page 1309
- Verifying Chassis Cluster Interfaces on page 1309
- Verifying Chassis Cluster Statistics on page 1310
- Verifying Chassis Cluster Control Plane Statistics on page 1311
- Verifying Chassis Cluster Data Plane Statistics on page 1311
- Verifying Chassis Cluster Redundancy Group Status on page 1311
- Troubleshooting with Logs on page 1312

#### Verifying Chassis Cluster Status

**Purpose** Verify the chassis cluster status, failover status, and redundancy group information.

**Action** From operational mode, enter the **show chassis cluster status** command.

```

{primary:node0}
show chassis cluster status
Cluster ID: 1
Node                Priority    Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
node0                129       primary   no       no
node1                128       secondary no       no

Redundancy group: 1 , Failover count: 1
node0                129       primary   no       no
node1                128       secondary no       no

```

#### Verifying Chassis Cluster Interfaces

**Purpose** Verify information about chassis cluster interfaces.

**Action** From operational mode, enter the **show chassis cluster interfaces** command.

```

{primary:node0}
user@host> show chassis cluster interfaces

```

Control link name: fxp1

Redundant-ethernet Information:

Name	Status	Redundancy-group
reth0	Up	1
reth1	Up	1

Interface Monitoring:

Interface	Weight	Status	Redundancy-group
xe-6/0/0	255	Up	1
xe-6/1/0	255	Up	1
xe-18/0/0	255	Up	1
xe-18/1/0	255	Up	1

**Verifying Chassis Cluster Statistics**

**Purpose** Verify information about chassis cluster services and control link statistics (heartbeats sent and received), fabric link statistics (probes sent and received), and the number of RTOs sent and received for services.

**Action** From operational mode, enter the **show chassis cluster statistics** command.

```
{primary:node0}
```

```
user@host> show chassis cluster statistics
```

Control link statistics:

Control link 0:

```
Heartbeat packets sent: 258689
Heartbeat packets received: 258684
Heartbeat packets errors: 0
```

Fabric link statistics:

```
Probes sent: 258681
Probes received: 258681
Probe errors: 0
```

Services Synchronized:

Service name	RTOs sent	RTOs received
Translation context	0	0
Incoming NAT	0	0
Resource manager	6	0
Session create	161	0
Session close	148	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0
Session ageout refresh replies	0	0
IPSec VPN	0	0
Firewall user authentication	0	0
MGCP ALG	0	0
H323 ALG	0	0
SIP ALG	0	0
SCCP ALG	0	0
PPTP ALG	0	0
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

**Verifying Chassis Cluster Control Plane Statistics**

**Purpose** Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).

**Action** From operational mode, enter the **show chassis cluster control-plane statistics** command.

```
{primary:node0}
user@host> show chassis cluster control-plane statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
    Heartbeat packets errors: 0
Fabric link statistics:
  Probes sent: 258681
  Probes received: 258681
  Probe errors: 0
```

**Verifying Chassis Cluster Data Plane Statistics**

**Purpose** Verify information about the number of RTOs sent and received for services.

**Action** From operational mode, enter the **show chassis cluster data-plane statistics** command.

```
{primary:node0}
user@host> show chassis cluster data-plane statistics

Services Synchronized:
  Service name                RTOs sent  RTOs received
  Translation context         0           0
  Incoming NAT                 0           0
  Resource manager            6           0
  Session create              161         0
  Session close               148         0
  Session change              0           0
  Gate create                  0           0
  Session ageout refresh requests 0           0
  Session ageout refresh replies 0           0
  IPSec VPN                   0           0
  Firewall user authentication 0           0
  MGCP ALG                    0           0
  H323 ALG                    0           0
  SIP ALG                     0           0
  SCCP ALG                    0           0
  PPTP ALG                    0           0
  RPC ALG                     0           0
  RTSP ALG                    0           0
  RAS ALG                     0           0
  MAC address learning        0           0
  GPRS GTP                    0           0
```

**Verifying Chassis Cluster Redundancy Group Status**

**Purpose** Verify the state and priority of both nodes in a cluster and information about whether the primary node has been preempted or whether there has been a manual failover.

**Action** From operational mode, enter the `chassis cluster status redundancy-group` command.

```
{primary:node0}
user@host> show chassis cluster status redundancy-group 1
Cluster ID: 1
  Node           Priority   Status   Preempt  Manual failover

Redundancy-Group: 1, Failover count: 1
  node0          100      primary  no       no
  node1          50       secondary no       no
```

#### *Troubleshooting with Logs*

**Purpose** Use these logs to identify any chassis cluster issues. You should run these logs on both nodes.

**Action** From operational mode, enter these `show log` commands.

```
user@host> show log jsrpd
user@host> show log chassisd
user@host> show log messages
user@host> show log dcd
user@host> show traceoptions
```

#### **Related Documentation**

- Understanding Chassis Cluster Redundancy Groups on page 1139.
- Node Interfaces on Active SRX Series Chassis Clusters on page 1211
- Example: Configuring an SRX Series Services Gateway for the Branch as a Chassis Cluster on page 1283

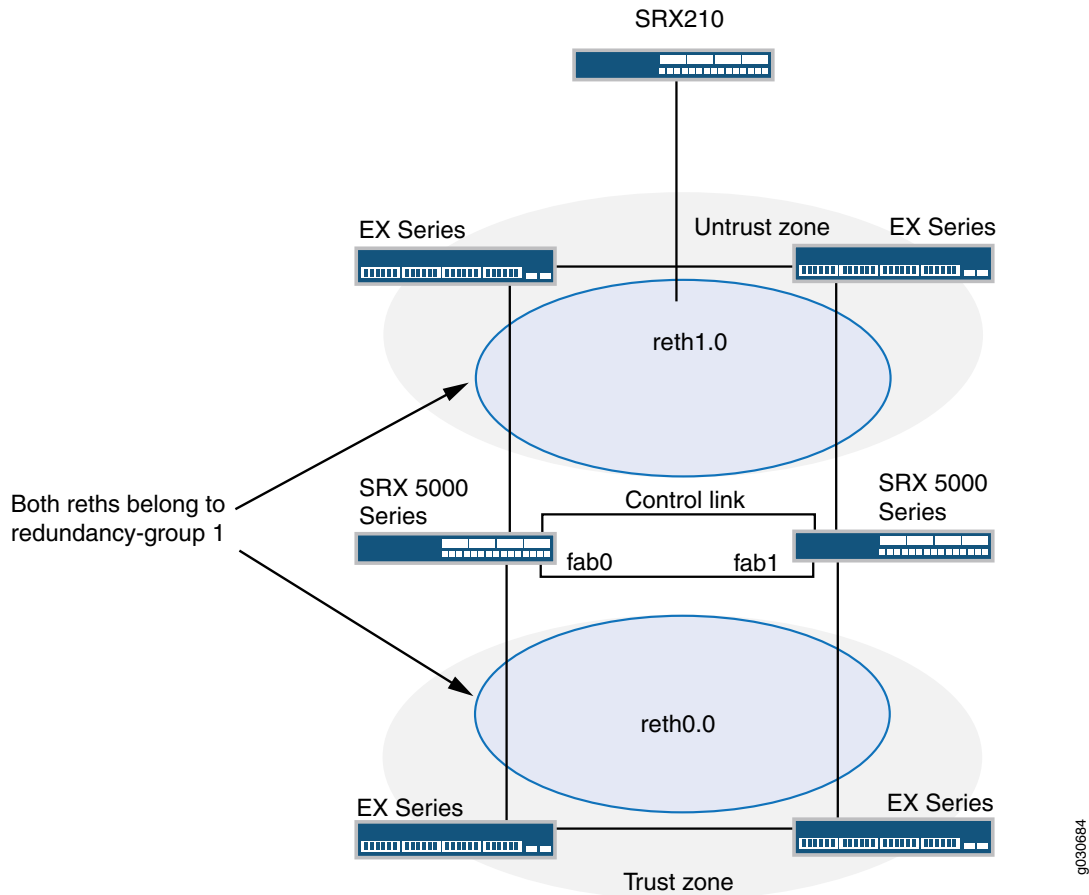
## Active/Passive Chassis Cluster Deployment with an IPsec Tunnel

- Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel on page 1312
- Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel on page 1314
- Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (J-Web) on page 1329

### Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel

In this case, a single device in the cluster terminates in an IPsec tunnel and is used to process all traffic while the other device is used only in the event of a failure (see Figure 122 on page 1313). When a failure occurs, the backup device becomes master and controls all forwarding.

Figure 122: Active/Passive Chassis Cluster with IPsec Tunnel Scenario (SRX Series Devices)



An active/passive chassis cluster can be achieved by using redundant Ethernet interfaces (reths) that are all assigned to the same redundancy group. If any of the interfaces in an active group in a node fails, the group is declared inactive and all the interfaces in the group fail over to the other node.

This configuration provides a way for a site-to-site IPsec tunnel to terminate in an active/passive cluster where a redundant Ethernet interface is used as the tunnel endpoint. In the event of a failure, the redundant Ethernet interface in the backup SRX Series device becomes active, forcing the tunnel to change endpoints to terminate in the new active SRX Series device. Because tunnel keys and session information are synchronized between the members of the chassis cluster, a failover does not require the tunnel to be renegotiated and all established sessions are maintained.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel on page 1314
- Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (J-Web) on page 1329

- Understanding What Happens When Chassis Cluster Is Enabled on page 1210
- Understanding Chassis Cluster Formation on page 1138

## Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel

This example shows how to configure active/passive chassis clustering with an IPsec tunnel for SRX Series devices.

- Requirements on page 1314
- Overview on page 1315
- Configuration on page 1319
- Verification on page 1326

### Requirements

---

Before you begin:

- Get two SRX5000 models with identical hardware configurations, one SRX210 edge router, and four EX Series Ethernet switches.
- Physically connect the two devices (back-to-back for the fabric and control ports) and ensure that they are the same models. You can configure both the fabric and control ports on the SRX5000 line.
- Set the two devices to cluster mode and reboot the devices. You must enter the following operational mode commands on both devices, for example:

- On node 0:

```
user@host> set chassis cluster cluster-id 1 node 0 reboot
```

- On node 1:

```
user@host> set chassis cluster cluster-id 1 node 1 reboot
```

The cluster ID is the same on both devices, but the node ID must be different because one device is node 0 and the other device is node 1. The range for the cluster ID is 0 through 15, and setting it to 0 is equivalent to disabling cluster mode.

If you have multiple SRX Series clusters on a single L3 broadcast domain, then you must assign different cluster IDs to each cluster, or else there will be a MAC address conflict.

- Get two SRX5000 models with identical hardware configurations, one SRX210 edge router, and four EX Series Ethernet switches.
- Physically connect the two devices (back-to-back for the fabric and control ports) and ensure that they are the same models. You can configure both the fabric and control ports on the SRX5000 line.
- Set the two devices to cluster mode and reboot the devices. You must enter the following operational mode commands on both devices, for example:
  - On node 0:



```
user@host> set chassis cluster cluster-id 1 node 0 reboot
```

- On node 1:

```
user@host> set chassis cluster cluster-id 1 node 1 reboot
```

The cluster ID is the same on both devices, but the node ID must be different because one device is node 0 and the other device is node 1. The range for the cluster ID is 0 through 15, and setting it to 0 is equivalent to disabling cluster mode.

If you have multiple SRX Series clusters on a single L3 broadcast domain, then you must assign different cluster IDs to each cluster, or else there will be a MAC address conflict.

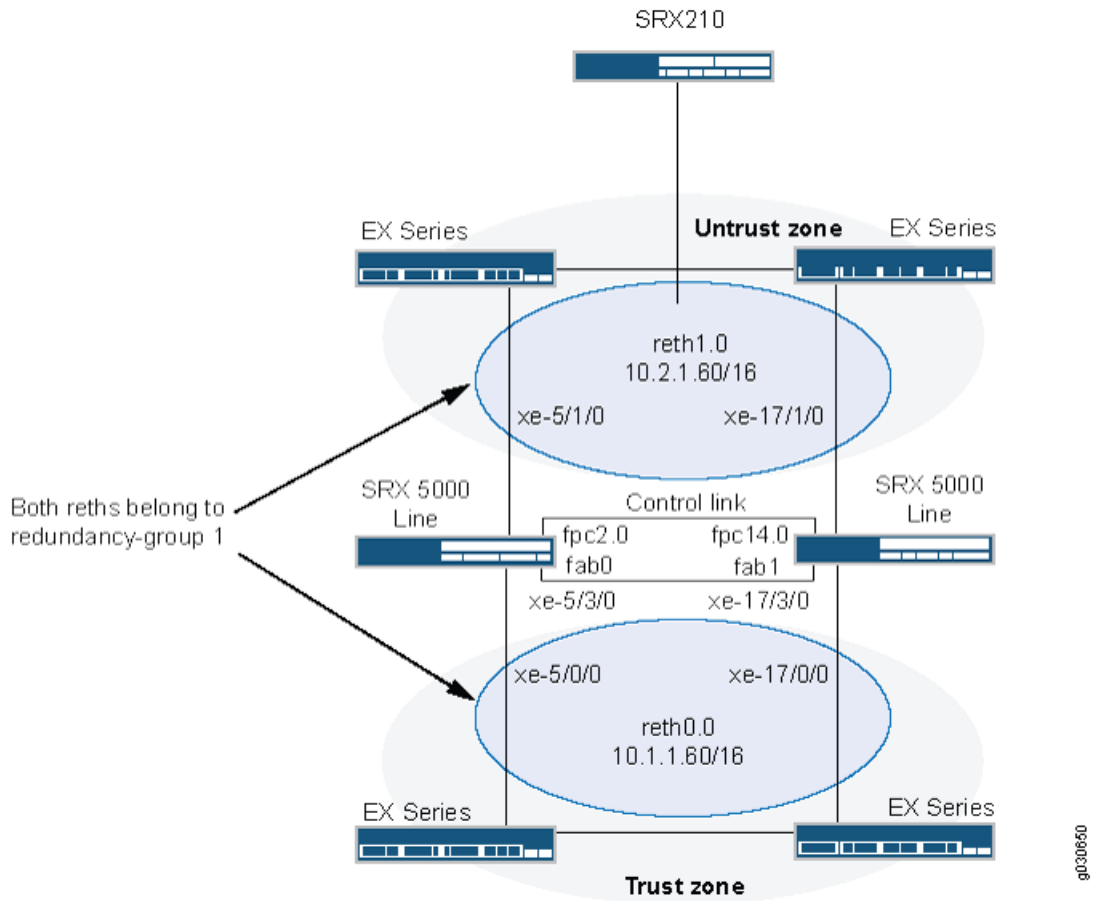
From this point forward, configuration of the cluster is synchronized between the node members and the two separate devices function as one device. Member-specific configurations (such as the IP address of the management port of each member) are entered using configuration groups.

### Overview

---

In this example, a single device in the cluster terminates in an IPsec tunnel and is used to process all traffic, and the other device is used only in the event of a failure. (See Figure 123 on page 1316.) When a failure occurs, the backup device becomes master and controls all forwarding.

Figure 123: Active/Passive Chassis Cluster with IPsec Tunnel Topology (SRX Series Devices)



In this example, you configure group (applying the configuration with the apply-groups command) and chassis cluster information. Then you configure IKE, IPsec, static route, security zone, and security policy parameters. See Table 127 on page 1316 through Table 133 on page 1319.

Table 127: Group and Chassis Cluster Configuration Parameters

Feature	Name	Configuration Parameters
Groups	node0	<ul style="list-style-type: none"> <li>• Hostname: SRX5800-1</li> <li>• Interface: fxp0                             <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 172.19.100.50/24</li> </ul> </li> </ul>
	node1	<ul style="list-style-type: none"> <li>• Hostname: SRX5800-2</li> <li>• Interface: fxp0                             <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 172.19.100.51/24</li> </ul> </li> </ul>

Table 128: Chassis Cluster Configuration Parameters

Feature	Name	Configuration Parameters
Fabric links	fab0	Interface: xe-5/3/0
	fab1	Interface: xe-17/3/0
Number of redundant Ethernet interfaces	–	2
Heartbeat interval	–	1000
Heartbeat threshold	–	3
Redundancy group	0	<ul style="list-style-type: none"> <li>• Priority: <ul style="list-style-type: none"> <li>• Node 0: 254</li> <li>• Node 1: 1</li> </ul> </li> </ul>
	1	<ul style="list-style-type: none"> <li>• Priority: <ul style="list-style-type: none"> <li>• Node 0: 254</li> <li>• Node 1: 1</li> </ul> </li> <li>•</li> </ul>
		Interface monitoring <ul style="list-style-type: none"> <li>• xe-5/0/0</li> <li>• xe-5/1/0</li> <li>• xe-17/0/0</li> <li>• xe-17/1/0</li> </ul>
Interfaces	xe-5/1/0	Redundant parent: reth1
	xe-5/1/0	Redundant parent: reth1
	xe-5/0/0	Redundant parent: reth0
	xe-17/0/0	Redundant parent: reth0
	reth0	Redundancy group: 1
		<ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 10.1.1.60/16</li> </ul>
	reth1	Redundancy group: 1
		<ul style="list-style-type: none"> <li>• Multipoint</li> <li>• Unit 0</li> <li>• 10.10.1.1/30</li> </ul>
	st0	

Table 128: Chassis Cluster Configuration Parameters (*continued*)

Feature	Name	Configuration Parameters
		<ul style="list-style-type: none"> <li>Unit 0</li> <li>10.10.1.1/30</li> </ul>

Table 129: IKE Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	proposal-set standard	-
Policy	preShared	<ul style="list-style-type: none"> <li>Mode: main</li> <li>Proposal reference: proposal-set standard</li> <li>IKE Phase 1 policy authentication method: pre-shared-key ascii-text</li> </ul>
Gateway	SRX210-1	<ul style="list-style-type: none"> <li>IKE policy reference: perShared</li> <li>External interface: reth0.0</li> <li>Gateway address: 10.1.1.90</li> </ul>

Table 130: IPsec Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	proposal-set standard	-
Policy	std	-
VPN	SRX210-1	<ul style="list-style-type: none"> <li>IKE gateway reference: SRX210-1</li> <li>IPsec policy reference: std</li> <li>Bind to interface: st0.0</li> <li>VPN monitoring: vpn-monitor optimized</li> <li>Tunnels established: establish-tunnels immediately</li> </ul>

Table 131: Static Route Configuration Parameters

Name	Configuration Parameters
0.0.0.0/0	Next hop: 10.2.1.1
10.3.0.0/16	Next hop: 10.10.1.2

Table 132: Security Zone Configuration Parameters

Name	Configuration Parameters
trust	<ul style="list-style-type: none"> <li>All system services are allowed.</li> <li>All protocols are allowed.</li> <li>The reth0.0 interface is bound to this zone.</li> </ul>

Table 132: Security Zone Configuration Parameters (*continued*)

Name	Configuration Parameters
untrust	<ul style="list-style-type: none"> <li>All system services are allowed.</li> <li>All protocols are allowed.</li> <li>The reth1.0 interface is bound to this zone.</li> </ul>
vpn	<ul style="list-style-type: none"> <li>All system services are allowed.</li> <li>All protocols are allowed.</li> <li>The st0.0 interface is bound to this zone.</li> </ul>

Table 133: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters
This security policy permits traffic from the trust zone to the untrust zone.	ANY	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address any</li> <li>destination-address any</li> <li>application any</li> </ul> </li> <li>Action: permit</li> </ul>
This security policy permits traffic from the trust zone to the vpn zone.	vpn-any	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address any</li> <li>destination-address any</li> <li>application any</li> </ul> </li> <li>Action: permit</li> </ul>

### Configuration

**CLI Quick Configuration** To quickly configure an active/passive chassis cluster pair with an IPsec tunnel, copy the following commands and paste them into the CLI:

```
{primary:node0}[edit]
set chassis cluster control-ports fpc 2 port 0
set chassis cluster control-ports fpc 14 port 0
set groups node0 system host-name SRX5800-1
set groups node0 interfaces fxp0 unit 0 family inet address 172.19.100.50/24
set groups node1 system host-name SRX5800-2
set groups node1 interfaces fxp0 unit 0 family inet address 172.19.100.51/24
set apply-groups "${node}"
set interfaces fab0 fabric-options member-interfaces xe-5/3/0
set interfaces fab1 fabric-options member-interfaces xe-17/3/0
set chassis cluster reth-count 2
set chassis cluster heartbeat-interval 1000
set chassis cluster heartbeat-threshold 3
set chassis cluster node 0
set chassis cluster node 1
set chassis cluster redundancy-group 0 node 0 priority 254
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 254
set chassis cluster redundancy-group 1 node 1 priority 1
```

```

set chassis cluster redundancy-group 1 preempt
set chassis cluster redundancy-group 1 interface-monitor xe-5/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-5/1/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-17/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-17/1/0 weight 255
set interfaces xe-5/1/0 gigether-options redundant-parent reth1
set interfaces xe-17/1/0 gigether-options redundant-parent reth1
set interfaces xe-5/0/0 gigether-options redundant-parent reth0
set interfaces xe-17/0/0 gigether-options redundant-parent reth0
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 10.1.1.60/16
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 10.2.1.60/16
set interfaces st0 unit 0 multipoint family inet address 10.10.1.1/30
set security ike policy preShared mode main
set security ike policy preShared proposal-set standard
set security ike policy preShared pre-shared-key ascii-text "juniper"## Encrypted password
set security ike gateway SRX210-1 ike-policy preShared
set security ike gateway SRX210-1 address 10.1.1.90
set security ike gateway SRX210-1 external-interface reth0.0
set security ipsec policy std proposal-set standard
set security ipsec vpn SRX210-1 bind-interface st0.0
set security ipsec vpn SRX210-1 vpn-monitor optimized
set security ipsec vpn SRX210-1 ike gateway SRX210-1
set security ipsec vpn SRX210-1 ike ipsec-policy std
set security ipsec vpn SRX210-1 establish-tunnels immediately
set routing-options static route 0.0.0.0/0 next-hop 10.2.1.1
set routing-options static route 10.3.0.0/16 next-hop 10.10.1.2
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces reth1.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces reth0.0
set security zones security-zone vpn host-inbound-traffic system-services all 144
set security zones security-zone vpn host-inbound-traffic protocols all
set security zones security-zone vpn interfaces st0.0
set security policies from-zone trust to-zone untrust policy ANY match source-address
any
set security policies from-zone trust to-zone untrust policy ANY match destination-address
any
set security policies from-zone trust to-zone untrust policy ANY match application any
set security policies from-zone trust to-zone vpn policy vpn-any then permit

```

**Step-by-Step Procedure** To configure an active/passive chassis cluster pair with an IPsec tunnel:

1. Configure control ports.

```

{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 2 port 0
user@host# set chassis cluster control-ports fpc 14 port 0

```

2. Configure the management interface.

```

{primary:node0}[edit]
user@host# set groups node0 system host-name SRX5800-1

```

```

user@host# set groups node0 interfaces fxp0 unit 0 family inet address
172.19.100.50/24
user@host# set groups node1 system host-name SRX5800-2
user@host# set groups node1 interfaces fxp0 unit 0 family inet address
172.19.100.51/24
user@host# set apply-groups "${node}"

```

3. Configure the fabric interface.

```

{primary:node0}[edit]
user@host# set interfaces fab0 fabric-options member-interfaces xe-5/3/0
user@host# set interfaces fab1 fabric-options member-interfaces xe-17/3/0

```

4. Configure redundancy groups.

```

{primary:node0}[edit]
user@host# set chassis cluster reth-count 2
user@host# set chassis cluster heartbeat-interval 1000
user@host# set chassis cluster heartbeat-threshold 3
user@host# set chassis cluster node 0
user@host# set chassis cluster node 1
user@host# set chassis cluster redundancy-group 0 node 0 priority 254
user@host# set chassis cluster redundancy-group 0 node 1 priority 1
user@host# set chassis cluster redundancy-group 1 node 0 priority 254
user@host# set chassis cluster redundancy-group 1 node 1 priority 1
user@host# set chassis cluster redundancy-group 1 preempt
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-5/0/0
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-5/1/0
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-17/0/0
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-17/1/0
weight 255

```

5. Configure redundant Ethernet interfaces.

```

{primary:node0}[edit]
user@host# set interfaces xe-5/1/0 gigether-options redundant-parent reth1
user@host# set interfaces xe-17/1/0 gigether-options redundant-parent reth1
user@host# set interfaces xe-5/0/0 gigether-options redundant-parent reth0
user@host# set interfaces xe-17/0/0 gigether-options redundant-parent reth0
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
user@host# set interfaces reth0 unit 0 family inet address 10.1.1.60/16
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
user@host# set interfaces reth1 unit 0 family inet address 10.2.1.60/16

```

6. Configure IPsec parameters.

```

{primary:node0}[edit]
user@host# set interfaces st0 unit 0 multipoint family inet address 10.10.1.1/30
user@host# set security ike policy preShared mode main
user@host# set security ike policy preShared proposal-set standard
user@host# set security ike policy preShared pre-shared-key ascii-text "juniper"##
Encrypted password
user@host# set security ike gateway SRX210-1 ike-policy preShared
user@host# set security ike gateway SRX210-1 address 10.1.1.90
user@host# set security ike gateway SRX210-1 external-interface reth0.0
user@host# set security ipsec policy std proposal-set standard

```

```

user@host# set security ipsec vpn SRX210-1 bind-interface st0.0
user@host# set security ipsec vpn SRX210-1 vpn-monitor optimized
user@host# set security ipsec vpn SRX210-1 ike gateway SRX210-1
user@host# set security ipsec vpn SRX210-1 ike ipsec-policy std
user@host# set security ipsec vpn SRX210-1 establish-tunnels immediately

```

- Configure static routes.

```

{primary:node0}[edit]
user@host# set routing-options static route 0.0.0.0/0 next-hop 10.2.1.1
user@host# set routing-options static route 10.3.0.0/16 next-hop 10.10.1.2

```

- Configure security zones.

```

{primary:node0}[edit]
user@host# set security zones security-zone untrust host-inbound-traffic
system-services all
user@host# set security zones security-zone untrust host-inbound-traffic protocols
all
user@host# set security zones security-zone untrust interfaces reth1.0
user@host# set security zones security-zone trust host-inbound-traffic
system-services all
user@host# set security zones security-zone trust host-inbound-traffic protocols
all
user@host# set security zones security-zone trust interfaces reth0.0
user@host# set security zones security-zone vpn host-inbound-traffic
system-services all
user@host# set security zones security-zone vpn host-inbound-traffic protocols all
user@host# set security zones security-zone vpn interfaces st0.0

```

- Configure security policies.

```

{primary:node0}[edit]
user@host# set security policies from-zone trust to-zone untrust policy ANY match
source-address any
user@host# set security policies from-zone trust to-zone untrust policy ANY match
destination-address any
user@host# set security policies from-zone trust to-zone untrust policy ANY match
application any
user@host# set security policies from-zone trust to-zone vpn policy vpn-any then
permit

```

**Results** From operational mode, confirm your configuration by entering the **show configuration** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```

user@host> show configuration
version x.xx.x;
groups {
  node0 {
    system {
      host-name SRX58001;
    }
    interfaces {

```



```

        fxp0 {
            unit 0 {
                family inet {
                    address 172.19.100.50/24;
                }
            }
        }
    }
}
node1 {
    system {
        host-name SRX58002;
    }
    interfaces {
        fxp0 {
            unit 0 {
                family inet {
                    address 172.19.100.51/24;
                }
            }
        }
    }
}
}
apply-groups "${node}";
system {
    root-authentication {
        encrypted-password "$1$zTMjraKG$qU8rjxoHzC6Y/WDmYpR9r.";
    }
}
chassis {
    cluster {
        reth-count 2;
        heartbeat-interval 1000;
        heartbeat-threshold 3;
        control-ports {
            fpc 2 port 0;
            fpc 14 port 0;
        }
        redundancy-group 0 {
            node 0 priority 254;
            node 1 priority 1;
        }
        redundancy-group 1 {
            node 0 priority 254;
            node 1 priority 1;
            preempt;
            interface-monitor {
                xe-6/0/0 weight 255;
                xe-6/1/0 weight 255;
                xe-18/0/0 weight 255;
                xe-18/1/0 weight 255;
            }
        }
    }
}
interfaces {
    xe-5/0/0 {
        together-options {
            redundant-parent reth0;
        }
    }
}

```

```
    }
  xe-5/1/0 {
    gigger-options {
      redundant-parent reth1;
    }
  }
  xe-17/0/0 {
    gigger-options {
      redundant-parent reth0;
    }
  }
  xe-17/1/0 {
    gigger-options {
      redundant-parent reth1;
    }
  }
}
fab0 {
  fabric-options {
    member-interfaces {
      xe-5/3/0;
    }
  }
}
fab1 {
  fabric-options {
    member-interfaces {
      xe-17/3/0;
    }
  }
}
reth0 {
  redundant-ether-options {
    redundancy-group 1;
  }
  unit 0 {
    family inet {
      address 10.1.1.60/16;
    }
  }
}
reth1 {
  redundant-ether-options {
    redundancy-group 1;
  }
  unit 0 {
    family inet {
      address 10.2.1.60/16;
    }
  }
}
st0 {
  unit 0 {
    multipoint;
    family inet {
      address 5.4.3.2/32;
    }
  }
}
}
routing-options {
  static {
```

```
        route 0.0.0.0/0 {
            next-hop 10.2.1.1;
        }
        route 10.3.0.0/16 {
            next-hop 10.10.1.2;
        }
    }
}
security {
    zones {
        security-zone trust {
            host-inbound-traffic {
                system-services {
                    all;
                }
            }
            interfaces {
                reth0.0;
            }
        }
        security-zone untrust
            host-inbound-traffic {
                system-services {
                    all;
                }
            }
            protocols {
                all;
            }
            interfaces {
                reth1.0;
            }
        }
    }
    security-zone vpn {
        host-inbound-traffic {
            system-services {
                all;
            }
        }
        protocols {
            all;
        }
        interfaces {
            st0.0;
        }
    }
}
policies {
    from-zone trust to-zone untrust {
        policy ANY {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit;
            }
        }
    }
}
```

```

    }
    from-zone trust to-zone vpn {
      policy vpn {
        match {
          source-address any;
          destination-address any;
          application any;
        }
        then {
          permit;
        }
      }
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Chassis Cluster Status on page 1326
- Verifying Chassis Cluster Interfaces on page 1326
- Verifying Chassis Cluster Statistics on page 1327
- Verifying Chassis Cluster Control Plane Statistics on page 1328
- Verifying Chassis Cluster Data Plane Statistics on page 1328
- Verifying Chassis Cluster Redundancy Group Status on page 1328
- Troubleshooting with Logs on page 1329

#### Verifying Chassis Cluster Status

**Purpose** Verify the chassis cluster status, failover status, and redundancy group information.

**Action** From operational mode, enter the **show chassis cluster status** command.

```

{primary:node0}
show chassis cluster status
Cluster ID: 1
Node                Priority    Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
  node0              0          primary  no       no
  node1              254       secondary no       no

Redundancy group: 1 , Failover count: 1
  node0              0          primary  yes      no
  node1              254       secondary yes      no

```

#### Verifying Chassis Cluster Interfaces

**Purpose** Verify the chassis cluster interfaces.

**Action** From operational mode, enter the **show chassis cluster interfaces** command.

```

{primary:node0}

```

```
user@host> show chassis cluster interfaces
```

```
Control link name: fxp1
```

```
Redundant-ethernet Information:
```

Name	Status	Redundancy-group
reth0	Up	1
reth1	Up	1

```
Interface Monitoring:
```

Interface	Weight	Status	Redundancy-group
xe-5/0/0	255	Up	1
xe-5/1/0	255	Up	1
xe-17/0/0	255	Up	1
xe-17/1/0	255	Up	1

### Verifying Chassis Cluster Statistics

**Purpose** Verify information about chassis cluster services and control link statistics (heartbeats sent and received), fabric link statistics (probes sent and received), and the number of RTOs sent and received for services.

**Action** From operational mode, enter the **show chassis cluster statistics** command.

```
{primary:node0}
```

```
user@host> show chassis cluster statistics
```

```
Control link statistics:
```

```
Control link 0:
```

```
Heartbeat packets sent: 258689
```

```
Heartbeat packets received: 258684
```

```
Heartbeat packets errors: 0
```

```
Fabric link statistics:
```

```
Probes sent: 258681
```

```
Probes received: 258681
```

```
Probe errors: 0
```

```
Services Synchronized:
```

Service name	RTOs sent	RTOs received
Translation context	0	0
Incoming NAT	0	0
Resource manager	6	0
Session create	161	0
Session close	148	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0
Session ageout refresh replies	0	0
IPSec VPN	0	0
Firewall user authentication	0	0
MGCP ALG	0	0
H323 ALG	0	0
SIP ALG	0	0
SCCP ALG	0	0
PPTP ALG	0	0
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

**Verifying Chassis Cluster Control Plane Statistics**

**Purpose** Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).

**Action** From operational mode, enter the **show chassis cluster control-plane statistics** command.

```
{primary:node0}
user@host> show chassis cluster control-plane statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
    Heartbeat packets errors: 0
Fabric link statistics:
  Probes sent: 258681
  Probes received: 258681
  Probe errors: 0
```

**Verifying Chassis Cluster Data Plane Statistics**

**Purpose** Verify information about the number of RTOs sent and received for services.

**Action** From operational mode, enter the **show chassis cluster data-plane statistics** command.

```
{primary:node0}
user@host> show chassis cluster data-plane statistics

Services Synchronized:
  Service name                RTOs sent  RTOs received
  Translation context         0           0
  Incoming NAT                 0           0
  Resource manager             6           0
  Session create               161         0
  Session close                 148         0
  Session change               0           0
  Gate create                   0           0
  Session ageout refresh requests 0           0
  Session ageout refresh replies 0           0
  IPSec VPN                    0           0
  Firewall user authentication 0           0
  MGCP ALG                     0           0
  H323 ALG                      0           0
  SIP ALG                       0           0
  SCCP ALG                      0           0
  PPTP ALG                      0           0
  RPC ALG                       0           0
  RTSP ALG                      0           0
  RAS ALG                       0           0
  MAC address learning         0           0
  GPRS GTP                     0           0
```

**Verifying Chassis Cluster Redundancy Group Status**

**Purpose** Verify the state and priority of both nodes in a cluster and information about whether the primary node has been preempted or whether there has been a manual failover.

**Action** From operational mode, enter the `chassis cluster status redundancy-group` command.

```
{primary:node0}
user@host> show chassis cluster status redundancy-group 1
Cluster ID: 1
  Node           Priority  Status  Preempt  Manual failover

Redundancy-Group: 1, Failover count: 1
  node0          0        primary yes      no
  node1          254     secondary yes      no
```

### *Troubleshooting with Logs*

**Purpose** Use these logs to identify any chassis cluster issues. You should run these logs on both nodes.

**Action** From operational mode, enter these `show` commands.

```
user@host> show log jsrpd
user@host> show log chassisd
user@host> show log messages
user@host> show log dcd
user@host> show traceoptions
```

### **Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel on page 1312
- Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (J-Web) on page 1329
- Understanding What Happens When Chassis Cluster Is Enabled on page 1210
- Understanding Chassis Cluster Formation on page 1138

## **Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (J-Web)**

1. Enable clusters. See Step 1 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel” on page 1314.
2. Configure the management interface. See Step 2 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel” on page 1314.
3. Configure the fabric interface. See Step 3 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel” on page 1314.
4. Configure the redundancy groups.
  - Select **Configure>System Properties>Chassis Cluster**.
  - Enter the following information, and then click **Apply**:
    - Redundant ether-Interfaces Count: **2**
    - Heartbeat Interval: **1000**
    - Heartbeat Threshold: **3**

Nodes: **0**

Group Number: **0**

Priorities: **254**

- Enter the following information, and then click **Apply**:

Nodes: **0**

Group Number: **1**

Priorities: **254**

- Enter the following information, and then click **Apply**:

Nodes: **1**

Group Number: **0**

Priorities: **1**

- Enter the following information, and then click **Apply**:

Nodes: **1**

Group Number: **1**

Priorities: **1**

Preempt: Select the check box.

Interface Monitor—Interface: **xe-5/0/0**

Interface Monitor—Weight: **255**

Interface Monitor—Interface: **xe-5/1/0**

Interface Monitor—Weight: **255**

Interface Monitor—Interface: **xe-17/0/0**

Interface Monitor—Weight: **255**

Interface Monitor—Interface: **xe-17/1/0**

Interface Monitor—Weight: **255**

5. Configure the redundant Ethernet interfaces.

- Select **Configure>System Properties>Chassis Cluster**.
- Select **xe-5/1/0**.
- Enter **reth1** in the Redundant Parent box.
- Click **Apply**.



- Select **xe-17/1/0**.
  - Enter **reth1** in the Redundant Parent box.
  - Click **Apply**.
  - Select **xe-5/0/0**.
  - Enter **reth0** in the Redundant Parent box.
  - Click **Apply**.
  - Select **xe-17/0/0**.
  - Enter **reth0** in the Redundant Parent box.
  - Click **Apply**.
  - See Step 5 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel” on page 1314.
6. Configure the IPsec configuration. See Step 6 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel” on page 1314.
  7. Configure the static routes .
    - Select **Configure>Routing>Static Routing**.
    - Click **Add**.
    - Enter the following information, and then click **Apply**:
      - Static Route Address: **0.0.0.0/0**
      - Next-Hop Addresses: **10.2.1.1**
    - Enter the following information, and then click **Apply**:
      - Static Route Address: **10.3.0.0/16**
      - Next-Hop Addresses: **10.10.1.2**
  8. Configure the security zones. See Step 8 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel” on page 1314.
  9. Configure the security policies. See Step 9 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel” on page 1314.
  10. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel on page 1312](#)

- [Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel on page 1314](#)
- [Understanding What Happens When Chassis Cluster Is Enabled on page 1210](#)
- [Understanding Chassis Cluster Formation on page 1138](#)

PART 12

# Network Address Translation

- Network Address Translation on page 1335



# Network Address Translation

- NAT Overview on page 1335
- Configuring NAT Using the NAT Wizard on page 1336
- Understanding NAT Rule Sets and Rules on page 1336
- Static NAT on page 1339
- Destination NAT on page 1350
- Source NAT on page 1368
- NAT for Multicast Flows on page 1417
- Configuring Proxy ARP (CLI Procedure) on page 1427
- Verifying NAT Configuration on page 1428

## NAT Overview

---

Network Address Translation (NAT) is a method for modifying or translating network address information in packet headers. Either or both source and destination addresses in a packet may be translated. NAT can include the translation of port numbers as well as IP addresses.

NAT is described in RFC 1631 to solve IP (version 4) address depletion problems. Since then, NAT has been found to be a useful tool for firewalls, traffic redirect, load sharing, network migrations, and so on.

The following types of NAT are supported on Juniper Networks devices:

- Static NAT
- Destination NAT
- Source NAT

You can use the NAT Wizard to perform basic NAT configuration. To perform more advanced configuration, use the J-Web interface or the CLI.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding NAT Rule Sets and Rules on page 1336
- Understanding Static NAT on page 1339

- Understanding Destination NAT on page 1350
- Understanding Source NAT on page 1369

## Configuring NAT Using the NAT Wizard

---

You can use the NAT Wizard to perform basic NAT configuration. To perform more advanced configuration, use the J-Web interface or the CLI.

To configure NAT using the NAT Wizard:

1. Select **Configure>Wizards>NAT Wizard** in the J-Web interface.
2. Click the Launch NAT Wizard button.
3. Follow the wizard prompts.

The upper left area of the wizard page shows where you are in the configuration process. The lower left area of the page shows field-sensitive help. When you click a link under the Resources heading, the document opens in your browser. If the document opens in a new tab, be sure to close only the tab (not the browser window) when you close the document.

## Understanding NAT Rule Sets and Rules

---

NAT processing centers on the evaluation of NAT rule sets and rules. A rule set determines the overall direction of the traffic to be processed. For example, a rule set can select traffic from a particular interface or to a specific zone. A rule set can contain multiple rules. Once a rule set is found that matches specific traffic, each rule in the rule set is evaluated for a match. Each rule in the rule set further specifies the traffic to be matched and the action to be taken when traffic matches the rule.

This topic includes the following sections:

- NAT Rule Sets on page 1336
- NAT Rules on page 1337
- Rule Processing on page 1338

### NAT Rule Sets

A rule set specifies a general set of matching conditions for traffic. For static NAT and destination NAT, a rule set specifies one of the following:

- Source interface
- Source zone
- Source routing instance

For source NAT rule sets, you configure both source and destination conditions:

- Source interface, zone, or routing instance

- Destination interface, zone, or routing instance

It is possible for a packet to match more than one rule set; in this case, the rule set with the more specific match is used. An interface match is considered more specific than a zone match, which is more specific than a routing instance match. If a packet matches both a destination NAT rule set that specifies a source zone and a destination NAT rule set that specifies a source interface, the rule set that specifies the source interface is the more specific match.

Source NAT rule set matching is more complex because you specify both source and destination conditions in a source NAT rule set. In the case where a packet matches more than one source NAT rule set, the rule set chosen is based on the following source/destination conditions (in order of priority):

1. Source interface/destination interface
2. Source zone/destination interface
3. Source routing instance/destination interface
4. Source interface/destination zone
5. Source zone/destination zone
6. Source routing instance/destination zone
7. Source interface/destination routing instance
8. Source zone/destination routing instance
9. Source routing instance/destination routing instance

For example, you can configure rule set A, which specifies a source interface and a destination zone, and rule set B, which specifies a source zone and a destination interface. If a packet matches both rule sets, rule set B is the more specific match.



**NOTE:** You cannot specify the same source and destination conditions for source NAT rule sets.

## NAT Rules

Once a rule set that matches the traffic has been found, each rule in the rule set is evaluated in order for a match. NAT rules can match on the following packet information:

- Destination address (for static NAT only)
- Source and destination address (for destination and source NAT)
- Destination port (for destination and source NAT)

The first rule in the rule set that matches the traffic is used. If a packet matches a rule in a rule set during session establishment, traffic is processed according to the action specified by that rule.

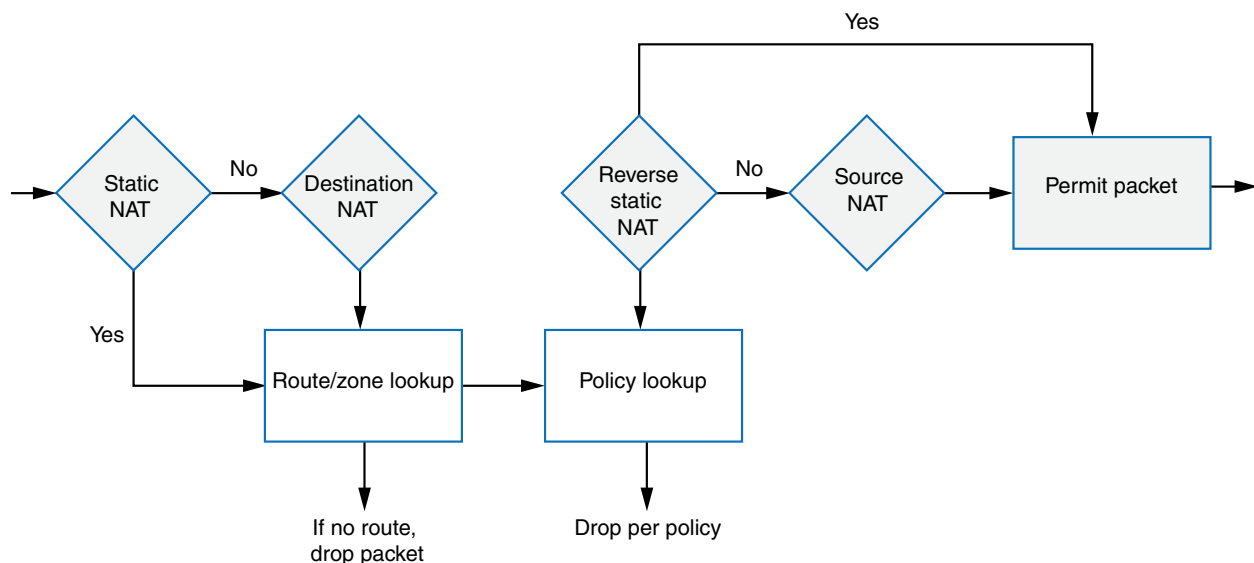
## Rule Processing

The NAT type determines the order in which NAT rules are processed. During the first packet processing for a flow, NAT rules are applied in the following order:

1. Static NAT rules
2. Destination NAT rules
3. Route lookup
4. Security policy lookup
5. Reverse mapping of static NAT rules
6. Source NAT rules

Figure 124 on page 1338 illustrates the order for NAT rule processing.

**Figure 124: NAT Rule Processing**



Static NAT and destination NAT rules are processed before route and security policy lookup. Static NAT rules take precedence over destination NAT rules. Reverse mapping of static NAT rules takes place after route and security policy lookup and takes precedence over source NAT rules. Source NAT rules are processed after route and security policy lookup and after reverse mapping of static NAT rules.

The configuration of rules and rule sets is basically the same for each type of NAT—source, destination, or static. But because both destination and static NAT are processed before route lookup, you cannot specify the destination zone, interface or routing instance in the rule set.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- NAT Overview on page 1335
- Static NAT Configuration Overview on page 1340



- Destination NAT Configuration Overview on page 1353
- Source NAT Configuration Overview on page 1374

## Static NAT

- Understanding Static NAT on page 1339
- Understanding Static NAT Rules on page 1339
- Static NAT Configuration Overview on page 1340
- Static NAT Configuration Examples on page 1340

### Understanding Static NAT

Static NAT defines a one-to-one mapping from one IP subnet to another IP subnet. The mapping includes destination IP address translation in one direction and source IP address translation in the reverse direction. From the NAT device, the original destination address is the virtual host IP address while the mapped-to address is the real host IP address.

Static NAT allows connections to be originated from either side of the network, but translation is limited to one-to-one or between blocks of addresses of the same size. For each private address, a public address must be allocated. No address pools are necessary.



**NOTE:** The original destination address, along with other addresses in source and destination NAT pools, must not overlap within the same routing instance.

Static NAT does not perform port address translation (PAT) and no address pools are needed for static NAT.

In NAT rule lookup, static NAT rules take precedence over destination NAT rules and reverse mapping of static NAT rules take precedence over source NAT rules.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Static NAT Configuration Overview on page 1340
- Example: Configuring Static NAT for Single Address Translation on page 1341
- Example: Configuring Static NAT for Subnet Translation on page 1345
- NAT Overview on page 1335
- Understanding Static NAT Rules on page 1339

### Understanding Static NAT Rules

Static NAT rules specify two layers of match conditions:

- Traffic direction—Allows you to specify **from interface**, **from zone**, or **from routing-instance**.
- Packet information—Destination IP address.

If multiple static NAT rules overlap in the match conditions, the most specific rule is chosen. For example, if rules A and B specify the same source and destination IP addresses, but rule A specifies traffic from zone 1 and rule B specifies traffic from interface **ge-0/0/0**, rule B is used to perform static NAT. An interface match is considered to be more specific than a zone match, which is more specific than a routing instance match.

For the static NAT rule action, specify the translated address and (optionally) the routing instance.

In NAT lookup, static NAT rules take precedence over destination NAT rules and reverse mapping of static NAT rules takes precedence over source NAT rules.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Static NAT on page 1339](#)
- [Static NAT Configuration Overview on page 1340](#)
- [Example: Configuring Static NAT for Single Address Translation on page 1341](#)
- [Example: Configuring Static NAT for Subnet Translation on page 1345](#)
- [Understanding NAT Rule Sets and Rules on page 1336](#)

## Static NAT Configuration Overview

The main configuration tasks for static NAT are as follows:

1. Configure static NAT rules that align with your network and security requirements.
2. Configure NAT proxy ARP entries for IP addresses in the same subnet of the ingress interface.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Static NAT on page 1339](#)
- [Configuring Proxy ARP \(CLI Procedure\) on page 1427](#)
- [Example: Configuring Static NAT for Single Address Translation on page 1341](#)
- [Example: Configuring Static NAT for Subnet Translation on page 1345](#)
- [Verifying NAT Configuration on page 1428](#)

## Static NAT Configuration Examples

- [Example: Configuring Static NAT for Single Address Translation on page 1341](#)
- [Example: Configuring Static NAT for Subnet Translation on page 1345](#)

---

### Example: Configuring Static NAT for Single Address Translation

---

This example describes how to configure a static NAT mapping of a single private address to a public address.

- Requirements on page 1341
- Overview on page 1341
- Configuration on page 1343
- Verification on page 1345

#### **Requirements**

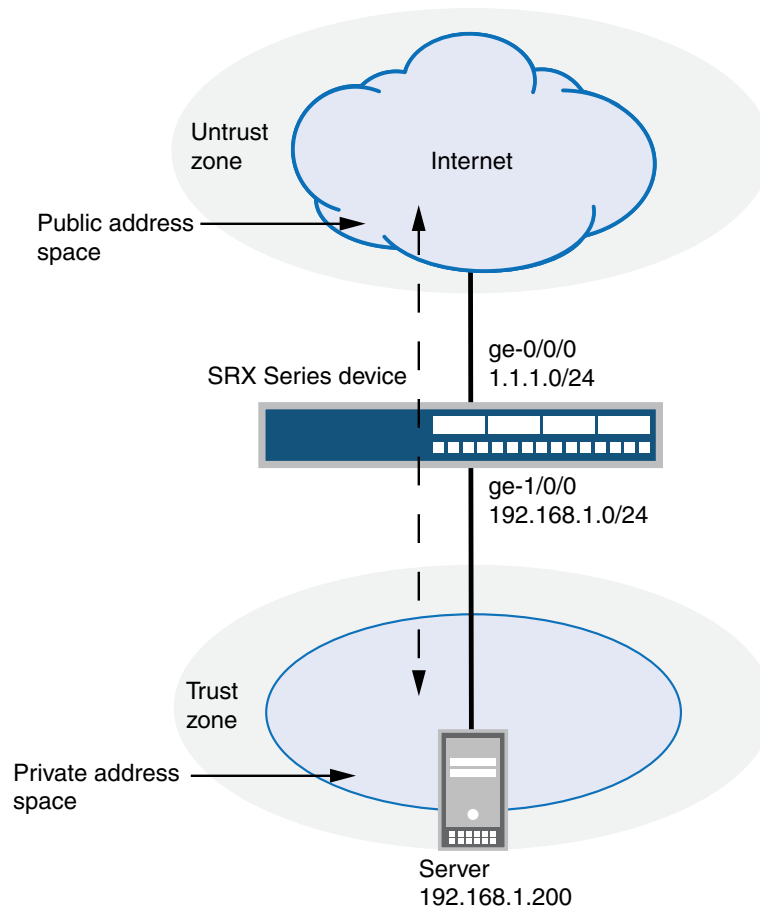
Before you begin:

1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 113.

#### **Overview**

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 125 on page 1342, devices in the untrust zone access a server in the trust zone by way of public address 1.1.1.200/32. For packets that enter the Juniper Networks security device from the untrust zone with the destination IP address 1.1.1.200/32, the destination IP address is translated to the private address 192.168.1.200/32. For a new session originating from the server, the source IP address in the outgoing packet is translated to the public address 1.1.1.200/32.

Figure 125: Static NAT Single Address Translation



Original Destination IP	Translated Destination IP
1.1.1.200/32	192.168.1.200/32

g030663

This example describes the following configurations:

- Static NAT rule set `rs1` with rule `r1` to match packets from the untrust zone with the destination address `1.1.1.200/32`. For matching packets, the destination IP address is translated to the private address `192.168.1.200/32`.
- Proxy ARP for the address `1.1.1.200` on interface `ge-0/0/0.0`. This allows the Juniper Networks security device to respond to ARP requests received on the interface for that address.
- Security policies to permit traffic to and from the `192.168.1.200` server.

**Configuration**

**CLI Quick Configuration** To quickly configure a static NAT mapping from a private address to a public address, copy the following commands and paste them into the CLI.

```
[edit]
set security nat static rule-set rs1 from zone untrust
set security nat static rule-set rs1 rule r1 match destination-address 1.1.1.200/32
set security nat static rule-set rs1 rule r1 then static-nat prefix 192.168.1.200/32
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.200/32
set security zones security-zone trust address-book address server-1 192.168.1.200/32
set security policies from-zone trust to-zone untrust policy permit-all match
  source-address server-1
set security policies from-zone trust to-zone untrust policy permit-all match
  destination-address any
set security policies from-zone trust to-zone untrust policy permit-all match application
  any
set security policies from-zone trust to-zone untrust policy permit-all then permit
set security policies from-zone untrust to-zone trust policy server-access match
  source-address any
set security policies from-zone untrust to-zone trust policy server-access match
  destination-address server-1
set security policies from-zone untrust to-zone trust policy server-access match application
  any
set security policies from-zone untrust to-zone trust policy server-access then permit
```

**Step-by-Step Procedure** The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a static NAT mapping from a private address to a public address:

1. Create a static NAT rule set.

```
[edit security nat static]
user@host# set rule-set rs1 from zone untrust
```

2. Configure a rule that matches packets and translates the destination address in the packets to a private address.

```
[edit security nat static]
user@host# set rule-set rs1 rule r1 match destination-address 1.1.1.200/32
user@host# set rule-set rs1 rule r1 then static-nat prefix 192.168.1.200/32
```

3. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.200
```

4. Configure an address book entry in the trust zone for the server's IP address.

```
[edit security]
user@host# set zones security-zone trust address-book address server-1
  192.168.1.200/32
```

5. Configure a security policy that allows traffic from the untrust zone to the server in the trust zone.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy server-access match source-address any destination-address
server-1 application any
user@host# set policy server-access then permit
```

6. Configure a security policy that allows all traffic from the server in the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy permit-all match source-address server-1 destination-address
any application any
user@host# set policy permit-all then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
static {
  rule-set rs1 {
    from zone untrust;
    rule r1 {
      match {
        destination-address 1.1.1.200/32;
      }
      then {
        static-nat prefix 192.168.1.200/32;
      }
    }
  }
}
proxy-arp {
  interface ge-0/0/0.0 {
    address {
      1.1.1.200/32;
    }
  }
}
user@host# show security policies
from-zone trust to-zone untrust {
  policy permit-all {
    match {
      source-address server-1;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone untrust to-zone trust {
  policy server-access {
    match {
      source-address any;
```

```

        destination-address server-1;
        application any;
    }
    then {
        permit;
    }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Static NAT Configuration on page 1345
- Verifying NAT Application to Traffic on page 1345

### Verifying Static NAT Configuration

**Purpose** Verify that there is traffic matching the static NAT rule set.

**Action** From operational mode, enter the **show security nat static rule** command. View the Translation hits field to check for traffic that matches the rule.

### Verifying NAT Application to Traffic

**Purpose** Verify that NAT is being applied to the specified traffic.

**Action** From operational mode, enter the **show security flow session** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding Static NAT on page 1339
  - Static NAT Configuration Overview on page 1340
  - Example: Configuring Static NAT for Subnet Translation on page 1345

### Example: Configuring Static NAT for Subnet Translation

This example describes how to configure a static NAT mapping of a private subnet address to a public subnet address.



**NOTE:** Address blocks for static NAT mapping must be of the same size.

- Requirements on page 1346
- Overview on page 1346
- Configuration on page 1348
- Verification on page 1350

### **Requirements**

Before you begin:

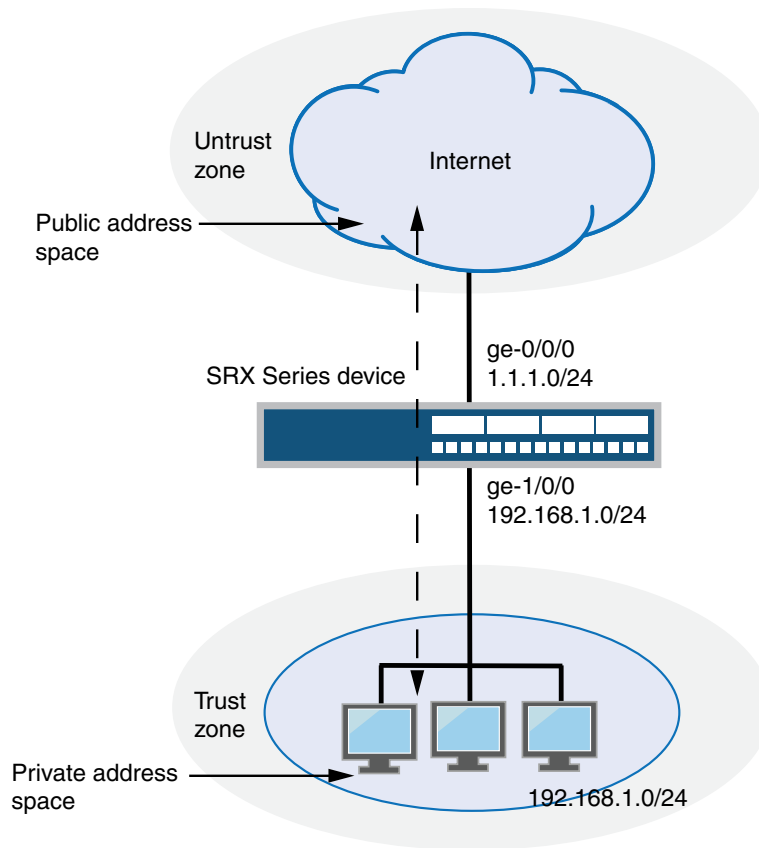
1. Configure network interfaces on the device. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 113.

### **Overview**

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 126 on page 1347, devices in the untrust zone access devices in the trust zone by way of public subnet address 1.1.1.0/24. For packets that enter the Juniper Networks security device from the untrust zone with a destination IP address in the 1.1.1.0/24 subnet, the destination IP address is translated to a private address on the 192.168.1.0/24 subnet. For new sessions originating from the 192.168.1.0/24 subnet, the source IP address in outgoing packets is translated to an address on the public 1.1.1.0/24 subnet.



Figure 126: Static NAT Subnet Translation



Original Destination IP	Translated Destination IP
1.1.1.0/24	192.168.1.0/24

g030664

This example describes the following configurations:

- Static NAT rule set `rs1` with rule `r1` to match packets received on interface `ge-0/0/0.0` with a destination IP address in the `1.1.1.0/24` subnet. For matching packets, the destination address is translated to an address on the `192.168.1.0/24` subnet.
- Proxy ARP for the address ranges `1.1.1.1/32` through `1.1.1.249/32` on interface `ge-0/0/0.0`. This allows the Juniper Networks security device to respond to ARP requests received on the interface for those addresses. The address `1.1.1.250/32` is assigned to the interface itself, so this address is not included in the proxy ARP configuration.
- Security policies to permit traffic to and from the `192.168.1.0/24` subnet.

**Configuration**

**CLI Quick Configuration** To quickly configure a static NAT mapping from a private subnet address to a public subnet address, copy the following commands and paste them into the CLI.

```
[edit]
set security nat static rule-set rs1 from interface ge-0/0/0.0
set security nat static rule-set rs1 rule r1 match destination-address 1.1.1.0/24
set security nat static rule-set rs1 rule r1 then static-nat prefix 192.168.1.0/24
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.1/32 to 1.1.1.249/32
set security zones security-zone trust address-book address server-group 192.168.1.0/24
set security policies from-zone trust to-zone untrust policy permit-all match
  source-address server-group
set security policies from-zone trust to-zone untrust policy permit-all match
  destination-address any
set security policies from-zone trust to-zone untrust policy permit-all match application
  any
set security policies from-zone trust to-zone untrust policy permit-all then permit
set security policies from-zone untrust to-zone trust policy server-access match
  source-address any
set security policies from-zone untrust to-zone trust policy server-access match
  destination-address server-group
set security policies from-zone untrust to-zone trust policy server-access match application
  any
set security policies from-zone untrust to-zone trust policy server-access then permit
```

**Step-by-Step Procedure** The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a static NAT mapping from a private subnet address to a public subnet address:

1. Create a static NAT rule set.

```
[edit security nat static]
user@host# set rule-set rs1 from interface ge-0/0/0.0
```

2. Configure a rule that matches packets and translates the destination address in the packets to an address in a private subnet.

```
[edit security nat static]
user@host# set rule-set rs1 rule r1 match destination-address 1.1.1.0/24
user@host# set rule-set rs1 rule r1 then static-nat prefix 192.168.1.0/24
```

3. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.1/32 to 1.1.1.249/32
```

4. Configure an address book entry in the trust zone for the subnet.

```
[edit security]
user@host# set zones security-zone trust address-book address server-group
  192.168.1.0/24
```

5. Configure a security policy that allows traffic from the untrust zone to the subnet in the trust zone.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy server-access match source-address any destination-address
server-group application any
user@host# set policy server-access then permit
```

6. Configure a security policy that allows all traffic from the subnet in the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy permit-all match source-address server-group
destination-address any application any
user@host# set policy permit-all then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
static {
  rule-set rs1 {
    from interface ge-0/0/0.0;
    rule r1 {
      match {
        destination-address 1.1.1.0/24;
      }
      then {
        static-nat prefix 192.168.1.0/24;
      }
    }
  }
}
proxy-arp {
  interface ge-0/0/0.0 {
    address {
      1.1.1.1/32;
    }
  }
}
user@host# show security policies
from-zone trust to-zone untrust {
  policy permit-all {
    match {
      source-address server-group;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone untrust to-zone trust {
  policy server-access {
    match {
      source-address any;
```

```
        destination-address server-group;
        application any;
    }
    then {
        permit;
    }
}
}
```

If you are done configuring the device, enter **commit** from configuration mode.

#### **Verification**

To confirm that the configuration is working properly, perform these tasks:

- [Verifying Static NAT Configuration on page 1350](#)
- [Verifying NAT Application to Traffic on page 1350](#)

#### **Verifying Static NAT Configuration**

**Purpose** Verify that there is traffic matching the static NAT rule set.

**Action** From operational mode, enter the **show security nat static rule** command. View the Translation hits field to check for traffic that matches the rule.

#### **Verifying NAT Application to Traffic**

**Purpose** Verify that NAT is being applied to the specified traffic.

**Action** From operational mode, enter the **show security flow session** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding Static NAT on page 1339](#)
  - [Static NAT Configuration Overview on page 1340](#)
  - [Example: Configuring Static NAT for Single Address Translation on page 1341](#)

---

## Destination NAT

- [Understanding Destination NAT on page 1350](#)
- [Understanding Destination NAT Address Pools on page 1351](#)
- [Understanding Destination NAT Rules on page 1352](#)
- [Destination NAT Configuration Overview on page 1353](#)
- [Destination NAT Configuration Examples on page 1353](#)

### Understanding Destination NAT

Destination NAT is the translation of the destination IP address of a packet entering the Juniper Networks device. Destination NAT is used to redirect traffic destined to a virtual

host (identified by the original destination IP address) to the real host (identified by the translated destination IP address).



**NOTE:** When destination NAT is performed, the destination IP address is translated according to configured destination NAT rules and then security policies are applied.

Destination NAT allows connections to be initiated only for incoming network connections—for example, from the Internet to a private network. Destination NAT is commonly used to perform the following actions:

- Translate a single IP address to another address (for example, to allow a device on the Internet to connect to a host on a private network).
- Translate a contiguous block of addresses to another block of addresses of the same size (for example, to allow access to a group of servers).
- Translate a destination IP address and port to another destination IP address and port (for example, to allow access to multiple services using the same IP address but different ports).

The following types of destination NAT are supported:

- Translation of the original destination IP address to an IP address from a user-defined pool. This type of translation does not include Port Address Translation (PAT). If the original destination IP address range is larger than the address range in the user-defined address pool, any untranslated packets are dropped.
- Translation of the original destination IP address (and optional port number) to one specific IP address (and port number) from a user-defined pool.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Destination NAT Configuration Overview on page 1353
- Example: Configuring Destination NAT for Single Address Translation on page 1353
- Example: Configuring Destination NAT for IP Address and Port Translation on page 1358
- Example: Configuring Destination NAT for Subnet Translation on page 1364
- NAT Overview on page 1335
- Understanding Destination NAT Address Pools on page 1351
- Understanding Destination NAT Rules on page 1352

## Understanding Destination NAT Address Pools

For destination NAT address pools, specify the following:

- Name of the destination NAT address pool
- Destination address or address range



**NOTE:** Do not overlap NAT addresses for source NAT, destination NAT, and static NAT within one routing instance.

- Destination port that is used for port forwarding
- Routing instance to which the pool belongs (the default is the main `inet.0` routing instance)

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Destination NAT on page 1350
- Destination NAT Configuration Overview on page 1353
- Example: Configuring Destination NAT for Single Address Translation on page 1353
- Example: Configuring Destination NAT for IP Address and Port Translation on page 1358
- Example: Configuring Destination NAT for Subnet Translation on page 1364

## Understanding Destination NAT Rules

Destination NAT rules specify two layers of match conditions:

- Traffic direction—Allows you to specify **from interface**, **from zone**, or **from routing-instance**.
- Packet information—Can be source IP addresses, destination IP address or subnet, or a single destination port number.

If multiple destination NAT rules overlap in the match conditions, the most specific rule is chosen. For example, if rules A and B specify the same source and destination IP addresses, but rule A specifies traffic from zone 1 and rule B specifies traffic from interface `ge-0/0/0`, rule B is used to perform destination NAT. An interface match is considered to be more specific than a zone match, which is more specific than a routing instance match.

The actions you can specify for a destination NAT rule are:

- `off`—Do not perform destination NAT.
- `pool`—Use the specified user-defined address pool to perform destination NAT.

Destination NAT rules are applied to traffic in the first packet that is processed for the flow or in the fast path for the ALG. Destination NAT rules are processed after static NAT rules but before source NAT rules.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Destination NAT on page 1350
- Destination NAT Configuration Overview on page 1353
- Example: Configuring Destination NAT for Single Address Translation on page 1353

- Example: Configuring Destination NAT for IP Address and Port Translation on page 1358
- Example: Configuring Destination NAT for Subnet Translation on page 1364
- Understanding NAT Rule Sets and Rules on page 1336

## Destination NAT Configuration Overview

The main configuration tasks for destination NAT are as follows:

1. Configure a destination NAT address pool that aligns with your network and security requirements.
2. Configure destination NAT rules that align with your network and security requirements.
3. Configure NAT proxy ARP entries for IP addresses in the same subnet of the ingress interface.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Destination NAT on page 1350
- Configuring Proxy ARP (CLI Procedure) on page 1427
- Example: Configuring Destination NAT for Single Address Translation on page 1353
- Example: Configuring Destination NAT for IP Address and Port Translation on page 1358
- Example: Configuring Destination NAT for Subnet Translation on page 1364
- Verifying NAT Configuration on page 1428

## Destination NAT Configuration Examples

- Example: Configuring Destination NAT for Single Address Translation on page 1353
- Example: Configuring Destination NAT for IP Address and Port Translation on page 1358
- Example: Configuring Destination NAT for Subnet Translation on page 1364

### Example: Configuring Destination NAT for Single Address Translation

This example describes how to configure a destination NAT mapping of a single public address to a private address.



**NOTE:** Mapping one destination IP address to another can also be accomplished with static NAT. Static NAT mapping allows connections to be established from either side of the gateway device, whereas destination NAT only allows connections to be established from one side. However, static NAT only allows translations from one address to another or between blocks of addresses of the same size.

- Requirements on page 1354
- Overview on page 1354

- Configuration on page 1356
- Verification on page 1358

### **Requirements**

Before you begin:

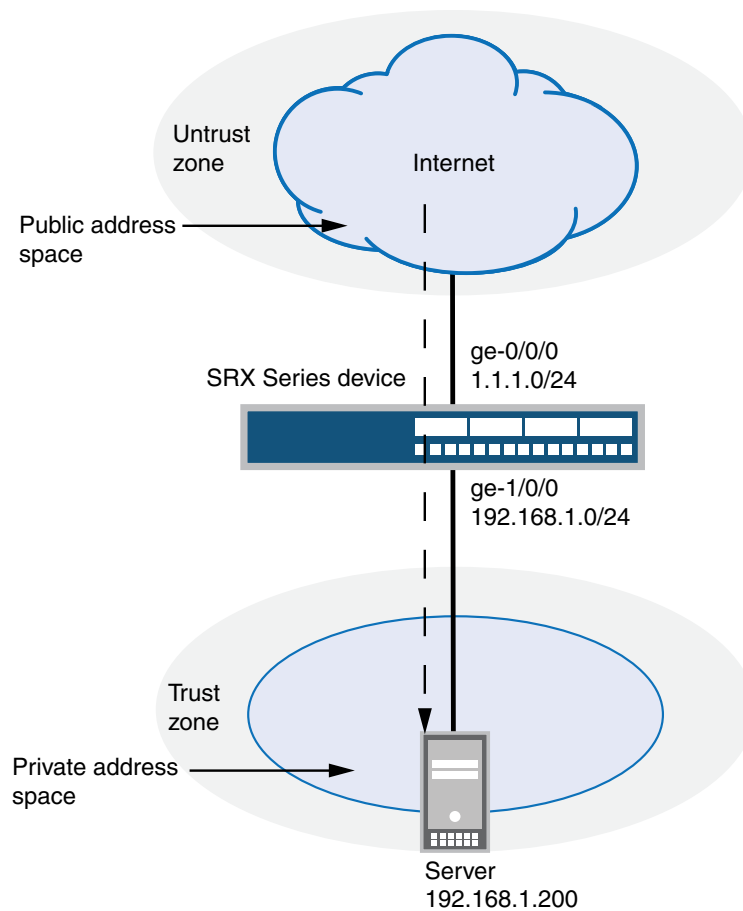
1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 113.

### **Overview**

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 127 on page 1355, devices in the untrust zone access a server in the trust zone by way of public address 1.1.1.200/32. For packets that enter the Juniper Networks security device from the untrust zone with the destination IP address 1.1.1.200/32, the destination IP address is translated to the private address 192.168.1.200/32.



Figure 127: Destination NAT Single Address Translation



Original Destination IP	Translated Destination IP
1.1.1.200/32	192.168.1.200/32

g030665

This example describes the following configurations:

- Destination NAT pool **dst-nat-pool-1** that contains the IP address 192.168.1.200/32.
- Destination NAT rule set **rs1** with rule **r1** to match packets received from the ge-0/0/0.0 interface with the destination IP address 1.1.1.200/32. For matching packets, the destination address is translated to the address in the **dst-nat-pool-1** pool.
- Proxy ARP for the address 1.1.1.200/32 on interface ge-0/0/0.0. This allows the Juniper Networks security device to respond to ARP requests received on the interface for that address.
- Security policies to permit traffic from the untrust zone to the translated destination IP address in the trust zone.

**Configuration**

**CLI Quick Configuration** To quickly configure a destination NAT mapping from a public address to a private address, copy the following commands and paste them into the CLI.

```
[edit]
set security nat destination pool dst-nat-pool-1 address 192.168.1.200/32
set security nat destination rule-set rs1 from interface ge-0/0/0.0
set security nat destination rule-set rs1 rule r1 match destination-address 1.1.1.200/32
set security nat destination rule-set rs1 rule r1 then destination-nat pool dst-nat-pool-1
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.200/32
set security zones security-zone trust address-book address server-1 192.168.1.200/32
set security policies from-zone untrust to-zone trust policy server-access match
  source-address any
set security policies from-zone untrust to-zone trust policy server-access match
  destination-address server-1
set security policies from-zone untrust to-zone trust policy server-access match application
  any
set security policies from-zone untrust to-zone trust policy server-access then permit
```

**Step-by-Step Procedure** The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a destination NAT mapping from a public address to a private address:

1. Create the destination NAT pool.

```
[edit security nat destination]
user@host# set pool dst-nat-pool-1 address 192.168.1.200/32
```

2. Create a destination NAT rule set.

```
[edit security nat destination]
user@host# set rule-set rs1 from interface ge-0/0/0.0
```

3. Configure a rule that matches packets and translates the destination address to the address in the pool.

```
[edit security nat destination]
user@host# set rule-set rs1 rule r1 match destination-address 1.1.1.200/32
user@host# set rule-set rs1 rule r1 then destination-nat pool dst-nat-pool-1
```

4. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.200/32
```

5. Configure an address book entry in the trust zone for the server.

```
[edit security]
user@host# set zones security-zone trust address-book address server-1
  192.168.1.200/32
```

6. Configure a security policy that allows traffic from the untrust zone to the server in the trust zone.

```
[edit security policies from-zone untrust to-zone trust]
```

```

user@host# set policy server-access match source-address any destination-address
server-1 application any
user@host# set policy server-access then permit

```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show security nat
destination {
  pool dst-nat-pool-1 {
    address 192.168.1.200/32;
  }
  rule-set rs1 {
    from interface ge-0/0/0.0;
    rule r1 {
      match {
        destination-address 1.1.1.200/32;
      }
      then {
        destination-nat pool dst-nat-pool-1;
      }
    }
  }
}
proxy-arp {
  interface ge-0/0/0.0 {
    address {
      1.1.1.200/32;
    }
  }
}
user@host# show security policies
from-zone untrust to-zone trust {
  policy server-access {
    match {
      source-address any;
      destination-address server-1;
      application any;
    }
    then {
      permit;
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

**Verification**

To confirm that the configuration is working properly, perform these tasks:

- Verifying Destination NAT Pool Usage on page 1358
- Verifying Destination NAT Rule Usage on page 1358
- Verifying NAT Application to Traffic on page 1358

**Verifying Destination NAT Pool Usage**

**Purpose** Verify that there is traffic using IP addresses from the destination NAT pool.

**Action** From operational mode, enter the **show security nat destination pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

**Verifying Destination NAT Rule Usage**

**Purpose** Verify that there is traffic matching the destination NAT rule.

**Action** From operational mode, enter the **show security nat destination rule all** command. View the Translation hits field to check for traffic that matches the rule.

**Verifying NAT Application to Traffic**

**Purpose** Verify that NAT is being applied to the specified traffic.

**Action** From operational mode, enter the **show security flow session** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Destination NAT on page 1350
- Destination NAT Configuration Overview on page 1353
- Example: Configuring Destination NAT for IP Address and Port Translation on page 1358
- Example: Configuring Destination NAT for Subnet Translation on page 1364

**Example: Configuring Destination NAT for IP Address and Port Translation**

---

This example describes how to configure destination NAT mappings of a public address to private addresses, depending on the port number.

- Requirements on page 1359
- Overview on page 1359
- Configuration on page 1361
- Verification on page 1363

**Requirements**

Before you begin:

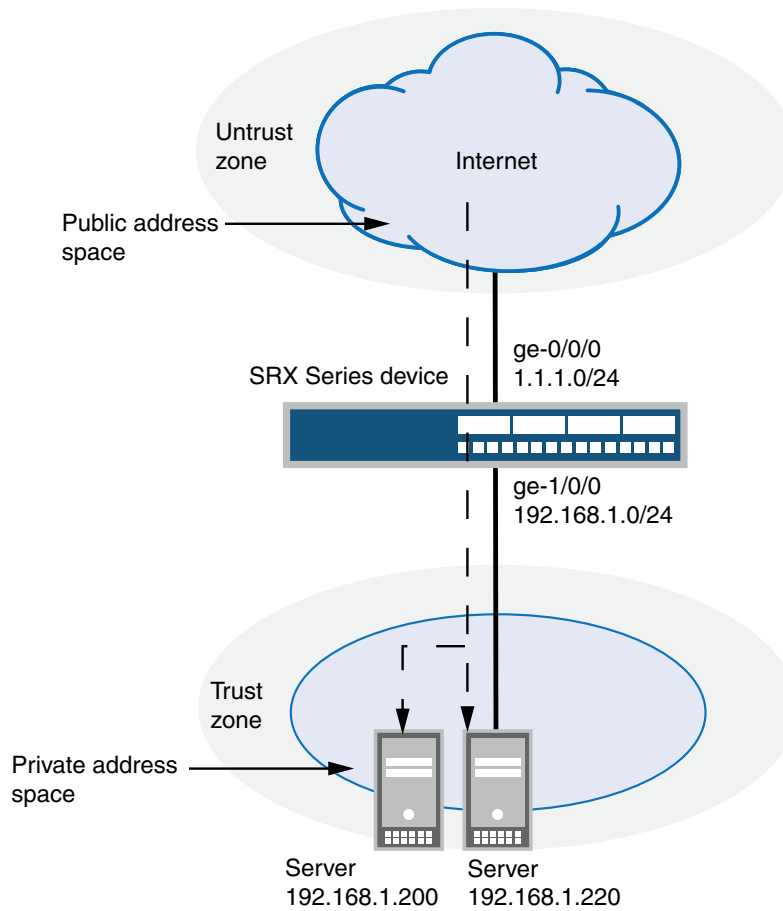
1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 113.

**Overview**

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 128 on page 1360, devices in the untrust zone access servers in the trust zone by way of public address 1.1.1.200 on port 80 or 8000. Packets entering the Juniper Networks security device from the untrust zone are mapped to the private addresses of the servers as follows:

- The destination IP address 1.1.1.200 and port 80 is translated to the private address 192.168.1.200 and port 80.
- The destination IP address 1.1.1.200 and port 8000 is translated to the private address 192.168.1.220 and port 8000.

Figure 128: Destination NAT Address and Port Translation



Original Destination IP	Translated Destination IP
1.1.1.200 port 80	192.168.1.200 port 80
1.1.1.200 port 8000	192.168.1.220 port 8000

g030666

This example describes the following configurations:

- Destination NAT pool **dst-nat-pool-1** that contains the IP address 192.168.1.200 port 80.
- Destination NAT pool **dst-nat-pool-2** that contains the IP address 192.168.1.220 and port 8000.
- Destination NAT rule set **rs1** with rule **r1** to match packets received from the untrust zone with the destination IP address 1.1.1.200 and destination port 80. For matching packets, the destination address is translated to the address in the **dst-nat-pool-1** pool.

- Destination NAT rule set `rs1` with rule `r2` to match packets received from the untrust zone with the destination IP address 1.1.1.200 and destination port 8000. For matching packets, the destination IP address and port are translated to the address and port in the `dst-nat-pool-2` pool.
- Proxy ARP for the address 1.1.1.200/32. This allows the Juniper Networks security device to respond to ARP requests received on the interface for that address.
- Security policies to permit traffic from the untrust zone to the translated destination IP addresses in the trust zone.

### Configuration

#### CLI Quick Configuration

To quickly configure a destination NAT mapping from a public address to a private address, copy the following commands and paste them into the CLI.

```
[edit]
set security nat destination pool dst-nat-pool-1 address 192.168.1.200/32
set security nat destination pool dst-nat-pool-1 address port 80
set security nat destination pool dst-nat-pool-2 address 192.168.1.220/32
set security nat destination pool dst-nat-pool-2 address port 8000
set security nat destination rule-set rs1 from zone untrust
set security nat destination rule-set rs1 rule r1 match destination-address 1.1.1.200/32
set security nat destination rule-set rs1 rule r1 match destination-port 80
set security nat destination rule-set rs1 rule r1 then destination-nat pool dst-nat-pool-1
set security nat destination rule-set rs1 rule r2 match destination-address 1.1.1.200/32
set security nat destination rule-set rs1 rule r2 match destination-port 8000
set security nat destination rule-set rs1 rule r2 then destination-nat pool dst-nat-pool-2
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.200/32
set security zones security-zone trust address-book address server-2 192.168.1.220/32
set security zones security-zone tru address-book address server-1 192.168.1.200/32
set security policies from-zone untrust to-zone trust policy server-access match
  source-address any
set security policies from-zone untrust to-zone trust policy server-access match
  destination-address server-1
set security policies from-zone untrust to-zone trust policy server-access match
  destination-address server-2
set security policies from-zone untrust to-zone trust policy server-access match application
  any
set security policies from-zone untrust to-zone trust policy server-access then permit
```

#### Step-by-Step Procedure

The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a destination NAT mapping from a public address to a private address:

1. Create destination NAT pools.

```
[edit security nat destination]
user@host# set pool dst-nat-pool-1 address 192.168.1.200 port 80
user@host# set pool dst-nat-pool-2 address 192.168.1.220 port 8000
```

2. Create a destination NAT rule set.

```
[edit security nat destination]
user@host# set rule-set rs1 from zone untrust
```

- Configure a rule that matches packets and translates the destination address to the address in the pool.

```
[edit security nat destination]
user@host# set rule-set rs1 rule r1 match destination-address 1.1.1.200
user@host# set rule-set rs1 rule r1 match destination-port 80
user@host# set rule-set rs1 rule r1 then destination-nat pool dst-nat-pool-1
```

- Configure a rule that matches packets and translates the destination address to the address in the pool.

```
[edit security nat destination]
user@host# set rule-set rs1 rule r2 match destination-address 1.1.1.200
user@host# set rule-set rs1 rule r2 match destination-port 8000
user@host# set rule-set rs1 rule r2 then destination-nat pool dst-nat-pool-2
```

- Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.200/32
```

- Configure address book entries in the trust zone for the server addresses.

```
[edit security]
user@host# set zones security-zone trust address-book address server-1
192.168.1.200/32
user@host# set zones security-zone trust address-book address server-2
192.168.1.220/32
```

- Configure a security policy that allows traffic from the untrust zone to the servers in the trust zone.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy server-access match source-address any destination-address
[server-1 server-2] application any
user@host# set policy server-access then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
destination {
  pool dst-nat-pool-1 {
    address 192.168.1.200/32 port 80;
  }
  pool dst-nat-pool-2 {
    address 192.168.1.220/32 port 8000;
  }
}
rule-set rs1 {
  from zone untrust;
  rule r1 {
    match {
      destination-address 1.1.1.200/32;
      destination-port 80;
    }
    then {
```





**Action** From operational mode, enter the **show security nat destination rule all** command. View the Translation hits field to check for traffic that matches the rule.

#### **Verifying NAT Application to Traffic**

**Purpose** Verify that NAT is being applied to the specified traffic.

**Action** From operational mode, enter the **show security flow session** command.

#### **Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Destination NAT on page 1350
- Destination NAT Configuration Overview on page 1353
- Example: Configuring Destination NAT for Single Address Translation on page 1353
- Example: Configuring Destination NAT for Subnet Translation on page 1364

#### **Example: Configuring Destination NAT for Subnet Translation**

This example describes how to configure a destination NAT mapping of a public subnet address to a private subnet address.



**NOTE:** Mapping addresses from one subnet to another can also be accomplished with static NAT. Static NAT mapping allows connections to be established from either side of the gateway device, whereas destination NAT allows connections to be established from only one side. However, static NAT only allows translations between blocks of addresses of the same size.

- Requirements on page 1364
- Overview on page 1364
- Configuration on page 1366
- Verification on page 1368

#### **Requirements**

Before you begin:

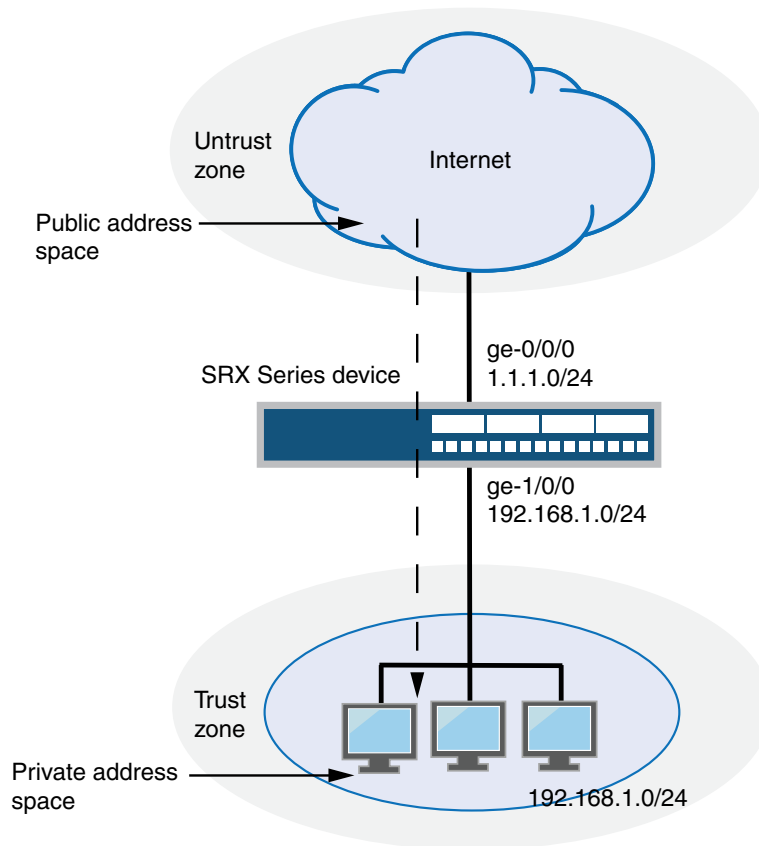
1. Configure network interfaces on the device. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 113.

#### **Overview**

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 129 on page 1365, devices in the untrust zone access devices in the trust zone by way of public subnet address 1.1.1.0/16. For packets that enter the Juniper Networks security device from the untrust zone with a

destination IP address in the 1.1.1.0/16 subnet, the destination IP address is translated to a private address on the 192.168.1.0/24 subnet.

Figure 129: Destination NAT Subnet Translation



g030667

This example describes the following configurations:

- Destination NAT pool **dst-nat-pool-1** that contains the IP address 192.168.1.0/24.
- Destination NAT rule set **rs1** with rule **r1** to match packets received from the ge-0/0/0.0 interface with the destination IP address on the 1.1.1.0/16 subnet. For matching packets, the destination address is translated to the address in the **dst-nat-pool-1** pool.
- Proxy ARP for the addresses 1.1.1.0/32 through 1.1.1.62/32 on the interface ge-0/0/0.0; these are the IP addresses of the hosts that should be translated from the 1.1.1.0/16 subnet. This allows the Juniper Networks security device to respond to ARP requests received on the interface for those addresses. The address 1.1.1.63/32 is assigned to

the interface itself, so this address is not included in the proxy ARP configuration. The addresses that are not in the 1.1.1.1/32 through 1.1.1.62/32 range are not expected to be present on the network and would not be translated.

- Security policies to permit traffic from the untrust zone to the translated destination IP addresses in the trust zone.

### Configuration

**CLI Quick Configuration** To quickly configure a destination NAT mapping from a public subnet address to a private subnet address, copy the following commands and paste them into the CLI.

```
[edit]
set security nat destination pool dst-nat-pool-1 address 192.168.1.0/24
set security nat destination rule-set rs1 from interface ge-0/0/0.0
set security nat destination rule-set rs1 rule r1 match destination-address 1.1.1.0/16
set security nat destination rule-set rs1 rule r1 then destination-nat pool dst-nat-pool-1
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.1/32 to 1.1.1.62/32
set security zones security-zone trust address-book address internal-net 192.168.1.0/24
set security policies from-zone untrust to-zone trust policy internal-access match
  source-address any
set security policies from-zone untrust to-zone trust policy internal-access match
  destination-address internal-net
set security policies from-zone untrust to-zone trust policy internal-access match
  application any
set security policies from-zone untrust to-zone trust policy internal-access then permit
```

**Step-by-Step Procedure** The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a destination NAT mapping from a public subnet address to a private subnet address:

1. Create the destination NAT pool.
 

```
[edit security nat destination]
user@host# set pool dst-nat-pool-1 address 192.168.1.0/24
```
2. Create a destination NAT rule set.
 

```
[edit security nat destination]
user@host# set rule-set rs1 from interface ge-0/0/0.0
```
3. Configure a rule that matches packets and translates the destination address to an address in the pool.
 

```
[edit security nat destination]
user@host# set rule-set rs1 rule r1 match destination-address 1.1.1.0/16
user@host# set rule-set rs1 rule r1 then destination-nat pool dst-nat-pool-1
```
4. Configure proxy ARP.
 

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.1/32 to 1.1.1.62/32
```
5. Configure an address book entry in the trust zone for the private subnet address.
 

```
[edit security]
```

```
user@host# set zones security-zone trust address-book address internal-net
192.168.1.0/24
```

- Configure a security policy that allows traffic from the untrust zone to the devices in the trust zone.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy internal-access match source-address any
destination-address internal-net application any
user@host# set policy internal-access then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
destination {
  pool dst-nat-pool-1 {
    address 192.168.1.0/24;
  }
  rule-set rs1 {
    from interface ge-0/0/0.0;
    rule r1 {
      match {
        destination-address 1.1.1.0/16;
      }
      then {
        destination-nat pool dst-nat-pool-1;
      }
    }
  }
}
proxy-arp {
  interface ge-0/0/0.0 {
    address {
      1.1.1.1/32 to 1.1.1.62/32;
    }
  }
}
user@host# show security policies
from-zone untrust to-zone trust {
  policy internal-access {
    match {
      source-address any;
      destination-address internal-net;
      application any;
    }
    then {
      permit;
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

**Verification**

To confirm that the configuration is working properly, perform these tasks:

- Verifying Destination NAT Pool Usage on page 1368
- Verifying Destination NAT Rule Usage on page 1368
- Verifying NAT Application to Traffic on page 1368

**Verifying Destination NAT Pool Usage**

**Purpose** Verify that there is traffic using IP addresses from the destination NAT pool.

**Action** From operational mode, enter the **show security nat destination pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

**Verifying Destination NAT Rule Usage**

**Purpose** Verify that there is traffic matching the destination NAT rule.

**Action** From operational mode, enter the **show security nat destination rule all** command. View the Translation hits field to check for traffic that matches the rule.

**Verifying NAT Application to Traffic**

**Purpose** Verify that NAT is being applied to the specified traffic.

**Action** From operational mode, enter the **show security flow session** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Destination NAT on page 1350
- Destination NAT Configuration Overview on page 1353
- Example: Configuring Destination NAT for Single Address Translation on page 1353
- Example: Configuring Destination NAT for IP Address and Port Translation on page 1358

---

**Source NAT**

---

- Understanding Source NAT on page 1369
- Source NAT Pools on page 1370
- Understanding Source NAT Rules on page 1373
- Source NAT Configuration Overview on page 1374
- Source NAT Configuration Examples on page 1374
- Disabling Port Randomization for Source NAT (CLI Procedure) on page 1411
- Persistent NAT on page 1412

## Understanding Source NAT

Source NAT is the translation of the source IP address of a packet leaving the Juniper Networks device. Source NAT is used to allow hosts with private IP addresses to access a public network.

Source NAT allows connections to be initiated only for outgoing network connections—for example, from a private network to the Internet. Source NAT is commonly used to perform the following translations:

- Translate a single IP address to another address (for example, to provide a single device in a private network with access to the Internet).
- Translate a contiguous block of addresses to another block of addresses of the same size.
- Translate a contiguous block of addresses to another block of addresses of smaller size.
- Translate a contiguous block of addresses to a single IP address or a smaller block of addresses using port translation.
- Translate a contiguous block of addresses to the address of the egress interface.

Translation to the address of the egress interface does not require an address pool; all other source NAT translations require configuration of an address pool. One-to-one and many-to-many translations for address blocks of the same size do not require port translation because there is an available address in the pool for every address that would be translated.

If the size of the address pool is smaller than the number of addresses that would be translated, either the total number of concurrent addresses that can be translated is limited by the size of the address pool or port translation must be used. For example, if a block of 253 addresses is translated to an address pool of 10 addresses, a maximum of 10 devices can be connected concurrently unless port translation is used.

The following types of source NAT are supported:

- Translation of the original source IP address to the egress interface's IP address (also called interface NAT). Port address translation is always performed.
- Translation of the original source IP address to an IP address from a user-defined address pool without port address translation. The association between the original source IP address to the translated source IP address is dynamic. However, once there is an association, the same association is used for the same original source IP address for new traffic that matches the same NAT rule.
- Translation of the original source IP address to an IP address from a user-defined address pool with port address translation. The association between the original source IP address to the translated source IP address is dynamic. Even if an association exists,

the same original source IP address may be translated to a different address for new traffic that matches the same NAT rule.

- Translation of the original source IP address to an IP address from a user-defined address pool by shifting the IP addresses. This type of translation is one-to-one, static, and without port address translation. If the original source IP address range is larger than the IP address range in the user-defined pool, untranslated packets are dropped.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Source NAT Pools on page 1370](#)
- [Understanding Source NAT Rules on page 1373](#)
- [Source NAT Configuration Overview on page 1374](#)
- [NAT Overview on page 1335](#)

## Source NAT Pools

- [Understanding Source NAT Pools on page 1370](#)
- [Understanding Source NAT Pools with PAT on page 1371](#)
- [Understanding Source NAT Pools Without PAT on page 1372](#)
- [Understanding Source NAT Pools with Address Shifting on page 1372](#)
- [Understanding Persistent Addresses on page 1373](#)

### Understanding Source NAT Pools

For source NAT address pools, specify the following:

- Name of the source NAT address pool.
- Up to eight address or address ranges.



**NOTE:** Do not overlap NAT addresses for source NAT, destination NAT, and static NAT within one routing instance.

- Routing instance to which the pool belongs (the default is the main **inet.0** routing instance).
- No port translation (optional)—By default, port address translation is performed with source NAT. If you specify the **port no-translation** option, the number of hosts that the source NAT pool can support is limited to the number of addresses in the pool.
- Overflow pool (optional)—Packets are dropped if there are no addresses available in the designated source NAT pool. To prevent that from happening when the **port no-translation** option is configured, you can specify an overflow pool. Once addresses from the original source NAT pool are exhausted, IP addresses and port numbers are allocated from the overflow pool. A user-defined source NAT pool or an egress interface



can be used as the overflow pool. (When the overflow pool is used, the pool ID is returned with the address.)

- IP address shifting (optional)—A range of original source IP addresses can be mapped to another range of IP addresses by shifting the IP addresses. Specify the **host-address-base** option with the base address of the original source IP address range.

When the **raise-threshold** option is configured for source NAT, an SNMP trap is triggered if the source NAT pool utilization rises above this threshold. If the optional **clear-threshold** option is configured, an SNMP trap is triggered if the source NAT pool utilization drops below this threshold. If **clear-threshold** is not configured it is set by default to 80 percent of the **raise-threshold** value.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Source NAT on page 1369](#)
- [Source NAT Configuration Overview on page 1374](#)
- [Understanding Source NAT Pools with PAT on page 1371](#)
- [Understanding Source NAT Pools Without PAT on page 1372](#)
- [Understanding Source NAT Pools with Address Shifting on page 1372](#)
- [Understanding Persistent Addresses on page 1373](#)

#### Understanding Source NAT Pools with PAT

Using the source pool with Port Address Translation (PAT), Junos OS translates both the source IP address and the port number of the packets. When PAT is used, multiple hosts can share the same IP address.

Junos OS maintains a list of assigned port numbers to distinguish what session belongs to which host. When PAT is enabled, up to 64,500 hosts can share a single IP address. Each source pool can contain multiple IP addresses, multiple IP address ranges, or both. For a source pool with PAT, Junos OS may assign different addresses to a single host for different concurrent sessions, unless the source pool or Junos OS has the persistent address feature enabled.

For interface source pool and source pool with PAT, range (1024, 65535) is available for port number mapping per IP address. Within range (1024, 63487) one port is allocated at a time. In range (63488, 65535), two ports are allocated at a time for RTP/RTCP applications such as SIP, H.323, and RTSP.

When a host initiates several sessions that match a policy that requires network address translation and is assigned an address from a source pool that has PAT enabled, the device assigns a different source IP address for each session. Such random address assignment can be problematic for services that create multiple sessions that require the same source IP address for each session. For example, it is important to have the same IP address for multiple sessions when using the AOL Instant Message (AIM) client.

To ensure that the router assigns the same IP address from a source pool to a host for multiple concurrent sessions, you can enable a persistent IP address per router.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Persistent Addresses on page 1373](#)
- [Understanding Source NAT on page 1369](#)
- [Understanding Source NAT Pools on page 1370](#)
- [Example: Configuring Source NAT for Single Address Translation on page 1378](#)
- [Example: Configuring Source NAT for Multiple Addresses with PAT on page 1383](#)
- [Example: Configuring Source NAT with Address Shifting on page 1393](#)
- [Example: Configuring Source NAT with Multiple Rules on page 1398](#)
- [Example: Configuring Source and Destination NAT Translations on page 1405](#)

---

### Understanding Source NAT Pools Without PAT

When you define a source pool, Junos OS enables PAT by default. To disable PAT, you must specify no port translation when you are defining a source pool.

When using a source pool without PAT, Junos OS performs source Network Address Translation for the IP address without performing PAT for the source port number. For applications that require that a particular source port number remain fixed, you must use source pool without PAT.

The source pool can contain multiple IP addresses, multiple IP address ranges, or both. For source pool without PAT, Junos OS assigns one translated source address to the same host for all its concurrent sessions.

Pool utilization for each source pool without PAT is computed. You can turn on pool utilization alarm by configuring alarm thresholds. An SNMP trap is triggered every time pool utilization rises above a threshold and goes below a threshold.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Configuring Source NAT for Multiple Addresses without PAT on page 1388](#)
- [Example: Configuring Source NAT with Multiple Rules on page 1398](#)
- [Understanding Source NAT on page 1369](#)
- [Understanding Source NAT Pools on page 1370](#)

---

### Understanding Source NAT Pools with Address Shifting

The match conditions for a source NAT rule set do not allow you to specify an address range; only address prefixes may be specified in a rule. When configuring a source NAT pool, you can specify the **host-base-address** option; this option specifies the IP address where the original source IP address range begins.

The range of original source IP addresses that are translated is determined by the number of addresses in the source NAT pool. For example, if the source NAT pool contains a range of ten IP addresses, then up to ten original source IP addresses can be translated,

starting with a specified base address. This type of translation is one-to-one, static, and without port address translation.

The match condition in a source NAT rule may define a larger address range than that specified in the source NAT pool. For example, a match condition might specify an address prefix that contains 256 addresses, but the source NAT pool contains a range of only ten IP addresses. A packet's source IP address can match a source NAT rule, but if the source IP address is not within the address range specified in the source NAT pool, the source IP address is not translated.

#### Related Documentation

- Junos OS Feature Support Reference for SRX Series and J Series Devices
- Understanding Source NAT
- Understanding Source NAT Pools
- Example: Configuring Source NAT with Address Shifting

#### Understanding Persistent Addresses

By default, port address translation is performed with source NAT. However, an original source address may not be translated to the same IP address for different traffic that originates from the same host. The source NAT **address-persistent** option ensures that the same IP address is assigned from the source NAT pool to a specific host for multiple concurrent sessions.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Source NAT on page 1369
- Source NAT Configuration Overview on page 1374
- Understanding Source NAT Pools with PAT on page 1371

## Understanding Source NAT Rules

Source NAT rules specify two layers of match conditions:

- Traffic direction—Allows you to specify combinations of **from interface**, **from zone**, or **from routing-instance** and **to interface**, **to zone**, or **to routing-instance**. You cannot configure the same **from** and **to** contexts for different rule sets.
- Packet information—Can be source and destination IP addresses or subnets.

If multiple source NAT rules overlap in the match conditions, the most specific rule is chosen. For example, if rules A and B specify the same source and destination IP addresses, but rule A specifies traffic from zone 1 to zone 2 and rule B specifies traffic from zone 1 to interface **ge-0/0/0**, rule B is used to perform source NAT. An interface match is considered to be more specific than a zone match, which is more specific than a routing instance match. For more information about rule set matching, see “Understanding NAT Rule Sets and Rules” on page 1336.

The actions you can specify for a source NAT rule are:

- off—Do not perform source NAT.
- pool—Use the specified user-defined address pool to perform source NAT.
- interface—Use the egress interface's IP address to perform source NAT.

Source NAT rules are applied to traffic in the first packet that is processed for the flow or in the fast path for the ALG. Source NAT rules are processed after static NAT rules, destination NAT rules, and reverse mapping of static NAT rules and after route and security policy lookup.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Source NAT on page 1369](#)
- [Source NAT Configuration Overview on page 1374](#)
- [Understanding NAT Rule Sets and Rules on page 1336](#)

## Source NAT Configuration Overview

The main configuration tasks for source NAT are as follows:

1. Configure a source NAT address pool that aligns with your network and security requirements (not needed for interface NAT).
2. Configure pool utilization alarms (optional)—Specify thresholds for pool utilization.
3. Configure address persistent (optional)—Ensures that the same IP address is assigned from the source NAT pool to a host for multiple concurrent sessions.
4. Configure source NAT rules that align with your network and security requirements.
5. Configure NAT proxy ARP entries for IP addresses in the same subnet of the ingress interface.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Configuring Source NAT for Egress Interface Translation on page 1375](#)
- [Example: Configuring Source NAT for Single Address Translation on page 1378](#)
- [Example: Configuring Source NAT for Multiple Addresses with PAT on page 1383](#)
- [Example: Configuring Source NAT for Multiple Addresses without PAT on page 1388](#)
- [Example: Configuring Source NAT with Address Shifting on page 1393](#)
- [Example: Configuring Source NAT with Multiple Rules on page 1398](#)
- [Example: Configuring Source and Destination NAT Translations on page 1405](#)
- [Verifying NAT Configuration on page 1428](#)

## Source NAT Configuration Examples

- [Example: Configuring Source NAT for Egress Interface Translation on page 1375](#)
- [Example: Configuring Source NAT for Single Address Translation on page 1378](#)

- Example: Configuring Source NAT for Multiple Addresses with PAT on page 1383
- Example: Configuring Source NAT for Multiple Addresses without PAT on page 1388
- Example: Configuring Source NAT with Address Shifting on page 1393
- Example: Configuring Source NAT with Multiple Rules on page 1398
- Example: Configuring Source and Destination NAT Translations on page 1405

### **Example: Configuring Source NAT for Egress Interface Translation**

This example describes how to configure a source NAT mapping of private addresses to the public address of an egress interface.

- Requirements on page 1375
- Overview on page 1375
- Configuration on page 1376
- Verification on page 1378

#### **Requirements**

Before you begin:

1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 113.

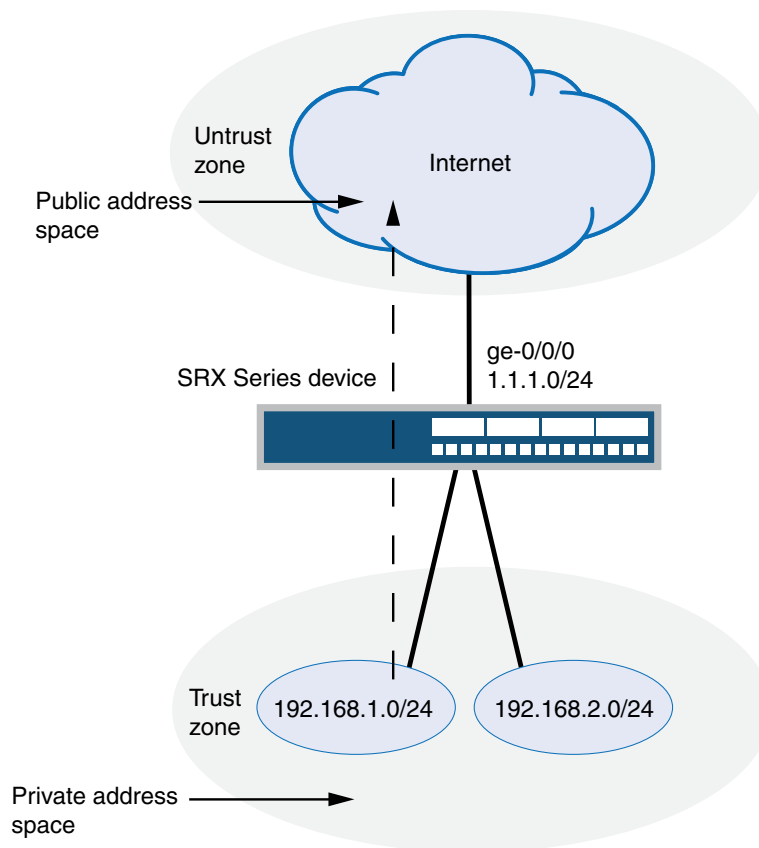
#### **Overview**

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 130 on page 1376, devices with private addresses in the trust zone access a public network through the egress interface ge-0/0/0. For packets that enter the Juniper Networks security device from the trust zone with a destination address in the untrust zone, the source IP address is translated to the IP address of the egress interface.



**NOTE:** No source NAT pool is required for source NAT using an egress interface. Proxy ARP does not need to be configured for the egress interface.

Figure 130: Source NAT Egress Interface Translation



Original Source IP	Translated Source IP
0.0.0.0/0	1.1.1.63 (Interface IP)

g030668

This example describes the following configurations:

- Source NAT rule set **rs1** with a rule **r1** to match any packet from the trust zone to the untrust zone. For matching packets, the source address is translated to the IP address of the egress interface.
- Security policies to permit traffic from the trust zone to the untrust zone.

**Configuration**

**CLI Quick Configuration** To quickly configure a source NAT mapping to an egress interface, copy the following commands and paste them into the CLI.

```
[edit]
set security nat source rule-set rs1 from zone trust
set security nat source rule-set rs1 to zone untrust
```

```

set security nat source rule-set rs1 rule r1 match source-address 0.0.0.0/0
set security nat source rule-set rs1 rule r1 match destination-address 0.0.0.0/0
set security nat source rule-set rs1 rule r1 then source-nat interface
set security policies from-zone trust to-zone untrust policy internet-access match
  source-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  destination-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  application any
set security policies from-zone trust to-zone untrust policy internet-access then permit

```

### Step-by-Step Procedure

The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a source NAT translation to an egress interface:

1. Create a source NAT rule set.

```

[edit security nat source]
user@host# set rule-set rs1 from zone trust
user@host# set rule-set rs1 to zone untrust

```

2. Configure a rule that matches packets and translates the source address to the address of the egress interface.

```

[edit security nat source]
user@host# set rule-set rs1 rule r1 match source-address 0.0.0.0/0
user@host# set rule-set rs1 rule r1 match destination-address 0.0.0.0/0
user@host# set rule-set rs1 rule r1 then source-nat interface

```

3. Configure a security policy that allows traffic from the trust zone to the untrust zone.

```

[edit security policies from-zone trust to-zone untrust]
user@host# set policy internet-access match source-address any
  destination-address any application any
user@host# set policy internet-access then permit

```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show security nat
source {
  rule-set rs1 {
    from zone trust;
    to zone untrust;
    rule r1 {
      match {
        source-address 0.0.0.0/0;
        destination-address 0.0.0.0/0;
      }
      then {
        source-nat {
          interface;

```

```

    }
  }
}
}
user@host# show security policies
from-zone trust to-zone untrust {
  policy internet-access {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### **Verification**

To confirm that the configuration is working properly, perform these tasks:

- Verifying Source NAT Rule Usage on page 1378
- Verifying NAT Application to Traffic on page 1378

#### **Verifying Source NAT Rule Usage**

**Purpose** Verify that there is traffic matching the source NAT rule.

**Action** From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

#### **Verifying NAT Application to Traffic**

**Purpose** Verify that NAT is being applied to the specified traffic.

**Action** From operational mode, enter the **show security flow session** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding Source NAT on page 1369
  - Source NAT Configuration Overview on page 1374

### **Example: Configuring Source NAT for Single Address Translation**

This example describes how to configure a source NAT mapping of a single private address to a public address.

- Requirements on page 1379
- Overview on page 1379



- Configuration on page 1381
- Verification on page 1383

**Requirements**

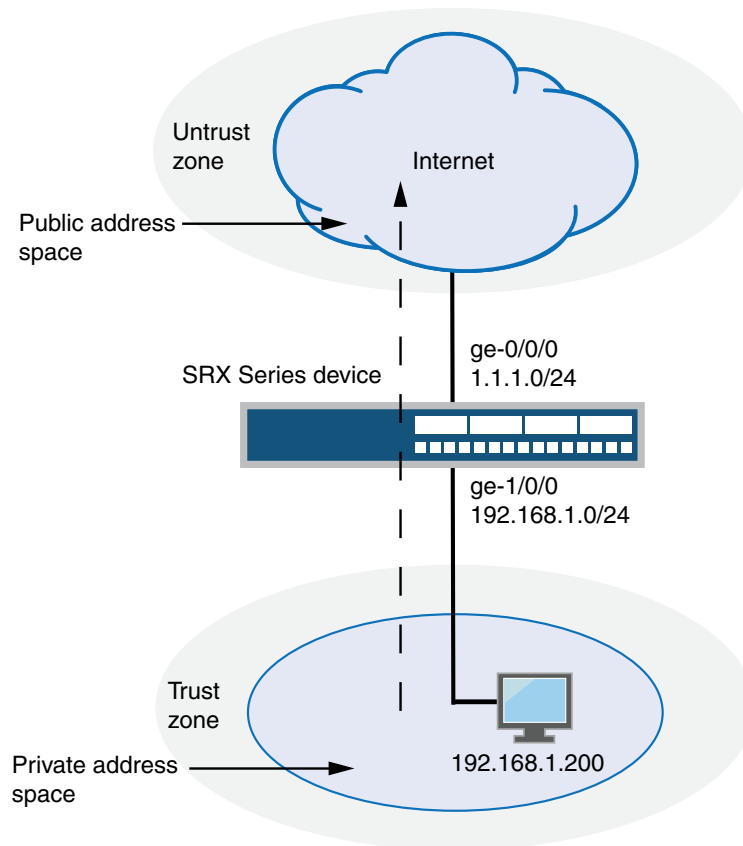
Before you begin:

1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 113.

**Overview**

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 131 on page 1380, a device with the private address 192.168.1.200 in the trust zone accesses a public network. For packets sent by the device to a destination address in the untrust zone, the Juniper Networks security device translates the source IP address to the public IP address 1.1.1.200/32.

Figure 131: Source NAT Single Address Translation



Original Source IP	Translated Source IP
192.168.1.200/32	1.1.1.200/32

g030669

This example describes the following configurations:

- Source NAT pool **src-nat-pool-1** that contains the IP address `1.1.1.200/32`.
- Source NAT rule set **rs1** with rule **r1** to match packets from the trust zone to the untrust zone with the source IP address `192.168.1.200/32`. For matching packets, the source address is translated to the IP address in **src-nat-pool-1** pool.
- Proxy ARP for the address `1.1.1.200` on interface `ge-0/0/0.0`. This allows the Juniper Networks security device to respond to ARP requests received on the interface for that address.
- Security policies to permit traffic from the trust zone to the untrust zone.

**Configuration**

**CLI Quick Configuration** To quickly configure a source NAT mapping for a single IP address, copy the following commands and paste them into the CLI.

```
[edit]
set security nat source pool src-nat-pool-1 address 1.1.1.200/32
set security nat source rule-set rs1 from zone trust
set security nat source rule-set rs1 to zone untrust
set security nat source rule-set rs1 rule r1 match source-address 192.168.1.200/32
set security nat source rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.200/32
set security policies from-zone trust to-zone untrust policy internet-access match
  source-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  destination-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  application any
set security policies from-zone trust to-zone untrust policy internet-access then permit
```

**Step-by-Step Procedure** The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a source NAT translation for a single IP address:

1. Create a source NAT pool.

```
[edit security nat source]
user@host# set pool src-nat-pool-1 address 1.1.1.200/32
```

2. Create a source NAT rule set.

```
[edit security nat source]
user@host# set rule-set rs1 from zone trust
user@host# set rule-set rs1 to zone untrust
```

3. Configure a rule that matches packets and translates the source address to the address in the pool.

```
[edit security nat source]
user@host# set rule-set rs1 rule r1 match source-address 192.168.1.200/32
user@host# set rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
```

4. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.200
```

5. Configure a security policy that allows traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy internet-access match source-address any
  destination-address any application any
user@host# set policy internet-access then permit
```



**Verification**

To confirm that the configuration is working properly, perform these tasks:

- Verifying Source NAT Pool Usage on page 1383
- Verifying Source NAT Rule Usage on page 1383
- Verifying NAT Application to Traffic on page 1383

**Verifying Source NAT Pool Usage**

**Purpose** Verify that there is traffic using IP addresses from the source NAT pool.

**Action** From operational mode, enter the **show security nat source pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

**Verifying Source NAT Rule Usage**

**Purpose** Verify that there is traffic matching the source NAT rule.

**Action** From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

**Verifying NAT Application to Traffic**

**Purpose** Verify that NAT is being applied to the specified traffic.

**Action** From operational mode, enter the **show security flow session** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding Source NAT on page 1369
  - Source NAT Configuration Overview on page 1374

**Example: Configuring Source NAT for Multiple Addresses with PAT**

This example describes how to configure a source NAT mapping of a private address block to a smaller public address block using port address translation.

- Requirements on page 1383
- Overview on page 1384
- Configuration on page 1386
- Verification on page 1388

**Requirements**

Before you begin:

1. Configure network interfaces on the device. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).
2. Create security zones and assign interfaces to them. See "Understanding Security Zones" on page 113.

**Overview**

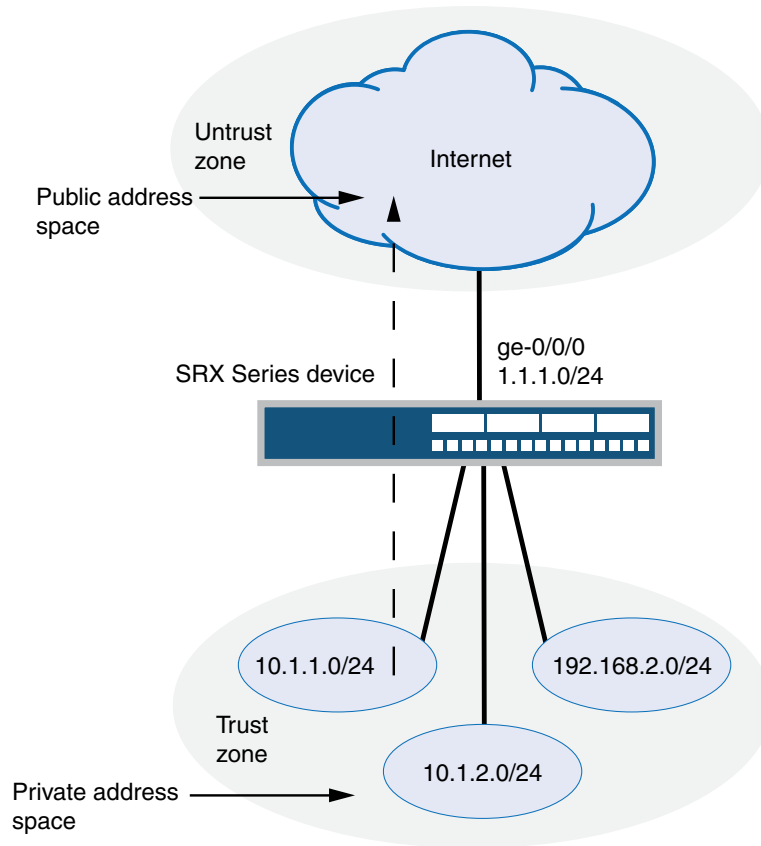
This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 132 on page 1385, the source IP address in packets sent from the trust zone to the untrust zone is mapped to a smaller block of public addresses in the range from 1.1.1.1/32 through 1.1.1.24/32. Because the size of the source NAT address pool is smaller than the number of potential addresses that might need to be translated, port address translation is used.



**NOTE:** Port address translation includes a source port number with the source IP address mapping. This allows multiple addresses on a private network to map to a smaller number of public IP addresses. Port address translation is enabled by default for source NAT pools.

---

Figure 132: Source NAT Multiple Addresses with PAT



Original Source IP	Translated Source IP
10.1.1.0/24	1.1.1.1 (with port address translation)
10.1.2.0/24	
192.168.1.0/24	

g030670

This example describes the following configurations:

- Source NAT pool **src-nat-pool-1** that contains the IP address range `1.1.1.0/32` through `1.1.1.24/32`.
- Source NAT rule set **rs1** to match all packets from the trust zone to the untrust zone. For matching packets, the source IP address is translated to an IP address in the **src-nat-pool-1** pool.

- Proxy ARP for the addresses 1.1.1/32 through 1.1.1.24/32 on interface ge-0/0/0.0. This allows the Juniper Networks security device to respond to ARP requests received on the interface for those addresses.
- Security policies to permit traffic from the trust zone to the untrust zone.

### Configuration

**CLI Quick Configuration** To quickly configure a source NAT mapping from a private address block to a smaller public address block using PAT, copy the following commands and paste them into the CLI.

```
[edit]
set security nat source pool src-nat-pool-1 address 1.1.1/32 to 1.1.1.24/32
set security nat source rule-set rs1 from zone trust
set security nat source rule-set rs1 to zone untrust
set security nat source rule-set rs1 rule r1 match source-address 10.1.1.0/24
set security nat source rule-set rs1 rule r1 match source-address 10.1.2.0/24
set security nat source rule-set rs1 rule r1 match source-address 192.168.1.0/24
set security nat source rule-set rs1 rule r1 match destination-address 0.0.0.0/0
set security nat source rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1/32 to 1.1.1.24/32
set security policies from-zone trust to-zone untrust policy internet-access match
  source-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  destination-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  application any
set security policies from-zone trust to-zone untrust policy internet-access then permit
```

**Step-by-Step Procedure** The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode.

To configure a source NAT mapping from a private address block to a smaller public address block using PAT:

1. Create a source NAT pool.

```
[edit security nat source]
user@host# set pool src-nat-pool-1 address 1.1.1 to 1.1.1.24
```

2. Create a source NAT rule set.

```
[edit security nat source]
user@host# set rule-set rs1 from zone trust
user@host# set rule-set rs1 to zone untrust
```

3. Configure a rule that matches packets and translates the source address to an address in the pool.

```
[edit security nat source]
user@host# set rule-set rs1 rule r1 match source-address [10.1.1.0/24 10.1.2.0/24
  192.168.1.0/24]
user@host# set rule-set rs1 rule r1 match destination-address 0.0.0.0/0
user@host# set rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
```



4. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.1 to 1.1.1.24
```

5. Configure a security policy that allows traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy internet-access match source-address any
destination-address any application any
user@host# set policy internet-access then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
source {
  pool src-nat-pool-1 {
    address {
      1.1.1.1/32 to 1.1.1.24/32;
    }
  }
}
rule-set rs1 {
  from zone trust;
  to zone untrust;
  rule r1 {
    match {
      source-address [10.1.1.0/24 10.1.2.0/24 192.168.1.0/24];
      destination-address 0.0.0.0/0;
    }
    then {
      source-nat {
        pool {
          src-nat-pool-1;
        }
      }
    }
  }
}
}
}
proxy-arp {
  interface ge-0/0/0.0 {
    address {
      1.1.1.1/32 to 1.1.1.24/32;
    }
  }
}
}
user@host# show security policies
from-zone trust to-zone untrust {
  policy internet-access {
    match {
      source-address any;
      destination-address any;
      application any;
    }
  }
}
```

```
    }  
    then {  
      permit;  
    }  
  }  
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### **Verification**

To confirm that the configuration is working properly, perform these tasks:

- Verifying Source NAT Pool Usage on page 1388
- Verifying Source NAT Rule Usage on page 1388
- Verifying NAT Application to Traffic on page 1388

#### **Verifying Source NAT Pool Usage**

**Purpose** Verify that there is traffic using IP addresses from the source NAT pool.

**Action** From operational mode, enter the **show security nat source pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

#### **Verifying Source NAT Rule Usage**

**Purpose** Verify that there is traffic matching the source NAT rule.

**Action** From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

#### **Verifying NAT Application to Traffic**

**Purpose** Verify that NAT is being applied to the specified traffic.

**Action** From operational mode, enter the **show security flow session** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding Source NAT on page 1369
  - Source NAT Configuration Overview on page 1374
  - Understanding Source NAT Pools with PAT on page 1371

### **Example: Configuring Source NAT for Multiple Addresses without PAT**

---

This example describes how to configure a source NAT mapping of a private address block to a smaller public address block without port address translation.



**NOTE:** Port address translation is enabled by default for source NAT pools. When port address translation is disabled, the number of translations that the source NAT pool can concurrently support is limited to the number of addresses in the pool. Packets are dropped if there are no addresses available in the source NAT pool. You can optionally specify an overflow pool from which IP addresses and port numbers are allocated when there are no addresses available in the original source NAT pool.

- Requirements on page 1389
- Overview on page 1389
- Configuration on page 1391
- Verification on page 1393

### **Requirements**

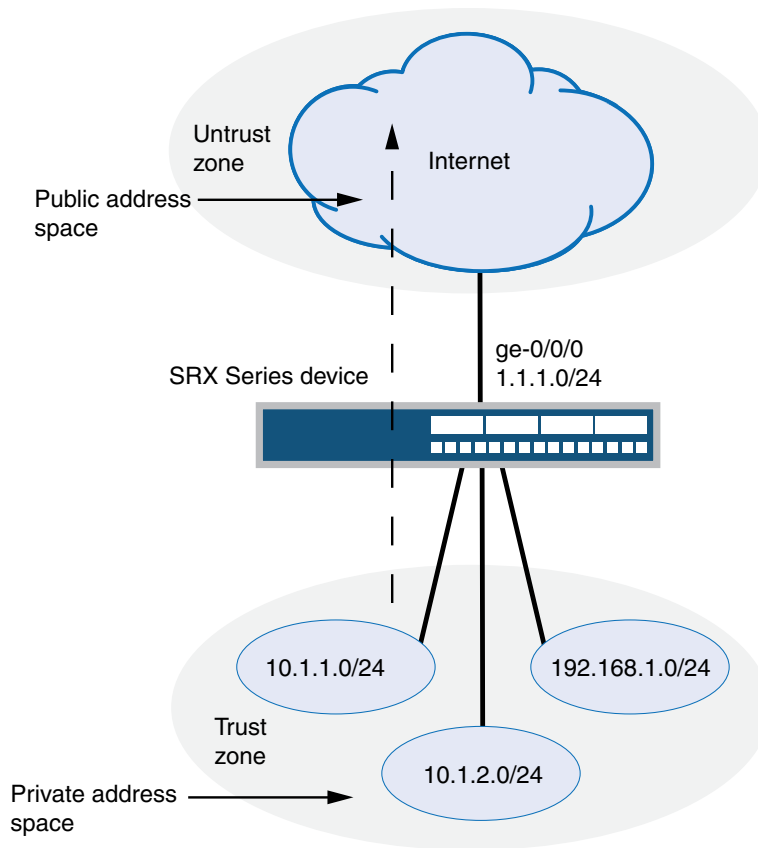
Before you begin:

1. Configure network interfaces on the device. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 113.

### **Overview**

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 133 on page 1390, the source IP address in packets sent from the trust zone to the untrust zone is mapped to a smaller block of public addresses in the range from 1.1.1/32 through 1.1.1.24/32.

Figure 133: Source NAT Multiple Addresses without PAT



Original Source IP	Translated Source IP
10.1.1.0/24 10.1.2.0/24 192.168.1.0/24	1.1.1.1 (no port address translation)

g030671

This example describes the following configurations:

- Source NAT pool **src-nat-pool-1** that contains the IP address range 1.1.1.1/32 through 1.1.1.24/32. The **port no-translation** option is specified for the pool.
- Source NAT rule set **rs1** to match all packets from the trust zone to the untrust zone. For matching packets, the source IP address is translated to an IP address in the **src-nat-pool-1** pool.

- Proxy ARP for the addresses 1.1.1/32 through 1.1.1.24/32 on interface ge-0/0/0.0. This allows the Juniper Networks security device to respond to ARP requests received on the interface for those addresses.
- Security policies to permit traffic from the trust zone to the untrust zone.

### Configuration

#### CLI Quick Configuration

To quickly configure a source NAT mapping from a private address block to a smaller public address block without PAT, copy the following commands and paste them into the CLI.

```
[edit]
set security nat source pool src-nat-pool-1 address 1.1.1/32 to 1.1.1.24/32
set security nat source pool src-nat-pool-1 port no-translation
set security nat source rule-set rs1 from zone trust
set security nat source rule-set rs1 to zone untrust
set security nat source rule-set rs1 rule r1 match source-address 0.0.0.0/0
set security nat source rule-set rs1 rule r1 match destination-address 0.0.0.0/0
set security nat source rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1/32 to 1.1.1.24/32
set security policies from-zone trust to-zone untrust policy internet-access match
  source-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  destination-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  application any
set security policies from-zone trust to-zone untrust policy internet-access then permit
```

#### Step-by-Step Procedure

The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a source NAT mapping from a private address block to a smaller public address block without PAT:

1. Create a source NAT pool.

```
[edit security nat source]
user@host# set pool src-nat-pool-1 address 1.1.1 to 1.1.1.24
```

2. Specify the **port no-translation** option.

```
[edit security nat source]
user@host# set pool src-nat-pool-1 port no-translation
```

3. Create a source NAT rule set.

```
[edit security nat source]
user@host# set rule-set rs1 from zone trust
user@host# set rule-set rs1 to zone untrust
```

4. Configure a rule that matches packets and translates the source address to an address in the pool.

```
[edit security nat source]
user@host# set rule-set rs1 rule r1 match source-address 0.0.0.0/0
user@host# set rule-set rs1 rule r1 match destination-address 0.0.0.0/0
```

```
user@host# set rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
```

5. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.1 to 1.1.1.24
```

6. Configure a security policy that allows traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy internet-access match source-address any
destination-address any application any
user@host# set policy internet-access then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
source {
  pool src-nat-pool-1 {
    address {
      1.1.1.1/32 to 1.1.1.24/32;
    }
    port no-translation;
  }
  rule-set rs1 {
    from zone trust;
    to zone untrust;
    rule r1 {
      match {
        source-address 0.0.0.0/0;
        destination-address 0.0.0.0/0;
      }
      then {
        source-nat {
          pool {
            src-nat-pool-1;
          }
        }
      }
    }
  }
}
proxy-arp {
  interface ge-0/0/0.0 {
    address {
      1.1.1.1/32 to 1.1.1.24/32;
    }
  }
}
user@host# show security policies
from-zone trust to-zone untrust {
  policy internet-access {
    match {
```

```

        source-address any;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### **Verification**

To confirm that the configuration is working properly, perform these tasks:

- Verifying Source NAT Pool Usage on page 1393
- Verifying Source NAT Rule Usage on page 1393
- Verifying NAT Application to Traffic on page 1393

#### **Verifying Source NAT Pool Usage**

**Purpose** Verify that there is traffic using IP addresses from the source NAT pool.

**Action** From operational mode, enter the **show security nat source pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

#### **Verifying Source NAT Rule Usage**

**Purpose** Verify that there is traffic matching the source NAT rule.

**Action** From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

#### **Verifying NAT Application to Traffic**

**Purpose** Verify that NAT is being applied to the specified traffic.

**Action** From operational mode, enter the **show security flow session** command.

### **Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Source NAT on page 1369
- Source NAT Configuration Overview on page 1374
- Understanding Source NAT Pools Without PAT on page 1372

### **Example: Configuring Source NAT with Address Shifting**

This example describes how to configure a source NAT mapping of a private address range to public addresses, with optional address shifting. This mapping is one-to-one between the original source IP addresses and translated IP addresses and no port translation is performed.



**NOTE:** The match conditions for a source NAT rule set do not allow you to specify an address range; only address prefixes may be specified in a rule. When configuring a source NAT pool, you can specify the `host-base-address` option; this option specifies the IP address where the original source IP address range begins.

The range of original source IP addresses that are translated is determined by the number of addresses in the source NAT pool. For example, if the source NAT pool contains a range of ten IP addresses, then up to ten original source IP addresses can be translated, starting with a specified base address.

The match condition in a source NAT rule may define a larger address range than that specified in the source NAT pool. For example, a match condition might specify an address prefix that contains 256 addresses, but the source NAT pool contains a range of only ten IP addresses. A packet's source IP address can match a source NAT rule, but if the source IP address is not within the address range specified in the source NAT pool, the source IP address is not translated.

- 
- Requirements on page 1394
  - Overview on page 1394
  - Configuration on page 1396
  - Verification on page 1398

### **Requirements**

Before you begin:

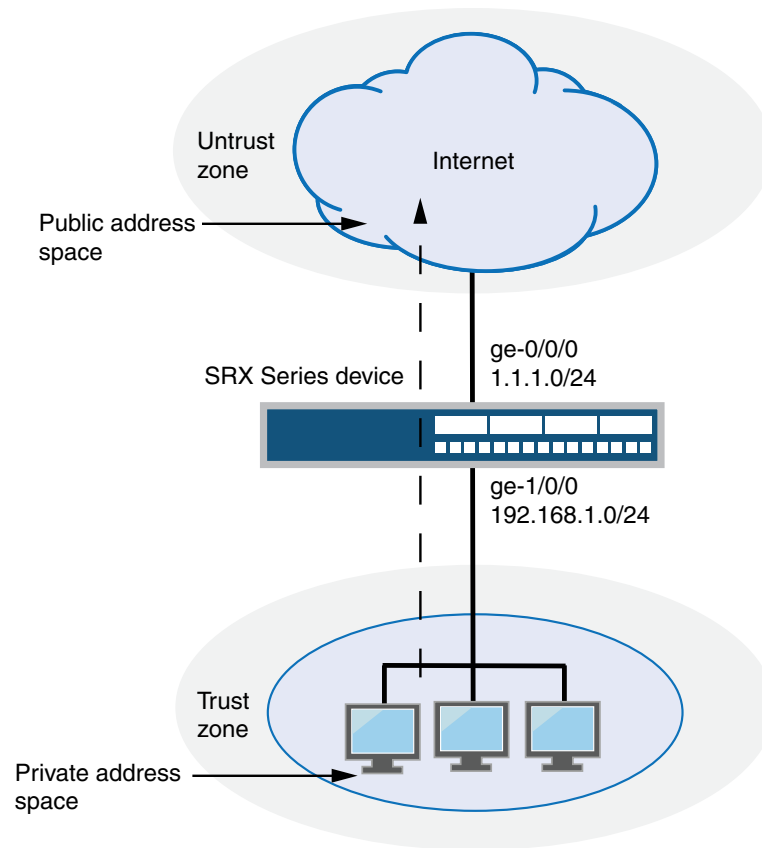
1. Configure network interfaces on the device. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 113.

### **Overview**

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 134 on page 1395, a range of private addresses in the trust zone is mapped to a range of public addresses in the untrust zone. For packets sent from the trust zone to the untrust zone, a source IP address in the range of 192.168.1.10/32 through 192.168.1.20/32 is translated to a public address in the range of 1.1.1.30/32 through 1.1.1.40/32.



Figure 134: Source NAT with Address Shifting



Original Source IP	Translated Source IP
192.168.1.10/32 - 192.168.1.20/32	1.1.1.30/32 - 1.1.1.40/32

g030672

This example describes the following configurations:

- Source NAT pool **src-nat-pool-1** that contains the IP address range 1.1.1.30/32 through 1.1.1.40/32. For this pool, the beginning of the original source IP address range is 192.168.1.10/32 and is specified with the **host-address-base** option.
- Source NAT rule set **rs1** with rule **r1** to match packets from the trust zone to the untrust zone with a source IP address in the 192.168.1.0/24 subnet. For matching packets that fall within the source IP address range specified by the **src-nat-pool-1** configuration, the source address is translated to the IP address in **src-nat-pool-1** pool.

- Proxy ARP for the addresses 1.1.1.30/32 through 1.1.1.40/32 on interface ge-0/0/0.0. This allows the Juniper Networks security device to respond to ARP requests received on the interface for that address.
- Security policies to permit traffic from the trust zone to the untrust zone.

### Configuration

**CLI Quick Configuration** To quickly configure a source NAT mapping with address shifting, copy the following commands and paste them into the CLI.

```
[edit]
set security nat source pool src-nat-pool-1 address 1.1.1.30/32 to 1.1.1.40/32
set security nat source pool src-nat-pool-1 host-address-base 192.168.1.10/32
set security nat source rule-set rs1 from zone trust
set security nat source rule-set rs1 to zone untrust
set security nat source rule-set rs1 rule r1 match source-address 192.168.1.0/24
set security nat source rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.30/32 to 1.1.1.40/32
set security policies from-zone trust to-zone untrust policy internet-access match
  source-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  destination-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  application any
set security policies from-zone trust to-zone untrust policy internet-access then permit
```

**Step-by-Step Procedure** The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a source NAT mapping with address shifting:

1. Create a source NAT pool.
 

```
[edit security nat source]
user@host# set pool src-nat-pool-1 address 1.1.1.30/32 to 1.1.1.40/32
```
2. Specify the beginning of the original source IP address range.
 

```
[edit security nat source]
user@host# set pool src-nat-pool-1 host-address-base 192.168.1.10/32
```
3. Create a source NAT rule set.
 

```
[edit security nat source]
user@host# set rule-set rs1 from zone trust
user@host# set rule-set rs1 to zone untrust
```
4. Configure a rule that matches packets and translates the source address to an address in the pool.
 

```
[edit security nat source]
user@host# set rule-set rs1 rule r1 match source-address 192.168.1.0/24
user@host# set rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
```
5. Configure proxy ARP.
 

```
[edit security nat]
```

```
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.30/32 to 1.1.1.40/32
```

6. Configure a security policy that allows traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy internet-access match source-address any
destination-address any application any
user@host# set policy internet-access then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
source {
  pool src-nat-pool-1 {
    address {
      1.1.1.30/32 to 1.1.1.40/32;
    }
    host-address-base 192.168.1.10/32;
  }
  rule-set rs1 {
    from zone trust;
    to zone untrust;
    rule r1 {
      match {
        source-address 192.168.1.0/24;
      }
      then {
        source-nat {
          pool {
            src-nat-pool-1;
          }
        }
      }
    }
  }
}
proxy-arp {
  interface ge-0/0/0.0 {
    address {
      1.1.1.30/32 to 1.1.1.40/32;
    }
  }
}
user@host# show security policies
from-zone trust to-zone untrust {
  policy internet-access {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
```

```

    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### **Verification**

To confirm that the configuration is working properly, perform these tasks:

- Verifying Source NAT Pool Usage on page 1398
- Verifying Source NAT Rule Usage on page 1398
- Verifying NAT Application to Traffic on page 1398

#### **Verifying Source NAT Pool Usage**

**Purpose** Verify that there is traffic using IP addresses from the source NAT pool.

**Action** From operational mode, enter the **show security nat source pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

#### **Verifying Source NAT Rule Usage**

**Purpose** Verify that there is traffic matching the source NAT rule.

**Action** From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

#### **Verifying NAT Application to Traffic**

**Purpose** Verify that NAT is being applied to the specified traffic.

**Action** From operational mode, enter the **show security flow session** command.

#### **Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Source NAT on page 1369
- Source NAT Configuration Overview on page 1374
- Understanding Source NAT Pools with Address Shifting on page 1372

#### **Example: Configuring Source NAT with Multiple Rules**

---

This example describes how to configure source NAT mappings with multiple rules.

- Requirements on page 1399
- Overview on page 1399
- Configuration on page 1401
- Verification on page 1404

**Requirements**

Before you begin:

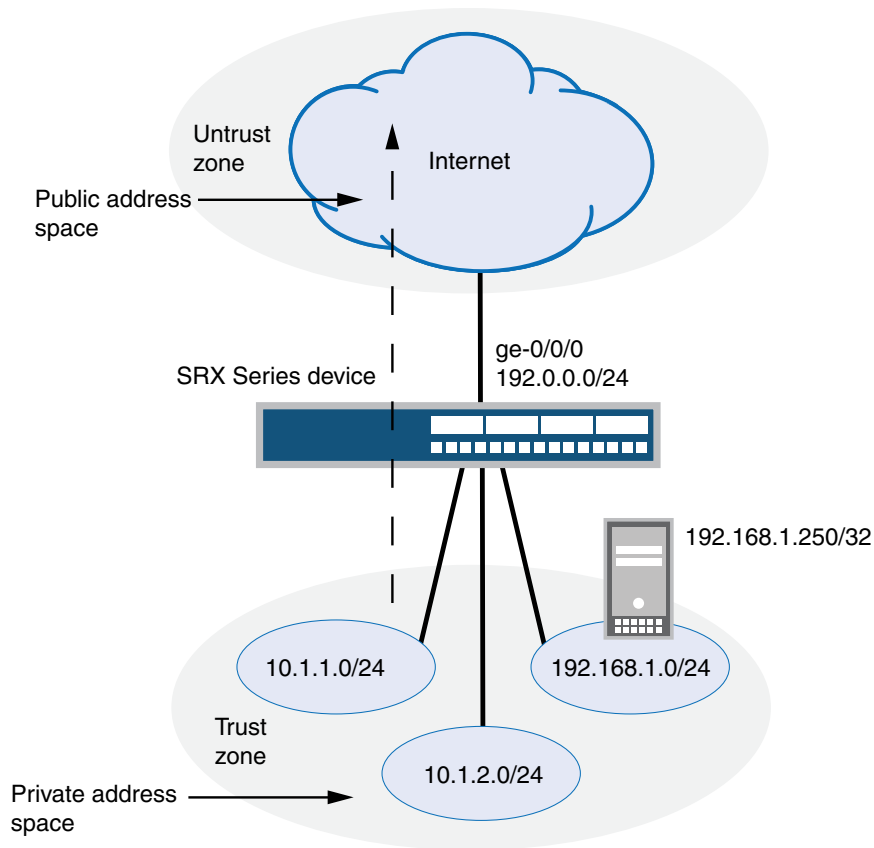
1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 113.

**Overview**

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 135 on page 1400, the following translations are performed on the Juniper Networks security device for the source NAT mapping for traffic from the trust zone to the untrust zones:

- The source IP address in packets sent by the 10.1.1.0/24 and 10.1.2.0/24 subnets to any address in the untrust zone is translated to a public address in the range from 192.0.0.1 to 192.0.0.24 with port translation.
- The source IP address in packets sent by the 192.168.1.0/24 subnet to any address in the untrust zone is translated to a public address in the range from 192.0.0.100 to 192.0.0.249 with no port translation.
- The source IP address in packets sent by the 192.168.1.250/32 host device is not translated.

Figure 135: Source NAT with Multiple Translation Rules



Original Source IP	Translated Source IP
10.1.1.0/24, 10.1.2.0/24	192.0.0.1 - 192.0.0.24 (w/port translation)
192.168.1.0/24	192.0.0.100 - 192.0.0.249 (no port translation)
192.168.1.250/32	(no source NAT translation)

g030673

This example describes the following configurations:

- Source NAT pool **src-nat-pool-1** that contains the IP address range 192.0.0.1 through 192.0.0.24.
- Source NAT pool **src-nat-pool-2** that contains the IP address range 192.0.0.100 through 192.0.0.249, with port address translation disabled.



**NOTE:** When port address translation is disabled, the number of translations that the source NAT pool can support concurrently is limited to the number of addresses in the pool. Packets are dropped if there are no addresses available in the source NAT pool. You can optionally specify an overflow pool from which IP addresses and port numbers are allocated when there are no addresses available in the original source NAT pool.

- Source NAT rule set **rs1** to match packets from the trust zone to the untrust zone. Rule set **rs1** contains multiple rules:
  - Rule **r1** to match packets with a source IP address in either the 10.1.1.0/24 or 10.1.2.0/24 subnets. For matching packets, the source address is translated to an IP address in the **src-nat-pool-1** pool.
  - Rule **r2** to match packets with a source IP address of 192.168.1.250/32. For matching packets, there is no NAT translation performed.
  - Rule **r3** to match packets with a source IP address in the 192.168.1.0/24 subnet. For matching packets, the source address is translated to an IP address in the **src-nat-pool-2** pool.



**NOTE:** The order of rules in a rule set is important, as the first rule in the rule set that matches the traffic is used. Therefore, rule **r2** to match a specific IP address must be placed before rule **r3** that matches the subnet on which the device is located.

- Proxy ARP for the addresses 192.0.0.1 through 192.0.0.24 and 192.0.0.100 through 192.0.0.249 on interface ge-0/0/0.0. This allows the Juniper Networks security device to respond to ARP requests received on the interface for those addresses.
- Security policies to permit traffic from the trust zone to the untrust zone.

### Configuration

**CLI Quick Configuration** To quickly configure a source NAT mapping with multiple rules, copy the following commands and paste them into the CLI.

```
[edit]
set security nat source pool src-nat-pool-1 address 192.0.0.1/32 to 192.0.0.24/32
set security nat source pool src-nat-pool-2 address 192.0.0.100/32 to 192.0.0.249/32
set security nat source pool src-nat-pool-2 port no-translation
set security nat source rule-set rs1 from zone trust
set security nat source rule-set rs1 to zone untrust
set security nat source rule-set rs1 rule r1 match source-address 10.1.1.0/24
set security nat source rule-set rs1 rule r1 match source-address 10.1.2.0/24
set security nat source rule-set rs1 rule r1 match destination-address 0.0.0.0/0
set security nat source rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
set security nat source rule-set rs1 rule r2 match source-address 192.168.1.250/32
set security nat source rule-set rs1 rule r2 match destination-address 0.0.0.0/0
set security nat source rule-set rs1 rule r2 then source-nat off
```

```

set security nat source rule-set rs1 rule r3 match source-address 192.168.1.0/24
set security nat source rule-set rs1 rule r3 match destination-address 0.0.0.0/0
set security nat source rule-set rs1 rule r3 then source-nat pool src-nat-pool-2
set security nat proxy-arp interface ge-0/0/0.0 address 192.0.0.1/32 to 192.0.0.24/32
set security nat proxy-arp interface ge-0/0/0.0 address 192.0.0.100/32 to 192.0.0.249/32
set security policies from-zone trust to-zone untrust policy internet-access match
  source-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  destination-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  application any
set security policies from-zone trust to-zone untrust policy internet-access then permit

```

**Step-by-Step Procedure** The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure multiple source NAT rules in a rule set:

1. Create a source NAT pool.

```

[edit security nat source]
user@host# set pool src-nat-pool-1 address 192.0.0.1 to 192.0.0.24

```

2. Create a source NAT pool with no port translation.

```

[edit security nat source]
user@host# set pool src-nat-pool-2 address 192.0.0.100 to 192.0.0.249
user@host# set pool src-nat-pool-2 port no-translation

```



**NOTE:** To configure an overflow pool for src-nat-pool-2 using the egress interface:

```

[edit security nat source]
user@host# set pool src-nat-pool-2 overflow-pool interface

```

3. Create a source NAT rule set.

```

[edit security nat source]
user@host# set rule-set rs1 from zone trust
user@host# set rule-set rs1 to zone untrust

```

4. Configure a rule that matches packets and translates the source address to an address in the pool.

```

[edit security nat source]
user@host# set rule-set rs1 rule r1 match source-address [10.1.1.0/24 10.1.2.0/24]
user@host# set rule-set rs1 rule r1 match destination-address 0.0.0.0/0
user@host# set rule-set rs1 rule r1 then source-nat pool src-nat-pool-1

```

5. Configure a rule to match packets for which the source address is not translated.

```

[edit security nat source]
user@host# set rule-set rs1 rule r2 match source-address 192.168.1.250/32
user@host# set rule-set rs1 rule r2 match destination-address 0.0.0.0/0
user@host# set rule-set rs1 rule r2 then source-nat off

```



- Configure a rule to match packets and translate the source address to an address in the pool with no port translation.

```
[edit security nat source]
user@host# set rule-set rs1 rule r3 match source-address 192.168.1.0/24
user@host# set rule-set rs1 rule r3 match destination-address 0.0.0.0/0
user@host# set rule-set rs1 rule r3 then source-nat pool src-nat-pool-2
```

- Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 192.0.0.1 to 192.0.0.24
user@host# set proxy-arp interface ge-0/0/0.0 address 192.0.0.100 to 192.0.0.249
```

- Configure a security policy that allows traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy internet-access match source-address any
destination-address any application any
user@host# set policy internet-access then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
source {
  pool src-nat-pool-1 {
    address {
      192.0.0.1/32 to 192.0.0.24/32;
    }
  }
  pool src-nat-pool-2 {
    address {
      192.0.0.100/32 to 192.0.0.249/32;
    }
    port no-translation;
  }
}
rule-set rs1 {
  from zone trust;
  to zone untrust;
  rule r1 {
    match {
      source-address [ 10.1.1.0/24 10.1.2.0/24 ];
      destination-address 0.0.0.0/0;
    }
    then {
      source-nat {
        pool {
          src-nat-pool-1;
        }
      }
    }
  }
}
rule r2 {
  match {
```



**Verifying Source NAT Pool Usage**

**Purpose** Verify that there is traffic using IP addresses from the source NAT pool.

**Action** From operational mode, enter the **show security nat source pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

**Verifying Source NAT Rule Usage**

**Purpose** Verify that there is traffic matching the source NAT rule.

**Action** From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

**Verifying NAT Application to Traffic**

**Purpose** Verify that NAT is being applied to the specified traffic.

**Action** From operational mode, enter the **show security flow session** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding Source NAT on page 1369
  - Source NAT Configuration Overview on page 1374
  - Understanding Source NAT Rules on page 1373

**Example: Configuring Source and Destination NAT Translations**

This example describes how to configure both source and destination NAT mappings.

- Requirements on page 1405
- Overview on page 1405
- Configuration on page 1407
- Verification on page 1410

**Requirements**

Before you begin:

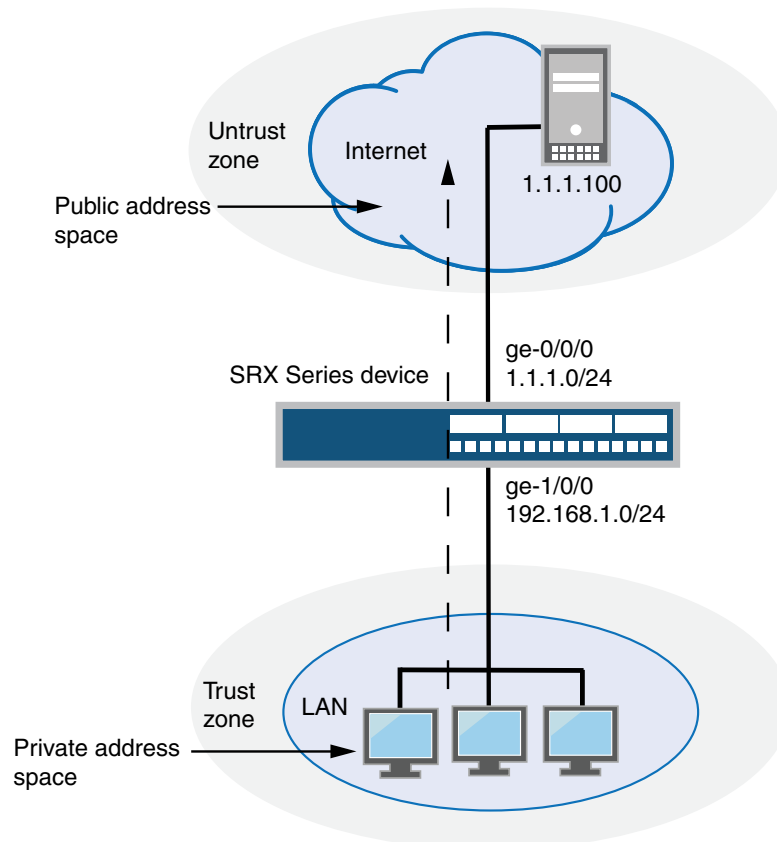
1. Configure network interfaces on the device. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 113.

**Overview**

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 136 on page 1406, the following translations are performed on the Juniper Networks security device:

- The source IP address in packets sent by the device with the private address 192.168.1.200 in the trust zone to any address in the untrust zone is translated to a public address in the range from 1.1.1.10 through 1.1.1.14.
- The destination IP address 1.1.1.100/32 in packets sent from the trust zone to the untrust zone is translated to the address 10.1.1.200/32.

Figure 136: Source and Destination NAT Translations



Original Source IP 192.168.1.0/24	Translated Source IP 1.1.1.10 - 1.1.1.14
Original Destination IP 1.1.1.100/32	Translated Destination IP 10.1.1.200/32

g030674

This example describes the following configurations:

- Source NAT pool **src-nat-pool-1** that contains the IP address range 1.1.1.10 through 1.1.1.14.
- Source NAT rule set **rs1** with rule **r1** to match any packets from the trust zone to the untrust zone. For matching packets, the source address is translated to an IP address in the **src-nat-pool-1** pool.

- Destination NAT pool **dst-nat-pool-1** that contains the IP address 10.1.1.200/32.
- Destination NAT rule set **rs1** with rule **r1** to match packets from the trust zone with the destination IP address 1.1.1.100. For matching packets, the destination address is translated to the IP address in the **dst-nat-pool-1** pool.
- Proxy ARP for the addresses 1.1.1.10 through 1.1.1.14 and 1.1.1.100/32 on interface ge-0/0/0.0. This allows the Juniper Networks security device to respond to ARP requests received on the interface for those addresses.
- Security policy to permit traffic from the trust zone to the untrust zone.
- Security policy to permit traffic from the untrust zone to the translated destination IP addresses in the trust zone.

### Configuration

**CLI Quick Configuration** To quickly configure source and destination NAT mappings, copy the following commands and paste them into the CLI.

```
[edit]
set security nat source pool src-nat-pool-1 address 1.1.1.10/32 to 1.1.1.14/32
set security nat source rule-set rs1 from zone trust
set security nat source rule-set rs1 to zone untrust
set security nat source rule-set rs1 rule r1 match source-address 0.0.0.0/0
set security nat source rule-set rs1 rule r1 match destination-address 0.0.0.0/0
set security nat source rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
set security nat destination pool dst-nat-pool-1 address 10.1.1.200/32
set security nat destination rule-set rs1 from zone untrust
set security nat destination rule-set rs1 rule r1 match destination-address 1.1.1.100/32
set security nat destination rule-set rs1 rule r1 then destination-nat pool dst-nat-pool-1
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.10/32 to 1.1.1.24/32
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.100/32
set security policies from-zone trust to-zone untrust policy internet-access match
  source-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  destination-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  application any
set security policies from-zone trust to-zone untrust policy internet-access then permit
set security zones security-zone trust address-book address dst-nat-pool-1 10.1.1.200/32
set security policies from-zone untrust to-zone trust policy dst-nat-pool-1-access match
  source-address any
set security policies from-zone untrust to-zone trust policy dst-nat-pool-1-access match
  destination-address dst-nat-pool-1
set security policies from-zone untrust to-zone trust policy dst-nat-pool-1-access match
  application any
set security policies from-zone untrust to-zone trust policy dst-nat-pool-1-access then
  permit
```

**Step-by-Step Procedure** The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode.

To configure the source and destination NAT translations:

1. Create a source NAT pool.

```
[edit security nat source]
user@host# set pool src-nat-pool-1 address 1.1.1.10 to 1.1.1.14
```

2. Create a source NAT rule set.

```
[edit security nat source]
user@host# set rule-set rs1 from zone trust
user@host# set rule-set rs1 to zone untrust
```

3. Configure a rule that matches packets and translates the source address to an address in the source NAT pool.

```
[edit security nat source]
user@host# set rule-set rs1 rule r1 match source-address 0.0.0.0/0
user@host# set rule-set rs1 rule r1 match destination-address 0.0.0.0/0
user@host# set rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
```

4. Create a destination NAT pool.

```
[edit security nat destination]
user@host# set pool dst-nat-pool-1 address 10.1.1.200/32
```

5. Create a destination NAT rule set.

```
[edit security nat destination]
user@host# set rule-set rs1 from zone untrust
```

6. Configure a rule that matches packets and translates the destination address to the address in the destination NAT pool.

```
[edit security nat destination]
user@host# set rule-set rs1 rule r1 match destination-address 1.1.1.100/32
user@host# set rule-set rs1 rule r1 then destination-nat pool dst-nat-pool-1
```

7. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.10 to 1.1.1.14
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.100
```

8. Configure a security policy that allows traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy internet-access match source-address any
destination-address any application any
user@host# set policy internet-access then permit
```

9. Configure an address book entry in the trust zone for the translated destination IP address.

```
[edit security]
user@host# set zones security-zone trust address-book address dst-nat-pool-1
10.1.1.200/32
```

10. Configure a security policy that allows traffic from the untrust zone to the trust zone.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy dst-nat-pool-1-access match source-address any
destination-address dst-nat-pool-1 application any
user@host# set policy dst-nat-pool-1-access then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
source {
  pool src-nat-pool-1 {
    address {
      1.1.1.10/32 to 1.1.1.14/32;
    }
  }
}
rule-set rs1 {
  to zone untrust;
  rule r1 {
    match {
      source-address 0.0.0.0/0;
      destination-address 0.0.0.0/0;
    }
    then {
      source-nat {
        pool {
          src-nat-pool-1;
        }
      }
    }
  }
}
}
}
destination {
  pool dst-nat-pool-1 {
    address 10.1.1.200/32;
  }
  rule-set rs1 {
    from zone untrust;
    rule r1 {
      match {
        destination-address 1.1.1.100/32;
      }
      then {
        destination-nat pool dst-nat-pool-1;
      }
    }
  }
}
}
proxy-arp {
  interface ge-0/0/0.0 {
    address {
```

```

        1.1.1.10/32 to 1.1.1.24/32;
        1.1.1.100/32;
    }
}
}
user@host# show security policies
from-zone trust to-zone untrust {
  policy internet-access {
    match {
      source-address any;
      destination-address any;
      application any;
    }
  }
  policy internet-access {
    then {
      permit;
    }
  }
}
from-zone untrust to-zone trust {
  policy dst-nat-pool-1-access {
    match {
      source-address any;
      destination-address dst-nat-pool-1;
      application any;
    }
    then {
      permit;
    }
  }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### **Verification**

To confirm that the configuration is working properly, perform these tasks:

- Verifying Source NAT Pool Usage on page 1410
- Verifying Source NAT Rule Usage on page 1411
- Verifying Destination NAT Pool Usage on page 1411
- Verifying Destination NAT Rule Usage on page 1411
- Verifying NAT Application to Traffic on page 1411

### **Verifying Source NAT Pool Usage**

**Purpose** Verify that there is traffic using IP addresses from the source NAT pool.

**Action** From operational mode, enter the **show security nat source pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.



**Verifying Source NAT Rule Usage**

**Purpose** Verify that there is traffic matching the source NAT rule.

**Action** From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

**Verifying Destination NAT Pool Usage**

**Purpose** Verify that there is traffic using IP addresses from the destination NAT pool.

**Action** From operational mode, enter the **show security nat destination pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

**Verifying Destination NAT Rule Usage**

**Purpose** Verify that there is traffic matching the destination NAT rule.

**Action** From operational mode, enter the **show security nat destination rule all** command. View the Translation hits field to check for traffic that matches the rule.

**Verifying NAT Application to Traffic**

**Purpose** Verify that NAT is being applied to the specified traffic.

**Action** From operational mode, enter the **show security flow session** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Source NAT on page 1369
- Source NAT Configuration Overview on page 1374
- Understanding Destination NAT on page 1350
- Destination NAT Configuration Overview on page 1353

**Disabling Port Randomization for Source NAT (CLI Procedure)**

For pool-based source NAT and interface NAT, port numbers are allocated randomly by default. Although randomized port number allocation can provide protection from security threats such as DNS poison attacks, it can also affect performance and memory usage for pool-based source NAT.

You can disable port randomization by using the **port-randomization disable** statement at the **[edit security nat source]** hierarchy level. To re-enable port randomization, use the **port-randomization** statement at the **[edit security nat source]** hierarchy level.

```
user@host# set security nat source port-randomization disable
```

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Source NAT on page 1369

- Source NAT Configuration Overview on page 1374

## Persistent NAT

- Understanding Persistent NAT on page 1412
- Understanding Session Traversal Utilities for NAT (STUN) Protocol on page 1413
- Persistent NAT Configuration Overview on page 1414
- Example: Configuring Persistent NAT with Source NAT Address Pool (CLI) on page 1415
- Example: Configuring Persistent NAT with Interface NAT (CLI) on page 1416

### Understanding Persistent NAT

Persistent NAT allows applications to use the Session Traversal Utilities for NAT (STUN) protocol when passing through NAT firewalls (see “Understanding Session Traversal Utilities for NAT (STUN) Protocol” on page 1413). Persistent NAT ensures that all requests from the same internal transport address are mapped to the same *reflexive transport address* (the public IP address and port created by the NAT device closest to the STUN server).

The following types of persistent NAT can be configured on the Juniper Networks device:

- Any remote host—All requests from a specific internal IP address and port are mapped to the same reflexive transport address. Any external host can send a packet to the internal host by sending the packet to the reflexive transport address.
- Target host—All requests from a specific internal IP address and port are mapped to the same reflexive transport address. An external host can send a packet to an internal host by sending the packet to the reflexive transport address. The internal host must have previously sent a packet to the external host’s IP address.
- Target host port—All requests from a specific internal IP address and port are mapped to the same reflexive transport address. An external host can send a packet to an internal host by sending the packet to the reflexive transport address. The internal host must have previously sent a packet to the external host’s IP address and port.

You configure any of the persistent NAT types with source NAT rules. The source NAT rule action can use a source NAT pool (with or without port translation) or an egress interface. Persistent NAT is not applicable for destination NAT, because persistent NAT bindings are based on outgoing sessions from internal to external.



**NOTE:** Port overloading is used in Junos OS only for normal interface NAT traffic. Persistent NAT does not support port overloading, and you must explicitly disable port overloading with the `port-overloading off` option at the `[edit security nat source]` hierarchy level.

To configure security policies to permit or deny persistent NAT traffic, you can use two new predefined services—`junos-stun` and `junos-persistent-nat`.



**NOTE:** Persistent NAT is different from the persistent address feature (see “Understanding Persistent Addresses” on page 1373). The persistent address feature applies to address mappings for source NAT pools configured on the device. The persistent NAT feature applies to address mappings on an external NAT device, and is configured for a specific source NAT pool or egress interface. Also, persistent NAT is intended for use with STUN client/server applications.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Session Traversal Utilities for NAT \(STUN\) Protocol on page 1413](#)
- [Persistent NAT and NAT64 Configuration Overview on page 1414](#)
- [Understanding Source NAT on page 1369](#)
- [Example: Configuring Persistent NAT with Source NAT Address Pool \(CLI\) on page 1415](#)
- [Example: Configuring Persistent NAT with Interface NAT \(CLI\) on page 1416](#)

#### Understanding Session Traversal Utilities for NAT (STUN) Protocol

Many video and voice applications do not work properly in a NAT environment. For example, Session Initiation Protocol (SIP), used with VoIP, encodes IP addresses and port numbers within application data. If a NAT firewall exists between the requestor and receiver, the translation of the IP address and port number in the data invalidates the information.

Also, a NAT firewall does not maintain a pinhole for incoming SIP messages. This forces the SIP application to either constantly refresh the pinhole with SIP messages or use an ALG to track registration, a function that may or may not be supported by the gateway device.

The Session Traversal Utilities for NAT (STUN) protocol, first defined in *RFC 3489, Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)* and then later in *RFC 5389, Session Traversal Utilities for NAT*, is a simple client/server protocol. A STUN client sends requests to a STUN server, which returns responses to the client. A STUN client is usually part of an application that requires a public IP address and/or port. STUN clients can reside in an end system such as a PC or in a network server whereas STUN servers are usually attached to the public Internet.



**NOTE:** Both the STUN client and STUN server must be provided by the application. Juniper Networks does not provide a STUN client or server.

The STUN protocol allows a client to:

- Discover whether the application is behind a NAT firewall.
- Determine the type of NAT binding being used (see “Understanding Persistent NAT and NAT64” on page 1412).

- Learn the reflexive transport address, which is the IP address and port binding allocated by NAT device closest to the STUN server. (There may be multiple levels of NAT between the STUN client and the STUN server.)

The client application can use the IP address binding information within protocols such as SIP and H.323.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Persistent NAT and NAT64 Configuration Overview on page 1414
- Understanding Persistent NAT and NAT64 on page 1412

### Persistent NAT Configuration Overview

---

To configure persistent NAT, specify the following options with the source NAT rule action (for either a source NAT pool or an egress interface):

- The type of persistent NAT—One of the following: any remote host, target host, or target host port (see “Understanding Persistent NAT and NAT64” on page 1412).
- (Optional) Address mapping—This option allows requests from a specific internal IP address to be mapped to the same reflexive IP address; internal and reflexive ports can be any ports. An external host using any port can send a packet to the internal host by sending the packet to the reflexive IP address (with a configured incoming policy that allows external to internal traffic). If this option is not configured, the persistent NAT binding is for specific internal and reflexive transport addresses.

You can only specify the **address-mapping** option when the persistent NAT type is any remote host and the source NAT rule action is one of the following actions:

- Source NAT pool with IP address shifting
- Source NAT pool with no port translation and no overflow pool
- (Optional) Inactivity timeout—Time, in seconds, that the persistent NAT binding remains in the device’s memory when all the sessions of the binding entry have expired. When the configured timeout is reached, the binding is removed from memory. The default value is 300 seconds. Configure a value from 60 through 7200 seconds.

When all sessions of a persistent NAT binding have expired, the binding remains in a query state in the SRX Series device’s memory for the specified inactivity timeout period. The query binding is automatically removed from memory when the inactivity timeout period expires (the default is 300 seconds). You can explicitly remove all or specific persistent NAT query bindings with the **clear security nat source persistent-nat-table** command.

- (Optional) Maximum session number—Maximum number of sessions with which a persistent NAT binding can be associated. The default is 30 sessions. Configure a value from 8 through 100.

For interface NAT, you need to explicitly disable port overloading with the **port-overloading off** option at the `[edit security nat source]` hierarchy level.

Finally, there are two predefined services that you can use in security policies to permit or deny STUN and persistent NAT traffic:

- **junos-stun**—STUN protocol traffic.
- **junos-persistent-nat**—Persistent NAT traffic.

For the **any remote host** persistent NAT type, the direction of the security policy is from external to internal. For target host or target host port persistent NAT types, the direction of the security policy is from internal to external.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Example: Configuring Persistent NAT with Source NAT Address Pool (CLI) on page 1415
- Example: Configuring Persistent NAT with Interface NAT (CLI) on page 1416
- Understanding Persistent NAT and NAT64 on page 1412
- Understanding Session Traversal Utilities for NAT (STUN) Protocol on page 1413

#### Example: Configuring Persistent NAT with Source NAT Address Pool (CLI)

You can configure any of the persistent NAT types with source NAT rules. The example in this section shows how to configure persistent NAT when source NAT is performed with a user-defined address pool.

The following example configures the target host persistent NAT type when source NAT is performed. In the following configuration, the source NAT address pool **sp1** consists of the address 30.1.1.5/32. The source NAT rule set **srs1** configures the following:

- Traffic direction is from zone **internal** to zone **external**.
- For packets with source address in the 40.1.1.0/24 subnet (internal phones) and destination address 20.20.20.0/24 (including STUN server, SIP proxy server and external phones), use the source NAT pool **sp1** to perform source NAT with the target host persistent NAT type.
- Set the persistent NAT **inactivity-timeout** to 180 seconds.

To configure the source NAT address pool:

```
user@host# set security nat source pool sp1 address 30.1.1.5/32
```

To configure the source NAT rule set:

```
user@host# set security nat source rule-set srs1 from zone internal
user@host# set security nat source rule-set srs1 to zone external
user@host# set security nat source rule-set srs1 rule sr1 match source-address 40.1.1.0/24
user@host# set security nat source rule-set srs1 rule sr1 match destination-address
  20.20.20.0/24
user@host# set security nat source rule-set srs1 rule sr1 then source-nat pool sp1
user@host# set security nat source rule-set srs1 rule sr1 then source-nat pool persistent-nat
  permit target-host
user@host# set security nat source rule-set srs1 rule sr1 then source-nat pool persistent-nat
  inactivity-timeout 180
```

For the target host persistent NAT type, configure a security policy to allow persistent NAT traffic from the internal network (internal zone) to the external network (external zone).

To configure a security policy to allow STUN traffic from internal SIP phones to an external STUN server:

```
user@host# set security policies from-zone internal to-zone external policy stun_traffic
  match source-address internal_phones destination-address stun_server application
  junos-stun
user@host# set security policies from-zone internal to-zone external policy stun_traffic
  then permit
```

To configure a security policy to allow SIP proxy traffic from internal SIP phones to an external SIP proxy server:

```
user@host# set security policies from-zone internal to-zone external policy
  sip_proxy_traffic match source-address internal_phones destination-address
  sip_proxy_server application junos-sip
user@host# set security policies from-zone internal to-zone external policy
  stun_proxy_traffic then permit
```

To configure a security policy to allow SIP traffic from internal to external SIP phones:

```
user@host# set security policies from-zone internal to-zone external policy sip_traffic
  match source-address internal_phones destination-address external_phones application
  junos-persistent-nat
user@host# set security policies from-zone internal to-zone external policy sip_traffic
  then permit
```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding Persistent NAT and NAT64 on page 1412
- Persistent NAT and NAT64 Configuration Overview on page 1414

#### Example: Configuring Persistent NAT with Interface NAT (CLI)

You can configure any of the persistent NAT types with source NAT rules. The example in this section shows how to configure persistent NAT when interface NAT is used to perform source NAT. For interface NAT, port overloading must be disabled.

The following example configures the **any remote host** persistent NAT type when interface NAT is performed. The interface NAT rule set **int1** configures the following:

- Traffic direction is from interface **ge-0/0/1.0** to interface **ge-0/0/2.0**.
- For packets with source address 40.1.1.0/24 (internal phones) and destination address 20.20.20.0/24 (including STUN server, SIP proxy server and external phones), perform interface NAT with the **any remote host** persistent NAT type.

You must also disable port overloading for interface NAT.

To configure the interface NAT rule set:

```
user@host# set security nat source rule-set int1 from interface ge-0/0/1.0
```

```

user@host# set security nat source rule-set int1 to interface ge-0/0/2.0
user@host# set security nat source rule-set int1 rule in1 match source-address 40.1.1.0/24
user@host# set security nat source rule-set int1 rule in1 match destination-address
20.20.20.0/24
user@host# set security nat source rule-set int1 rule in1 then source-nat interface
persistent-nat permit any-remote-host

```

To disable port overloading for interface NAT:

```

user@host# set security nat source interface port-overloading off

```

For the any remote host persistent NAT type, configure a security policy to allow persistent NAT traffic from the external network (external zone) to the internal network (internal zone).

To configure a security policy to allow STUN traffic from the internal SIP phones to the external STUN server:

```

user@host# set security policies from-zone internal to-zone external policy stun_traffic
match source-address internal_phones destination-address stun_server application
junos-stun
user@host# set security policies from-zone internal to-zone external policy stun_traffic
then permit

```

To configure a security policy to allow SIP proxy traffic from the internal SIP phones to the external SIP proxy server:

```

user@host# set security policies from-zone internal to-zone external policy
sip_proxy_traffic match source-address internal_phones destination-address
sip_proxy_server application junos-sip
user@host# set security policies from-zone internal to-zone external policy
stun_proxy_traffic then permit

```

To configure a security policy to allow SIP traffic from external SIP phones to internal SIP phones:

```

user@host# set security policies from-zone external to-zone internal policy sip_traffic
match source-address external_phones destination-address internal_phones application
junos-persistent-nat
user@host# set security policies from-zone external to-zone internal policy sip_traffic
then permit

```

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Persistent NAT and NAT64 on page 1412](#)
- [Persistent NAT and NAT64 Configuration Overview on page 1414](#)

## NAT for Multicast Flows

- [Understanding NAT for Multicast Flows on page 1418](#)
- [Example: Configuring NAT for Multicast Flows on page 1418](#)

## Understanding NAT for Multicast Flows

Network Address Translation (NAT) can be used to translate source addresses in IPv4 multicast flows and to translate IPv4 multicast group destination addresses.

Either static NAT or destination NAT can be used to perform multicast group address translation. Static NAT allows connections to be originated from either side of the network, but translation is limited to one-to-one addresses or between blocks of addresses of the same size. No address pools are necessary. Use the **static** configuration statement at the [edit security nat] hierarchy level to configure static NAT rule sets for multicast traffic. Destination NAT allows connections to be initiated only for incoming network connections—for example, from the Internet to a private network. Use the **destination** configuration statement at the [edit security nat] hierarchy level to configure destination NAT pools and rule sets.

Source NAT for multicast traffic is supported only by using IP address shifting to translate the original source IP address to an IP address from a user-defined address pool. This type of translation is one-to-one, static, and without port address translation. If the original source IP address range is larger than the IP address range in the user-defined pool, untranslated packets are dropped. The mapping does not provide bidirectional mapping, which static NAT provides. Use the **source** configuration statement at the [edit security nat] hierarchy level to configure source NAT pools and rule sets. When you define the source NAT pool for this type of source NAT, use the **host-address-base** option to specify the start of the original source IP address range.

### Related Documentation

- [Understanding Static NAT on page 1339](#)
- [Understanding Destination NAT on page 1350](#)
- [Understanding Source NAT Pools with Address Shifting on page 1372](#)
- [Example: Configuring NAT for Multicast Flows on page 1418](#)
- [Junos OS Multicast Protocols Configuration Guide](#)

## Example: Configuring NAT for Multicast Flows

This example shows how to configure a Juniper Networks device for address translation of multicast flows.

- [Requirements on page 1418](#)
- [Overview on page 1419](#)
- [Configuration on page 1421](#)
- [Verification on page 1426](#)

### Requirements

---

1. Configure network interfaces on the device. See the [Junos OS Interfaces Configuration Guide for Security Devices](#).
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 113.



3. Configure the device for multicast forwarding. See the *Junos OS Multicast Protocols Configuration Guide*.

### Overview

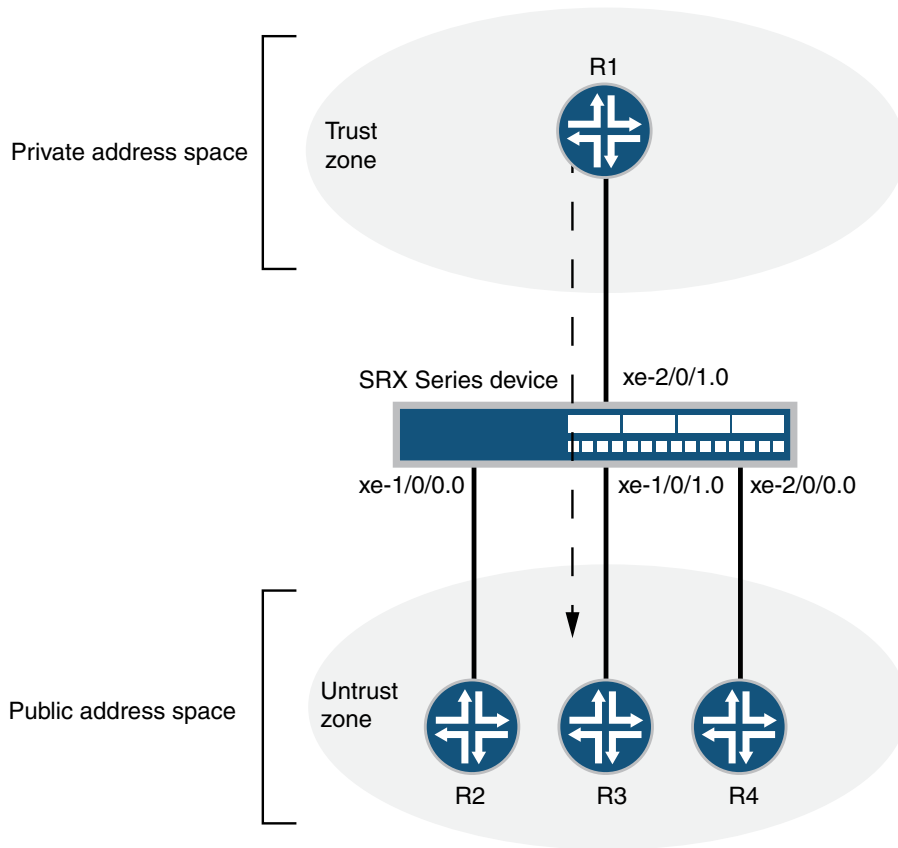
---

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. Figure 137 on page 1420 depicts a typical deployment of the Juniper Networks device for multicast forwarding. The source router R1 sends multicast packets with source addresses in the range 11.1.1.100 through 11.1.1.110 and the group address 225.0.0.1/32 toward the Juniper Networks device. The source router R1 is in the private network (trust zone) upstream of the Juniper Networks device. There are several receivers in the public network (untrust zone) downstream of the device.

The Juniper Networks device translates incoming multicast packets from R1 before forwarding them out on the downstream interfaces. The following translations are applied:

- For the interface to R2, the source address is untranslated, and the group address is translated to 226.0.0.1/32.
- For the interface to R3, the source address is translated to an address in the range 50.50.50.200 through 50.50.50.210, and the group address is translated to 226.0.0.1/32.
- For the interface to R4, the source address is translated to an address in the range 10.10.10.100 through 10.10.10.110, and the group address is translated to 226.0.0.1/32.

Figure 137: NAT Translations for Multicast Flows



From R1	To R2	To R3	To R4
Original Group IP	Group IP	Group IP	Group IP
225.0.0.1/32	226.0.0.1/32	226.0.0.1/32	226.0.0.1/32
Original Source IP	Source IP	Source IP	Source IP
11.1.1.100- 11.1.1.110	11.1.1.100- 11.1.1.110	50.50.50.200- 50.50.50.210	10.10.10.100- 10.10.10.110

6830688

This example describes the following configurations:

- Destination NAT pool **dst-nat-pool** that contains the IP address 226.0.0.1/32.
- Destination NAT rule set **rs1** with rule **r1** to match packets arriving on interface **xe-2/0/1.0** with the destination IP address 225.0.0.1/32. For matching packets, the destination address is translated to the IP address in the **dst-nat-pool** pool.
- Source NAT pool **src-nat-shift-1** that contains the IP address range 50.50.50.200/32 through 50.50.50.210/32. For this pool, the beginning of the original source IP address range is 11.1.1.100/32 and is specified with the **host-address-base** option.
- Source NAT rule set **rs-shift1** with rule **r1** to match packets from the trust zone to interface **xe-1/0/1.0** with a source IP address in the 11.1.1.96/28 subnet. For matching packets that fall within the source IP address range specified by the **src-nat-shift-1** configuration, the source address is translated to the IP address in the **src-nat-shift-1** pool.
- Source NAT pool **src-nat-shift-2** that contains the IP address range 10.10.10.100/32 through 10.10.10.110/32. For this pool, the beginning of the original source IP address range is 11.1.1.100/32 and is specified with the **host-address-base** option.
- Source NAT rule set **rs-shift2** with rule **r1** to match packets from the trust zone to interface **xe-2/0/0.0** with a source IP address in the 11.1.1.96/28 subnet. For matching packets that fall within the source IP address range specified by the **src-nat-shift-2** configuration, the source address is translated to the IP address in the **src-nat-shift-2** pool.
- Proxy ARP for the addresses 11.1.1.100 through 11.1.1.110 on interface **xe-1/0/0.0**, addresses 50.50.50.200 through 50.50.50.210 on interface **xe-1/0/1.0**, and addresses 10.10.10.100 through 10.10.10.110 on interface **xe-2/0/0.0**. This allows the Juniper Networks security device to respond to ARP requests received on the interface for those addresses.
- Security policy to permit traffic from the trust zone to the untrust zone.
- Security policy to permit traffic from the untrust zone to the translated destination IP address in the trust zone.

### Configuration

#### CLI Quick Configuration

To quickly configure the destination and source NAT translations for multicast flows, copy the following commands and paste them into the CLI:

```
[edit]
set security nat source pool src-nat-shift-1 address 50.50.50.200/32 to 50.50.50.210/32
set security nat source pool src-nat-shift-1 host-address-base 11.1.1.100/32
set security nat source pool src-nat-shift-2 address 10.10.10.100/32 to 10.10.10.110/32
set security nat source pool src-nat-shift-2 host-address-base 11.1.1.100/32
set security nat source rule-set rs-shift1 from zone trust
set security nat source rule-set rs-shift1 to interface xe-1/0/1.0
set security nat source rule-set rs-shift1 rule r1 match source-address 11.1.1.96/28
set security nat source rule-set rs-shift1 rule r1 then source-nat pool src-nat-shift1
set security nat source rule-set rs-shift2 from zone trust
set security nat source rule-set rs-shift2 to interface xe-2/0/0.0
set security nat source rule-set rs-shift2 rule r2 match source-address 11.1.1.96/28
set security nat source rule-set rs-shift2 rule r2 then source-nat pool src-nat-shift2
```

```

set security nat destination pool dst-nat-pool address 226.0.0.1/32
set security nat destination rule-set rs1 from interface xe-2/0/1.0
set security nat destination rule-set rs1 rule r1 match destination-address 225.0.0.1/32
set security nat destination rule-set rs1 rule r1 then destination-nat pool dst-nat-pool
set security nat proxy-arp interface xe-1/0/0.0 address 11.1.1.100/32 to 11.1.1.110/32
set security nat proxy-arp interface xe-1/0/1.0 address 50.50.50.200/32 to
  50.50.50.210/32
set security nat proxy-arp interface xe-2/0/0.0 address 10.10.10.100/32 to 10.10.10.110/32
set security policies from-zone trust to-zone untrust policy internet-access match
  source-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  destination-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  application any
set security policies from-zone trust to-zone untrust policy internet-access then permit
set security policies from-zone untrust to-zone trust policy dst-nat-pool-access match
  source-address any
set security policies from-zone untrust to-zone trust policy dst-nat-pool-access match
  destination-address 226.0.0.1/21
set security policies from-zone untrust to-zone trust policy dst-nat-pool-access match
  application any
set security policies from-zone untrust to-zone trust policy dst-nat-pool-access then
  permit

```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see “Using the CLI Editor in Configuration Mode.”

To configure the destination and source NAT translations for multicast flows:

1. Create a destination NAT pool.
 

```
[edit security nat destination]
user@host# set pool dst-nat-pool address 226.0.0.1/32
```
2. Create a destination NAT rule set.
 

```
[edit security nat destination]
user@host# set rule-set rs1 from interface xe-2/0/1.0
```
3. Configure a rule that matches packets and translates the destination address to the address in the destination NAT pool.
 

```
[edit security nat destination]
user@host# set rule-set rs1 rule r1 match destination-address 225.0.0.1/32
user@host# set rule-set rs1 rule r1 then destination-nat pool dst-nat-pool
```
4. Create a source NAT pool.
 

```
[edit security nat source]
user@host# set pool src-nat-shift-1 address 50.50.50.200 to 50.50.50.210
```
5. Specify the beginning of the original source IP address range.
 

```
[edit security nat source]
user@host# set pool src-nat-shift-1 host-address-base 11.1.1.100
```
6. Create a source NAT rule set.

- ```
[edit security nat source]
user@host# set rule-set rs-shift1 from zone trust
user@host# set rule-set rs-shift1 to interface xe-1/0/1.0
```
7. Configure a rule that matches packets and translates the destination address to the address in the source NAT pool.
 

```
[edit security nat source]
user@host# set rule-set rs-shift1 rule r1 match source-address 11.1.1.96/28
user@host# set rule-set rs-shift1 rule r1 then source-nat pool src-nat-shift1
```
  8. Create a source NAT pool.
 

```
[edit security nat source]
user@host# set pool src-nat-shift-2 address 10.10.10.100 to 10.10.10.110
```
  9. Specify the beginning of the original source IP address range.
 

```
[edit security nat source]
user@host# set pool src-nat-shift-2 host-address-base 11.1.1.100
```
  10. Create a source NAT rule set.
 

```
[edit security nat source]
user@host# set rule-set rs-shift2 from zone trust
user@host# set rule-set rs-shift2 to interface xe-2/0/0.0
```
  11. Configure a rule that matches packets and translates the destination address to the address in the source NAT pool.
 

```
[edit security nat source]
user@host# set rule-set rs-shift2 rule r2 match source-address 11.1.1.96/28
user@host# set rule-set rs-shift2 rule r2 then source-nat pool src-nat-shift2
```
  12. Configure proxy ARP.
 

```
[edit security nat]
user@host# set proxy-arp interface xe-1/0/0.0 address 11.1.1.100 to 11.1.1.110
user@host# set proxy-arp interface xe-1/0/1.0 address 50.50.50.200 to 50.50.50.210
user@host# set proxy-arp interface xe-2/0/0.0 address 10.10.10.100 to 10.10.10.110
```
  13. Configure a security policy that allows traffic from the trust zone to the untrust zone.
 

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy internet-access match source-address any
destination-address any application any
user@host# set policy internet-access then permit
```
  14. Configure a security policy that allows traffic from the untrust zone to the trust zone.
 

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy dst-nat-pool-access match source-address any
destination-address 226.0.0.1/32 application any
user@host# set policy dst-nat-pool-access then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
```

```
source {
  pool src-nat-shift-1 {
    address {
      50.50.50.200/32 to 50.50.50.210/32;
    }
    host-address-base 11.1.1.100/32;
  }
  pool src-nat-shift-2 {
    address {
      10.10.10.100/32 to 10.10.10.110/32;
    }
    host-address-base 11.1.1.100/32;
  }
  rule-set trust-to-untrust {
    from zone trust;
    to zone untrust;
    rule source-nat-rule {
      match {
        source-address 0.0.0.0/0;
      }
      then {
        source-nat {
          interface;
        }
      }
    }
  }
}
rule-set rs-shift1 {
  from zone trust;
  to interface xe-1/0/1.0;
  rule r1 {
    match {
      source-address 11.1.1.96/28;
    }
    then {
      source-nat {
        pool {
          src-nat-shift1;
        }
      }
    }
  }
}
rule-set rs-shift2 {
  from zone trust;
  to interface xe-2/0/0.0;
  rule r2 {
    match {
      source-address 11.1.1.96/28;
    }
    then {
      source-nat {
        pool {
          src-nat-shift2;
        }
      }
    }
  }
}
```

```

    }
  }
}
destination {
  pool dst-nat-pool {
    address 226.0.0.1/32;
  }
  rule-set rs1 {
    from interface xe-2/0/1.0;
    rule r1 {
      match {
        destination-address 225.0.0.1/32;
      }
      then {
        destination-nat pool dst-nat-pool;
      }
    }
  }
}
proxy-arp {
  interface xe-1/0/0.0 {
    address {
      11.1.1.100/32 to 11.1.1.110/32;
    }
  }
  interface xe-1/0/1.0 {
    address {
      50.50.50.200/32 to 50.50.50.210/32;
    }
  }
  interface xe-2/0/0.0 {
    address {
      10.10.10.100/32 to 10.10.10.110/32;
    }
  }
}
[edit]
user@host# show security policies
from-zone trust to-zone untrust {
  policy trust-to-untrust {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
  policy internet-access {
    match {
      source-address any;
      destination-address any;
      application any;
    }
  }
}

```

```

    }
    then {
      permit;
    }
  }
  from-zone untrust to-zone trust {
    policy dst-nat-pool-access {
      match {
        source-address any;
        destination-address 226.0.0.1/21;
        application any;
      }
      then {
        permit;
      }
    }
  }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

## Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Destination NAT Pool Usage on page 1426
- Verifying Destination NAT Rule Usage on page 1426
- Verifying Source NAT Pool Usage on page 1426
- Verifying Source NAT Rule Usage on page 1427
- Verifying NAT Application to Traffic on page 1427

### *Verifying Destination NAT Pool Usage*

**Purpose** Verify that there is traffic using IP addresses from the destination NAT pool.

**Action** From operational mode, enter the **show security nat destination pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

### *Verifying Destination NAT Rule Usage*

**Purpose** Verify that there is traffic matching the destination NAT rule.

**Action** From operational mode, enter the **show security nat destination rule all** command. View the Translation hits field to check for traffic that matches the rule.

### *Verifying Source NAT Pool Usage*

**Purpose** Verify that there is traffic using IP addresses from the source NAT pool.

**Action** From operational mode, enter the **show security nat source pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.



**Verifying Source NAT Rule Usage**

**Purpose** Verify that there is traffic matching the source NAT rule.

**Action** From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

**Verifying NAT Application to Traffic**

**Purpose** Verify that NAT is being applied to the specified traffic.

**Action** From operational mode, enter the **show security flow session** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Understanding NAT for Multicast Flows on page 1418
  - Source NAT Configuration Overview on page 1374
  - Destination NAT Configuration Overview on page 1353
  - [Junos OS Multicast Protocols Configuration Guide](#)

## Configuring Proxy ARP (CLI Procedure)

---

You use NAT proxy ARP functionality to configure proxy ARP entries for IP addresses that require either source or destination NAT and that are in the same subnet as the ingress interface.



**NOTE:** On SRX Series devices, you must explicitly configure NAT proxy ARP.

When configuring NAT proxy ARP, you must specify the logical interface on which to configure proxy ARP. Then you enter an address or address range.

The device performs proxy ARP for the following conditions:

- When addresses defined in the static NAT and source NAT pool are in the same subnet as that of the ingress interface
- When addresses in the original destination address entry in the destination NAT rules are in the same subnet as that of the ingress interface

```
user@host# set security nat proxy-arp interface fe-0/0/0.0 address 10.1.1.10 to 10.1.1.20
```

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - Static NAT Configuration Overview on page 1340
  - Destination NAT Configuration Overview on page 1353
  - Source NAT Configuration Overview on page 1374

## Verifying NAT Configuration

**Purpose** The NAT trace options hierarchy configures trace file and flags for verification purposes. J Series and SRX Series devices have two main components. Those are the Routing Engine (RE) and the Packet Forwarding Engine (PFE). The PFE is divided into the ukernel portion and the real-time portion. For verification, you can turn on flags individually to debug NAT functionality on the RE, ukernel PFE, or real-time PFE. The trace data is written to `/var/log/security-trace` by default.



**NOTE:** If session logging has been enabled in the policy configurations on the device, the session logs will include specific NAT details for each session. See “Monitoring Policy Statistics” on page 177 for information on how to enable session logging and “Information Provided in Session Log Entries for SRX Series Services Gateways” on page 16 for a description of information provided in session logs.

Use the `security nat traceoptions` command to verify if the NAT configurations are correctly updated to the device upon commit. To verify if NAT translations are being applied to the traffic and to view individual traffic flow processing with NAT translations, use the `security flow traceoptions` command.

**Action**

```

user@host# set security nat traceoptions flag all
user@host# set security nat traceoptions flag destination-nat-pfe
user@host# set security nat traceoptions flag destination-nat-re
user@host# set security nat traceoptions flag destination-nat-rti
user@host# set security nat traceoptions flag destination-nat-pfe
user@host# set security nat traceoptions flag source-nat-pfe
user@host# set security nat traceoptions flag source-nat-re
user@host# set security nat traceoptions flag source-nat-rt
user@host# set security nat traceoptions flag static-nat-pfe
user@host# set security nat traceoptions flag static-nat-re
user@host# set security nat traceoptions flag static-nat-rt

```

To filter a specific flow, you can define a packet filter and use it as a traceoption :

```

root@host# set security flow traceoptions packet-filter packet-filter
root@host# set security flow traceoptions packet-filter packet-filter apply-groups
root@host# set security flow traceoptions packet-filter packet-filter apply-groups-except
root@host# set security flow traceoptions packet-filter packet-filter destination-port
root@host# set security flow traceoptions packet-filter packet-filter destination-prefix
root@host# set security flow traceoptions packet-filter packet-filter interface
root@host# set security flow traceoptions packet-filter packet-filter protocol
root@host# set security flow traceoptions packet-filter packet-filter source-port
root@host# set security flow traceoptions packet-filter packet-filter source-prefix

```

To verify NAT traffic and to enable all traffic trace in data plane, use the traceoption `set security flow traceoptions flag basic-datapath` command.

- Related Documentation**
- [\*Junos OS Feature Support Reference for SRX Series and J Series Devices\*](#)
  - [Static NAT Configuration Overview on page 1340](#)
  - [Destination NAT Configuration Overview on page 1353](#)
  - [Source NAT Configuration Overview on page 1374](#)



PART 13

# GPRS

- [General Packet Radio Service on page 1433](#)



# General Packet Radio Service

- GPRS Overview on page 1433
- Policy-Based GTP on page 1437
- GTP Inspection Objects on page 1442
- GTP Message Filtering on page 1444
- GTP Information Elements on page 1453
- Understanding GGSN Redirection on page 1462

## GPRS Overview

---

General Packet Radio Service (GPRS) networks connect to several external networks including those of roaming partners, corporate customers, GPRS Roaming Exchange (GRX) providers, and the public Internet. GPRS network operators face the challenge of protecting their network while providing and controlling access to and from these external networks. Juniper Networks provides solutions to many of the security problems plaguing GPRS network operators.

In the GPRS architecture, the fundamental cause of security threats to an operator's network is the inherent lack of security in the GPRS tunneling protocol (GTP). GTP is the protocol used between GPRS support nodes (GSNs). GTP is used to establish a GTP tunnel for individual mobile stations (MSs) and between a Serving GPRS Support Node (SGSN) and a gateway GPRS support node (GGSN). A GTP tunnel is a channel between GSNs through which two hosts exchange data. The SGSN receives packets from the MS and encapsulates them within a GTP header before forwarding them to the GGSN through the GTP tunnel. When the GGSN receives the packets, it decapsulates them and forwards them to the external host.

Communication between different GPRS networks is not secure because GTP does not provide any authentication, data integrity, or confidentiality protection. Implementing IP Security (IPsec) for connections between roaming partners, setting traffic rate limits, and using stateful inspection can eliminate a majority of the GTP's security risks. The GTP firewall features in Junos OS address key security issues in mobile operators' networks.

Juniper Networks security devices mitigate a wide variety of attacks on the following types of GPRS interfaces:

- Gn—The Gn interface is the connection between an SGSN and a GGSN within the same public land mobile network (PLMN).
- Gp—The Gp interface is the connection between two PLMNs.
- Gi—The Gi interface is the connection between a GGSN and the Internet or destination networks connected to a PLMN.



**NOTE:** The term *interface* has different meanings in Junos OS and in GPRS technology. In Junos OS, an interface is a doorway to a security zone that allows traffic to enter and exit the zone. In GPRS, an interface is a connection, or a reference point, between two components of a GPRS infrastructure, for example, an SGSN and a GGSN.

This topic contains the following sections:

- Gp and Gn Interfaces on page 1434
- Gi Interface on page 1435
- Operational Modes on page 1436

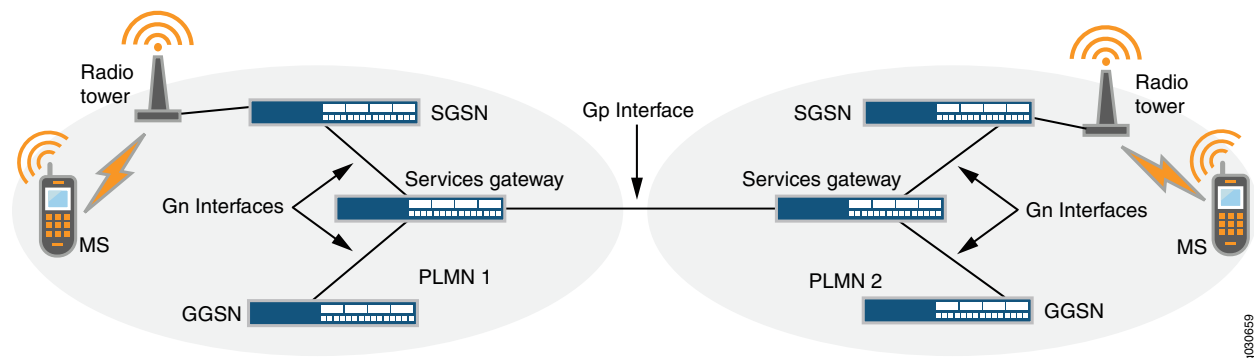
## Gp and Gn Interfaces

You implement a security device on the Gn interface to protect core network assets such as the SGSN and GGSN. To secure GTP tunnels on the Gn interface, you place the security device between SGSNs and GGSNs within a common PLMN.

When you implement a security device to the Gp interface, you protect a PLMN from another PLMN. To secure GTP tunnels on the Gp interface, you place the SGSNs and GGSNs of a PLMN behind the security device so that all traffic, incoming and outgoing, goes through the firewall.

Figure 138 on page 1434 illustrates the placement of Juniper Networks SRX Series devices used to protect PLMNs on the Gp and Gn interfaces.

Figure 138: Gp and Gn Interfaces





## Gi Interface

When you implement a security device on the Gi interface, you can simultaneously control traffic for multiple networks, protect a PLMN against the Internet and external networks, and protect mobile users from the Internet and other networks. Junos OS provides a great number of virtual routers, making it possible for you to use one virtual router per customer network and thereby allow the separation of traffic for each customer network.

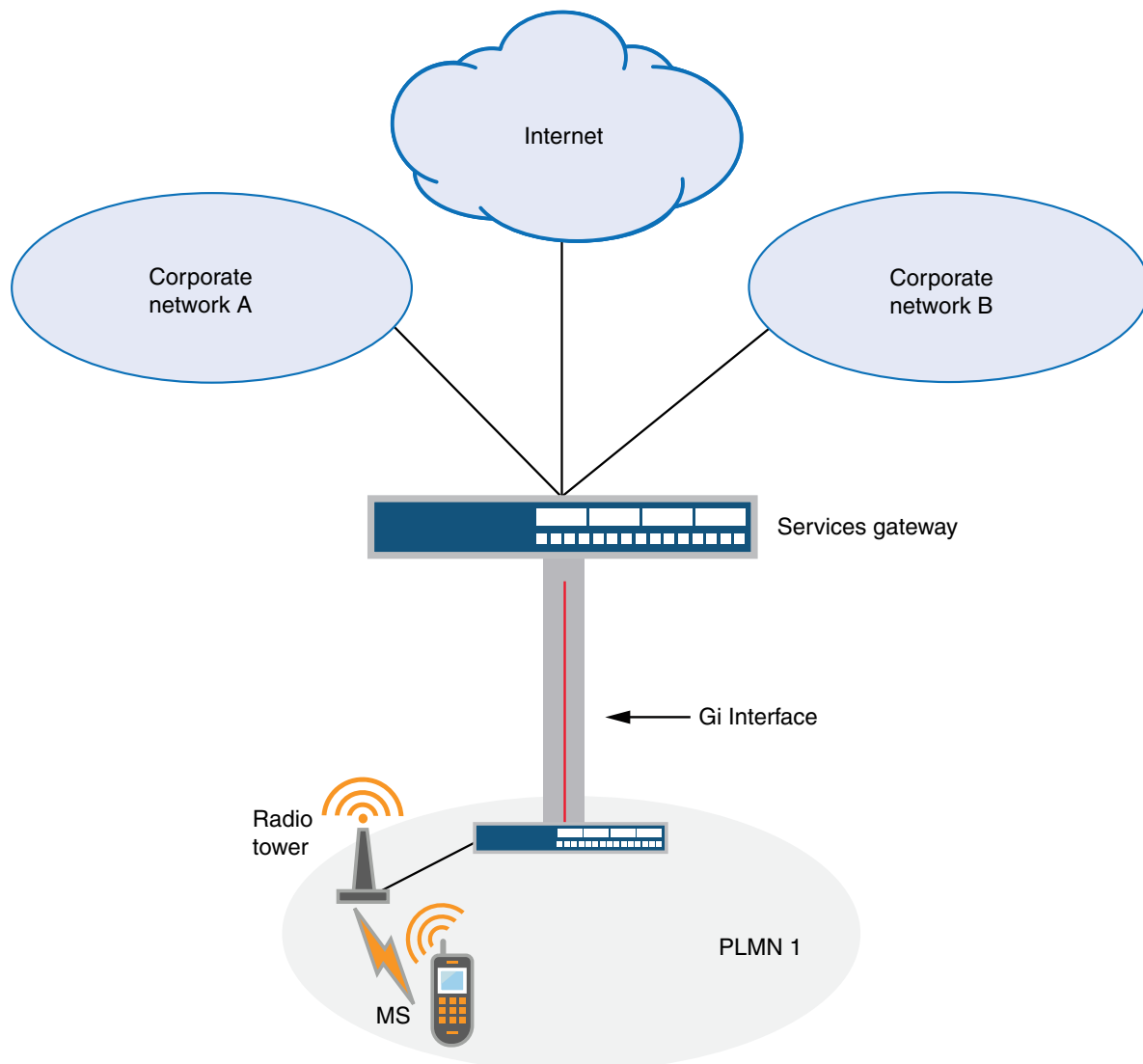
The security device can securely forward packets to the Internet or destination networks using the Layer 2 Tunneling Protocol (L2TP) for IPsec virtual private network (VPN) tunnels.



NOTE: SRX Series devices do not support full L2TP.

Figure 139 on page 1436 illustrates the implementation of a security device to protect a PLMN on the Gi interface.

Figure 139: Gi Interface



## Operational Modes

Junos OS supports two interface operational modes with GTP: transparent mode and route mode. If you want the security device to participate in the routing infrastructure of your network, you can run it in route mode. This requires a certain amount of network redesign. Alternatively, you can implement the security device into your existing network in transparent mode without having to reconfigure the entire network. In transparent mode, the security device functions as a Layer 2 switch or bridge, and the IP addresses of interfaces are set at 0.0.0.0, making the presence of the security device invisible, or *transparent*, to users.

Junos OS supports Network Address Translation (NAT) on interfaces and policies that do not have GTP inspection enabled.

Currently in Junos OS, route mode supports active/passive, and active/active chassis cluster. Transparent mode supports active/passive only.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Junos OS Layer 2 Bridging and Switching Configuration Guide for Security Devices](#)
- Chassis Cluster Overview on page 1137
- Understanding Policy-Based GTP on page 1437
- Understanding GTP Inspection Objects on page 1442
- Understanding GTP Message Filtering on page 1444
- Supported GTP Message Types on page 1447

## Policy-Based GTP

---

- Understanding Policy-Based GTP on page 1437
- Example: Enabling GTP Inspection in Policies on page 1438

### Understanding Policy-Based GTP

By default, the public land mobile network (PLMN) that the Juniper Networks device protects is in the Trust zone. The device protects the PLMN in the Trust zone against other PLMNs in other zones. You can place all the PLMNs against which you are protecting your PLMN in the Untrust zone, or you can create user-defined zones for each PLMN. A PLMN can occupy one security zone or multiple security zones.

You must create policies to enable traffic to flow between zones and PLMNs. Policies contain rules that permit, deny, or tunnel traffic. The device performs GPRS tunneling protocol (GTP) policy filtering by checking every GTP packet against policies that regulate GTP traffic and by then forwarding, dropping, or tunneling the packet based on these policies.

By selecting the GTP service in a policy, you enable the device to permit, deny, or tunnel GTP traffic. However, this does not enable the device to inspect GTP traffic. For the device to inspect GTP traffic, you must apply a GTP configuration, also referred to as a *GTP inspection object*, to a policy.

You can apply only one GTP inspection object per policy, but you can apply a GTP inspection object to multiple policies. Using policies, you can permit or deny the establishment of GTP tunnels from certain peers such as a Serving GPRS Support Node (SGSN).

You can configure policies that specify “Any” as the source or destination zone (thereby including all hosts in the zone), and you can configure policies that specify multiple source and destination addresses.

In policies, you can enable traffic logging.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [GPRS Overview on page 1433](#)
  - [Understanding GTP Inspection Objects on page 1442](#)
  - [Understanding GTP Message Filtering on page 1444](#)
  - [Supported GTP Message Types on page 1447](#)
  - [Example: Enabling GTP Inspection in Policies on page 1438](#)

## Example: Enabling GTP Inspection in Policies

This example shows how to enable GTP inspection in policies.

- [Requirements on page 1438](#)
- [Overview on page 1438](#)
- [Configuration on page 1438](#)
- [Verification on page 1442](#)

### Requirements

Before you begin, the device must be restarted after enabling GTP. By default GTP is disabled.

### Overview

In this example, you configure interfaces as ge-0/0/1 and ge-0/0/2, the addresses are 2.0.0.254/8 and 3.0.0.254/8. You then configure the security zone and specify address as 2.0.0.5/32 and 3.0.0.6/32. You enable the GTP service in the security policies to allow bidirectional traffic between two networks within the same PLMN.

### Configuration

- CLI Quick Configuration** To quickly configure GTP inspection in policies, copy the following commands and paste them into the CLI:

```
[edit]
set security gprs gtp profile gtp1
set interfaces ge-0/0/1 unit 0 family inet address 2.0.0.254/8
set interfaces ge-0/0/2 unit 0 family inet address 3.0.0.254/8
set security zones security-zone sgsn interfaces ge-0/0/1.0 host-inbound-traffic
  system-services all
set security zones security-zone sgsn host-inbound-traffic protocols all
set security zones security-zone ggsn interfaces ge-0/0/2.0 host-inbound-traffic
  system-services all
set security zones security-zone ggsn host-inbound-traffic protocols all
set security zones security-zone sgsn address-book address local-sgsn 2.0.0.5/32
set security zones security-zone ggsn address-book address remote-ggsn 3.0.0.6/32
set security policies from-zone sgsn to-zone ggsn policy sgsn_to_ggsn match
  source-address local-sgsn destination-address remote-ggsn application junos-gprs-gtp
set security policies from-zone sgsn to-zone ggsn policy sgsn_to_ggsn then permit
  application-services gprs-gtp-profile gtp1
```

```
set security policies from-zone ggsn to-zone sgsn policy ggsn_to_sgsn match
  source-address remote-ggsn destination-address local-sgsn application junos-gprs-gtp
set security policies from-zone ggsn to-zone sgsn policy sgsn_to_ggsn then permit
  application-services gprs-gtp-profile gtp1
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure GTP inspection in policies:

1. Enable GTP.

```
[edit]
user@host# set security gprs gtp enable
user@host# commit
user@host# exit
user@host# request system reboot
```



#### NOTE:

2. Create the GTP inspection object.

```
[edit]
user@host# set security gprs gtp profile gtp1
```

3. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family inet address 2.0.0.254/8
user@host# set ge-0/0/2 unit 0 family inet address 3.0.0.254/8
```

4. Configure security zones.

```
[edit security zones]
user@host# set security-zone sgsn interfaces ge-0/0/1.0
user@host# set security-zone sgsn host-inbound-traffic system-services all
user@host# set security-zone sgsn host-inbound-traffic protocols all
user@host# set security-zone ggsn interfaces ge-0/0/2.0
user@host# set security-zone ggsn host-inbound-traffic system-services all
user@host# set security-zone ggsn host-inbound-traffic protocols all
```

5. Specify addresses.

```
[edit security zones]
user@host# set security-zone sgsn address-book address local-sgsn 2.0.0.5/32
user@host# set security-zone ggsn address-book address remote-ggsn 3.0.0.6/32
```

6. Enable the GTP service in the security policies.

```
[edit security policies]
user@host# set from-zone sgsn to-zone ggsn policy sgsn_to_ggsn match
  source-address local-sgsn destination-address remote-ggsn application
  junos-gprs-gtp
user@host# set from-zone sgsn to-zone ggsn policy sgsn_to_ggsn then permit
  application-services gprs-gtp-profile gtp1
```

```

user@host# set from-zone ggsn to-zone sgsn policy ggsn_to_sgsn match
source-address remote-ggsn destination-address local-sgsn application
junos-gprs-gtp
user@host# set from-zone ggsn to-zone sgsn policy sgsn_to_ggsn then permit
application-services gprs-gtp-profile gtp1

```

**Results** From configuration mode, confirm your configuration by entering the **show security** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show security** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```

[edit]
user@host# show security
...
gprs {
  gtp {
    profile gtp1;
  }
}
zones {
  security-zone Trust {
  host-inbound-traffic {
  system-services {
    all;
  }
  protocols {
    all;
  }
}
  interfaces {
    ge-0/0/1.0;
  }
}
...
security-zone sgsn {
  address-book {
    address local-sgsn 2.0.0.5/32;
  }
  host-inbound-traffic {
    system-services {
      all;
    }
  }
  protocols {
    all;
  }
}
  interfaces {
    ge-0/0/1.0;
  }
}
security-zone ggsn {
  address-book {

```

```
    address remoteggsn 3.0.0.6/32;
  }
  host-inbound-traffic {
    system-services {
      all;
    }
  }
  protocols {
    all;
  }
}
interfaces {
  ge-0/0/2.0;
}
}
}
policies {
  from-zone sgsn to-zone ggsn {
    policy sgsn_to_ggsn {
      match {
        source-address local-sgsn;
        destination-address remoteggsn;
        application junos-gprs-gtp;
      }
      then {
        permit {
          application-services {
            gprs-gtp-profile gtp1;
          }
        }
      }
    }
  }
  from-zone ggsn to-zone sgsn {
    policy ggsn_to_sgsn {
      match {
        source-address remoteggsn;
        destination-address localsgsn;
        application junos-gprs-gtp;
      }
    }
  }
  policy sgsn_to_ggsn {
    then {
      permit {
        application-services {
          gprs-gtp-profile gtp1;
        }
      }
    }
  }
}
default-policy {
  permit-all;
}
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

---

To confirm that the configuration is working properly, perform this task:

- [Verifying GTP Inspection in Policies on page 1442](#)

#### *Verifying GTP Inspection in Policies*

**Purpose** Verify that GTP is enabled.

**Action** From operational mode, enter the **show security** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [GPRS Overview on page 1433](#)
  - [Understanding Policy-Based GTP on page 1437](#)
  - [Understanding GTP Message Filtering on page 1444](#)
  - [Supported GTP Message Types on page 1447](#)

## GTP Inspection Objects

---

- [Understanding GTP Inspection Objects on page 1442](#)
- [Example: Creating a GTP Inspection Object on page 1443](#)

### Understanding GTP Inspection Objects

For the device to perform the inspection of GPRS tunneling protocol (GTP) traffic, you must create a GTP inspection object and then apply it to a policy. GTP inspection objects provide more flexibility in that they allow you to configure multiple policies that enforce different GTP configurations. You can configure the device to control GTP traffic differently based on source and destination zones and addresses, action, and so on.

To configure GTP features, you must enter the context of a GTP configuration. To save your settings in the CLI, you must first exit the GTP configuration, then enter the **commit** command.

- Related Documentation**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [GPRS Overview on page 1433](#)
  - [Understanding Policy-Based GTP on page 1437](#)
  - [Understanding GTP Message Filtering on page 1444](#)
  - [Supported GTP Message Types on page 1447](#)
  - [Example: Creating a GTP Inspection Object on page 1443](#)



## Example: Creating a GTP Inspection Object

This example shows how to create a GTP inspection object.

- Requirements on page 1443
- Overview on page 1443
- Configuration on page 1443
- Verification on page 1443

### Requirements

---

No special configuration beyond device initialization is required before configuring this feature.

### Overview

---

In this example, you create a GTP inspection object named LA-NY. You preserve most of the default values, and enable the sequence number validation feature.

### Configuration

---

#### Step-by-Step Procedure

To configure a GTP inspection object:

1. Create a GTP inspection object.  

```
[edit]  
user@host# set security gprs gtp profile la-ny
```
2. Enable the sequence number validation.  

```
[edit]  
user@host# set security gprs gtp profile la-ny seq-number-validated
```
3. If you are done configuring the device, commit the configuration.  

```
[edit]  
user@host# commit
```

### Verification

---

To verify the configuration is working properly, enter the **show security gprs** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding GTP Inspection Objects on page 1442](#)
- [GPRS Overview on page 1433](#)
- [Understanding Policy-Based GTP on page 1437](#)
- [Understanding GTP Message Filtering on page 1444](#)
- [Supported GTP Message Types on page 1447](#)

## GTP Message Filtering

---

- [Understanding GTP Message Filtering on page 1444](#)
- [GTP Message-Length Filtering on page 1444](#)
- [GTP Message-Type Filtering on page 1446](#)
- [GTP Message-Rate Limiting on page 1450](#)
- [GTP Sequence Number Validation on page 1451](#)
- [Understanding GTP IP Fragmentation on page 1453](#)

### Understanding GTP Message Filtering

When the device receives a GPRS tunneling protocol (GTP) packet, it checks the packet against policies configured on the device. If the packet matches a policy, the device inspects the packet according to the GTP configuration applied to the policy. If the packet fails to meet any of the GTP configuration parameters, the device will pass or drop the packets based on the configuration of the GTP inspection object.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding GTP Inspection Objects on page 1442](#)
- [GPRS Overview on page 1433](#)
- [Understanding Policy-Based GTP on page 1437](#)
- [Understanding GTP Message-Length Filtering on page 1444](#)
- [Supported GTP Message Types on page 1447](#)

### GTP Message-Length Filtering

- [Understanding GTP Message-Length Filtering on page 1444](#)
- [Example: Setting the GTP Message Lengths on page 1445](#)

#### Understanding GTP Message-Length Filtering

---

You can configure the device to drop packets that do not meet your specified minimum or maximum message lengths. In the GPRS tunneling protocol (GTP) header, the message length field indicates the length, in octets, of the GTP payload. It does not include the length of the GTP header itself, the UDP header, or the IP header. The default minimum and maximum GTP message lengths are 0 and 65,535 bytes, respectively.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding GTP Inspection Objects on page 1442](#)
- [GPRS Overview on page 1433](#)
- [Understanding Policy-Based GTP on page 1437](#)
- [Example: Setting the GTP Message Lengths on page 1445](#)

- Supported GTP Message Types on page 1447

### Example: Setting the GTP Message Lengths

This example shows how to set the GTP message lengths.

- Requirements on page 1445
- Overview on page 1445
- Configuration on page 1445
- Verification on page 1445

#### Requirements

No special configuration beyond device initialization is required before configuring this feature.

#### Overview

In this example, you configure the minimum GTP message length to 8 octets and the maximum GTP message length to 1200 octets for the GTP inspection object.

#### Configuration

#### Step-by-Step Procedure

To configure the GTP message lengths:

1. Specify the GTP profile.
 

```
[edit]
user@host# set security gprs gtp profile gtp1
```
2. Specify the minimum message length.
 

```
[edit]
user@host# set security gprs gtp profile gtp1 min-message-length 8
```
3. Specify the maximum message length.
 

```
[edit]
user@host# set security gprs gtp profile gtp1 max-message-length 1200
```
4. If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

#### Verification

To verify the configuration is working properly, enter the **show security gprs** command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding GTP Inspection Objects on page 1442
- GPRS Overview on page 1433
- Understanding Policy-Based GTP on page 1437
- Understanding GTP Message-Length Filtering on page 1444
- Supported GTP Message Types on page 1447

## GTP Message-Type Filtering

- [Understanding GTP Message-Type Filtering](#) on page 1446
- [Example: Permitting and Denying GTP Message Types](#) on page 1446
- [Supported GTP Message Types](#) on page 1447

### Understanding GTP Message-Type Filtering

You can configure the device to filter GPRS tunneling protocol (GTP) packets and permit or deny them based on their message type. By default, the device permits all GTP message types.

A GTP message type includes one or many messages. When you permit or deny a message type, you automatically permit or deny all messages of the specified type. For example, if you select to drop the `sgsn-context` message type, you thereby drop `sgsn-context-request`, `sgsn-context-response`, and `sgsn-context-acknowledge` messages.

You permit and deny message types based on the GTP version number. For example, you can deny message types for one version while you permit them for the other version.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding GTP Inspection Objects](#) on page 1442
- [GPRS Overview](#) on page 1433
- [Understanding Policy-Based GTP](#) on page 1437
- [Example: Permitting and Denying GTP Message Types](#) on page 1446
- [Supported GTP Message Types](#) on page 1447

### Example: Permitting and Denying GTP Message Types

This example shows how to permit and deny GTP message types.

- [Requirements](#) on page 1446
- [Overview](#) on page 1446
- [Configuration](#) on page 1446
- [Verification](#) on page 1447

#### **Requirements**

No special configuration beyond device initialization is required before configuring this feature.

#### **Overview**

In this example, for the `gtp1` profile, you configure the device to drop the `error-indication` and `failure-report` message types for version 1.

#### **Configuration**

**Step-by-Step Procedure** To permit and deny GTP message types:

1. Configure the device.

```
[edit]
user@host# set security gprs gtp profile gtp1
```

- Drop the error indication.

```
[edit]
user@host# set security gprs gtp profile gtp1 drop error-indication 1
```

- Drop the failure report messages.

```
[edit]
user@host# set security gprs gtp profile gtp1 drop failure-report 1
```

- If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security gprs** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding GTP Inspection Objects on page 1442
- GPRS Overview on page 1433
- Understanding Policy-Based GTP on page 1437
- Understanding GTP Message-Type Filtering on page 1446
- Supported GTP Message Types on page 1447

### Supported GTP Message Types

Table 134 on page 1447 lists the GTP messages supported in GTP Releases 1997 and 1999 (including charging messages for GTP) and the message types that you can use to configure GTP message-type filtering.

**Table 134: GTP Messages**

| Message                        | Message Type  | Version 0 | Version 1 |
|--------------------------------|---------------|-----------|-----------|
| create AA pdp context request  | create-aa-pdp | b         |           |
| create AA pdp context response | create-aa-pdp | b         |           |
| create pdp context request     | create-pdp    | b         | b         |
| create pdp context response    | create-pdp    | b         | b         |
| data record request            | data-record   | b         | b         |
| data record response           | data-record   | b         | b         |
| delete AA pdp context request  | delete-aa-pdp | b         |           |

Table 134: GTP Messages (continued)

| Message                                    | Message Type     | Version 0 | Version 1 |
|--------------------------------------------|------------------|-----------|-----------|
| delete AA pdp context response             | delete-aa-pdp    | b         |           |
| delete pdp context request                 | delete-pdp       | b         | b         |
| delete pdp context response                | delete-pdp       | b         | b         |
| echo request                               | echo             | b         | b         |
| echo response                              | echo             | b         | b         |
| error indication                           | error-indication | b         | b         |
| failure report request                     | failure-report   | b         | b         |
| failure report response                    | failure-report   | b         | b         |
| forward relocation request                 | fwd-relocation   | b         | b         |
| forward relocation response                | fwd-relocation   | b         | b         |
| forward relocation complete                | fwd-relocation   | b         | b         |
| forward relocation complete<br>acknowledge | fwd-relocation   | b         | b         |
| forward SRNS context                       | fwd-srns-context | b         | b         |
| forward SRNS context acknowledge           | fwd-srns-context | b         | b         |
| identification request                     | identification   | b         | b         |
| identification response                    | identification   | b         | b         |
| node alive request                         | node-alive       | b         | b         |
| node alive response                        | node-alive       | b         | b         |
| note MS GPRS present request               | note-ms-present  | b         | b         |
| note MS GPRS present response              | note-ms-present  | b         | b         |
| pdu notification request                   | pdu-notification | b         | b         |
| pdu notification response                  | pdu-notification | b         | b         |
| pdu notification reject request            | pdu-notification | b         | b         |

Table 134: GTP Messages (*continued*)

| Message                                  | Message Type          | Version 0 | Version 1 |
|------------------------------------------|-----------------------|-----------|-----------|
| pdu notification reject response         | pdu-notification      | b         | b         |
| RAN info relay                           | ran-info              | b         | b         |
| redirection request                      | redirection           | b         | b         |
| redirection response                     | redirection           | b         | b         |
| relocation cancel request                | relocation-cancel     | b         | b         |
| relocation cancel response               | relocation-cancel     | b         | b         |
| send route info request                  | send-route            | b         | b         |
| send route info response                 | send-route            | b         | b         |
| sgsn context request                     | sgsn-context          | b         | b         |
| sgsn context response                    | sgsn-context          | b         | b         |
| sgsn context acknowledge                 | sgsn-context          | b         | b         |
| supported extension headers notification | supported-extension   | b         | b         |
| g-pdu                                    | gtp-pdu               | b         | b         |
| update pdp context request               | update-pdp            | b         | b         |
| updated pdp context response             | update-pdp            | b         | b         |
| version not supported                    | version-not-supported | b         | b         |

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding GTP Inspection Objects on page 1442
- GPRS Overview on page 1433
- Understanding Policy-Based GTP on page 1437
- Understanding GTP Message-Type Filtering on page 1446
- Example: Setting the GTP Message Lengths on page 1445

## GTP Message-Rate Limiting

- [Understanding GTP Message-Rate Limiting on page 1450](#)
- [Example: Limiting the GTP Message Rate on page 1450](#)

### [Understanding GTP Message-Rate Limiting](#)

---

You can configure the device to limit the rate of network traffic going to a GPRS support node (GSN). You can set separate thresholds, in packets per second, for GGSN tunneling protocol, control (GTP-C) messages. Because GTP-C messages require processing and replies, they can potentially overwhelm a GSN. By setting a rate limit on GTP-C messages, you can protect your GSNs from possible denial-of-service (DoS) attacks such as the following:

- **Border gateway bandwidth saturation**—A malicious operator connected to the same GPRS Roaming Exchange (GRX) as your public land mobile network (PLMN) can direct so much network traffic at your Border Gateway that legitimate traffic is starved for bandwidth in or out of your PLMN, thus denying roaming access to or from your network.
- **GTP flood**—GPRS tunneling protocol (GTP) traffic can flood a GSN, forcing it to spend its CPU cycles processing illegitimate data. This can prevent subscribers from roaming and forwarding data to external networks, and it can prevent a General Packet Radio Service (GPRS) from attaching to the network.

This feature limits the rate of traffic sent to each GSN from the Juniper Networks device. The default rate is unlimited.

#### **Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding GTP Inspection Objects on page 1442](#)
- [GPRS Overview on page 1433](#)
- [Understanding Policy-Based GTP on page 1437](#)
- [Example: Limiting the GTP Message Rate on page 1450](#)
- [Supported GTP Message Types on page 1447](#)

### [Example: Limiting the GTP Message Rate](#)

---

This example shows how to limit the GTP message rate.

- [Requirements on page 1450](#)
- [Overview on page 1451](#)
- [Configuration on page 1451](#)
- [Verification on page 1451](#)

#### **Requirements**

No special configuration beyond device initialization is required before configuring this feature.



**Overview**

In this example, you limit the rate of incoming GTP messages to 300 packets per second.

**Configuration****Step-by-Step Procedure**

To configure the GTP message rate:

1. Specify the GTP profile.

```
[edit]
user@host# set security gprs gtp profile gtp1
```

2. Set the limit rate.

```
[edit]
user@host# set security gprs gtp profile gtp1 rate-limit 300
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

**Verification**

To verify the configuration is working properly, enter the **show security gprs** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding GTP Inspection Objects on page 1442](#)
- [GPRS Overview on page 1433](#)
- [Understanding Policy-Based GTP on page 1437](#)
- [Understanding GTP Message-Rate Limiting on page 1450](#)
- [Supported GTP Message Types on page 1447](#)

**GTP Sequence Number Validation**

- [Understanding GTP Sequence Number Validation on page 1451](#)
- [Example: Enabling GTP Sequence Number Validation on page 1452](#)

**Understanding GTP Sequence Number Validation**

You can configure the device to perform sequence-number validation.

The header of a GPRS tunneling protocol (GTP) packet contains a Sequence Number field. This number indicates to the gateway GPRS support node (GGSN) receiving the GTP packets the order of the packets. During the packet data protocol (PDP) context-activation stage, a sending GGSN uses zero (0) as the sequence number for the first G-PDU it sends through a tunnel to another GGSN. The sending GGSN increments the sequence number for each following G-PDU it sends. The value resets to zero when it reaches 65,535.

During the PDP context-activation stage, the receiving GGSN sets its counter to zero. Subsequently, whenever the receiving GGSN receives a valid G-PDU, the GGSN increments its counter by one. The counter resets to zero when it reaches 65,535.

Normally, the receiving GGSN compares the sequence number in the packets it received with the sequence number from its counter. If the numbers correspond, the GGSN forwards the packet. If they differ, the GGSN drops the packet. By implementing a Juniper Networks device between the GGSNs, the device can perform this validation for the GGSN and drop packets that arrive out of sequence. This feature helps conserve GGSN resources by preventing the unnecessary processing of invalid packets.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding GTP Inspection Objects on page 1442](#)
- [GPRS Overview on page 1433](#)
- [Understanding Policy-Based GTP on page 1437](#)
- [Example: Enabling GTP Sequence Number Validation on page 1452](#)
- [Supported GTP Message Types on page 1447](#)

### [Example: Enabling GTP Sequence Number Validation](#)

---

This example shows how to enable GTP sequence number validation feature.

- [Requirements on page 1452](#)
- [Overview on page 1452](#)
- [Configuration on page 1452](#)
- [Verification on page 1453](#)

**Requirements**

No special configuration beyond device initialization is required before configuring this feature.

**Overview**

In this example, you set the gtp profile as gtp1 and you also enable the sequence number validation feature.

**Configuration****Step-by-Step Procedure**

To enable GTP sequence number validation feature:

1. Set the GTP profile.  

```
[edit]  
user@host# set security gprs gtp profile gtp1
```
2. Enable the sequence number validation.  

```
[edit]  
user@host# set security gprs gtp profile gtp1 seq-number-validated
```
3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security gprs** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding GTP Sequence Number Validation on page 1451
- Understanding GTP Inspection Objects on page 1442
- GPRS Overview on page 1433
- Understanding Policy-Based GTP on page 1437
- Supported GTP Message Types on page 1447

## Understanding GTP IP Fragmentation

A GPRS tunneling protocol (GTP) packet consists of the message body and three headers: GTP, UDP, and IP. If the resulting IP packet is larger than the maximum transmission unit (MTU) on the transferring link, the sending Serving GPRS Support Node (SGSN) or gateway GPRS support node (GGSN) performs an IP fragmentation.

By default, the device buffers IP fragments until it receives a complete GTP message, and then inspects the GTP message.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding GTP Information Elements on page 1453
- Understanding GTP Inspection Objects on page 1442
- GPRS Overview on page 1433
- Understanding Policy-Based GTP on page 1437
- Supported GTP Message Types on page 1447

## GTP Information Elements

- Understanding GTP Information Elements on page 1453
- GTP APN Filtering on page 1454
- GTP IMSI Prefix Filtering on page 1456
- GTP R6 Information Elements on page 1458

## Understanding GTP Information Elements

Information elements (IEs) are included in all GPRS tunneling protocol (GTP) control message packets. IEs provide information about GTP tunnels, such as creation, modification, deletion, and status. Junos OS supports IEs consistent with Third-Generation Partnership Project (3GPP) Release 6. If you have contractual agreements with operators

running earlier releases of 3GPP, you can reduce network overhead by restricting control messages containing unsupported IEs.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding GTP IP Fragmentation on page 1453](#)
- [Understanding GTP Inspection Objects on page 1442](#)
- [GPRS Overview on page 1433](#)
- [Understanding Policy-Based GTP on page 1437](#)
- [Supported GTP Message Types on page 1447](#)

## GTP APN Filtering

- [Understanding GTP APN Filtering on page 1454](#)
- [Example: Setting a GTP APN and a Selection Mode on page 1455](#)

### Understanding GTP APN Filtering

An access point name (APN) is an information element (IE) included in the header of a GPRS tunneling protocol (GTP) packet that provides information about how to reach a network. An APN comprises two elements:

- Network ID—Identifies the name of an external network such as `mobiphone.com`.
- Operator ID—Uniquely identifies the operators' public land mobile network (PLMN) such as `mnc123.mcc456`.

By default, the device permits all APNs. However, you can configure the device to perform APN filtering to restrict access to roaming subscribers to external networks.

To enable APN filtering, you must specify one or more APNs. To specify an APN, you need to know the domain name of the network (for example, `mobiphone.com`) and, optionally, the operator ID. Because the domain name (network ID) portion of an APN can potentially be very long and contain many characters, you can use the wildcard (\*) as the first character of the APN. The wildcard indicates that the APN is not limited only to `mobiphone.com` but also includes all the characters that might precede it.

You may also set a *selection mode* for the APN. The selection mode indicates the origin of the APN and whether or not the Home Location Register (HLR) has verified the user subscription. You set the selection mode according to the security needs of your network. Possible selection modes include the following:

- Mobile Station—Mobile station-provided APN, subscription not verified.

This selection mode indicates that the mobile station (MS) provided the APN and that the HLR did not verify the user's subscription to the network.

- Network—Network-provided APN, subscription not verified.

This selection mode indicates that the network provided a default APN because the MS did not specify one, and that the HLR did not verify the user's subscription to the network.

- Verified—MS or network-provided APN, subscription verified.

This selection mode indicates that the MS or the network provided the APN and that the HLR verified the user's subscription to the network.

APN filtering applies only to create-pdp-request messages. When performing APN filtering, the device inspects GTP packets to look for APNs that match APNs that you set. If the APN of a GTP packet matches an APN that you specified, the device then verifies the selection mode and only forwards the GTP packet if both the APN and the selection mode match the APN and the selection mode that you specified. Because APN filtering is based on perfect matches, using the wildcard (\*) when setting an APN suffix can prevent the inadvertent exclusion of APNs that you would otherwise authorize. The device automatically denies all other APNs that do not match.

Additionally, the device can filter GTP packets based on the combination of an International Mobile Subscriber Identity (IMSI) prefix and an APN.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Example: Setting a GTP APN and a Selection Mode on page 1455](#)
- [Understanding GTP Inspection Objects on page 1442](#)
- [GPRS Overview on page 1433](#)
- [Understanding Policy-Based GTP on page 1437](#)
- [Supported GTP Message Types on page 1447](#)

#### Example: Setting a GTP APN and a Selection Mode

This example shows how to set a GTP APN and a selection mode.

- [Requirements on page 1455](#)
- [Overview on page 1455](#)
- [Configuration on page 1455](#)
- [Verification on page 1456](#)

##### Requirements

No special configuration beyond device initialization is required before configuring this feature.

##### Overview

In this example, you set a GTP APN as `mobiphone.com.mnc123.mcc456.gprs` and use the wildcard (\*) character. You also set the selection mode as `network`.

##### Configuration

#### Step-by-Step Procedure

To configure a GTP APN and a selection mode:

1. Specify the GTP profile.

[edit]

```
user@host# set security gprs gtp profile gtp1
```

2. Set a selection mode for the APN.

```
[edit]
user@host# set security gprs gtp profile gtp1 apn
      *mobiphone.com.mnc123.mcc456.gprs mcc-mnc * action selection net
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security gprs** command.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding GTP APN Filtering on page 1454
- Understanding GTP Inspection Objects on page 1442
- GPRS Overview on page 1433
- Understanding Policy-Based GTP on page 1437
- Supported GTP Message Types on page 1447

## GTP IMSI Prefix Filtering

- Understanding IMSI Prefix Filtering of GTP Packets on page 1456
- Example: Setting a Combined IMSI Prefix and APN Filter on page 1457

### Understanding IMSI Prefix Filtering of GTP Packets

A GPRS support node (GSN) identifies a mobile station (MS) by its International Mobile Station Identity (IMSI). An IMSI consists of three elements: the mobile country code (MCC), the mobile network code (MNC), and the Mobile Subscriber Identification Number (MSIN). The MCC and MNC combined constitute the IMSI prefix and identify the mobile subscriber's home network, or public land mobile network (PLMN).

By setting IMSI prefixes, you can configure the device to deny GPRS tunneling protocol (GTP) traffic coming from nonroaming partners. By default, a device does not perform IMSI prefix filtering on GTP packets. By setting IMSI prefixes, you configure the device to filter create-pdp-request messages and permit only GTP packets with IMSI prefixes that match the ones you set. The device allows GTP packets with IMSI prefixes that do not match any of the IMSI prefixes that you set. To block GTP packets with IMSI prefixes that do not match any of the IMSI prefixes set, use an explicit wildcard for the IMSI filter, and the drop action should be the last IMSI prefix filtering policy.

When you filter GTP packets based on an IMSI prefix, you must also specify an APN.

### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Example: Setting a Combined IMSI Prefix and APN Filter on page 1457
- Understanding GTP Inspection Objects on page 1442

- GPRS Overview on page 1433
- Understanding Policy-Based GTP on page 1437
- Supported GTP Message Types on page 1447

### Example: Setting a Combined IMSI Prefix and APN Filter

This example shows how to set and combine IMSI prefix and APN filter.

- Requirements on page 1457
- Overview on page 1457
- Configuration on page 1457
- Verification on page 1457

#### Requirements

No special configuration beyond device initialization is required before configuring this feature.

#### Overview

In this example, you set `mobiphone.com.mnc123.mcc456.gprs` as an APN and use the wildcard (\*). You permit all selection modes for this APN. You also set the IMSI prefix for a known PLMN, which is 246565. The MCC-MNC pair can be five or six digits.

#### Configuration

#### Step-by-Step Procedure

To set and combine IMSI prefix and APN filter:

1. Set the GTP profile.
 

```
[edit]
user@host# set security gprs gtp profile gtp1
```
2. Set the selection mode for APN.
 

```
[edit]
user@host# set security gprs gtp profile gtp1 apn
*mobiphone.com.mnc123.mcc456.gprs mcc-mnc 246565 action pass
```
3. If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

#### Verification

To verify the configuration is working properly, enter the `show security gprs` command.

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Understanding IMSI Prefix Filtering of GTP Packets on page 1456
- Understanding GTP Inspection Objects on page 1442
- GPRS Overview on page 1433
- Understanding Policy-Based GTP on page 1437

- Supported GTP Message Types on page 1447

## GTP R6 Information Elements

- Understanding R6 Information Elements Removal on page 1458
- Example: Removing R6 Information Elements from GTP Messages on page 1458
- Supported R6 Information Elements on page 1459

### Understanding R6 Information Elements Removal

The Third-Generation Partnership Project (3GPP) R6 information element (IE) removal feature allows you to retain interoperability in roaming between Second-Generation Partnership Project (2GPP) and 3GPP networks. You can configure the GPRS tunneling protocol (GTP)-aware Juniper Networks device, residing on the border of a public land mobile network (PLMN) and a GPRS Roaming Exchange (GRX) and acting as a Gp firewall, to remove 3GPP-specific attributes from the GTP packet header when the packet passes into a 2GPP network. You can configure the device to remove the RAT, RAI, ULI, IMEI-SV, and access point name (APN) restriction IEs from GTP messages prior to forwarding these messages to the gateway GPRS support node (GGSN).

#### Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- Example: Removing R6 Information Elements from GTP Messages on page 1458
- Understanding GTP Inspection Objects on page 1442
- GPRS Overview on page 1433
- Understanding Policy-Based GTP on page 1437
- Supported GTP Message Types on page 1447

### Example: Removing R6 Information Elements from GTP Messages

This example shows how to remove R6 information elements from GTP messages.

- Requirements on page 1458
- Overview on page 1458
- Configuration on page 1459
- Verification on page 1459

#### **Requirements**

No special configuration beyond device initialization is required before configuring this feature.

#### **Overview**

In this example, you configure the Gp interface of the security device to remove newly added R6 IEs (RAT, ULI, IMEI-SV, and APN restrictions) from the GTP message.



**Configuration****Step-by-Step Procedure**

To remove R6 information elements from GTP messages:

- Specify the GTP profile.
 

```
[edit]
user@host# set security gprs gtp profile gtp1
```
- Specify the information element.
 

```
[edit]
user@host# set security gprs gtp profile gtp1 remove-r6
```
- If you are done configuring the device, commit the configuration.
 

```
[edit]
user@host# commit
```

**Verification**

To verify the configuration is working properly, enter the **show security gprs** command.

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding R6 Information Elements Removal on page 1458](#)
- [Understanding GTP Inspection Objects on page 1442](#)
- [GPRS Overview on page 1433](#)
- [Understanding Policy-Based GTP on page 1437](#)
- [Supported GTP Message Types on page 1447](#)

**Supported R6 Information Elements**

Junos OS supports all 3GPP R6 IEs for GTP), as listed in Table 135 on page 1459.

**Table 135: Supported Information Elements**

| IE Type Value | Information Element                             |
|---------------|-------------------------------------------------|
| 1             | Cause                                           |
| 2             | International Mobile Subscriber Identity (IMSI) |
| 3             | Routing Area Identity (REI)                     |
| 4             | Temporary Logical Link Identity (TLLI)          |
| 5             | Packet TMSI (P-TMSI)                            |
| 8             | Reordering Required                             |
| 9             | Authentication Triplet                          |
| 11            | MAP Cause                                       |

Table 135: Supported Information Elements (*continued*)

| IE Type Value | Information Element                      |
|---------------|------------------------------------------|
| 12            | P-TMSI Signature                         |
| 13            | MS Validated                             |
| 14            | Recovery                                 |
| 15            | Selection Mode                           |
| 16            | Tunnel Endpoint Identifier Data I        |
| 17            | Tunnel Endpoint Identifier Control Plane |
| 18            | Tunnel Endpoint Identifier Data II       |
| 19            | Teardown ID                              |
| 20            | NSAPI                                    |
| 21            | RANAP Cause                              |
| 22            | RAB Context                              |
| 23            | Radio Priority SMS                       |
| 24            | Radio Priority                           |
| 25            | Packet Flow ID                           |
| 26            | Charging Characteristics                 |
| 27            | Trace Reference                          |
| 28            | Trace Type                               |
| 29            | MS Not Reachable Reason                  |
| 127           | Charging ID                              |
| 128           | End User Address                         |
| 129           | MM Context                               |
| 130           | PDP Context                              |
| 131           | Access Point Name                        |
| 132           | Protocol Configuration Options           |

Table 135: Supported Information Elements (*continued*)

| IE Type Value | Information Element                        |
|---------------|--------------------------------------------|
| 133           | GSN Address                                |
| 134           | MS International PSTN/ISDN Number (MSISDN) |
| 135           | Quality of Service Profile                 |
| 136           | Authentication Quintuplet                  |
| 137           | Traffic Flow Template                      |
| 138           | Target Identification                      |
| 139           | UTRAN Transparent Container                |
| 140           | RAB Setup Information                      |
| 141           | Extension Header Type List                 |
| 142           | Trigger Id                                 |
| 143           | OMC Identity                               |
| 144           | RAN Transparent Container                  |
| 145           | PDP Context Prioritization                 |
| 146           | Additional RAB Setup Information           |
| 147           | SGSN Number                                |
| 148           | Common Flags                               |
| 149           | APN Restriction                            |
| 150           | Radio Priority LCS                         |
| 151           | RAT Type                                   |
| 152           | User Location Information                  |
| 153           | MS Time Zone                               |
| 154           | IMEI-SV                                    |
| 155           | CAMEL Charging Information Container       |
| 156           | MBMS UE Context                            |

Table 135: Supported Information Elements (*continued*)

| IE Type Value | Information Element                    |
|---------------|----------------------------------------|
| 157           | Temporary Mobile Group Identity (TMGI) |
| 158           | RIM Routing Address                    |
| 159           | MBMS Protocol Configuration Options    |
| 160           | MBMS Service Area                      |
| 161           | Source TNC PDCP context Information    |
| 162           | Additional Trace Information           |
| 163           | Hop Counter                            |
| 164           | Selected PLMN ID                       |
| 165           | MBMS Session Identifier                |
| 166           | MBMS2G/3G Indicator                    |
| 167           | Enhanced NSAPI                         |
| 168           | MBMS Session Duration                  |
| 169           | Additional MBMS Trace Information      |
| 251           | Charging Gateway Address               |
| 255           | Private Extension                      |

**Related Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding R6 Information Elements Removal on page 1458](#)
- [Example: Removing R6 Information Elements from GTP Messages on page 1458](#)
- [GPRS Overview on page 1433](#)
- [Understanding Policy-Based GTP on page 1437](#)
- [Supported GTP Message Types on page 1447](#)

## Understanding GGSN Redirection

Junos OS supports GPRS tunneling protocol (GTP) traffic and gateway GPRS support node (GGSN) redirection. A GGSN (X) can send create-pdp-context responses in which it can specify different GGSN IP addresses (GGSN Y and GGSN Z) for subsequent GTP-C and GTP-U messages. Consequently, the SGSN sends the subsequent GGSN tunneling

protocol, control (GTP-C) and GGSN tunneling protocol, user plane (GTP-U) messages to GGSNs Y and Z, instead of X.

**Related  
Documentation**

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [GPRS Overview on page 1433](#)
- [Understanding Policy-Based GTP on page 1437](#)
- [Supported GTP Message Types on page 1447](#)



PART 14

# Index

- Index on page 1467





# Index

## Symbols

|                                              |       |
|----------------------------------------------|-------|
| #, comments in configuration statements..... | xlili |
| ( ), in syntax descriptions.....             | xlili |
| 3DES.....                                    | 457   |
| < >, in syntax descriptions.....             | xlili |
| [ ], in configuration statements.....        | xlili |
| { }, in configuration statements.....        | xlili |
| (pipe), in syntax descriptions.....          | xlili |

## A

|                                                 |          |
|-------------------------------------------------|----------|
| AAA.....                                        | 406      |
| Access Manager                                  |          |
| client-side files.....                          | 647      |
| error messages.....                             | 650      |
| logging.....                                    | 650      |
| overview.....                                   | 597      |
| system requirements.....                        | 647      |
| Windows registry changes.....                   | 650      |
| Access Point Name <i>See</i> APN                |          |
| accommodating end-to-end TCP communication      |          |
| end-to-end TCP communication.....               | 102      |
| address pools.....                              | 612, 613 |
| address sweep.....                              | 1019     |
| Advanced Encryption Standard (AES).....         | 457      |
| AES.....                                        | 457      |
| agentless access <i>See</i> UAC, Infranet agent |          |
| agents, zombie.....                             | 1065     |
| aggressive mode.....                            | 468      |
| AH (authentication header) protocol             |          |
| overview.....                                   | 451      |
| ALGs                                            |          |
| MS RPC.....                                     | 383      |
| SIP.....                                        | 269      |
| SIP NAT.....                                    | 283      |
| Sun RPC.....                                    | 380      |
| allowing                                        |          |
| unknown SIP ALG message types.....              | 280      |
| antireplay                                      |          |
| group VPN.....                                  | 662      |

|                                    |               |
|------------------------------------|---------------|
| antispam filtering.....            | 851           |
| local list.....                    | 859           |
| message handling.....              | 867           |
| server-based.....                  | 852, 853, 861 |
| antivirus                          |               |
| sophos overview.....               | 951           |
| verifying.....                     | 924           |
| antivirus, express.....            | 929           |
| EICAR file.....                    | 930           |
| limitations.....                   | 930           |
| testing.....                       | 930           |
| updating antivirus patterns.....   | 931           |
| antivirus, full.....               | 869           |
| application protocol scanning..... | 886           |
| content size limits.....           | 883           |
| decompression layer limit.....     | 884           |
| file extension scanning.....       | 879           |
| intelligent prescreening.....      | 881           |
| notification options.....          | 899, 900, 901 |
| scan session throttling.....       | 886           |
| scanning timeout.....              | 885           |
| signature database support.....    | 870           |
| updating antivirus patterns.....   | 870           |
| antivirus, Sophos                  |               |
| custom objects.....                | 955           |
| feature profile.....               | 959           |
| managing antivirus data files..... | 954           |
| security policies.....             | 966           |
| updating antivirus data files..... | 953           |
| utm policies.....                  | 965           |
| APN                                |               |
| filtering.....                     | 1454          |
| selection mode.....                | 1454          |
| appDDoS                            |               |
| application-level DDoS protection  |               |
| overview.....                      | 763           |
| AppDDoS                            |               |
| understanding logging.....         | 828           |
| AppDDoS Protection                 |               |
| enabling example.....              | 768           |
| application binding.....           | 736, 796      |

- application identification.....795
  - application binding.....796
  - application package manual download.....1104
  - configuring policies (IDP).....798
  - custom application signatures.....1111
  - disable.....799, 1127
  - memory limit.....804
  - nested applications.....800, 1109
  - overview.....795
  - service binding.....796
  - session limit.....804
  - system cache.....801, 1122
  - system caching for nested application
    - identification.....802, 1123
  - understanding application package.....1104
  - verifying application package.....1108
  - verifying cache statistics.....803, 1124
  - verifying counters.....806
  - See also* IDP
- application identification (Junos)
  - overview.....1103
- application identification services
  - memory limit.....1125
  - session limit.....1125
- application package
  - automatic update.....1106
  - manually update.....1104
  - understanding.....1104
  - updating, overview.....1104
  - verifying.....1108
- application sets
  - IDP, configuring.....733
  - overview.....730
- application system cache.....801, 1122
  - overview.....801, 1122
- application tracking
  - AppTrack.....1129
- application-level DDoS.....763
  - application-level DDoS protection
    - overview.....763
    - statistics reporting overview.....772
- Application-Level DDoS
  - understanding logging.....828
- application-level DDoS protection
  - configuration.....768
- applications
  - IDP, configuring.....731
- AppTrack
  - application tracking.....1129
- attack detection
  - overview.....1017
- attack object groups.....781
  - predefined.....781
- attack objects
  - predefined.....781
- attacks
  - DOS.....1065, 1066, 1098
  - ICMP
    - floods.....1088, 1089
    - fragments.....1052
  - IP packet fragments.....1059
  - Land.....1092, 1093
  - large ICMP packets.....1053
  - Ping of Death.....1095
  - replay.....469
  - session table floods.....1039, 1066
  - SYN floods.....1074, 1078
  - SYN fragments.....1061
  - Teardrop.....1096, 1098
  - UDP floods.....1090, 1091
  - unknown protocols.....1057
  - WinNuke.....1098, 1100
- auth users
  - groups.....415
- authenticating users
  - pass-through authentication.....390
- authentication
  - administrative.....406
  - algorithms.....456
  - client groups.....415
  - configuring
    - external authentication servers.....407
    - SecurID server.....407
  - pass-through.....390
  - Web.....397
- authentication tables *See* UAC, authentication tables
- authentication, authorization, and accounting servers.....406
- AutoKey IKE VPN.....455
  - management.....455
- B**
- banners.....418
- braces, in configuration statements.....xlili
- brackets
  - angle, in syntax descriptions.....xlili
  - square, in configuration statements.....xlili

## C

- CA certificates.....570
- captive portal
  - configuration.....440
  - overview.....439
  - redirect URL configure.....445
  - redirect URL options.....444
- certificates.....455
  - CA.....570
  - loading.....585
  - local.....574
  - revocation.....587
  - self-signed.....592
  - UAC deployments See UAC, device authentication
- changing session characteristics.....7, 97
- chassis cluster
  - ISSU upgrading.....1251
- chassis clusters
  - about.....1137
  - cluster ID and node ID configuration example.....1240
  - conditional route advertising configuration example.....1178
  - control interfaces.....1222
  - control link recovery configuration example.....1195
  - control port configuration example.....1184
  - control ports for dual control links configuration example.....1186
  - creating a J Series cluster.....1239
  - creating an SRX Series cluster.....1235
  - dampening time configuration example.....1160
  - disabling.....1254
  - enabling.....1210
  - fabric configuration example.....1201
  - fabric interfaces.....1222
  - formation.....1138
  - hardware setup for J Series devices.....1238
  - hardware setup for SRX Series devices.....1223
  - interface monitoring example.....1147
  - J Series active/passive configuration example.....1257, 1270, 1314
  - LAG and LACP configuration example.....1232
  - layer 2 Ethernet switching.....1227
  - management interface configuration example.....1242
  - management interfaces on J Series devices.....1221
  - management interfaces on SRX Series devices.....1221
  - minimum links configuration example.....1174
  - node interfaces on J Series devices.....1219
  - node interfaces on SRX Series devices.....1211
  - number of redundant Ethernet interfaces configuration example.....1245
  - redundancy group configuration.....1144
  - redundancy group IP address monitoring configuration example.....1151
  - redundancy groups.....1139
  - redundant Ethernet interface configuration example.....1164
  - redundant Ethernet interface link aggregation group configuration example.....1171
  - switching fabric interfaces configuration example.....1228
  - verifying.....1246
  - verifying statistics.....1247
  - verifying status.....1249
  - VLAN and IRB configuration example.....1229
- client groups for firewall authentication.....415
- cold sync
  - monitoring.....1155
- colocation mode.....679
- comments, in configuration statements.....xlili
- compiling IDP policy.....790
- compound attack sample.....751
- conditional route advertising configuration.....1176
- configuring
  - anomaly attack objects.....758
  - application identification services, memory limit.....1125
  - application identification services, session limit.....1125
  - application identification, memory limit.....804
  - application identification, session limit.....804
  - conditional route advertising.....1176
  - DSCP in IDP policy.....727
  - exempt rulebase.....722
  - external authentication servers.....407
  - group VPN.....663
  - group VPN colocation mode.....680
  - group VPN multicast rekey.....695
  - group VPN unicast rekey.....694
  - group VPNs.....663
  - host inbound traffic.....116
    - protocols.....122
  - IDP application sets.....733

|                                                |            |
|------------------------------------------------|------------|
| IDP applications.....                          | 731        |
| IDP in security policy.....                    | 702        |
| IDP policy, application identification.....    | 798        |
| IDP services.....                              | 731        |
| interface monitoring.....                      | 1147       |
| IPS rulebase.....                              | 717        |
| log suppression.....                           | 830        |
| protocol anomaly-based attack.....             | 759        |
| redundant Ethernet interfaces.....             | 1164       |
| SCCP DoS attack protection.....                | 334, 335   |
| signature attack objects.....                  | 755        |
| signature database automatic download.....     | 788        |
| signature database manual download.....        | 784        |
| SIP DoS attack protection.....                 | 278        |
| SIP proxy                                      |            |
| private zone.....                              | 309        |
| TCP-reset parameter.....                       | 125        |
| terminal rules.....                            | 725        |
| three-zone SIP scenario.....                   | 314        |
| configuring DSCP rewriting.....                | 818        |
| connection, deleting.....                      | 644        |
| Content Filtering.....                         | 969        |
| filter types.....                              | 969        |
| protocol support.....                          | 970        |
| verifying.....                                 | 982        |
| control link.....                              | 1183       |
| failure and recovery.....                      | 1193       |
| control plane                                  |            |
| overview.....                                  | 1182       |
| controlling session termination.....           | 98         |
| conventions                                    |            |
| notice icons.....                              | xlii       |
| text and syntax.....                           | xlii       |
| cookies, SYN.....                              | 1085       |
| CoS features.....                              | 6, 97      |
| counters, verifying                            |            |
| for application identification.....            | 806        |
| creating a J Series chassis cluster.....       | 1239       |
| creating an SRX Series chassis cluster.....    | 1235       |
| curly braces, in configuration statements..... | xliii      |
| custom attacks                                 |            |
| application binding.....                       | 736        |
| compound.....                                  | 749        |
| configuring.....                               | 755, 758   |
| name.....                                      | 736        |
| protocol anomaly.....                          | 748        |
| protocol binding.....                          | 740        |
| service binding.....                           | 736        |
| severity.....                                  | 736        |
| signature.....                                 | 743        |
| time binding.....                              | 742        |
| customer support.....                          | xliv       |
| contacting JTAC.....                           | xliv       |
| <b>D</b>                                       |            |
| data                                           |            |
| fabric.....                                    | 1199       |
| fabric (dual).....                             | 1200       |
| forwarding.....                                | 1198       |
| plane.....                                     | 1197       |
| Data Encryption Standard (DES).....            | 457        |
| data path.....                                 | 104        |
| fast-path processing.....                      | 106        |
| forward processing.....                        | 105        |
| session-based processing.....                  | 105        |
| data processing, stateful and stateless.....   | 3, 93      |
| DDoS.....                                      | 1065       |
| application-level.....                         | 763        |
| defining                                       |            |
| exempt rulebase.....                           | 722        |
| IPS rulebase.....                              | 717        |
| DES.....                                       | 457        |
| destination NAT.....                           | 1350       |
| address and port translation configuration     |            |
| example.....                                   | 1358       |
| address pools.....                             | 1351       |
| configuration overview.....                    | 1353       |
| overview.....                                  | 1350       |
| rules.....                                     | 1352       |
| single address translation configuration       |            |
| example.....                                   | 1353       |
| subnet translation configuration               |            |
| example.....                                   | 1364       |
| with source NAT configuration example.....     | 1405       |
| Diffie-Hellman.....                            | 455        |
| Diffserv                                       |            |
| configuring in IDP policy.....                 | 727        |
| digital signature.....                         | 570        |
| disabling                                      |            |
| chassis clusters.....                          | 1254       |
| disabling TCP packet security checks.....      | 101        |
| documentation                                  |            |
| comments on.....                               | xliv       |
| DoS                                            |            |
| firewall.....                                  | 1072       |
| session table floods.....                      | 1039, 1066 |
| DoS attacks.....                               | 1065       |

- DoS Attacks
    - network.....1073
    - OS-specific.....1094
  - download
    - signature database automatic.....788
    - signature database manually.....784
    - signature database overview.....783
  - DSCP rewriting.....817
  - dual control links
    - about.....1188
    - connecting.....1189
    - upgrading the second routing engine.....1191
  - dynamic auth table provisioning *See* UAC, dynamic
    - auth table provisioning
  - dynamic packet filtering.....1017
  - dynamic policies *See* group VPNs
  - dynamic VPNs
    - address pools.....612, 613
    - client access.....599
    - configuration example.....603
    - configuration overview.....601
    - group IKE IDs.....618, 619
    - individual user IKE IDs.....626
    - local authentication.....612, 613
    - overview.....597
    - shared IKE IDs.....618, 619
    - supported options.....600
    - tunnels.....600
- E**
- enabling chassis clusters.....1210
  - encryption algorithms.....457
  - ESP.....456, 457
  - ESP (Encapsulating Security Payload) protocol
    - overview.....451
  - exempt rulebase
    - configuring.....722
  - expansion of session capacity.....26
- F**
- fabric data link.....1199
  - fabric data link (dual).....1200
  - fabric data-link failure.....1199
  - fabric interfaces.....1222
  - fast-path processing.....106
  - filters, stateless firewall.....6, 96
  - FIN scans.....1038
  - FIN without ACK flag attack detection
    - overview.....1033
  - firewall users, pass-through
    - authentication process.....390
  - floods
    - ICMP.....1088, 1089
    - session table.....1066
    - SYN.....1074, 1078, 1085
    - UDP.....1090, 1091
  - flow-based packet processing
    - defined.....3
  - flow-based processing
    - enabling.....66
  - flowd
    - monitoring.....1154
  - font conventions.....xlii
  - forward processing.....105
  - forwarding features.....107
- G**
- gatekeeper devices.....223
  - GDOI protocol *See* group VPNs
  - Gi interface.....1433
  - Gp interface.....1433
  - gprs
    - about.....1433
    - tunneling protocol.....1433
  - group IKE IDs.....618, 619
  - group keys
    - KEK.....690
    - TEK.....690
  - group policies *See* group VPNs
  - group VPNs
    - antireplay.....662
    - colocation configuration.....680
    - colocation mode.....679
    - configuration.....663
    - configuration overview.....663
    - dynamic policies.....660
    - GDOI protocol.....657
    - group keys.....690
    - group policies.....660
    - heartbeat messages.....693
    - IKE Phase 1 configuration.....659
    - interoperability with GET VPN.....698
    - IPsec SA configuration.....659
    - key activation.....692
    - limitations.....697
    - member.....658
    - member reregistration.....692
    - multicast rekey configuration.....695

|                                             |               |                                        |          |
|---------------------------------------------|---------------|----------------------------------------|----------|
| overview.....                               | 655           | detector.....                          | 752      |
| rekey messages.....                         | 691           | DSCP.....                              | 727      |
| scope policies.....                         | 660           | DSCP rewriting.....                    | 817      |
| server.....                                 | 658           | dscp rewriting.....                    | 818      |
| server-member communication.....            | 689           | enabling IDP.....                      | 702      |
| unicast rekey configuration.....            | 694           | inserting rule.....                    | 714      |
| VPN group configuration.....                | 662           | log suppression.....                   | 827      |
| GTP                                         |               | logging, overview.....                 | 827      |
| access point name (APN) filtering.....      | 1454          | maximize-idp-sessions.....             | 825      |
| inspection objects.....                     | 1437          | packet capture.....                    | 832      |
| IP fragmentation.....                       | 1453          | performance and capacity tuning.....   | 825      |
| policy-based.....                           | 1437          | policy.....                            | 701      |
| GTP messages.....                           | 1449          | policy, manage.....                    | 701      |
| length, filtering by.....                   | 1444          | policy, overview.....                  | 701      |
| rate, limiting by.....                      | 1450          | protocol decoder.....                  | 752      |
| type, filtering by.....                     | 1446          | rewriting.....                         | 818      |
| types.....                                  | 1446          | rulebase, application-level DDoS.....  | 715      |
| versions 0 and 1.....                       | 1449          | rulebase, DDoS.....                    | 715      |
| H                                           |               | rulebase, exempt.....                  | 721      |
| hardware                                    |               | rulebase, IPS.....                     | 716      |
| supported platforms.....                    | xlii          | rulebase, overview.....                | 713      |
| hardware setup, chassis cluster.....        | 1223, 1238    | rules, actions.....                    | 710      |
| hash-based message authentication code..... | 456           | rules, IP actions.....                 | 711      |
| heartbeats.....                             | 1192          | rules, match conditions.....           | 707      |
| group VPN.....                              | 693           | rules, objects.....                    | 708      |
| high availability.....                      | 1436          | rules, overview.....                   | 707      |
| HMAC.....                                   | 456           | send attack logs to the IC.....        | 838      |
| Host Checker See UAC, Host Checker policy   |               | setting terminal rules.....            | 725      |
| enforcement                                 |               | signature database.....                | 777      |
| hub-and-spoke.....                          | 506           | terminal rules, overview.....          | 724      |
| I                                           |               | verify load status.....                | 790      |
| ICMP                                        |               | verify policy compilation.....         | 790      |
| floods.....                                 | 1088, 1089    | verify signature database version..... | 792      |
| fragments.....                              | 1052          | IDP application-level DDoS             |          |
| IPv6.....                                   | 59            | configuring statistics reporting.....  | 775      |
| large packets.....                          | 1053          | statistics reporting overview.....     | 772      |
| Path MTU.....                               | 61            | IDP policy                             |          |
| ICMP header flags.....                      | 747           | application identification.....        | 798      |
| IDP                                         |               | overview.....                          | 701      |
| application and services.....               | 731           | rulebase, exempt.....                  | 721      |
| application identification.....             | 795           | IDP, inline tap mode                   |          |
| application sets.....                       | 730           | configuring.....                       | 706      |
| application sets, configuring.....          | 733           | overview.....                          | 705      |
| custom attacks, properties.....             | 743, 748, 749 | IKE.....                               | 455      |
| deactivating rules.....                     | 714           | group IDs.....                         | 618, 619 |
| defining exempt rulebase.....               | 722           | individual user IDs.....               | 626      |
| defining IPS rulebase.....                  | 717           | Phase 1 proposals                      |          |
|                                             |               | group VPN.....                         | 659      |
|                                             |               | predefined.....                        | 467      |

- Phase 2 proposals
  - predefined.....468
  - proxy IDs.....468
  - shared IDs.....618, 619
- IKE IDs.....626
- in-service upgrade
  - chassis cluster.....1251
- Infranet agent *See* UAC, Infranet agent
- Infranet Controller *See* UAC, Infranet Controller
- Infranet Enforcer *See* UAC, Junos OS Enforcer
- initiating manual redundancy group failover.....1158
- inline tap mode
  - overview.....705
- inspections.....1017
- interface monitoring configuration.....1147
- interfaces.....112, 165
  - control.....1222
  - fabric.....1222
- interfaces on J Series devices
  - management.....1221
  - node.....1219
- interfaces on SRX Series devices
  - management.....1221
  - node.....1211
- Internet Key Exchange session.....573
- intrusion detection and prevention *See* IDP
- IP options
  - incorrectly formatted.....1055
  - loose source route.....1025
  - record route.....1025, 1027
  - security.....1025, 1027
  - source route.....1043
  - stream ID.....1025, 1027
  - strict source route.....1025
  - timestamp.....1025, 1027
- IP packet fragments.....1059
- IP protocol header.....745
- IP spoofing.....1041
- IPS rulebase
  - configuring.....717
- IPsec
  - digital signature.....570
  - overview.....451
  - SAs.....451, 457, 468, 655
    - group VPN configuration.....659*See also* group VPNs
- security protocols
  - Authentication Header (AH).....456
  - Encapsulating Security Protocol (ESP).....456
- tunnel.....451
  - creating through dynamic VPN
    - feature.....597
  - tunnel mode.....459
  - tunnel negotiation.....457
  - UAC support.....427
- IPv6
  - address examples.....50
  - address format.....50
  - address space.....48
  - address types.....48, 49
  - addressing.....48
  - anycast addresses.....49
  - basic packet header fields.....52
  - enabling.....66
  - features.....48
  - flow module sanity checks.....55
  - host-inbound traffic.....49
  - ICMP overview.....59
  - multicast addresses.....49
  - overview.....48
  - packet fragmentation.....62
  - packet header extension fields.....54
  - packet header overview.....51
  - Path MTU.....61
  - sessions.....63
  - SRX Series high-end devices.....63
  - unicast addresses.....49
- J**
  - JUEP *See* UAC, device authentication
  - Junos OS Enforcer *See* UAC, Junos OS Enforcer
- K**
  - KEK *See* group VPNs
  - key activation
    - group VPN.....692
- L**
  - L2TP.....1435
  - land attack detection
    - configuration.....1093
    - overview.....1092
  - local authentication.....612, 613
  - local certificate.....574

|                                     |            |                                               |            |
|-------------------------------------|------------|-----------------------------------------------|------------|
| log suppression.....                | 827        | destination NAT configuration.....            | 1353       |
| configuring.....                    | 830        | destination NAT configuration examples.....   | 1353       |
| logging                             |            | destination NAT overview.....                 | 1350       |
| IDP, overview.....                  | 827        | destination NAT rules.....                    | 1352       |
| logging, traffic.....               | 1437       | disabling port randomization.....             | 1411       |
| loose source route IP detection     |            | multicast flow.....                           | 1417, 1418 |
| configuration.....                  | 1025       | multicast flow configuration example.....     | 1418       |
|                                     |            | overview.....                                 | 1335       |
| <b>M</b>                            |            | persistent addresses.....                     | 1373       |
| main mode.....                      | 467        | persistent NAT.....                           | 1412       |
| management interfaces.....          | 1221       | persistent NAT configuration overview.....    | 1414       |
| manual key management               |            | persistent NAT overview.....                  | 1412       |
| overview.....                       | 454        | port address translation.....                 | 1371       |
| manuals                             |            | proxy ARP.....                                | 1427       |
| comments on.....                    | xliv       | rule sets and rules.....                      | 1336       |
| MD5.....                            | 456        | source.....                                   | 1368       |
| Message Digest version 5 (MD5)..... | 456        | source NAT address pools.....                 | 1370       |
| MGCP ALG.....                       | 347        | source NAT configuration.....                 | 1374       |
| commands.....                       | 350        | source NAT configuration examples.....        | 1374       |
| entities.....                       | 348        | source NAT overview.....                      | 1369       |
| security.....                       | 348        | source NAT rules.....                         | 1373       |
| MGCP timeouts                       |            | static.....                                   | 1339       |
| inactivity.....                     | 353        | static NAT configuration.....                 | 1340       |
| Mobile Station (MS) mode.....       | 1454       | static NAT configuration examples.....        | 1340       |
| modes                               |            | static NAT overview.....                      | 1339       |
| aggressive.....                     | 468        | static NAT rules.....                         | 1339       |
| main.....                           | 467        | STUN protocol.....                            | 1413       |
| tunnel.....                         | 459        | verify configuration.....                     | 1428       |
| modes, operational                  |            | without port address translation.....         | 1372       |
| NAT.....                            | 1436       | NAT mode.....                                 | 1436       |
| route.....                          | 1436       | Network Address Translation See NAT           |            |
| transparent.....                    | 1436       | network mode.....                             | 1454       |
| modes, selection                    |            | node interfaces on J Series devices.....      | 1219       |
| APN.....                            | 1454       | node interfaces on SRX Series devices.....    | 1211       |
| Mobile Station (MS).....            | 1454       | notice icons.....                             | xlii       |
| network.....                        | 1454       |                                               |            |
| verified.....                       | 1454       | <b>O</b>                                      |            |
| modulus.....                        | 455        | Odyssey Access Client See UAC, Infranet agent |            |
| MS RPC ALG, defined.....            | 383        | operational modes                             |            |
| multicast flow                      |            | NAT.....                                      | 1436       |
| NAT.....                            | 1417, 1418 | route.....                                    | 1436       |
| NAT configuration example.....      | 1418       | transparent.....                              | 1436       |
| multimedia sessions, SIP.....       | 269        |                                               |            |
|                                     |            | <b>P</b>                                      |            |
| <b>N</b>                            |            | packet capture                                |            |
| NAT.....                            | 1335       | IDP.....                                      | 832        |
| address shifting.....               | 1372       | packet filtering.....                         | 3, 93      |
| destination.....                    | 1350       | packet fragmentation                          |            |
| destination NAT address pools.....  | 1351       | IPv6.....                                     | 62         |



- packet processing.....3, 93
    - stateful.....3, 93
    - stateless.....3, 93
  - packet-based processing.....5, 96
  - parentheses, in syntax descriptions.....xlili
  - pass-through authentication.....390
  - Path MTU
    - Path MTU.....61
  - Perfect Forward Secrecy *See* PFS
  - PFS.....469
  - Phase 1.....467
    - proposals.....467
    - proposals, predefined.....467
  - Phase 2.....468
    - proposals.....468
    - proposals, predefined.....468
  - ping of death attack protection
    - configuration.....1096
    - overview.....1095
  - pinholes.....272
  - PKI.....570
    - using SCEP.....574
  - policies.....1437
    - application services processing order.....427
    - core section.....1039
    - shadowing.....175
  - policies, configuring.....1437
  - policy
    - IDP *See* IDP
  - policy templates
    - predefined.....778
  - policy-based VPN configuration example.....489
  - port scan attack protection
    - overview.....1022
  - predefined attack objects.....781
  - predefined policy templates.....778
    - overview.....778
  - preshared key.....455
  - Primary-level entry
    - secondary-level
      - et 51162069364085652760889089022894990
  - Primary-level entry
    - et 51162069364085652760889089022894990
  - probes
    - network.....1019
    - open ports.....1022
    - operating systems.....1030, 1035
  - processing
    - data.....3, 93
    - flow-based.....4, 94
    - packet-based.....5, 96
  - proposals
    - Phase 1.....467
    - Phase 2.....468
  - protocol anomaly.....748
  - protocol anomaly attack.....749
    - direction.....749
    - expression (boolean expression).....750
    - member index.....751
    - member index sample.....751
    - order.....750
    - reset.....750
    - sample.....749, 751
    - scope.....750
    - test condition.....749
  - protocol anomaly attack sample.....749
  - protocol anomaly-based attack
    - configuring.....759
  - protocol binding.....740
    - sample format.....742
  - proxy IDs.....468
  - public/private key pair.....572
- R**
- rate limiting, GTP-C messages.....1450
  - reconnaissance
    - address sweep.....1019
    - FIN scans.....1038
    - IP options.....1025
    - port scan.....1022
    - SYN and FIN flags set.....1030
    - TCP packet without flags.....1035
  - reconnaissance deterrence
    - IP address sweeps.....1019
      - blocking.....1019
    - overview.....1019
  - record route IP option.....1025, 1027
  - redundancy group
    - initiating manual failover.....1158
  - redundancy groups
    - about.....1139
    - group 0.....1140
    - groups 1 through 128.....1141
    - interface monitoring.....1146
    - IP address monitoring.....1148
  - redundant Ethernet interface LAG.....1169



- security checks, disabling TCP packet.....101
- security IP option.....1025, 1027
- security policy
  - enabling IDP.....702
- security zones.....111
  - creating.....113
  - functional.....113
  - host inbound traffic.....116
    - protocols.....122
  - interfaces.....112, 165
    - ports.....112
  - TCP-reset parameter.....125
- selection modes
  - APN.....1454
  - Mobile Station (MS).....1454
  - Network.....1454
  - verified.....1454
- self-signed certificates
  - about.....592
  - automatically generated.....593
  - manually generated.....593
- sequence-number validation.....1451
- service binding.....736, 796
- services
  - IDP, configuring.....731
  - timeout threshold.....194
- session
  - changing characteristics.....7, 97
  - controlling termination.....98
- session capacity expansion.....26
- session limits.....1067
  - source-based.....1067, 1068, 1071
- session lookup.....105
- session table floods.....1039, 1066
- session-based processing.....105
- SHA-1.....456
- shared IKE IDs.....618, 619
- show security idp application-identification
  - application-system-cache command.....803
- signature attack sample.....748
- signature custom attack.....743
  - context.....743
  - direction.....744
  - ICMP header.....747
  - IP protocol flags.....745
  - pattern.....745
  - protocol-specific parameters.....745
  - sample.....748
- TCP header.....746
- UDP header.....747
- signature database.....777
  - attack object groups.....781
  - automatic update.....788
  - manually update.....784
  - overview.....777
  - predefined attack objects.....781
  - predefined policy templates.....778
  - updating, overview.....783
  - verify.....790
  - verify load status.....790
  - verify policy compilation.....790
  - verify version.....792
  - version, overview.....783
  - See also IDP
- SIP
  - connection information.....271
  - defined.....269
  - media announcements.....271
  - messages.....269
  - multimedia sessions.....269
  - pinholes.....270
  - request methods.....274
  - response codes.....290
  - RTCP.....271
  - RTP.....271
  - SDP.....270
  - signaling.....270
- SIP ALG.....272
  - call duration and timeouts.....275
- SIP NAT
  - call setup.....283
  - defined.....283
- SIP timeouts
  - inactivity.....275
  - media inactivity.....276, 331, 831
  - session inactivity.....275
  - signaling inactivity.....275
- SNMP failover traps.....1161
- Sophos antivirus
  - configuration overview.....955
  - custom objects.....955
  - feature profile.....959
  - overview.....951
  - security policies.....966
  - utm policies.....965
- Sophos Antivirus
  - features.....952

|                                             |            |
|---------------------------------------------|------------|
| sophos Antivirus                            |            |
| comparison to kaspersky.....                | 953        |
| source IP route attack protection           |            |
| overview.....                               | 1043       |
| source NAT.....                             | 1368       |
| address pools.....                          | 1370       |
| address shifting.....                       | 1372       |
| address shifting configuration example..... | 1393       |
| addresses with PAT configuration            |            |
| example.....                                | 1383       |
| addresses without PAT configuration         |            |
| example.....                                | 1388       |
| configuration overview.....                 | 1374       |
| disabling port randomization.....           | 1411       |
| egress interface translation configuration  |            |
| example.....                                | 1375       |
| multiple rules configuration example.....   | 1398       |
| overview.....                               | 1369       |
| persistent addresses.....                   | 1373       |
| persistent NAT.....                         | 1412       |
| persistent NAT configuration overview.....  | 1414       |
| persistent NAT overview.....                | 1412       |
| port address translation.....               | 1371       |
| rules.....                                  | 1373       |
| single address translation configuration    |            |
| example.....                                | 1378       |
| STUN protocol.....                          | 1413       |
| with destination NAT configuration          |            |
| example.....                                | 1405       |
| without port address translation.....       | 1372       |
| SPUs                                        |            |
| monitoring.....                             | 1154       |
| stateful.....                               | 1017       |
| stateful and stateless data processing..... | 3, 93      |
| stateful inspection.....                    | 1017       |
| stateful packet processing .....            | 3, 93      |
| stateless firewall filters.....             | 6, 96      |
| stateless packet processing.....            | 3, 93      |
| static NAT.....                             | 1339       |
| configuration overview.....                 | 1340       |
| overview.....                               | 1339       |
| rules.....                                  | 1339       |
| single address translation configuration    |            |
| example.....                                | 1341       |
| subnet translation configuration            |            |
| example.....                                | 1345       |
| statistics                                  |            |
| application-level DDoS overview.....        | 772        |
| statistics, verifying                       |            |
| for application identification.....         | 803, 1124  |
| stream ID IP option.....                    | 1025, 1027 |
| strict source route IP option.....          | 1025       |
| Sun RPC ALG.....                            | 380        |
| call scenarios.....                         | 380        |
| defined.....                                | 380        |
| support, technical See technical support    |            |
| SYN and FIN flags protection                |            |
| overview.....                               | 1030       |
| SYN checking.....                           | 1038       |
| asymmetric routing.....                     | 1039       |
| reconnaissance hole.....                    | 1039       |
| session table floods.....                   | 1039       |
| SYN cookies.....                            | 1085       |
| SYN floods.....                             | 1074, 1078 |
| alarm threshold.....                        | 1076       |
| attack threshold.....                       | 1076       |
| destination threshold.....                  | 1076       |
| source threshold.....                       | 1076       |
| SYN cookies.....                            | 1085       |
| threshold.....                              | 1075       |
| timeout.....                                | 1076       |
| SYN fragment protection                     |            |
| overview.....                               | 1061       |
| SYN-ACK-ACK proxy floods.....               | 1072       |
| SYN-ACK-ACK-proxy flood protection          |            |
| configuration.....                          | 1072       |
| syntax conventions.....                     | xlii       |
| <b>T</b>                                    |            |
| TCP header flag attack protection           |            |
| configuration.....                          | 746        |
| overview.....                               | 1035       |
| teardrop attack protection                  |            |
| configuration.....                          | 1098       |
| overview.....                               | 1096       |
| technical support                           |            |
| contacting JTAC.....                        | xliv       |
| TEK See group VPNs                          |            |
| terminal rules                              |            |
| overview.....                               | 724        |
| setting.....                                | 725        |
| three-way handshakes.....                   | 1074       |
| time binding.....                           | 742        |
| count.....                                  | 743        |
| scope.....                                  | 742        |
| timestamp IP option.....                    | 1025, 1027 |

- traffic
    - counting.....1437
    - logging.....1437
  - transparent mode.....1436
  - transport mode.....459
  - Triple DES.....457
  - tunnel mode
    - overview.....459
- U**
- UAC**
- authentication tables
    - failover processing.....447
    - overview.....427
  - captive portal.....439
    - See also captive portal
  - certificates See UAC, device authentication
  - clustering See UAC, failover processing
  - device authentication
    - configuring.....424
    - overview.....424
  - dynamic auth table provisioning.....427
  - failover processing
    - configuring timeout actions.....447
    - connecting to cluster.....424
    - overview.....447
  - Host Checker policy enforcement.....437
  - Infranet agent
    - agentless access.....437
    - Odyssey Access Client.....437
    - overview.....421, 437
    - support information.....437
  - Infranet Controller
    - communications with Junos OS
      - Enforcer.....424
    - configuring access to.....424
    - overview.....421
  - IPsec support.....427
  - JUEP See UAC, device authentication
  - Junos OS Enforcer
    - communications with Infranet
      - Controller.....424
    - enabling.....424
    - overview.....421
  - logging.....428
  - overview.....421
  - policies
    - application services processing
      - order.....427
      - enforcement overview.....427
    - resource access policies
      - failover processing.....447
      - overview.....427
    - show commands.....427
    - test-only mode.....428
    - timeout actions See UAC, failover processing
    - user roles.....427
  - UDP header attack protection
    - configuration.....747
  - Unified Access Control See UAC
  - Unified Threat Management
    - antispam filtering.....851
    - antivirus protection, express.....929
    - antivirus protection, full.....869
    - content filtering.....969
    - licensing.....845
    - overview.....843
    - platform support.....845
    - sophos antivirus overview.....951
    - web filtering.....985
  - unknown protocol attack protection
    - overview.....1057
  - upgrading
    - chassis cluster ISSU.....1251
  - user roles See UAC, user roles
  - UTM
    - WELF support for log files.....846
- V**
- verification
    - application system cache.....803, 806, 1124
  - verified mode.....1454
  - verifying
    - chassis cluster statistics.....1247
    - chassis cluster status.....1249
    - chassis clusters.....1246
    - IDP policy compilation.....790
    - IDP policy load status.....790
    - signature database.....790
    - signature database version.....792
  - version
    - signature database.....783
  - VPNs
    - aggressive mode.....468
    - AutoKey IKE.....455

- Diffie-Hellman exchange.....455
- Diffie-Hellman groups.....455
- dynamic VPN See dynamic VPNs
- group See group VPNs
- group configuration.....662
- group VPN See group VPNs
- hub-and-spoke configuration example.....507
- main mode.....467
- Phase 1.....467
- Phase 2.....468
- policy-based configuration example.....489
- replay protection.....469
- route-based configuration example.....470

**W**

- Web Filtering.....985
  - cache.....987
  - integrated.....986
  - local.....1005
  - profiles.....988, 1006
  - verifying.....1012
- wildcards.....1454
- Windows registry changes, Access Manager.....650
- WinNuke attack protection
  - configuration.....1100
  - overview.....1098

**Z**

- zombie agents.....1065
- zones
  - functional.....113
  - security.....111