Engineering
Simplicity

Junos® OS

Security Services Administration Guide

Published
2025-07-10

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

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.

Junos® OS Security Services Administration Guide

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. 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.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

6 MAC Limiting and Move Limiting

11

Unicast Forwarding

12

Storm Control

About This Guide

The Junos operating system (Junos OS) supports the IP Security (IPsec) associations and the Internet Key Exchange (IKE) security services features. The IPsec suite provides network layer data security with functions such as authentication of origin, data integrity, confidentiality, replay protection, and non-repudiation of source. IKE defines mechanisms for key generation and exchange and manages security associations (SAs). An SA is a simplex connection that allows two hosts to communicate with each other securely by means of IPsec.

Junos OS Distributed Denial-of-Service (DDoS) protection identifies and suppresses malicious control packets while enabling legitimate control traffic to be processed. This protection enables the router to continue functioning while under attack from multiple sources. Junos OS DDoS protection provides a single point of protection management that enables network administrators to customize a profile appropriate for the control traffic on their networks.

Use the topics in this section to configure essential security services.

1

PART

Port Security

CHAPTER 1

Port Security Overview

IN THIS CHAPTER

Overview of Port Security

IN THIS SECTION

Port Security Features

Ethernet LANs are vulnerable to attacks such as address spoofing (forging) and Layer 2 denial of service (DoS) on network devices. Port security features help protect the access ports on your device against the loss of information and productivity that such attacks can cause.

Junos OS is hardened through the separation of control forwarding and services planes, with each function running in protected memory. The control-plane CPU is protected by rate limiting, routing policy, and firewall filters to ensure switch uptime even under severe attack.

Junos OS provides features to help secure ports on a device. Ports can be categorized as either trusted or untrusted. You apply policies appropriate to each category to protect ports against various types of attacks.

Access port security features such as dynamic Address Resolution Protocol (ARP) inspection, DHCP snooping, and MAC limiting are controlled through a single Junos OS CLI command. Basic port security features are enabled in the device's default configuration. You can configure additional features with minimal configuration steps. Depending on the particular feature, you can configure the feature either on VLANs or bridge domain interfaces.

Starting with Junos OS Release 18.4R1, DHCP snooping occurs on trusted ports for the following Juniper Series switches, EX2300, EX4600, and QFX5K. Prior to Junos OS Release 18.4R1, for these devices, this was true only for DHCPv6 snooping. In addition, DHCP snooping occurs on trusted ports for EX9200 Series switches, and Fusion Enterprises, that are running Junos OS Release 19.1R1 and later.

Juniper Networks EX Series Ethernet Switches provide the following hardware and software security features:

Console Port—Allows use of the console port to connect to the Routing Engine through an RJ-45 cable. You then use the command-line interface (CLI) to configure the switch.

Out-of-Band Management—A dedicated management Ethernet port on the rear panel allows out-of-band management.

Software Images—All Junos OS images are signed by Juniper Networks certificate authority (CA) with public key infrastructure (PKI).

User Authentication, Authorization, and Accounting (AAA)—Features include:

- User and group accounts with password encryption and authentication.
- Access privilege levels configurable for login classes and user templates.
- RADIUS authentication, TACACS+ authentication, or both, for authenticating users who attempt to access the switch.
- Auditing of configuration changes through system logging or RADIUS/TACACS+.

802.1X Authentication—Provides network access control. Supplicants (hosts) are authenticated when they initially connect to a LAN. Authenticating supplicants before they receive an IP address from a DHCP server prevents unauthorized supplicants from gaining access to the LAN. EX Series switches support Extensible Authentication Protocol (EAP) methods, including EAP-MD5, EAP-TLS, EAP-TTLS, and EAP-PEAP.

Port Security—Access Port security features supported on switching devices are::

- DHCP snooping—Filters and blocks ingress Dynamic Host Configuration Protocol (DHCP) server messages on untrusted ports, and builds and maintains a database of DHCP lease information, which is called the DHCP snooping database.

> **NOTE**: DHCP snooping is not enabled in the default configuration of the switching device. DHCP snooping is enabled on a VLAN or bridge domain. The details of enabling DHCP snooping depend on the particular device.

- Trusted DHCP server—Configuring the DHCP server on a trusted port protects against rogue DHCP servers sending leases. You enable this feature on an interface (port). By default, access ports are untrusted, and trunk ports are trusted. (Access ports are the switch ports that connect to Ethernet endpoints such as user PCs and laptops, servers, and printers. Trunk ports are the switch ports that connect an Ethernet switch to other switches or to routers.)

- DHCPv6 snooping—DHCP snooping for IPv6.

- DHCP option 82—Also known as the DHCP Relay Agent Information option. This DHCPv4 feature helps protect the switching device against attacks such as spoofing of IP addresses and MAC addresses and DHCP IP address starvation. Option 82 provides information about the network location of a DHCP client, and the DHCP server uses this information to implement IP addresses or other parameters for the client.

- DHCPv6 option 37—Option 37 is the remote ID option for DHCPv6 and is used to insert information about the network location of the remote host into DHCPv6 packets. You enable option 37 on a VLAN.

> **NOTE**: DHCPv6 snooping with option 37 is not supported on the MX Series.

- DHCPv6 option 18—Option 18 is the circuit ID option for DHCPv6 and is used to insert information about the client port into DHCPv6 packets. This option includes other details that can be optionally configured, such as the prefix and the interface description.

- DHCPv6 option 16—Option 16 is the vendor ID option for DHCPv6 and is used to insert information about the vendor of the client hardware into DHCPv6 packets.

- Dynamic ARP inspection (DAI)—Prevents Address Resolution Protocol (ARP) spoofing attacks. ARP requests and replies are compared against entries in the DHCP snooping database, and filtering decisions are made on the basis of the results of those comparisons. You enable DAI on a VLAN.

- IPv6 neighbor discovery inspection—Prevents IPv6 address spoofing attacks. Neighbor discovery requests and replies are compared against entries in the DHCPv6 snooping database, and filtering decisions are made on the basis of the results of those comparisons. You enable neighbor discovery inspection on a VLAN.

- IP source guard—Mitigates the effects of IP address spoofing attacks on the Ethernet LAN. With IP source guard enabled, the source IP address in the packet sent from an untrusted access interface is validated against the DHCP snooping database. If the packet cannot be validated, it is discarded. You enable IP source guard on a VLAN or bridge domain.

- IPv6 source guard—IP source guard for IPv6.
- MAC limiting—Protects against flooding of the Ethernet switching table (also known as the MAC forwarding table or Layer 2 forwarding table). You can enable MAC limiting on an interface.
- MAC move limiting—Tracks MAC movement and detects MAC spoofing on access ports. You enable this feature on a VLAN or bridge domain.
- Persistent MAC learning—Also known as sticky MAC. Persistent MAC learning enables interfaces to retain dynamically learned MAC addresses across switch reboots. You enable this feature on an interface.
- Unrestricted proxy ARP—The switch responds to all ARP messages with its own MAC address. Hosts that are connected to the switch's interfaces cannot communicate directly with other hosts. Instead, all communications between hosts go through the switch.
- Restricted proxy ARP—The switch does not respond to an ARP request if the physical networks of the source and target of the ARP request are the same. It does not matter whether the destination host has the same IP address as the incoming interface or a different (remote) IP address. An ARP request for a broadcast address elicits no reply.

Device Security—Storm control permits the switch to monitor unknown unicast and broadcast traffic and drop packets, or shut down, or temporarily disable the interface when a specified traffic level is exceeded, thus preventing packets from proliferating and degrading the LAN. You can enable storm control on access interfaces or trunk interfaces.

Encryption Standards—Supported standards include:

- 128-, 192-, and 256-bit Advanced Encryption Standard (AES)
- 56-bit Data Encryption Standard (DES) and 168-bit 3DES

SEE ALSO

Understanding DHCP Snooping (non-ELS) | **445**

Understanding DHCP Snooping (ELS) | **435**

Understanding DHCP Option 82 | **501**

IPv6 Neighbor Discovery Inspection | **605**

Understanding ARP Spoofing and Inspection | **529**

Understanding IP Source Guard for Port Security on Switches | **538**

Understanding MAC Limiting and MAC Move Limiting | **357**

802.1X for Switches Overview
Understanding Proxy ARP
Understanding Storm Control | 744

Understanding How to Protect Access Ports from Common Attacks

IN THIS SECTION

Port security features can protect the Juniper Networks EX Series and QFX10000 Ethernet Switches against various types of attacks. Protection methods against some common attacks are:

Mitigation of Ethernet Switching Table Overflow Attacks

In an overflow attack on the Ethernet switching table, an intruder sends so many requests from new MAC addresses that the table cannot learn all the addresses. When the switch can no longer use information in the table to forward traffic, it is forced to broadcast messages. Traffic flow on the switch is disrupted, and packets are sent to all hosts on the network. In addition to overloading the network with traffic, the attacker might also be able to sniff that broadcast traffic.

To mitigate such attacks, configure both a MAC limit for learned MAC addresses and some specific allowed MAC addresses. Use the MAC limiting feature to control the total number of MAC addresses that can be added to the Ethernet switching table for the specified interface or interfaces. By setting the MAC addresses that are explicitly allowed, you ensure that the addresses of network devices whose network access is critical are guaranteed to be included in the Ethernet switching table. See "Example: Protecting against Ethernet Switching Table Overflow Attacks" on page 392.

NOTE: You can also configure learned MAC addresses to persist on each interface. Used in combination with a configured MAC limit, this persistent MAC learning helps prevent traffic loss after a restart or an interface-down event and also increases port security by limiting the MAC addresses allowed on the interface.

Mitigation of Rogue DHCP Server Attacks

If an attacker sets up a rogue DHCP server to impersonate a legitimate DHCP server on the LAN, the rogue server can start issuing leases to the network's DHCP clients. The information provided to the clients by this rogue server can disrupt their network access, causing DoS. The rogue server might also assign itself as the default gateway device for the network. The attacker can then sniff the network traffic and perpetrate a man-in-the-middle attack—that is, it misdirects traffic intended for a legitimate network device to a device of its choice.

To mitigate a rogue DHCP server attack, set the interface to which that rogue server is connected as untrusted. That action will block all ingress DHCP server messages from that interface. See "Example: Protecting against Rogue DHCP Server Attacks" on page 387.

NOTE: The switch logs all DHCP server packets that are received on untrusted ports—for example:

5 untrusted DHCPOFFER received, interface ge-0/0/0.0[65], vlan v1[10] server ip/mac 12.12.12.1/00:00:00:00:01:12 offer ip/client mac 12.12.12.253/00:AA:BB:CC:DD:01

You can use these messages to detect malicious DHCP servers on the network.

NOTE: For QFX Series switches, including QFX10000, if you attach a DHCP server to an access port, you must configure the port as trusted.

Protection Against ARP Spoofing Attacks (Does not apply to QFX10000 Series Switches)

In ARP spoofing, an attacker sends faked ARP messages on the network. The attacker associates its own MAC address with the IP address of a network device connected to the switch. Any traffic sent to that IP address is instead sent to the attacker. Now the attacker can create various types of mischief, including sniffing the packets that were meant for another host and perpetrating man-in-the middle attacks. (In a man-in-the-middle attack, the attacker intercepts messages between two hosts, reads them, and perhaps alters them, all without the original hosts knowing that their communications have been compromised.)

To protect against ARP spoofing on your switch, enable both DHCP snooping and dynamic ARP inspection (DAI). DHCP snooping builds and maintains the DHCP snooping table. That table contains the MAC addresses, IP addresses, lease times, binding types, VLAN information, and interface information for the untrusted interfaces on the switch. DAI uses the information in the DHCP snooping table to validate ARP packets. Invalid ARP packets are blocked and, when they are blocked, a system log message is recorded that includes the type of ARP packet and the sender’s IP address and MAC address.

See "Example: Protecting Against ARP Spoofing Attacks" on page 482.

Protection Against DHCP Snooping Database Alteration Attacks (Does not apply to QFX10000 Series Switches)

In an attack designed to alter the DHCP snooping database, an intruder introduces a DHCP client on one of the switch's untrusted access interfaces that has a MAC address identical to that of a client on another untrusted port. The intruder acquires the DHCP lease, which results in changes to the entries in the DHCP snooping table. Subsequently, what would have been valid ARP requests from the legitimate client are blocked.

To protect against this type of alteration of the DHCP snooping database, configure MAC addresses that are explicitly allowed on the interface. See "Example: Protecting Against DHCP Snooping Database Attacks" on page 476.

Protection Against DHCP Starvation Attacks

In a DHCP starvation attack, an attacker floods an Ethernet LAN with DHCP requests from spoofed (counterfeit) MAC addresses so that the switch's trusted DHCP servers cannot keep up with requests from legitimate DHCP clients on the switch. The address space of those servers is completely used up, so they can no longer assign IP addresses and lease times to clients. DHCP requests from those clients are either dropped—that is, the result is a denial of service (DoS)—or directed to a rogue DHCP server set up by the attacker to impersonate a legitimate DHCP server on the LAN.

To protect the switch from DHCP starvation attacks, use the MAC limiting feature. Specify the maximum number of MAC addresses that the switch can learn on the access interfaces to which those clients connect. The switch's DHCP server or servers will then be able to supply the specified number of IP addresses and leases to those clients and no more. If a DHCP starvation attack occurs after the maximum number of IP addresses has been assigned, the attack will fail. See "Example: Protecting against DHCP Starvation Attacks" on page 382.

NOTE: For additional protection on EX Series switches, you can configure learned MAC addresses on each interface to persist across restarts of the switch by enabling persistent MAC learning. This persistent MAC learning both helps to prevent traffic loss after a restart and ensures that even after a restart or an interface-down event, the persistent MAC addresses are re-entered into the forwarding database rather than the switch learning new MAC addresses.

SEE ALSO

Understanding DHCP Snooping (non-ELS) | **445**

Example: Setting Up DHCP Option 82 | **508**

Understanding and Using Trusted DHCP Servers | **415**

Understanding MAC Limiting and MAC Move Limiting | 357

Understanding ARP Spoofing and Inspection | 529

Configuring Port Security (non-ELS) | 11

Configuring Port Security (ELS)

NOTE: The features described are supported on EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Configuring Port Security (non-ELS)" on page 11. For ELS details, see Using the Enhanced Layer 2 Software CLI.

Ethernet LANs are vulnerable to attacks such as address spoofing and Layer 2 denial of service (DoS) on network devices. DHCP port security features help protect the access ports on the switch against the losses of information and productivity that can result from such attacks.

The following port security features are supported for DHCPv4:

- DHCP snooping
- Dynamic ARP inspection (DAI)
- IP source guard
- DHCP option 82

The following port security features are supported for DHCPv6:

- DHCPv6 snooping
- IPv6 Neighbor discovery inspection
- IPv6 source guard
- DHCPv6 option 37, option 18 and option 16

DHCP snooping and DHCPv6 snooping are disabled by default on any VLAN. No explicit CLI configuration is used to enable DHCP snooping or DHCPv6 snooping. When you configure any of the port security features for a VLAN at the `[edit vlans *vlan-name* forwarding-options dhcp-security]` hierarchy level, DHCP snooping and DHCPv6 snooping are automatically enabled on that VLAN.

NOTE: Starting in Junos OS Release 14.1X53-D47 and 15.1R6, you can enable DHCP snooping or DHCPv6 snooping on a VLAN without configuring other port security features by configuring the `dhcp-security` CLI statement at the `[edit vlans *vlan-name* forwarding-options]` hierarchy level.

DAI, IPv6 neighbor discovery inspection, IP source guard, IPv6 source guard, DHCP option 82 and DHCPv6 options are configured per VLAN. You must configure a VLAN before configuring these DHCP port security features. See Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure).

The DHCP port security features that you specify for the VLAN apply to all the interfaces included within that VLAN. However, you can assign different attributes to an access interface or a group of access interfaces within the VLAN. The access interface or interfaces must first be configured as a group using the `group` statement at the `[edit vlans` *`vlan-name`* `forwarding-options dhcp-security]` hierarchy level. A group must have at least one interface.

> **NOTE**: Configuring a group of access interfaces on a VLAN at the `[edit vlans` *`vlan-name`* `forwarding-options dhcp-security]` hierarchy level automatically enables DHCP snooping for all interfaces in the VLAN.

Attributes that can be specified for access interfaces using the `group` statement are:

- Specifying that the interface have a static IP-MAC address (`static-ip or static-ipv6`)
- Specifying an access interface to act as a trusted interface to a DHCP server (`trusted`)
- Specifying an interface not to transmit DHCP option 82 (`no-option82`) or DHCPv6 options (`no-option37`)

> **NOTE**: Trunk interfaces are trusted by default. However, you can override this default behavior and set a trunk interface as `untrusted`.

For additional details, see:

- "Enabling Dynamic ARP Inspection (ELS)" on page 531
- "IPv6 Neighbor Discovery Inspection" on page 605
- "Configuring IP Source Guard (ELS)" on page 545
- "Setting Up DHCP Option 82 on the Switch with No Relay (ELS)" on page 517

You can override the general port security settings for the VLAN by configuring a group of access interfaces within that VLAN. For details, see:

- "Configuring Static DHCP IP Addresses for DHCP snooping (ELS)" on page 460
- "Enabling a Trusted DHCP Server (ELS)" on page 416

SEE ALSO

Port Security Features | 2
Understanding DHCP Snooping (non-ELS) | 445

Configuring Port Security (non-ELS)

IN THIS SECTION

Ethernet LANs are vulnerable to attacks such as address spoofing and Layer 2 denial of service (DoS) on network devices. Port security features such as DHCP snooping, DAI (dynamic ARP inspection), MAC limiting, MAC move limiting, and persistent MAC learning, as well as trusted DHCP server, help protect the access ports on the switch against the loss of information and productivity that such attacks can cause.

Depending on the particular feature, you can configure the port security feature either on:

- VLANs—A specific VLAN or all VLANs

- Interfaces—A specific interface or all interfaces

NOTE: If you configure one of the port security features on all VLANs or all interfaces, the switch software enables that port security feature on all VLANs and all interfaces that are not explicitly configured with other port security features.

However, if you do explicitly configure one of the port security features on a specific VLAN or on a specific interface, you must explicitly configure any additional port security features that you want to apply to that VLAN or interface. Otherwise, the switch software automatically applies the default values for the feature.

For example, if you disable DHCP snooping on all VLANs and decide to explicitly enable IP source guard only on a specific VLAN, you must also explicitly enable DHCP snooping on that specific VLAN. Otherwise, the default value of no DHCP snooping applies to that VLAN.

To configure port security features by using the CLI:

Enabling DHCP Snooping

You can configure DHCP snooping to enable the device to monitor DHCP messages received, ensure that hosts use only the IP addresses that are assigned to them, and allow access only to authorized DHCP servers.

To enable DHCP snooping:

- On a specific VLAN:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan vlan-name  examine-dhcp
```

- On all VLANs:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan all examine-dhcp
```

To enable DHCPv6 snooping:

- On a specific VLAN:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan vlan-name  examine-dhcpv6
```

- On all VLANs:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan all examine-dhcpv6
```

Enabling Dynamic ARP Inspection (DAI)

You can enable DAI to protect against ARP snooping. To enable DAI:

- On a single VLAN:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan vlan-name arp-inspection
```

- On all VLANs:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan all arp-inspection
```

Enabling IPv6 Neighbor Discovery Inspection

You can enable neighbor discovery inspection to protect against IPv6 address spoofing.

- To enable neighbor discovery on a single VLAN:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan vlan-name neighbor-discovery-inspection
```

- To enable neighbor discovery on all VLANs:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan all neighbor-discovery-inspection
```

Limiting Dynamic MAC Addresses on an Interface

Limit the number of dynamic MAC addresses allowed on an interface and specify the action to take if the limit is exceeded:

- On a single interface:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface interface-name mac-limit limit action action
```

- On all interfaces:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface all mac-limit limit action action
```

Enabling Persistent MAC Learning on an Interface

You can configure learned MAC addresses to persist on an interface across restarts of the switch:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface interface-name persistent-learning
```

Limiting MAC Address Movement

You can limit the number of times a MAC address can move from its original interface in 1 second:

- On a single VLAN:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan vlan-name mac-move-limit limit action action
```

- On all VLANs:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan all mac-move-limit limit action action
```

Restricting a VoIP Client MAC Address in a VoIP VLAN

To restrict a VoIP client MAC address from being learned in a configured VoIP VLAN:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interfaceinterface-name voip-mac-exlusive
```

Any MAC address learned on that interface for the VoIP VLAN is not learned on a data VLAN with that same interface. If a MAC address has been learned on a data VLAN interface and then the MAC address is learned on a VoIP VLAN with that same interface, the MAC address is removed from the data VLAN interface.

Configuring Trusted DHCP Servers on an Interface

Configure a trusted DHCP server on an interface:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface interface-name dhcp-trusted
```

RELATED DOCUMENTATION

- Configuring Autorecovery for Port Security Events | **758**
- Example: Configuring Port Security (non-ELS) | **15**
- Example: Protecting Against Address Spoofing and Layer 2 DoS Attacks | **464**
- Monitoring Port Security
- Port Security Features | **2**
- *secure-access-port*

Example: Configuring Port Security (non-ELS)

IN THIS SECTION

You can configure DHCP snooping, dynamic ARP inspection (DAI), MAC limiting, persistent MAC learning, and MAC move limiting on the untrusted ports of switches to protect the switches and the Ethernet LAN against address spoofing and Layer 2 denial-of-service (DoS) attacks. You can also configure a trusted DHCP server and specific (allowed) MAC addresses for the switch interfaces.

> **NOTE**: The switches used in this example do not support the ELS configuration style. For information on configuring port security on ELS switches, see "Configuring Port Security (ELS)" on page 9.

This example describes how to configure basic port security features on a switch:

Requirements

This example uses the following hardware and software components:

- One EX Series or QFX Series.
- Junos OS Release 11.4 or later for EX Series switches or Junos OS Release 12.1 or later for the QFX Series
- A DHCP server to provide IP addresses to network devices on the switch

Before you configure basic port security features, be sure you have:

- Connected the DHCP server to the switch.
- Configured a VLAN on the switch. See the task for your platform:
  - Configuring VLANs for EX Series Switches

NOTE: In this example, the DHCP server and its clients are all members of a single VLAN on the switch.

Overview and Topology

IN THIS SECTION

Ethernet LANs are vulnerable to address spoofing and DoS attacks on network devices. To protect the devices from such attacks, you can configure:

- DHCP snooping to validate DHCP server messages
- DAI to protect against MAC spoofing
- MAC limiting to constrain the number of MAC addresses the switch adds to its MAC address cache
- MAC move limiting to help prevent MAC spoofing
- Persistent MAC learning (sticky MAC) to constrain the MAC addresses that can be learned on an interface to the first ones learned, even after a reboot of the switch

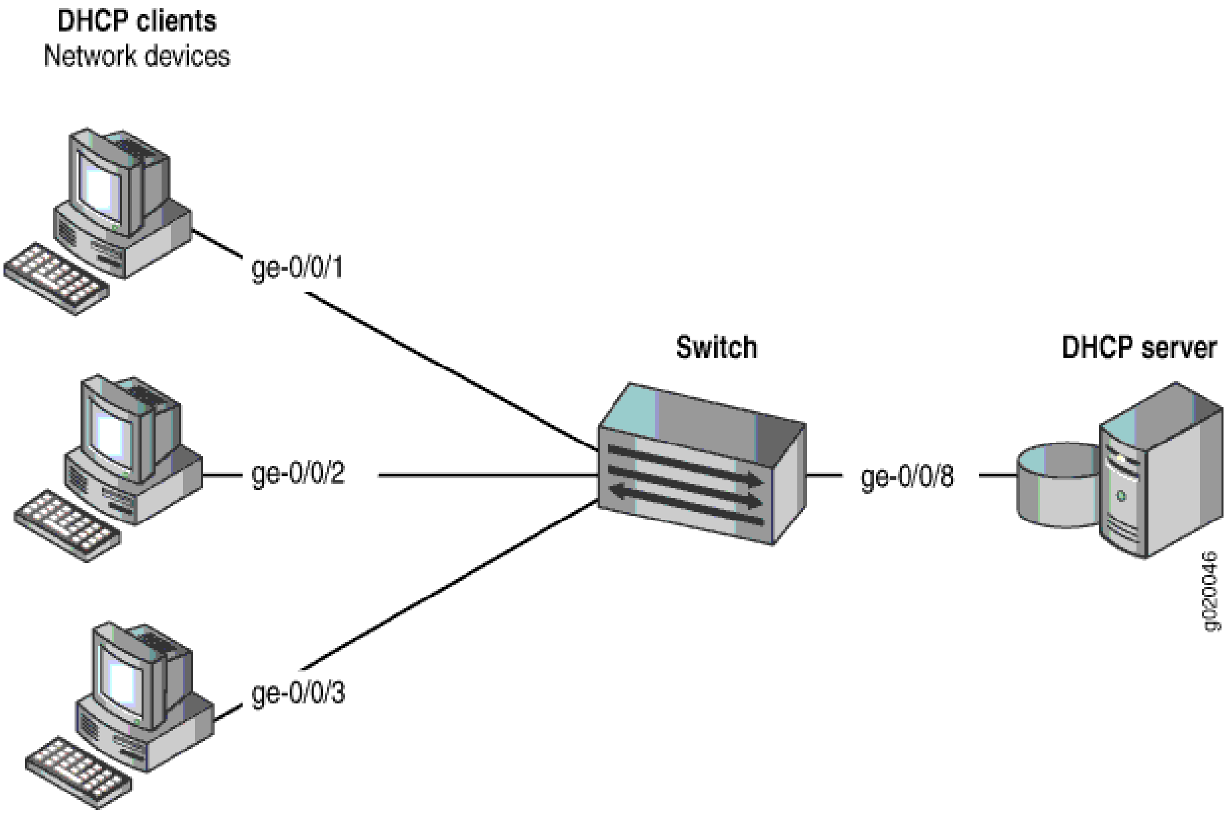
- Trusted DHCP server configured on a trusted port to protect against rogue DHCP servers sending leases

This example shows how to configure these security features on a switch connected to a DHCP server.

The setup for this example includes the VLAN **employee-vlan** on the switch. Figure 1 on page 17 illustrates the topology for this example.

Topology

Figure 1: Network Topology for Basic Port Security



The components of the topology for this example are shown in Table 1 on page 17.

Table 1: Components of the Port Security Topology

Properties	Settings
Switch hardware	One EX Series or QFX series switch

Table 1: Components of the Port Security Topology *(Continued)*

Properties	Settings
VLAN name and ID	**employee-vlan**, tag **20**
VLAN subnets	**192.0.2.16/28** **192.0.2.17** through **192.0.2.30** **192.0.2.31** is subnet's broadcast address
Interfaces in **employee-vlan**	**ge-0/0/1, ge-0/0/2, ge-0/0/3, ge-0/0/8**
Interface for DHCP server	**ge-0/0/8**

In this example, the switch is initially configured with the default port security setup. In the default switch configuration:

- Secure port access is activated on the switch.
- DHCP snooping and DAI are disabled on all VLANs.
- All access ports are untrusted, and all trunk ports are trusted for DHCP snooping.

In the configuration tasks for this example, you set the DHCP server as trusted; you enable DHCP snooping, DAI, and MAC move limiting on a VLAN; you set a value for a MAC limit on some interfaces; you configure some specific (allowed) MAC addresses on an interface; and you configure persistent MAC learning on an interface.

Configuration

IN THIS SECTION

To configure basic port security on a switch whose DHCP server and client ports are in a single VLAN:

Procedure

CLI Quick Configuration

To quickly configure basic port security on the switch, copy the following commands and paste them into the switch terminal window:

```
[edit ethernet-switching-options secure-access-port]
set interface ge-0/0/1 mac-limit 4
set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:80
set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:81
set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:83
set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:85
set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:88
set interface ge-0/0/2 mac-limit 4
set interface ge-0/0/1 persistent-learning
set interface ge-0/0/8 dhcp-trusted
set vlan employee-vlan arp-inspection
 set vlan employee-vlan examine-dhcp
 set vlan employee-vlan mac-move-limit 5
```

Step-by-Step Procedure

Configure basic port security on the switch:

1. Enable DHCP snooping on the VLAN:

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set vlan employee-vlan examine-dhcp
   ```

2. Specify the interface (port) from which DHCP responses are allowed:

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set interface ge-0/0/8 dhcp-trusted
   ```

3. Enable dynamic ARP inspection (DAI) on the VLAN:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan employee-vlan arp-inspection
```

4. Configure a MAC limit of 4 and use the default action, **drop**. (Packets are dropped, and the MAC address is not added to the Ethernet switching table if the MAC limit is exceeded on the interfaces):

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface ge-0/0/1 mac-limit 4
user@switch# set interface ge-0/0/2 mac-limit 4
```

5. Allow learned MAC addresses for a particular interface to persist across restarts of the switch and interface-down events by enabling persistent MAC learning:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface ge-0/0/1 persistent-learning
```

6. Configure a MAC move limit of 5 and use the default action, **drop**. (Packets are dropped, and the MAC address is not added to the Ethernet switching table if a MAC address has exceeded the MAC move limit):

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan employee-vlan mac-move-limit 5
```

7. Configure allowed MAC addresses:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:80
user@switch# set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:81
user@switch# set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:83
user@switch# set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:85
user@switch# set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:88
```

Results

Check the results of the configuration:

```
[edit ethernet-switching-options secure-access-port]
user@switch# show
interface ge-0/0/1.0 {
    mac-limit 4;
    persistent-learning;
}
interface ge-0/0/2.0 {
    allowed-mac [ 00:05:85:3a:82:80 00:05:85:3a:82:81 00:05:85:3a:82:83 00:05:85:3a:82:85
00:05:85:3a:82:88 ];
    mac-limit 4;
}
interface ge-0/0/8.0 {
    dhcp-trusted;
}
vlan employee-vlan {
    arp-inspection
    examine-dhcp;
    mac-move-limit 5;
}
```

Verification

IN THIS SECTION

To confirm that the configuration is working properly:

Verifying That DHCP Snooping Is Working Correctly on the Switch

Purpose

Verify that DHCP snooping is working on the switch.

Action

Send some DHCP requests from network devices (here they are DHCP clients) connected to the switch.

Display the DHCP snooping information when the interface on which the DHCP server connects to the switch is trusted. The following output results when requests are sent from the MAC addresses and the server has provided the IP addresses and leases:

```
user@switch> show dhcp snooping binding
DHCP Snooping Information:
MAC Address         IP Address    Lease    Type     VLAN          Interface


-----------------   ----------    -----    ----     ----          ---------


00:05:85:3A:82:77    192.0.2.17    600    dynamic  employee—vlan   ge-0/0/1.0


00:05:85:3A:82:79    192.0.2.18    653    dynamic  employee—vlan   ge-0/0/1.0


00:05:85:3A:82:80    192.0.2.19    720    dynamic  employee—vlan   ge-0/0/2.0


00:05:85:3A:82:81    192.0.2.20    932    dynamic  employee—vlan   ge-0/0/2.0


00:05:85:3A:82:83    192.0.2.21   1230    dynamic  employee—vlan   ge-0/0/2.0


00:05:85:27:32:88    192.0.2.22   3200    dynamic  employee—vlan   ge-0/0/2.0
```

Meaning

When the interface on which the DHCP server connects to the switch has been set to trusted, the output (see preceding sample) shows, for each MAC address, the assigned IP address and lease time—that is, the time, in seconds, remaining before the lease expires.

If the DHCP server had been configured as untrusted, no entries would be added to the DHCP snooping database, and nothing would be shown in the output of the `show dhcp snooping binding` command.

Verifying That DAI Is Working Correctly on the Switch

Purpose

Verify that DAI is working on the switch.

Action

Send some ARP requests from network devices connected to the switch.

Display the DAI information:

```
user@switch> show arp inspection statistics
ARP inspection statistics:
Interface        Packets received  ARP inspection pass   ARP inspection failed
---------------  ---------------   -------------------- ---------------------
ge-0/0/1.0                      7                    5                     2
ge-0/0/2.0                     10                   10                     0
ge-0/0/3.0                     12                   12                     0
```

Meaning

The sample output shows the number of ARP packets received and inspected per interface, with a listing of how many packets passed and how many failed the inspection on each interface. The switch compares the ARP requests and replies against the entries in the DHCP snooping database. If a MAC address or IP address in the ARP packet does not match a valid entry in the database, the packet is dropped.

Verifying That MAC Limiting, MAC Move Limiting, and Persistent MAC Learning Are Working Correctly on the Switch

Purpose

Verify that MAC limiting, MAC move limiting, and persistent MAC learning are working on the switch.

Action

Suppose that two packets have been sent from hosts on **ge-0/0/1** and five packets from hosts on **ge-0/0/2**, with both interfaces set to a MAC limit of **4** with the default action **drop** and **ge-0/0/1** enabled for persistent MAC learning.

Display the MAC addresses learned:

```
user@switch> show ethernet-switching table
Ethernet-switching table:  7 entries, 4 learned, 2 persistent entries

  VLAN                 MAC address        Type         Age    Interfaces

  employee-vlan        *                  Flood          -    All-members
  employee-vlan        00:05:85:3A:82:77   Persistent     0    ge-0/0/1.0
  employee-vlan        00:05:85:3A:82:79   Persistent     0    ge-0/0/1.0
  employee-vlan        00:05:85:3A:82:80   Learn          0    ge-0/0/2.0
  employee-vlan        00:05:85:3A:82:81   Learn          0    ge-0/0/2.0
  employee-vlan        00:05:85:3A:82:83   Learn          0    ge-0/0/2.0
  employee-vlan        00:05:85:3A:82:85   Learn          0    ge-0/0/2.0
```

Now suppose packets have been sent from two of the hosts on **ge-0/0/2** after they have been moved to other interfaces more than five times in 1 second, with **employee-vlan** set to a MAC move limit of **5** with the default action **drop**.

Display the MAC addresses in the table:

```
user@switch> show ethernet-switching table
```

```
Ethernet-switching table:  7 entries, 2 learned, 2 persistent entries

  VLAN                 MAC address        Type         Age    Interfaces

  employee-vlan        *                  Flood          -    All-members
  employee-vlan        00:05:85:3A:82:77   Persistent     0    ge-0/0/1.0
  employee-vlan        00:05:85:3A:82:79   Persistent     0    ge-0/0/1.0
  employee-vlan        00:05:85:3A:82:80   Learn          0    ge-0/0/2.0
  employee-vlan        00:05:85:3A:82:81   Learn          0    ge-0/0/2.0
  employee-vlan        *                  Flood          -    ge-0/0/2.0
  employee-vlan        *                  Flood          -    ge-0/0/2.0
```

Meaning

The first sample output shows that with a MAC limit of **4** for each interface, the fifth MAC address on **ge-0/0/2** was not learned because it exceeded the MAC limit. The second sample output shows that MAC addresses for three of the hosts on **ge-/0/0/2** were not learned, because the hosts had been moved back more than five times in 1 second.

Interface ge-0/0/1.0 was enabled for persistent MAC learning, so the MAC addresses associated with this interface are of the type **persistent**.

Verifying That Allowed MAC Addresses Are Working Correctly on the Switch

Purpose

Verify that allowed MAC addresses are working on the switch.

Action

Display the MAC cache information after five allowed MAC addresses have been configured on interface **ge-0/0/2**:

```
user@switch> show ethernet-switching table
Ethernet-switching table:  5 entries, 4 learned

  VLAN                 MAC address        Type         Age    Interfaces


  employee-vlan        00:05:85:3A:82:80   Learn          0    ge-0/0/2.0
  employee-vlan        00:05:85:3A:82:81   Learn          0    ge-0/0/2.0
  employee-vlan        00:05:85:3A:82:83   Learn          0    ge-0/0/2.0
  employee-vlan        00:05:85:3A:82:85   Learn          0    ge-0/0/2.0
  employee-vlan        *                   Flood          -    ge-0/0/2.0
```

Meaning

Because the MAC limit value for this interface has been set to **4**, only four of the five configured allowed addresses are learned.

SEE ALSO

Example: Protecting against Rogue DHCP Server Attacks | **387**

Example: Protecting Against ARP Spoofing Attacks | **482**

Example: Protecting Against DHCP Snooping Database Attacks | **476**

Configuring Port Security (non-ELS) | **11**

2

PART

IPSec

CHAPTER 2

Understanding IPsec and Security Associations

IN THIS CHAPTER

IPSec Terms and Acronyms

IN THIS SECTION

Triple Data Encryption Standard (3DES)

An enhanced DES algorithm that provides 168-bit encryption by processing data three times with three different keys.

Adaptive Services PIC

A next-generation Physical Interface Card (PIC) that provides IPsec services and other services, such as Network Address Translation (NAT) and stateful firewall, on M Series and T Series platforms.

Advanced Encryption Standard (AES)

A next-generation encryption method that is based on the Rijndael algorithm and uses a 128-bit block, three different key sizes (128, 192, and 256 bits), and multiple rounds of processing to encrypt data.

authentication header (AH)

A component of the IPsec protocol used to verify that the contents of a packet have not changed (data integrity), and to validate the identity of the sender (data source authentication). For more information about AH, see RFC 2402.

certificate authority (CA)

A trusted third-party organization that generates, enrolls, validates, and revokes digital certificates. The CA guarantees the identity of a user and issues public and private keys for message encryption and decryption.

certificate revocation list (CRL)

A list of digital certificates that have been invalidated before their expiration date, including the reasons for their revocation and the names of the entities that have issued them. A CRL prevents usage of digital certificates and signatures that have been compromised.

cipher block chaining (CBC)

A cryptographic method that encrypts blocks of ciphertext by using the encryption result of one block to encrypt the next block. Upon decryption, the validity of each block of ciphertext depends on the validity of all the preceding ciphertext blocks. For more information on how to use CBC with DES and ESP to provide confidentiality, see RFC 2405.

Data Encryption Standard (DES)

An encryption algorithm that encrypts and decrypts packet data by processing the data with a single shared key. DES operates in increments of 64-bit blocks and provides 56-bit encryption.

digital certificate

Electronic file that uses private and public key technology to verify the identity of a certificate creator and distribute keys to peers.

ES PIC

A PIC that provides first-generation encryption services and software support for IPsec on M Series and T Series platforms.

Encapsulating Security Payload (ESP)

A component of the IPsec protocol used to encrypt data in an IPv4 or IPv6 packet, provide data integrity, and ensure data source authentication. For more information about ESP, see RFC 2406.

Hashed Message Authentication Code (HMAC)

A mechanism for message authentication using cryptographic hash functions. HMAC can be used with any iterative cryptographic hash function, such as MD5 or SHA-1, in combination with a secret shared key. For more information on HMAC, see RFC 2104.

Internet Key Exchange (IKE)

Establishes shared security parameters for any hosts or routers using IPsec. IKE establishes the SAs for IPsec. For more information about IKE, see RFC 2407.

Message Digest 5 (MD5)

An authentication algorithm that takes a data message of arbitrary length and produces a 128-bit message digest. For more information, see RFC 1321.

Perfect Forward Secrecy (PFS)

Provides additional security by means of a Diffie-Hellman shared secret value. With PFS, if one key is compromised, previous and subsequent keys are secure because they are not derived from previous keys.

public key infrastructure (PKI)

A trust hierarchy that enables users of a public network to securely and privately exchange data through the use of public and private cryptographic key pairs that are obtained and shared with peers through a trusted authority.

registration authority (RA)

A trusted third-party organization that acts on behalf of a CA to guarantee the identity of a user.

Routing Engine

A PCI-based architectural portion of a Junos OS-based router that handles the routing protocol process, the interface process, some of the chassis components, system management, and user access.

security association (SA)

Specifications that must be agreed upon between two network devices before IKE or IPsec are allowed to function. SAs primarily specify protocol, authentication, and encryption options.

Security Association Database (SADB)

A database where all SAs are stored, monitored, and processed by IPsec.

Secure Hash Algorithm 1 (SHA-1)

An authentication algorithm that takes a data message of less than 264 bits in length and produces a 160-bit message digest. For more information on SHA-1, see RFC 3174.

Secure Hash Algorithm 2 (SHA-2)

A successor to the SHA-1 authentication algorithm that includes a group of SHA-1 variants (SHA-224, SHA-256, SHA-384, and SHA-512). SHA-2 algorithms use larger hash sizes and are designed to work with enhanced encryption algorithms such as AES.

Security Policy Database (SPD)

A database that works with the SADB to ensure maximum packet security. For inbound packets, IPsec checks the SPD to verify if the incoming packet matches the security configured for a particular policy. For outbound packets, IPsec checks the SPD to see if the packet needs to be secured.

Security Parameter Index (SPI)

An identifier that is used to uniquely identify an SA at a network host or router.

Simple Certificate Enrollment Protocol (SCEP)

A protocol that supports CA and registration authority (RA) public key distribution, certificate enrollment, certificate revocation, certificate queries, and certificate revocation list (CRL) queries.

Security Associations Overview

To use *IPsec* security services, you create *SA*s between hosts. An SA is a simplex connection that allows two hosts to communicate with each other securely by means of IPsec. There are two types of SAs: manual and dynamic.

- Manual SAs require no negotiation; all values, including the keys, are static and specified in the configuration. Manual SAs statically define the Security Parameter Index (*SPI*) values, algorithms, and keys to be used, and require matching configurations on both ends of the tunnel. Each peer must have the same configured options for communication to take place.

- Dynamic SAs require additional configuration. With dynamic SAs, you configure *IKE* first and then the SA. IKE creates dynamic security associations; it negotiates SAs for IPsec. The IKE configuration defines the algorithms and keys used to establish the secure IKE connection with the peer security gateway. This connection is then used to dynamically agree upon keys and other data used by the dynamic IPsec SA. The IKE SA is negotiated first and then used to protect the negotiations that determine the dynamic IPsec SAs.

- Set up user-level tunnels or SAs, including tunnel attribute negotiations and key management. These tunnels can also be refreshed and terminated on top of the same secure channel.

The Junos OS implementation of IPsec supports two modes of security (*transport mode* and *tunnel mode*).

RELATED DOCUMENTATION

IKE Key Management Protocol Overview | 64

IPsec Requirements for Junos-FIPS | 66

[edit security] Hierarchy Level

IKE Key Management Protocol Overview

IKE is a key management protocol that creates dynamic *SA*s; it negotiates SAs for *IPsec*. An IKE configuration defines the algorithms and keys used to establish a secure connection with a peer security gateway.

IKE does the following:

- Negotiates and manages IKE and IPsec parameters
- Authenticates secure key exchange
- Provides mutual peer authentication by means of shared secrets (not passwords) and public keys
- Provides identity protection (in main mode)

IKE occurs over two phases. In the first phase, it negotiates security attributes and establishes shared secrets to form the bidirectional IKE SA. In the second phase, inbound and outbound IPsec SAs are established. The IKE SA secures the exchanges in the second phase. IKE also generates keying material, provides Perfect Forward Secrecy, and exchanges identities.

NOTE: Starting in Junos OS Release 14.2, when you perform an SNMP walk of the jnxIkeTunnelEntry object in the jnxIkeTunnelTable table, the `Request failed: OID not increasing` error message might be generated. This problem occurs only when simultaneous Internet Key Exchange security associations (IKE SAs) are created, which occurs when both ends of the SA initiate IKE SA negotiations at the same time. When an SNMP MIB walk is performed to display IKE SAs, the snmpwalk tool expects the object identifiers (OIDs) to be in increasing order. However, in the case of simultaneous IKE SAs, the OIDs in the SNMP table might not be in increasing order. This behavior occurs because the tunnel IDs, which are part of the OIDs, are allocated based on the initiator of the IKE SA, which can be on either side of the IKE tunnel.

The following is an example of an SNMP MIB walk that is performed on IKE simultaneous SAs:

```
jnxIkeTunLocalRole."ipsec_ss_cust554".ipv4."192.0.2.41".47885 = INTEGER:
responder(2)   >>> This is Initiator SA
```

```
jnxIkeTunLocalRole."ipsec_ss_cust554".ipv4."192.0.2.41".47392 = INTEGER:
initiator(1)   >>> This is Responder's SA
```

The OID comparison fails when the SNMP walk is tunnel ID (47885 and 47392). It cannot be ensured when an SNMP walk is performed that the tunnel IDs are in increasing order because tunnels might be initiated from either side.

To work around this problem, the SNMP MIB walk contains an option, -Cc, to disable check for increasing OIDs. The following is an example of the MIB walk performed on the jnxIkeTunnelEntry table with the -Cc option:

```
snmpwalk -Os -Cc -c public -v 1 vira jnxIkeTunnelEntry
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.2	Starting in Junos OS Release 14.2, when you perform an SNMP walk of the jnxIkeTunnelEntry object in the jnxIkeTunnelTable table, the `Request failed: OID not increasing` error message might be generated.

RELATED DOCUMENTATION

Security Associations Overview | 63

IPsec Requirements for Junos-FIPS | 66

[edit security] Hierarchy Level

IPsec Requirements for Junos-FIPS

In a *Junos-FIPS* environment, hardware configurations with two *Routing Engine*s must be configured to use *IPsec* and a private routing instance for all communications between the Routing Engines. IPsec communication between the Routing Engines and AS II *FIPS PIC*s is also required.

RELATED DOCUMENTATION

Security Associations Overview | 63

IKE Key Management Protocol Overview | 64

[edit security] Hierarchy Level

CHAPTER 3

IPsec Configurations and Examples

IN THIS CHAPTER

Considering General IPsec Issues

Before you configure IPsec, it is helpful to understand some general guidelines.

- IPv4 and IPv6 traffic and tunnels—You can configure IPsec tunnels to carry traffic in the following ways: IPv4 traffic traveling over IPv4 IPsec tunnels, IPv6 traffic traveling over IPv4 IPsec tunnels, IPv4 traffic traveling over IPv6 IPsec tunnels, and IPv6 traffic traveling over IPv6 IPsec tunnels.

- Configuration syntax differences between the AS and MultiServices PICs and the ES PIC—There are slight differences in the configuration statements and operational mode commands that are used with the PICs that support IPsec. As a result, the syntax for the AS and MultiServices PICs cannot be used interchangeably with the syntax for the ES PIC. However, the syntax for one type of PIC can be converted to its equivalent syntax on the other PIC for interoperability. The syntax differences are highlighted in Table 2 on page 37.

- Configuring keys for authentication and encryption—When preshared keys are required for authentication or encryption, you must use the guidelines shown in Table 3 on page 39 to implement the correct key size.

- Rejection of weak and semiweak keys—The DES and 3DES encryption algorithms will reject weak and semiweak keys. As a result, do not create and use keys that contain the patterns listed in Table 4 on page 40.

Table 2: Comparison of IPsec Configuration Statements and Operational Mode Commands for the AS and MultiServices PICs and ES PIC

AS and MultiServices PICs Statements and Commands	ES PIC Statements and Commands
Configuration Mode Statements	
[edit service-set *name*]	–
[edit services ipsec-vpn ike] • policy {...} • proposal {...}	[edit security ike] • policy {...} • proposal {...}
[edit services ipsec-vpn ipsec] • policy {...} • proposal {...}	[edit security ipsec] • policy {...} • proposal {...}
[edit services ipsec-vpn rule *rule-name*] • remote-gateway *address*	[edit interface es- *fpc* / *pic* / *port*] • tunnel destination *address*
[edit services ipsec-vpn rule *rule-name* term *term-name*] • from *match-conditions* {...} then dynamic {...} • from *match-conditions* {...} then manual {...}	[edit security ipsec] • security-association *name* dynamic {...} • security-association *name* manual {...}
[edit services ipsec-vpn rule-set]	–
[edit services service-set ipsec-vpn] • local-gateway *address*	[edit interface es- *fpc* / *pic* / *port*] • tunnel source *address*
Operational Mode Commands	

Table 2: Comparison of IPsec Configuration Statements and Operational Mode Commands for the AS and MultiServices PICs and ES PIC *(Continued)*

AS and MultiServices PICs Statements and Commands	ES PIC Statements and Commands
clear security pki ca-certificate	–
clear security pki certificate-request	–
clear security pki local-certificate	–
clear services ipsec-vpn certificates	–
request security pki ca-certificate enroll	request security certificate (unsigned)
request security pki ca-certificate load	request system certificate add
request security pki generate-certificate-request	–
request security pki generate-key-pair	request security key-pair
request security pki local-certificate enroll	request security certificate (signed)
request security pki local-certificate load	request system certificate add
`show security pki ca-certificate`	`show system certificate`
`show security pki certificate-request`	–
`show security pki crl`	–
`show security pki local-certificate`	`show system certificate`
`show services ipsec-vpn certificates`	`show ipsec certificates`

Table 2: Comparison of IPsec Configuration Statements and Operational Mode Commands for the AS and MultiServices PICs and ES PIC *(Continued)*

AS and MultiServices PICs Statements and Commands	ES PIC Statements and Commands
`show services ipsec-vpn ike security-associations`	`show ike security-associations`
`show services ipsec-vpn ipsec security-associations`	`show ipsec security-associations`

Table 3: Authentication and Encryption Key Lengths

	Number of Hexadecimal Characters	Number of ASCII Characters
Authentication		
HMAC-MD5-96	32	16
HMAC-SHA1-96	40	20
Encryption		
AES-128-CBC	16	32
AES-192-CBC	24	48
AES-256-CBC	32	64
DES-CBC	16	8
3DES-CBC	48	24

Table 4: Weak and Semiweak Keys

Weak Keys			
0101	0101	0101	0101
1F1F	1F1F	1F1F	1F1F
E0E0	E0E0	E0E0	E0E0
FEFE	FEFE	FEFE	FEFE
Semiweak Keys			
01FE	01FE	01FE	01FE
1FE0	1FE0	0EF1	0EF1
01E0	01E0	01F1	01F1
1FFE	1FFE	0EFE	0EFE
011F	011F	010E	010E
E0FE	E0FE	F1FE	F1FE
FE01	FE01	FE01	FE01
E01F	E01F	F10E	F10E
E001	E001	F101	F101
FEF1	FEF1	FE0E	FE0E
1F01	1F01	0E01	0E01

Table 4: Weak and Semiweak Keys *(Continued)*

Weak Keys			
FEE0	FEE0	FEF1	FEF1

Keep in mind the following limitations of IPsec services on the AS PIC:

- The AS PIC does not transport packets containing IPv4 options across IPsec tunnels. If you try to send packets containing IP options across an IPsec tunnel, the packets are dropped. Also, if you issue a `ping` command with the **record-route** option across an IPsec tunnel, the `ping` command fails.

- The AS PIC does not transport packets containing the following IPv6 options across IPsec tunnels: hop-by-hop, destination (Type 1 and 2), and routing. If you try to send packets containing these IPv6 options across an IPsec tunnel, the packets are dropped.

- Destination class usage is not supported with IPsec services on the AS PIC.

IPsec Configuration for an ES PIC Overview

IN THIS SECTION

IPsec Configuration for an ES PIC Overview

IP Security (IPsec) provides a secure way to authenticate senders and encrypt *IPv4* and *IPv6* traffic between network devices, such as routers and hosts. The following sections show how to configure IPsec for an ES PIC.

The key management process (**kmd**) provides IPsec authentication services for ES PICs. The key management process starts only when IPsec is configured on the router.

SEE ALSO

- Configuring Manual SAs on an ES PIC | 42
- Configuring a Digital Certificate for IKE on an ES PIC | 43
- Configuring Security Associations for IPsec on an ES PIC | 44
- Configuring an IKE Proposal for Dynamic SAs
- Example: Configuring an IKE Proposal

Configuring Manual SAs on an ES PIC

To define a manual security association (*SA*) configuration for an ES *PIC*, include at least the following statements at the `[edit security ipsec]` hierarchy level:

```
[edit security ipsec]
security-association sa-name {
    manual {
        direction (inbound | outbound | bidirectional) {
            authentication {
                algorithm (hmac-md5-96 | hmac-sha1-96);
                key (ascii-text key | hexadecimal key);
            }
            encryption {
                algorithm (des-cbc | 3des-cbc);
                key (ascii-text key | hexadecimal key);
            }
            protocol (ah | esp | bundle);
            spi spi-value;
        }
    }
}
```

SEE ALSO

- IPsec Configuration for an ES PIC Overview | 41

Configuring IKE Requirements on an ES PIC

To define an *IKE* configuration for an ES *PIC*, include at least the following statements at the `[edit security]` hierarchy level:

```
[edit security ike]
proposal ike-proposal-name {
    authentication-method (dsa-signatures | pre-shared-keys | rsa-signatures);
    dh-group (group1 | group2);
    encryption-algorithm (3des-cbd | des-cbc | aes-128-cbc | aes-192-cbc | aes-256-cbc);
}
policy ike-peer-address {
    proposals [ ike-proposal-names ];
    pre-shared-key (ascii-text key | hexadecimal key);
}
```

SEE ALSO

IPsec Configuration for an ES PIC Overview | **41**

Configuring a Digital Certificate for IKE on an ES PIC

To define a *digital certificate* configuration for *IKE* for an encryption interface on M Series and T Series routers, include at least the following statements at the `[edit security certificates]` and `[edit security ike]` hierarchy levels:

```
[edit security]
certificates {
    certification-authority ca-profile-name {
        ca-name ca-identity;
        crl filename;
        enrollment-url url-name;
        file certificate-filename;
        ldap-url url-name;
    }
}
ike {
    policy ike-peer-address {
        local-certificate certificate-filename;
        local-key-pair private-public-key-file;
        proposal [ ike-proposal-names ];
    }
    proposal ike-proposal-name {
        authentication-method rsa-signatures;
    }
}
```

SEE ALSO

IPsec Configuration for an ES PIC Overview | 41

Configuring Security Associations for IPsec on an ES PIC

IN THIS SECTION

To use *IPsec* security services, you create an *SA* between hosts. An SA is a simplex connection that allows two hosts to communicate with each other securely by means of IPsec. You can configure two types of SAs:

- Manual—Requires no negotiation; all values, including the keys, are static and specified in the configuration. As a result, each peer must have the same configured options for communication to take place. For information about how to configure a manual SA, see "Configuring Manual IPsec Security Associations for an ES PIC" on page 47.

- Dynamic—Specify proposals to be negotiated with the tunnel peer. The keys are generated as part of the negotiation and therefore do not need to be specified in the configuration. The dynamic SA includes one or more **proposal** statements, which allow you to prioritize a list of protocols and algorithms to be negotiated with the peer. For information about how to configure a dynamic SA, see "Associating the Configured Security Association with a Logical Interface" on page 218.

NOTE: The Junos OS does not perform a commit check when an SA name referenced in the Border Gateway Protocol (*BGP*) protocol section is not configured at the **[edit security ipsec]** hierarchy level.

We recommend that you configure no more than 512 dynamic security associations per ES Physical Interface Card (*PIC*).

To configure an SA for IPsec for an ES PIC, include the **security-association** statement at the **[edit security ipsec]** hierarchy level:

```
[edit security ipsec]
 security-associationsa-name;
```

NOTE: You configure a dynamic SA for the AS and MultiServices PICs at the **[edit services ipsec-vpn rule** ***rule-name*** **term** ***term-name*** **then dynamic]**, **[edit services ipsec-vpn ike]**, and **[edit services ipsec-vpn ipsec]** hierarchy levels.

For more information, see the “IPsec Services Configuration Guidelines” chapter of the Junos OS Services Interfaces Library for Routing Devices.

Tasks to configure SAs for IPsec for an ES PIC are:

Configuring the Description for an SA

To specify a description for an IPsec SA, include the **description** statement at the **edit security ipsec security-association** ***sa-name*****]** hierarchy level:

```
[edit security ipsec security-association sa-name]
 descriptiondescription;
```

Configuring IPsec Transport Mode

In *transport mode*, the data portion of the IP packet is encrypted, but the IP header is not. Transport mode can be used only when the communication endpoint and cryptographic endpoint are the same. Virtual private network (*VPN*) gateways that provide encryption and decryption services for protected hosts cannot use transport mode for protected VPN communications. You configure manual SAs, and you must configure static values on both ends of the SA.

NOTE: When you use transport mode, the Junos OS supports both BGP and OSPFv3 for manual SAs.

To configure IPsec security for transport mode, include the **mode** statement with the **transport** option at the **edit security ipsec security-association** ***sa-name***] hierarchy level:

```
[edit security ipsec security-association sa-name]
mode transport;
```

To apply *tunnel mode*, you configure manual SAs in transport mode and then reference the SA by name at the **[edit protocols bgp]** hierarchy level to protect a session with a given peer.

NOTE: You can configure BGP to establish a peer relationship over encrypted tunnels.

Configuring IPsec Tunnel Mode

You use tunnel mode when you use preshared keys with *IKE* to authenticate peers, or digital certificates with IKE to authenticate peers.

When you use preshared keys, you manually configure a preshared key, which must match that of its peer. With digital certificates, each router is dynamically or manually enrolled with a certificate authority (CA). When a tunnel is established, the public keys used for IPsec are dynamically obtained through IKE and validated against the CA certificate. This avoids the manual configuration of keys on routers within the topology. Adding a new router to the topology does not require any security configuration changes to existing routers.

To configure the IPsec in tunnel mode, include the **mode** statement with the **tunnel** option at the **edit security ipsec security-association** ***sa-name***] hierarchy level:

```
[edit security ipsec security-association sa-name]
mode tunnel;
```

NOTE: The Junos OS supports both both BGP and OSPFv3 in transport mode.

To enable tunnel mode, follow the steps in these sections:

- Configuring an IKE Proposal for Dynamic SAs
- "Associating the Configured Security Association with a Logical Interface" on page 218

- "IPsec Tunnel Traffic Configuration Overview" on page 219

Configuring IPsec Security Associations

IN THIS SECTION

Configuring Manual IPsec Security Associations for an ES PIC

IN THIS SECTION

To use IPsec security services, you create security associations (SAs) between hosts. An SA is a simplex connection that allows two hosts to communicate with each other securely by means of IPsec. There are two types of SAs: manual and dynamic.

Manual SAs require no negotiation; all values, including the keys, are static and specified in the configuration. As a result, peers can communicate only when they all share the same configured options.

To configure the manual IPsec SA for an ES PIC, include the **manual** statement at the **edit security ipsec security-association** ***sa-name***] hierarchy level:

```
[edit security ipsec security-association sa-name]
manual {
    direction (inbound | outbound | bi-directional) {
        authentication {
            algorithm (hmac-md5-96 | hmac-sha1-96);
            key (ascii-text key | hexadecimal key);
        }
         auxiliary-spiauxiliary-spi-value;
        encryption {
            algorithm (des-cbc | 3des-cbc);
            key (ascii-text key | hexadecimal key);
        }
        protocol (ah | esp | bundle);
        spi spi-value;
    }
}
```

Tasks to configure a manual SA are:

Configuring the Processing Direction

The **direction** statement sets inbound and outbound IPsec processing. If you want to define different algorithms, keys, or security parameter index (SPI) values for each direction, you configure the **inbound** and **outbound** options. If you want the same attributes in both directions, use the **bidirectional** option.

To configure the direction of IPsec processing, include the **direction** statement and specify the direction at the **[edit security ipsec security-association *sa-name* manual]** hierarchy level:

```
[edit security ipsec security-association sa-name manual]
direction (inbound | outbound | bidirectional);
```

The following example shows how to define different algorithms, keys, and security parameter index values for inbound and outbound processing directions:

```
[edit security ipsec security-association sa-name]
manual {
    direction inbound {
        encryption {
            algorithm 3des-cbc;
            key ascii-text 234567890123456789012345678901234;
        }
        protocol esp;
        spi 16384;
    }
    direction outbound {
        encryption {
            algorithm 3des-cbc;
            key ascii-text 123456789012345678901abcd;
        }
        protocol esp;
        spi 24576;
    }
}
```

The following example shows how to define the same algorithms, keys, and security parameter index values for bidirectional processing:

```
[edit security ipsec security-association sa-name manual]
direction bidirectional {
    authentication {
        algorithm hmac-md5-96;
        key ascii-text 123456789012abcd;
    }
    protocol ah;
    spi 20001;
}
```

Configuring the Protocol for a Manual SA

IPsec uses two protocols to protect IP traffic: Encapsulating Security Payload (*ESP*) and authentication header (*AH*). For transport mode SAs, both ESP and AH are supported. The AH protocol is used for strong authentication. The **bundle** option uses AH authentication and ESP encryption; it does not use ESP authentication because AH provides stronger authentication of IP packets.

NOTE: The AH protocol is supported only on M Series routers.

To configure the IPsec protocol on an ES PIC, include the **protocol** statement at the **edit security ipsec security-association *sa-name* manual direction (inbound | outbound | bidirectional)]** hierarchy level and specify the **ah**, **bundle**, or **esp** option:

```
[edit security ipsec security-association sa-name  manual direction (inbound |
outbound | bi-directional)]
protocol (ah | bundle | esp);
```

Configuring the Security Parameter Index

An *SPI* is an arbitrary value that uniquely identifies which SA to use at the receiving host. The sending host uses the SPI to identify and select which SA to use to secure every packet. The receiving host uses the SPI to identify and select the encryption algorithm and key used to decrypt packets.

NOTE: Each manual SA must have a unique SPI and protocol combination.

Use the auxiliary SPI when you configure the **protocol** statement to use the **bundle** option.

To configure the SPI on an ES PIC, include the **spi** statement and specify a value (256 through 16,639) at the **[edit security ipsec security-association *sa-name* manual direction (inbound | outbound | bi-directional]** hierarchy level:

```
[edit security ipsec security-association sa-name manual direction (inbound |
outbound | bidirectional)]
spi spi-value;
```

Configuring the Auxiliary Security Parameter Index

When you configure the **protocol statement to use the bundle** option, the Junos OS uses the auxiliary SPI for the ESP and the SPI for the AH.

NOTE: Each manual SA must have a unique SPI and protocol combination.

To configure the auxiliary SPI, include the **auxiliary-spi** statement at the **[edit security ipsec security-association *sa-name* manual direction (inbound | outbound | bi-directional)]** hierarchy level and set the value to an integer between 256 and 16,639:

```
[edit security ipsec security-association sa-name manual direction (inbound |
outbound | bidirectional)]
auxiliary-spi auxiliary-spi-value;
```

Configuring the Authentication Algorithm and Key

To configure an authentication algorithm and key, include the **authentication** statement at the **[edit security ipsec security-association *sa-name* manual direction (inbound | outbound | bi-directional)]** hierarchy level:

```
[edit security ipsec security-association sa-name manual direction (inbound | outbound |
bidirectional)]
authentication {
    algorithm (hmac-md5-96 | hmac-sha1-96);
    key (ascii-text key | hexadecimal key);
}
```

The algorithm can be one of the following:

- **hmac-md5-96**—Hash algorithm that authenticates packet data. It produces a 128-bit authenticator value and 96-bit digest.
- **hmac-sha1-96**—Hash algorithm that authenticates packet data. It produces a 160-bit authenticator value and a 96-bit digest.

The key can be one of the following:

- **ascii-text *key***—ASCII text key. With the **hmac-md5-96** option, the key contains
- 16 ASCII characters. With the **hmac-sha1-96** option, the key contains 20 ASCII characters.
- **hexadecimal *key***—Hexadecimal key. With the **hmac-md5-96** option, the key contains 32 hexadecimal characters. With the **hmac-sha1-96** option, the key contains 40 hexadecimal characters.

Configuring the Encryption Algorithm and Key

To configure IPsec encryption, include the **encryption** statement and specify an algorithm and key at the **[edit security ipsec security-association *sa-name* manual direction (inbound | outbound | bi-directional)]** hierarchy level:

```
[edit security ipsec security-association sa-name manual direction (inbound | outbound | bi-
directional)]
encryption {
    algorithm (des-cbc | 3des-cbc);
    key (ascii-text key | hexadecimal key);
}
```

The algorithm can be one of the following:

- **des-cbc**—Encryption algorithm that has a block size of 8 bytes; its key size is 64 bits long.
- **3des-cbc**—Encryption algorithm that has a block size of 24 bytes; its key size is 192 bits long.

> **NOTE**: For a list of Data Encryption Standard (*DES*) encryption algorithm weak and semiweak keys, see RFC 2409. For **3des-cbc**, we recommend that the first 8 bytes not be the same as the second 8 bytes, and that the second 8 bytes be the same as the third 8 bytes.

The key can be one of the following:

- **ascii-text**—ASCII text key. With the **des-cbc** option, the key contains 8 ASCII characters. With the **3des-cbc** option, the key contains 24 ASCII characters.
- **hexadecimal**—Hexadecimal key. With the **des-cbc** option, the key contains 16 hexadecimal characters. With the **3des-cbc** option, the key contains 48 hexadecimal characters.

> **NOTE**: You cannot configure encryption when you use the AH protocol.

SEE ALSO

Configuring Dynamic IPsec Security Associations | 52

Configuring Dynamic IPsec Security Associations

You configure dynamic *SA*s with a set of proposals that are negotiated by the security gateways. The keys are generated as part of the negotiation and do not need to be specified in the configuration. The dynamic SA includes one or more proposals, which allow you to prioritize a list of protocols and algorithms to be negotiated with the peer.

To configure a dynamic SA, include the **dynamic** statement at the **[edit security ipsec security-association *sa-name*]** hierarchy level. Specify an *IPsec* policy name, and optionally, a 32-packet or 64-packet replay window size.

```
[edit security ipsec security-association sa-name]
dynamic {
    ipsec-policy policy-name;
    replay-window-size (32 | 64);
}
```

SEE ALSO

Configuring Manual IPsec Security Associations for an ES PIC | 47

Configuring an IKE Policy

IN THIS SECTION

Configuring an IKE Policy for Preshared Keys

IN THIS SECTION

An *IKE* policy defines a combination of security parameters (IKE proposals) to be used during IKE negotiation. It defines a peer address, the preshared key for the given peer, and the proposals needed for that connection. During the IKE negotiation, IKE looks for an IKE policy that is the same on both peers. The peer that initiates the negotiation sends all its policies to the remote peer, and the remote peer tries to find a match.

A match is made when both policies from the two peers have a proposal that contains the same configured attributes. If the lifetimes are not identical, the shorter lifetime between the two policies (from the host and peer) is used. The configured preshared key must also match its peer.

To ensure that at least one proposal will match a remote peer's proposal, you can create multiple, prioritized proposals at each peer. Do this by configuring the proposal(s) and associating them with an IKE policy, and, optionally, prioritizing the list in the `policy` statement, where they are evaluated in list order.

Include the `policy` statement at the `[edit security ike]` hierarchy level and specify an *IPsec* tunnel destination as the peer address:

```
[edit security ike]
policy ike-peer-address;
```

Tasks for configuring an IKE policy are:

Configuring the Description for an IKE Policy

To specify a description for an IKE policy, include the `description` statement at the `[edit security ike policy ike-peer-address]` hierarchy level:

```
[edit security ike policy ike-peer-address]
description description;
```

Configuring the Mode for an IKE Policy

IKE policy has two modes: aggressive and main. By default, *main mode* is enabled. Main mode uses six messages, in three exchanges, to establish the IKE SA. (These three steps are IKE SA negotiation, a *Diffie-Hellman key exchange*, and authentication of the peer.) Main mode also allows a peer to hide its identity.

Aggressive mode also establishes an authenticated IKE SA and keys. However, aggressive mode uses half the number of messages, has less negotiation power, and does not provide identity protection. The peer can use the aggressive or main mode to start IKE negotiation; the remote peer accepts the mode sent by the peer.

To configure IKE policy mode, include the `mode` statement and specify `aggressive` or `main` at the `[edit security ike policy ike-peer-address]` hierarchy level:

```
[edit security ike policy ike-peer-address ]
mode (aggressive | main);
```

For Junos OS in FIPS mode, the aggressive option for IKEv1 is not supported with the mode statement at the `[edit services ipsec-vpn ike policy policy-name]` hierarchy level.

Configuring the Preshared Key for an IKE Policy

IKE policy preshared keys authenticate peers. You must manually configure a preshared key, which must match that of its peer. The preshared key can be an ASCII text (alphanumeric) key or a hexadecimal key.

A local certificate is an alternative to the preshared key. A commit operation fails if either a preshared key or a local certificate is not configured.

To configure an IKE policy preshared key, include the `pre-shared-key` statement at the `[edit security ike policy *ike-peer-address*]` hierarchy level:

```
[edit security ike policy ike-peer-address]
pre-shared-key (ascii-text key | hexadecimal key);
```

Associating Proposals with an IKE Policy

The IKE policy proposal is a list of one or more proposals associated with an IKE policy.

To configure an IKE policy proposal, include the `proposals` statement at the `[edit security ike policy *ike-peer-address*]` hierarchy level and specify one or more proposal names:

```
[edit security ike policy ike-peer-address]
proposals [ proposal-names ];
```

RELATED DOCUMENTATION

Example: Configuring an IKE Policy | 55

Example: Configuring an IKE Policy

Define two *IKE* policies: policy 10.1.1.2 and policy 10.1.1.1. Each policy is associated with `proposal-1` and `proposal-2`.

```
[edit security]
ike {
    proposal proposal-1 {
        authentication-method pre-shared-keys;
        dh-group group1;
        authentication-algorithm sha1;
        encryption-algorithm 3des-cbc;
        lifetime-seconds 1000;
    }
    proposal proposal-2 {
        authentication-method pre-shared-keys;
        dh-group group2;
        authentication-algorithm md5;
        encryption-algorithm des-cbc;
        lifetime-seconds 10000;
    }
    proposal proposal-3 {
        authentication-method rsa-signatures;
        dh-group group2;
        authentication-algorithm md5;
        encryption-algorithm des-cbc;
        lifetime-seconds 10000;
    }
    policy 10.1.1.2 {
        mode main;
        proposals [ proposal-1 proposal-2 ];
        pre-shared-key ascii-text example-pre-shared-key;
    }
    policy 10.1.1.1 {
        local-certificate certificate-filename;
        local-key-pair private-public-key-file;
        mode aggressive;
        proposals [ proposal-2 proposal-3 ]
        pre-shared-key hexadecimal 0102030abbcd;
    }
}
```

NOTE: Updates to the current IKE proposal and policy configuration are not applied to the current IKE *SA*; updates are applied to new IKE SAs.

If you want the new updates to take immediate effect, you must clear the existing IKE security associations so that they will be reestablished with the changed configuration. For information about how to clear the current IKE security association, see the CLI Explorer.

NOTE: When configuring multiple IPSec tunnels between IPSec peers, the IPSec tunnels can terminate on multiple local addresses on a physical interface of an IPSec peer and vice-versa.

SEE ALSO

Configuring an IKE Policy for Preshared Keys | 53

Configuring an IPsec Proposal for an ES PIC

IN THIS SECTION

An *IPsec* proposal lists protocols and algorithms (security services) to be negotiated with the remote IPsec peer.

To configure an IPsec proposal and define its properties, include the following statements at the `[edit security ipsec]` hierarchy level:

```
[edit security ipsec]
proposal ipsec-proposal-name {
    authentication-algorithm (hmac-md5-96 | hmac-sha1-96);
    description description ;
    encryption-algorithm (3des-cbc | des-cbc);
    lifetime-seconds seconds;
    protocol (ah | esp | bundle);
}
```

Tasks to configure an IPsec proposal for an ES PIC include:

Configuring the Authentication Algorithm for an IPsec Proposal

To configure an IPsec authentication algorithm, include the `authentication-algorithm` statement at the `[edit security ipsec proposal` *`ipsec-proposal-name`*`]` hierarchy level:

```
[edit security ipsec proposal ipsec-proposal-name]
authentication-algorithm (hmac-md5-96 | hmac-sha1-96);
```

The authentication algorithm can be one of the following:

- `hmac-md5-96`—Hash algorithm that authenticates packet data. It produces a 128-bit digest. Only 96 bits are used for authentication.
- `hmac-sha1-96`—Hash algorithm that authenticates packet data. It produces a 160-bit digest. Only 96 bits are used for authentication.

Configuring the Description for an IPsec Proposal

To specify a description for an IPsec proposal, include the `description` statement at the `[edit security ipsec proposal` *`ipsec-proposal-name`*`]` hierarchy level:

```
[edit security ike policy ipsec-proposal-name]
description description;
```

Configuring the Encryption Algorithm for an IPsec Proposal

To configure the IPsec encryption algorithm, include the `encryption-algorithm` statement at the `[edit security ipsec proposal` *`ipsec-proposal-name`*`]` hierarchy level:

```
[edit security ipsec proposal ipsec-proposal-name ]
encryption-algorithm (3des-cbc | des-cbc);
```

The encryption algorithm can be one of the following:

- `3des-cbc`—Encryption algorithm that has a block size of 24 bytes; its key size is 192 bits long.
- `des-cbc`—Encryption algorithm that has a block size of 8 bytes; its key size is
- 48 bits long.

NOTE: We recommend that you use the triple DES cipher block chaining (*3DES-CBC*) encryption algorithm.

Configuring the Lifetime for an IPsec SA

The IPsec lifetime option sets the lifetime of an IPsec SA. When the IPsec SA expires, it is replaced by a new SA (and SPI) or is terminated. A new SA has new authentication and encryption keys, and SPI; however, the algorithms may remain the same if the proposal is not changed. If you do not configure a lifetime and a lifetime is not sent by a responder, the lifetime is 28,800 seconds.

To configure the IPsec lifetime, include the `lifetime-seconds` statement and specify the number of seconds (180 through 86,400) at the `[edit security ipsec proposal` *`ipsec-proposal-name`*`]` hierarchy level:

```
[edit security ipsec proposal ipsec-proposal-name]
lifetime-seconds seconds;
```

NOTE: When a dynamic SA is created, two types of lifetimes are used: hard and soft. The hard lifetime specifies the lifetime of the SA. The soft lifetime, which is derived from the hard lifetime, informs the IPsec key management system that that the SA is about to expire. This allows the key management system to negotiate a new SA before the hard lifetime expires. When you specify the lifetime, you specify a hard lifetime.

Configuring the Protocol for a Dynamic IPsec SA

The `protocol` statement sets the protocol for a dynamic SA. The ESP protocol can support authentication, encryption, or both. The AH protocol is used for strong authentication. AH also authenticates the IP packet. The `bundle` option uses AH authentication and ESP encryption; it does not use ESP authentication because AH provides stronger authentication of IP packets.

To configure the protocol for a dynamic SA, include the `protocol` statement at the `[edit security ipsec proposal` *`ipsec-proposal-name`*`]` hierarchy level:

```
[edit security ipsec proposal ipsec-proposal-name ] protocol (ah | esp | bundle);
```

SEE ALSO

IPsec Configuration for an ES PIC Overview | 41

Configuring an IPsec Policy

IN THIS SECTION

Configuring the IPsec Policy for an ES PIC

IN THIS SECTION

An *IPsec* policy defines a combination of security parameters (IPsec proposals) used during IPsec negotiation. It defines Perfect Forward Secrecy (*PFS*) and the proposals needed for the connection. During the IPsec negotiation, IPsec looks for an IPsec proposal that is the same on both peers. The peer that initiates the negotiation sends all its policies to the remote peer, and the remote peer tries to find a match.

A match is made when both policies from the two peers have a proposal that contains the same configured attributes. If the lifetimes are not identical, the shorter lifetime between the two policies (from the host and peer) is used.

You can create multiple, prioritized IPsec proposals at each peer to ensure that at least one proposal will match a remote peer's proposal.

First, you configure one or more IPsec proposals; then you associate these proposals with an IPsec policy. You can prioritize the proposals in the list by listing them in the order in which the IPsec policy uses them (first to last).

To configure an IPsec policy, include the `policy` statement at the `[edit security ipsec]` hierarchy level, specifying the policy name and one or more proposals you want to associate with this policy:

```
[edit security ipsec]
policy ipsec-policy-name {
    proposals [ proposal-names ];
}
```

Configuring Perfect Forward Secrecy

PFS provides additional security by means of a *Diffie-Hellman key exchange* shared secret value. With PFS, if one key is compromised, previous and subsequent keys are secure because they are not derived from previous keys. This statement is optional.

To configure PFS, include the `perfect-forward-secrecy` statement and specify a Diffie-Hellman group at the `[edit security ipsec policy` *`ipsec-policy-name`*`]` hierarchy level:

```
[edit security ipsec policy ipsec-policy-name]
perfect-forward-secrecy {
    keys (group1 | group2);
}
```

The key can be one of the following:

- `group1`—Specify that IKE use the 768-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.
- `group2`—Specify that IKE use the 1024-bit Diffie-Hellman prime modulus group when performing the new Diffie-Hellman exchange.

`group2` provides more security than `group1`, but requires more processing time.

RELATED DOCUMENTATION

Example: Configuring an IPsec Policy | **61**

IPsec Configuration for an ES PIC Overview | **41**

Example: Configuring an IPsec Policy

The following example shows how to configure an IPsec policy:

```
[edit security ipsec]
proposal dynamic-1 {
    protocol esp;
    authentication-algorithm hmac-md5-96;
    encryption-algorithm 3des-cbc;
    lifetime-seconds 6000;
}
proposal dynamic-2 {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
    lifetime-seconds 6000;
}
policy dynamic-policy-1 {
    perfect-forward-secrecy {
        keys group1;
    }
    proposals [ dynamic-1 dynamic-2 ];
}
security-association dynamic-sa1 {
    dynamic {
        replay-window-size 64;
        ipsec-policy dynamic-policy-1;
    }
}
```

NOTE: Updates to the current IPsec proposal and policy configuration are not applied to the current IPsec SA; updates are applied to new IPsec SAs.

If you want the new updates to take immediate effect, you must clear the existing IPsec security associations so that they will be reestablished with the changed configuration. For information about how to clear the current IPsec security association, see the CLI Explorer.

SEE ALSO

Configuring the IPsec Policy for an ES PIC | **60**

IPsec Configuration for an ES PIC Overview | **41**

CHAPTER 4

Configuring IPsec Security Associations

IN THIS CHAPTER

Overview of IPsec

IN THIS SECTION

Security Associations Overview

To use *IPsec* security services, you create *SA*s between hosts. An SA is a simplex connection that allows two hosts to communicate with each other securely by means of IPsec. There are two types of SAs: manual and dynamic.

- Manual SAs require no negotiation; all values, including the keys, are static and specified in the configuration. Manual SAs statically define the Security Parameter Index (*SPI*) values, algorithms, and keys to be used, and require matching configurations on both ends of the tunnel. Each peer must have the same configured options for communication to take place.

- Dynamic SAs require additional configuration. With dynamic SAs, you configure *IKE* first and then the SA. IKE creates dynamic security associations; it negotiates SAs for IPsec. The IKE configuration defines the algorithms and keys used to establish the secure IKE connection with the peer security gateway. This connection is then used to dynamically agree upon keys and other data used by the dynamic IPsec SA. The IKE SA is negotiated first and then used to protect the negotiations that determine the dynamic IPsec SAs.

- Set up user-level tunnels or SAs, including tunnel attribute negotiations and key management. These tunnels can also be refreshed and terminated on top of the same secure channel.

The Junos OS implementation of IPsec supports two modes of security (*transport mode* and *tunnel mode*).

SEE ALSO

IKE Key Management Protocol Overview | 64

IPsec Requirements for Junos-FIPS | 66

[edit security] Hierarchy Level

IKE Key Management Protocol Overview

IKE is a key management protocol that creates dynamic *SA*s; it negotiates SAs for *IPsec*. An IKE configuration defines the algorithms and keys used to establish a secure connection with a peer security gateway.

IKE does the following:

- Negotiates and manages IKE and IPsec parameters

- Authenticates secure key exchange

- Provides mutual peer authentication by means of shared secrets (not passwords) and public keys

- Provides identity protection (in main mode)

IKE occurs over two phases. In the first phase, it negotiates security attributes and establishes shared secrets to form the bidirectional IKE SA. In the second phase, inbound and outbound IPsec SAs are established. The IKE SA secures the exchanges in the second phase. IKE also generates keying material, provides Perfect Forward Secrecy, and exchanges identities.

NOTE: Starting in Junos OS Release 14.2, when you perform an SNMP walk of the jnxIkeTunnelEntry object in the jnxIkeTunnelTable table, the `Request failed: OID not increasing` error message might be generated. This problem occurs only when simultaneous Internet Key Exchange security associations (IKE SAs) are created, which occurs when both ends of the SA initiate IKE SA negotiations at the same time. When an SNMP MIB walk is performed to display IKE SAs, the snmpwalk tool expects the object identifiers (OIDs) to be in increasing order. However, in the case of simultaneous IKE SAs, the OIDs in the SNMP table might not be in increasing order. This behavior occurs because the tunnel IDs, which are part of the OIDs, are allocated based on the initiator of the IKE SA, which can be on either side of the IKE tunnel.

The following is an example of an SNMP MIB walk that is performed on IKE simultaneous SAs:

```
jnxIkeTunLocalRole."ipsec_ss_cust554".ipv4."192.0.2.41".47885 = INTEGER:
responder(2)   >>> This is Initiator SA
jnxIkeTunLocalRole."ipsec_ss_cust554".ipv4."192.0.2.41".47392 = INTEGER:
initiator(1)   >>> This is Responder's SA
```

The OID comparison fails when the SNMP walk is tunnel ID (47885 and 47392). It cannot be ensured when an SNMP walk is performed that the tunnel IDs are in increasing order because tunnels might be initiated from either side.

To work around this problem, the SNMP MIB walk contains an option, -Cc, to disable check for increasing OIDs. The following is an example of the MIB walk performed on the jnxIkeTunnelEntry table with the -Cc option:

```
snmpwalk -Os -Cc -c public -v 1 vira jnxIkeTunnelEntry
```

SEE ALSO

Security Associations Overview | 63

IPsec Requirements for Junos-FIPS | 66

[edit security] Hierarchy Level

IPsec Requirements for Junos-FIPS

In a *Junos-FIPS* environment, hardware configurations with two *Routing Engine*s must be configured to use *IPsec* and a private routing instance for all communications between the Routing Engines. IPsec communication between the Routing Engines and AS II *FIPS PIC*s is also required.

SEE ALSO

Security Associations Overview | **63**

IKE Key Management Protocol Overview | **64**

[edit security] Hierarchy Level

Overview of IPsec

IP Security (*IPsec*) is a standards based framework for ensuring secure private communication over IP networks. IPsec provides a secure way to authenticate senders and encrypt IP version 4 (IPv4) and version 6 (IPv6) traffic between network devices, such as routers and hosts. IPsec includes data integrity, sender authentication, source data confidentiality, and protection against data replay.

The main concepts you need to understand are as follows:

- "IPsec-Enabled Line Cards" on page 66
- "Authentication Algorithms" on page 68
- Encryption Algorithms
- "IPsec Protocols" on page 69
- "IPsec Security Associations" on page 72
- "IPSec Modes" on page 72
- *Digital Certificates*
- "Service Sets" on page 74

IPsec-Enabled Line Cards

The first choice you need to make when implementing IPsec on a Junos OS-based router is the type of line card you wish to use. The term line card includes Physical Interface Cards (PICs), Modular Interface Cards (MICs), Dense Port Concentrators (DPCs), and Modular Port Concentrators (MPCs). The following line cards support IPsec implementation.

NOTE: See the specific hardware documentation for your router to determine if the line cards on that router support IPsec.

The following line cards support IPsec:

- The Encryption Services (ES) PIC provides encryption services and software support for IPsec.
- The Adaptive Services (AS) PIC and the Adaptive Services (AS) II PIC provide IPsec services and other services, such as Network Address Translation (NAT) and stateful firewall.
- The AS II Federal Information Processing Standards (FIPS) PIC is a special version of the AS PIC that communicates securely with the Routing Engine by using internal IPsec. You must configure IPsec on the AS II FIPS PIC when you enable FIPS mode on the router. For more information about implementing IPsec on an AS II FIPS PIC installed in a router configured in FIPS mode, see the *Secure Configuration Guide for Common Criteria and Junos-FIPS*.
- The Multiservices PICs supply hardware acceleration for an array of packet processing-intensive services. These services include IPsec services and other services, such as stateful firewall, NAT, IPsec, anomaly detection, and tunnel services.
- The Multiservices Dense Port Concentrators (DPCs) provide IPsec services.
- The Multiservices Modular Port Concentrators (MS-MPCs) support IPsec services.
- The Multiservices Modular Interface Cards (MS-MICs) support IPsec services.

NOTE: Junos OS extension-provider packages, including the IPsec service package, come preinstalled and preconfigured on MS-MPCs and MS-MICs.

SEE ALSO

Overview of IPsec | **66**

Considering General IPsec Issues | **36**

Services PICs-Overview

Enabling Service Packages

Multiservices MIC and Multiservices MPC (MS-MIC and MS-MPC) Overview

Authentication Algorithms

Authentication is the process of verifying the identity of the sender. Authentication algorithms use a shared key to verify the authenticity of the IPsec devices. The Junos OS uses the following authentication algorithms:

- Message Digest 5 (MD5) uses a one-way hash function to convert a message of arbitrary length to a fixed-length message digest of 128 bits. Because of the conversion process, it is mathematically infeasible to calculate the original message by computing it backwards from the resulting message digest. Likewise, a change to a single character in the message will cause it to generate a very different message digest number.

  To verify that the message has not been tampered with, the Junos OS compares the calculated message digest against a message digest that is decrypted with a shared key. The Junos OS uses the MD5 hashed message authentication code (HMAC) variant that provides an additional level of hashing. MD5 can be used with authentication header (AH), Encapsulating Security Payload (ESP), and Internet Key Exchange (IKE).

- Secure Hash Algorithm 1 (SHA-1) uses a stronger algorithm than MD5. SHA-1 takes a message of less than 264 bits in length and produces a 160-bit message digest. The large message digest ensures that the data has not been changed and that it originates from the correct source. The Junos OS uses the SHA-1 HMAC variant that provides an additional level of hashing. SHA-1 can be used with AH, ESP, and IKE.

- SHA-256, SHA-384, and SHA-512 (sometimes grouped under the name SHA-2) are variants of SHA-1 and use longer message digests. The Junos OS supports the SHA-256 version of SHA-2, which can process all versions of Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Triple DES (3DES) encryption.

Encryption Algorithms

Encryption encodes data into a secure format so that it cannot be deciphered by unauthorized users. Like authentication algorithms, a shared key is used with encryption algorithms to verify the authenticity of the IPsec devices. The Junos OS uses the following encryption algorithms:

- Data Encryption Standard cipher-block chaining (DES-CBC) is a symmetric secret-key block algorithm. DES uses a key size of 64 bits, where 8 bits are used for error detection and the remaining 56 bits provide encryption. DES performs a series of simple logical operations on the shared key, including permutations and substitutions. CBC takes the first block of 64 bits of output from DES, combines this block with the second block, feeds this back into the DES algorithm, and repeats this process for all subsequent blocks.

- Triple DES-CBC (3DES-CBC) is an encryption algorithm that is similar to DES-CBC, but provides a much stronger encryption result because it uses three keys for 168-bit (3 x 56-bit) encryption. 3DES works by using the first key to encrypt the blocks, the second key to decrypt the blocks, and the third key to re-encrypt the blocks.

- Advanced Encryption Standard (AES) is a next-generation encryption method based on the Rijndael algorithm developed by Belgian cryptographers Dr. Joan Daemen and Dr. Vincent Rijmen. It uses a 128-bit block and three different key sizes (128, 192, and 256 bits). Depending on the key size, the algorithm performs a series of computations (10, 12, or 14 rounds) that include byte substitution, column mixing, row shifting, and key addition. The use of AES in conjunction with IPsec is defined in RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*.

- Starting In Junos OS Release 17.3R1, Advanced Encryption Standard in Galois/Counter Mode (AES-GCM) is supported for MS-MPCs and MS-MICs. However, in Junos FIPS mode, AES-GCM is not supported in Junos OS Release 17.3R1. Starting in Junos OS Release 17.4R1, AES-GCM is supported in Junos FIPS mode. AES-GCM is an authenticated encryption algorithm designed to provide both authentication and privacy. AES-GCM uses universal hashing over a binary Galois field to provide authenticated encryption and allows authenticated encryption at data rates of tens of Gbps.

SEE ALSO

Configuring IKE Proposals

Configuring IPsec Proposals

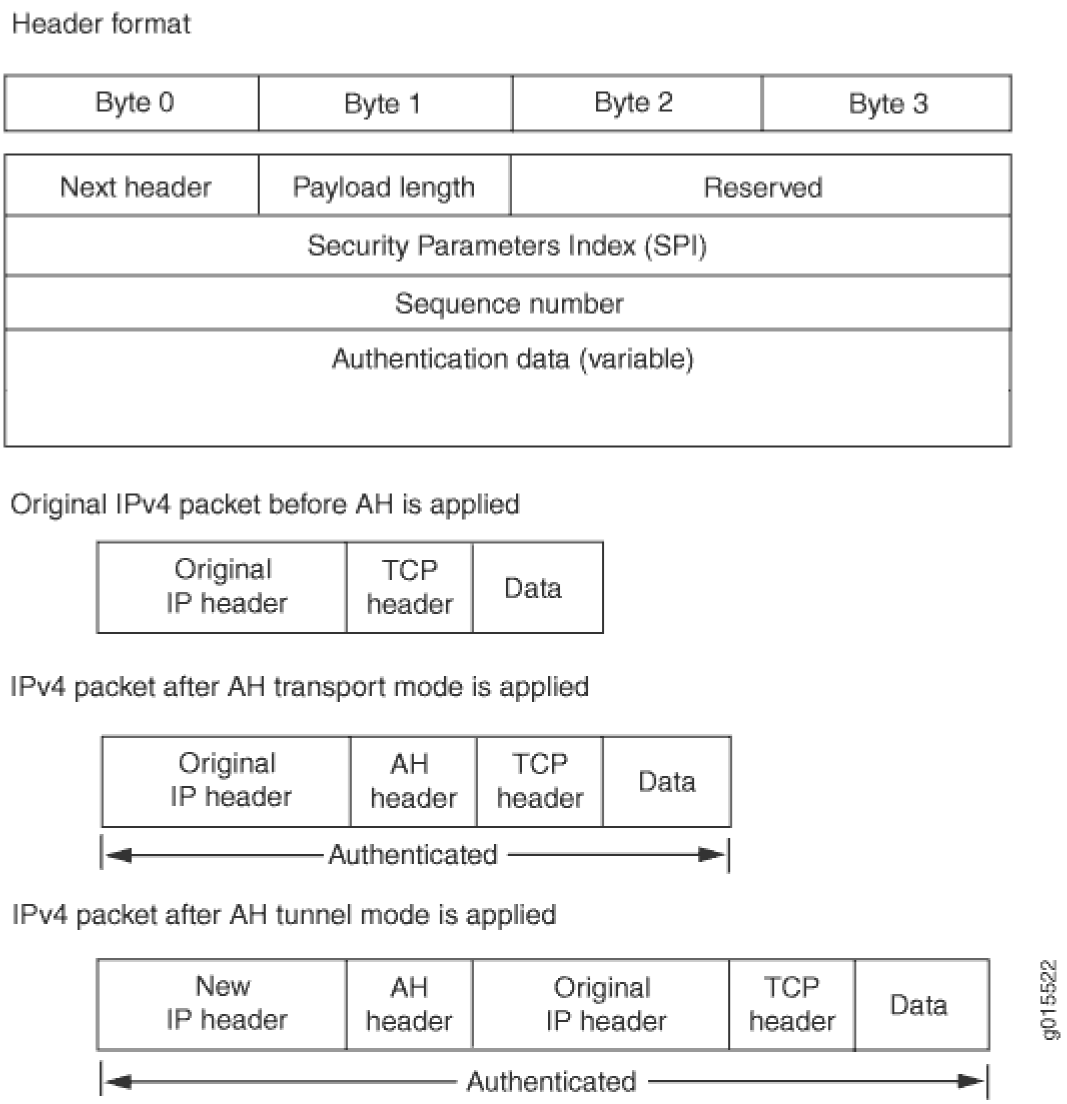
IPsec Protocols

IPsec protocols determine the type of authentication and encryption applied to packets that are secured by the router. The Junos OS supports the following IPsec protocols:

- AH—Defined in RFC 2402, AH provides connectionless integrity and data origin authentication for IPv4 and IPv6 packets. It also provides protection against replays. AH authenticates as much of the IP header as possible, as well as the upper-level protocol data. However, some IP header fields might change in transit. Because the value of these fields might not be predictable by the sender, they cannot be protected by AH. In an IP header, AH can be identified with a value of `51` in the `Protocol` field of an IPv4 packet and the `Next Header` field of an IPv6 packet. An example of the IPsec protection offered by AH is shown in Figure 2 on page 70.

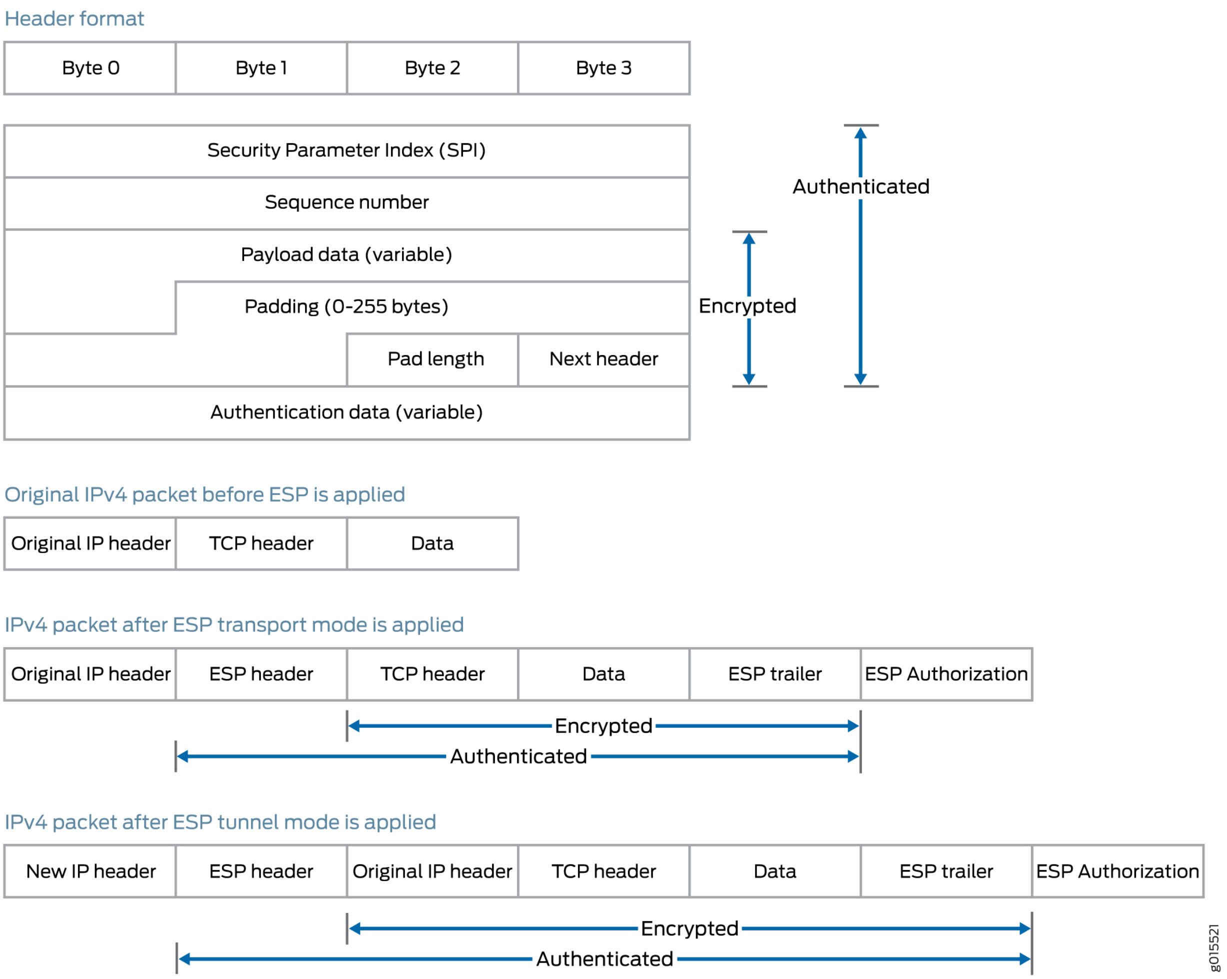
  NOTE: AH is not supported on the T Series, M120, and M320 routers.

Figure 2: AH Protocol



- ESP—Defined in RFC 2406, ESP can provide encryption and limited traffic flow confidentiality, or connectionless integrity, data origin authentication, and an anti-replay service. In an IP header, ESP can be identified a value of `50` in the `Protocol` field of an IPv4 packet and the `Next Header` field of an IPv6 packet. An example of the IPsec protection offered by ESP is shown in Figure 3 on page 71.

Figure 3: ESP Protocol



- Bundle—When you compare AH with ESP, there are some benefits and shortcomings in both protocols. ESP provides a decent level of authentication and encryption, but does so only for part of the IP packet. Conversely, although AH does not provide encryption, it does provide authentication for the entire IP packet. Because of this, the Junos OS offers a third form of IPsec protocol called a protocol bundle. The bundle option offers a hybrid combination of AH authentication with ESP encryption.

SEE ALSO

Configuring IPsec Proposals

Configuring Security Associations

protocol (IPsec)

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.4R1	Starting in Junos OS Release 17.4R1, AES-GCM is supported in Junos FIPS mode.
17.3R1	Starting In Junos OS Release 17.3R1, Advanced Encryption Standard in Galois/Counter Mode (AES-GCM) is supported for MS-MPCs and MS-MICs.
14.2	Starting in Junos OS Release 14.2, when you perform an SNMP walk of the jnxIkeTunnelEntry object in the jnxIkeTunnelTable table, the `Request failed: OID not increasing` error message might be generated.

IPsec Security Associations Overview

IN THIS SECTION

IPsec Security Associations

Another IPSec consideration is the type of security association (SA) that you wish to implement. An SA is a set of IPSec specifications that are negotiated between devices that are establishing an IPSec relationship. These specifications include preferences for the type of authentication, encryption, and IPSec protocol that should be used when establishing the IPSec connection. An SA can be either unidirectional or bidirectional, depending on the choices made by the network administrator. An SA is uniquely identified by a Security Parameter Index (SPI), an IPv4 or IPv6 destination address, and a security protocol (AH or ESP) identifier.

You can configure IPSec with a preset, preshared manual SA or use IKE to establish a dynamic SA. Manual SAs require you to specify all the IPSec requirements up front. Conversely, IKE dynamic SAs typically contain configuration defaults for the highest levels of authentication and encryption.

IPSec Modes

When configuring IPSec, the last major consideration is the type of IPSec mode you wish to implement in your network. The Junos OS supports the following IPSec modes:

- Tunnel mode is supported for both AH and ESP in the Junos OS and is the usual choice for a router. In tunnel mode, the SA and associated protocols are applied to tunneled IPv4 or IPv6 packets. For a tunnel mode SA, an outer IP header specifies the IPSec processing destination, and an inner IP header specifies the ultimate destination for the packet. The security protocol header appears after the outer IP header, and before the inner IP header. In addition, there are slight differences for tunnel mode when you implement it with AH and ESP:

  - For AH, portions of the outer IP header are protected, as well as the entire tunneled IP packet.

  - For ESP, only the tunneled packet is protected, not the outer header.

  When one side of a security association is a security gateway (such as a router), the SA must use tunnel mode. However, when traffic (for example, SNMP commands or BGP sessions) is destined for a router, the system acts as a host. Transport mode is allowed in this case because the system does not act as a security gateway and does not send or receive transit traffic.

- Transport mode provides a security association between two hosts. In transport mode, the protocols provide protection primarily for upper layer protocols. For IPv4 and IPv6 packets, a transport mode security protocol header appears immediately after the IP header and any options, and before any higher layer protocols (for example, TCP or UDP). There are slight differences for transport mode when you implement it with AH and ESP:

  - For AH, selected portions of the IP header are protected, as well as selected portions of the extension headers and selected options within the IPv4 header.

  - For ESP, only the higher layer protocols are protected, not the IP header or any extension headers preceding the ESP header.

NOTE: Support for IPSec transport mode is primarily limited to routing authentication and to certain configurations only application when Junos FIPs code is used.

SEE ALSO

Overview of IPsec | 66

Configuring Security Associations | 75

Understanding OSPFv3 Authentication

Example: Configuring IPsec Authentication for an OSPF Interface

Service Sets

IN THIS SECTION

Service Sets

The Adaptive Services PIC supports two types of service sets when you configure IPSec tunnels. Because they are used for different purposes, it is important to know the differences between these service set types.

- Next-hop service set—Supports multicast and multicast-style dynamic routing protocols (such as OSPF) over IPSec. Next-hop service sets allow you to use *inside* and *outside* logical interfaces on the Adaptive Services PIC to connect with multiple routing instances. They also allow the use of Network Address Translation (NAT) and stateful firewall capabilities. However, next-hop service sets do not monitor Routing Engine traffic by default and require configuration of multiple service sets to support traffic from multiple interfaces.

- Interface service set—Applied to a physical interface and similar to a stateless *firewall filter*. They are easy to configure, can support traffic from multiple interfaces, and can monitor Routing Engine traffic by default. However, they cannot support dynamic routing protocols or multicast traffic over the IPSec tunnel.

In general, we recommend that you use next-hop service sets because they support routing protocols and multicast over the IPSec tunnel, they are easier to understand, and the routing table makes forwarding decisions without administrative intervention.

SEE ALSO

Understanding Junos VPN Site Secure

Configuring Junos VPN Site Secure or IPSec VPN

Public Key Infrastructure User Guide

Configuring Security Associations

IN THIS SECTION

Configuring Security Associations

The first IPsec configuration step is to select a type of security association (SA) for your IPsec connection. You must statically configure all specifications for manual SAs, but you can rely on some defaults when you configure an IKE dynamic SA. To configure a security association, see the following sections.

Configuring Manual SAs

On the ES PIC, you configure a manual security association at the `[edit security ipsec security-association` *`name`*`]` hierarchy level. Include your choices for authentication, encryption, direction, mode, protocol, and SPI. Be sure that these choices are configured exactly the same way on the remote IPsec gateway.

```
[edit security]
ipsec {
    security-association                    sa-name                  {
        description                    description;
        manual {
            direction (inbound | outbound | bidirectional) {
                authentication {
                    algorithm (hmac-md5-96 | hmac-sha1-96);
                    key (ascii-text key | hexadecimal key);
                }
                auxiliary-spi auxiliary-spi;
                encryption {
                    algorithm (des-cbc | 3des-cbc);
                    key (ascii-text key  | hexadecimal key);
                }
                protocol (ah | esp | bundle);
                spi                        spi-value;
            }
        }
        mode (tunnel | transport);
    }
}
```

On the AS and MultiServices PICs, you configure a manual security association at the `[edit services ipsec-vpn rule rule-name]` hierarchy level. Include your choices for authentication, encryption, direction, protocol, and SPI. Be sure that these choices are configured exactly the same way on the remote IPsec gateway.

```
[edit services ipsec-vpn]
rule rule-name {
    match-direction (input | output);
    term                    term-name                  {
        from {
            destination-address                      address;
            source-address                      address;
        }
        then {
            backup-remote-gateway address;
            clear-dont-fragment-bit;
            manual {
                direction (inbound | outbound | bidirectional) {
                    authentication {
                        algorithm (hmac-md5-96 | hmac-sha1-96);
                        key (ascii-text key | hexadecimal key);
                    }
                    auxiliary-spi spi-value;
                    encryption {
                        algorithm algorithm; # This can be aes-128-cbc, aes-192-cbc,
                        # aes-256-cbc, des-cbc, or 3des-cbc.
                        key (ascii-text key | hexadecimal key);
                    }
                    protocol (ah | bundle | esp);
                    spi spi-value;
                }
            }
            no-anti-replay;
            remote-gateway                      address;
            syslog;
        }
    }
}
rule-set                  rule-set-name                {
    [ rule                  rule-names                ];
}
```

Configuring IKE Dynamic SAs

On the ES PIC, you configure an IKE dynamic SA at the `[edit security ike]` and `[edit security ipsec]` hierarchy levels. Include your choices for IKE policies and proposals, which include options for authentication algorithms, authentication methods, Diffie-Hellman groups, encryption, IKE modes, and preshared keys. The IKE policy must use the IP address of the remote end of the IPsec tunnel as the policy name. Also, include your choices for IPsec policies and proposals, which include options for authentication, encryption, protocols, Perfect Forward Secrecy (PFS), and IPsec modes. Be sure that these choices are configured exactly the same way on the remote IPsec gateway.

```
[edit security]
ike {
    proposal ike-proposal-name {
        authentication-algorithm (md5 | sha1 |sha-256 |sha-384);
        authentication-method (dsa-signatures | pre-shared-keys | rsa-signatures);
        description description;
        dh-group (group1 | group2);
        encryption-algorithm (3des-cbc | aes-128-cbc | aes-192-cbc | aes-256-cbc | des-cbc);
        lifetime-seconds seconds;
    }
    policy ike-peer-address {
        description description;
        encoding (binary | pem);
        identity identity-name;
        local-certificate certificate-filename;
        local-key-pair private-public-key-file;
        mode (aggressive | main);
        pre-shared-key (ascii-text key | hexadecimal key);
        proposals [ proposal-names ];
    }
}
ipsec {
    proposal ipsec-proposal-name  {
        authentication-algorithm (hmac-md5-96 | hmac-sha1-96 | hmac-sha-256-128);
        description description;
        encryption-algorithm (3des-cbc | aes-128-cbc | aes-192-cbc | aes-256-cbc | des-cbc);
```

```
            lifetime-seconds seconds;
            protocol (ah | esp | bundle);
        }
        policy ipsec-policy-name {
            description description;
            perfect-forward-secrecy {
                keys (group1 | group2);
            }
            proposals [ proposal-names ];
        }
        security-association sa-name {
            description description;
            dynamic {
                ipsec-policy policy-name;
                replay-window-size (32 | 64);
            }
            mode (tunnel | transport);
        }
    }
```

On the AS and MultiServices PICs, you configure an IKE dynamic security association at the `[edit services ipsec-vpn ike]`, `[edit services ipsec-vpn ipsec]`, and `[edit services ipsec-vpn rule` *`rule-name`*`]` hierarchy levels. Include your choices for IKE policies and proposals, which include options for authentication algorithms, authentication methods, Diffie-Hellman groups, encryption, IKE modes, and preshared keys. Also, include your choices for IPsec policies and proposals, which include options for authentication, encryption, protocols, PFS, and IPsec modes. Be sure that these choices are configured exactly the same way on the remote IPsec gateway.

If you choose not to explicitly configure IKE and IPsec policies and proposals on the AS and MultiServices PICs, your configuration can default to some preset values. These default values are shown in Table 5 on page 78.

Table 5: IKE and IPsec Proposal and Policy Default Values for the AS and MultiServices PICs

IKE Policy Statement	Default Value
mode	main
proposals	default
IKE Proposal Statement	**Default Value**

Table 5: IKE and IPsec Proposal and Policy Default Values for the AS and MultiServices PICs *(Continued)*

IKE Policy Statement	Default Value
authentication-algorithm	sha1
authentication-method	pre-shared-keys
dh-group	group2
encryption-algorithm	3des-cbc
lifetime-seconds	3600 (seconds)
IPsec Policy Statement	Default Value
perfect-forward-secrecy keys	group2
proposals	default
IPsec Proposal Statement	Default Value
authentication-algorithm	hmac-sha1-96
encryption-algorithm	3des-cbc
lifetime-seconds	28800 (seconds)
protocol	esp

NOTE: If you use the default IKE and IPsec policy and proposal values preset within the AS and MultiServices PICs, you must explicitly configure an IKE policy and include a preshared key. This is because the **pre-shared-keys** authentication method is one of the preset values in the default IKE proposal.

NOTE: Starting in Junos OS release 14.2, in an environment in which Juniper Networks MX Series routers interoperate with Cisco ASA devices, IKE security associations (SAs) and IPsec SAs are deleted immediately on the Cisco ASA devices, but they are retained on the MX Series routers.As a result, 100 percent traffic loss occurs on the MX routers when traffic is initiated from either the MX Series routers or Cisco ASA devices. This problem of excessive traffic loss occurs when a service PIC is restarted on MX Series routers, a line card is restarted on MX series routers, or when a shutdown/no shutdown command sequence or a change in speed setting is performed on the Cisco ASA devices. To prevent this problem of the preservation of IKE and IPsec SAs in such a deployment, you must manually delete the IPsec and IKE SAs by entering the `clear ipsec security-associations` and `clear ike security-associations` commands respectively.

If you decide to configure values manually, the following information shows the complete statement hierarchy and options for dynamic IKE SAs on the AS and MultiServices PICs:

```
[edit services ipsec-vpn]
ike {
    proposal proposal-name {
        authentication-algorithm (md5 | sha1 | sha256);
        authentication-method (pre-shared-keys | rsa-signatures);
        description description;
        dh-group (group1 | group2);
        encryption-algorithm algorithm; # This can be aes-128-cbc, aes-192-cbc,
        # aes-256-cbc, des-cbc, or 3des-cbc.
        lifetime-seconds seconds;
    }
    policy policy-name {
        description description;
        local-id {
            ipv4_addr [ values ];
            key_id [ values ];
        }
        local-certificate certificate-id-name;
        mode (aggressive | main);
        pre-shared-key (ascii-text key | hexadecimal key);
        proposals [ proposal-names ];
        remote-id {
            ipv4_addr [ values ];
            key_id [ values ];
        }
    }
}
ipsec {
    proposal proposal-name {
        authentication-algorithm (hmac-md5-96 | hmac-sha1-96);
        description description;
        encryption-algorithm algorithm; # This can be aes-128-cbc, aes-192-cbc,
        # aes-256-cbc, des-cbc, or 3des-cbc.
        lifetime-seconds seconds;
        protocol (ah | esp | bundle);
    }
    policy policy-name {
        description description;
        perfect-forward-secrecy {
            keys (group1 | group2);
        }
        proposals [ proposal-names ];
    }
}
rule rule-name {
    match-direction (input | output);
    term term-name {
        from {
            destination-address address;
            source-address address;
        }
        then {
            backup-remote-gateway address;
            clear-dont-fragment-bit;
            dynamic {
                ike-policy policy-name;
                ipsec-policy policy-name;
            }
            no-anti-replay;
            remote-gateway address;
            syslog;
        }
    }
}
rule-set rule-set-name {
    [ rule rule-names ];
}
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.2	Starting in Junos OS release 14.2, in an environment in which Juniper Networks MX Series routers interoperate with Cisco ASA devices, IKE security associations (SAs) and IPsec SAs are deleted immediately on the Cisco ASA devices, but they are retained on the MX Series routers.

Directing Traffic into an IPsec Tunnel

IN THIS SECTION

Using a Filter to Select Traffic to Be Secured

For the ES PIC, you need to configure a firewall filter to direct traffic into the IPsec tunnel. To apply a security association to traffic that matches a firewall filter, include the `ipsec-sa` *sa-name* statement at the `[edit firewall filter` *filter-name* `term` *term-name* `then]` hierarchy level.

```
[edit firewall filter filter-name]
term term-name {
    from {
        source-address {
            ip-address;
        }
        destination-address {
            ip-address;
        }
    }
    then {
        count counter-name;
        ipsec-sa sa-name;
    }
}
term other {
    then accept;
}
```

For the AS and MultiServices PICs, you do not need to configure a separate firewall filter. A filter is already built into the IPsec VPN `rule` statement at the `[edit services ipsec-vpn]` hierarchy level. To apply a security association to traffic that matches the IPsec VPN rule, include the **dynamic** or `manual` statement at the `[edit services rule rule-name term term-name then]` hierarchy level. To specify whether the rule should match input or output traffic, include the `match-direction` statement at the `[edit services rule rule-name]` hierarchy level.

After defining the rules for your IPsec VPNs, you must apply the rules to a service set. To do this, include the `ipsec-vpn-rules rule-name` statement at the `[edit services service-set service-set-name]` hierarchy level. Include an IPv4 or IPv6 IPsec gateway with the `local-gateway local-ip-address` statement at the `[edit services service-set service-set-name]` hierarchy level.

Also, you must select either a single interface or a pair of interfaces that participate in IPsec. To select a single interface, include the interface-service `interface-name` statement at the `[edit services service-set service-set-name]` hierarchy level. To select a pair of interfaces and a next hop, include the `next-hop-service` statement at the `[edit services service-set service-set-name]` hierarchy level and specify an inside interface and an outside interface. Only next-hop service sets support IPsec within Layer 3 VPNs and use of routing protocols over the IPsec tunnel.

```
[edit services]
service-set service-set-name {
    interface-service {
        service-interface interface-name;
    }
    next-hop-service {
        inside-service-interface interface-name;
        outside-service-interface interface-name;
    }
    ipsec-vpn-options {
        local-gateway local-ip-address <routing-instance instance-name>;
        trusted-ca ca-profile-name;
    }
    ipsec-vpn-rules rule-name;
}
```

```
ipsec-vpn {
    rule rule-name {
        term term-name {
            from {
                source-address {
                    ip-address;
                }
                destination-address {
                    ip-address;
                }
            }
            then {
                remote-gateway remote-ip-address;
                (dynamic | manual);
            }
        }
        match-direction output;
    }
}
```

Applying the Filter or Service Set to the Interface Receiving Traffic to Be Secured

For the ES PIC, apply your firewall filter on the input interface receiving the traffic that you wish to send to the IPsec tunnel. To do this, include the `filter` statement at the `[edit interfaces` *`interface-name`* `unit` *`unit-number`* `family inet]` hierarchy level.

```
[edit interfaces interface-name unit unit-number family inet]
filter {
    input filter-name;
}
```

For the AS and MultiServices PICs, apply your IPsec-based interface service set to the input interface receiving the traffic that you wish to send to the IPsec tunnel. To do this, include the `service-set` *`service-set-name`* statement at the `[edit interfaces` *`interface-name`* `unit` *`unit-number`* `family inet service (input | output)]` hierarchy level.

```
[edit interfaces interface-name unit unit-number family inet]
service {
    input {
        service-set service-set-name;
    }
    output {
        service-set service-set-name;
    }
}
```

To configure a next-hop-based service set on the AS and MultiServices PICs, include the `service-domain` statement at the `[edit interfaces interface-name unit unit-number]` hierarchy level and specify one logical interface on the AS PIC as an inside interface and a second logical interface on the AS PIC as an outside interface.

```
[edit interfaces sp-fpc/pic/port]
unit 0 {
    family inet {
        address ip-address;
    }
}
unit 1 {
    family inet;
    service-domain inside;
}
unit 2 {
    family inet;
    service-domain outside;
}
```

CHAPTER 5

Using Digital Certificates for IPsec

IN THIS CHAPTER

Using Digital Certificates for IPsec

IN THIS SECTION

Using Digital Certificates for IPsec

A popular way for network administrators to scale an IPsec network is to use digital certificates instead of preshared keys. To enable digital certificates in your network, you need to use a combination of operational mode commands and configuration statements. The following tasks enable you to implement digital certificates on AS and MultiServices PICs installed in M Series and T Series routers:

- "Configuring a CA Profile" on page 87
- "Configuring a Certificate Revocation List" on page 87
- Requesting a CA Digital Certificate
- Generating a Private/Public Key Pair
- Generating and Enrolling a Local Digital Certificate
- Applying the Local Digital Certificate to an IPsec Configuration

- Configuring Automatic Reenrollment of Digital Certificates
- *Monitoring Digital Certificates*
- *Clearing Digital Certificates*

SEE ALSO

> *Digital Certificates*

Configuring a CA Profile

The CA profile contains the name and URL of the CA or RA, as well as some retry timer settings. CA certificates issued by Entrust, VeriSign, and Microsoft are all compatible with M Series, and T Series routers. To configure the domain name of the CA or RA, include the `ca-identity` statement at the `[edit security pki ca-profile` *`ca-profile-name`*`]` hierarchy level. To configure the URL of the CA, include the `url` statement at the `[edit security pki ca-profile` *`ca-profile-name`* `enrollment]` hierarchy level. To configure the number of enrollment attempts the router should perform, include the `retry` statement at the `[edit security pki ca-profile` *`ca-profile-name`* `enrollment]` hierarchy level. To configure the amount of time the router should wait between enrollment attempts, include the `retry-interval` statement at the `[edit security pki ca-profile` *`ca-profile-name`* `enrollment]` hierarchy level.

```
[edit security pki]
ca-profile ca-profile-name {
    ca-identity ca-identity;
    enrollment {
        url url-name;
        retry number-of-enrollment-attempts; # The range is 0 though 100 attempts.
        retry-interval seconds; # The range is 0 though 3600 seconds.
    }
}
```

NOTE: When you delete the entire public key infrastructure (PKI) configuration, all the CA certificates in the device are not deleted as expected. These CA certificates are accessible after you create the CA profiles again.

Configuring a Certificate Revocation List

A certificate revocation list (CRL) contains a list of digital certificates that have been canceled before their expiration date. When a participating peer uses a digital certificate, it checks the certificate signature and validity. It also acquires the most recently issued CRL and checks that the certificate serial number is not on that CRL. By default, CRL verification is enabled on any CA profile running on Junos OS Release 8.1 or later. To disable CRL verification, include the `disable` statement at the [**edit security pki ca-profile** ***ca-profile-name*** **revocation-check**] hierarchy level.

To specify the URL for the Lightweight Directory Access Protocol (LDAP) server where your CA stores its current CRL, include the `url` statement at the [**edit security pki ca-profile** ***ca-profile-name*** **revocation-check crl**] hierarchy level. If the LDAP server requires a password to access the CRL, include the `password` statement at the [**edit security pki ca-profile** ***ca-profile-name*** **revocation-check crl url**] hierarchy level.

NOTE: You do not need to specify a URL for the LDAP server if the certificate includes a certificate distribution point (CDP). The CDP is a field within the certificate that contains information about how to retrieve the CRL for the certificate. The router uses this information to download the CRL automatically. Any LDAP URL you configure takes precedence over the CDP included in the certificate.

If you manually downloaded the CRL, you must manually install it on the router. To manually install the CRL, issue the `request security pki crl load ca-profile` *`ca-profile-name`* `filename` *`path/filename`* command.

To configure the time interval between CRL updates, include the `refresh-interval` statement at the [**edit security ca-profile** ***ca-profile-name*** **revocation-check crl**] hierarchy level.

To override the default behavior and permit IPsec peer authentication to continue when the CRL fails to download, include the `disable on-download-failure` statement at the [**edit security ca-profile** ***ca-profile-name*** **revocation-check crl**] hierarchy level.

```
[edit security pki ca-profile ca-profile-name]
revocation-check {
    disable;
    crl {
        disable on-download-failure;
        refresh-interval number-of-hours { # The range is 0 through 8784 hours.
            url {
                 url-name;
                password;
            }
        }
    }
}
```

CHAPTER 6

Additional IPsec Options

IN THIS CHAPTER

Using Filter-Based Forwarding to Select Traffic to Be Secured

Instead of using a firewall filter, you can also forward traffic into an IPsec security association by using a filter-based forwarding instance. First, configure the filter-based forwarding instance. Then, configure a routing table group to advertise the routes from the filter-based forwarding instance. Next, create a firewall filter for the ES PIC and reference the filter-based forwarding instance. Lastly, apply the filter and IPsec security association to the ES PIC.

```
[edit]
routing-instances {
    forwarding {
        instance-type forwarding;
        routing-options {
            static {
                route 10.10.10.0/24 next-hop 192.168.0.5;
            }
        }
    }
}
routing-options {
    rib-groups {
                                            group-name                  {
            import-rib [ inet.0 forwarding.inet.0 ];
        }
    }
}
firewall {
    family inet {
        filter filter-name {
            term term-name {
                then routing-instance instance-name;
            }
        }
    }
}
[edit]
interfaces {
    es-0/0/0 {
        unit 0 {
            tunnel {
                source source-ip-address;
                destination destination-ip-address;
            }
            family inet {
                ipsec-sa sa-name;
                filter {
                    input filter-name;
                }
                address ip-address;
            }
        }
    }
}
```

Using IPsec with a Layer 3 VPN

Some key concepts to keep in mind when configuring IPsec within a VPN include the following:

- Add the inside services interface for a next-hop style service set into the routing instance by including the `interface sp-`*`fpc/pic/port`* statement at the `[edit routing-instances `*`instance-name`*`]` hierarchy level.

- For interface style service sets, add the interface on which you apply the service set and the services interface by including both interfaces at the `[edit routing-instances `*`instance-name`*`]` hierarchy level.

- To define a routing instance for the local gateway within the service set, include the **routing-instance *instance-name*** option at the `[edit services service-set` *`service-set-name`* `ipsec-vpn-options local-gateway` *`address`*`]` hierarchy level.

The following configuration for an AS PIC on a provider edge (PE) router demonstrates the use of next-hop service sets with an IKE dynamic SA in a VPN routing and forwarding (VRF) routing instance.

```
[edit]
interfaces {
    so-0/0/0 {
        description "Interface connected to the customer edge (CE) router";
        unit 0 {
            family inet {
                address 10.6.6.6/32;
            }
        }
    }
    so-2/2/0 {
        description "Source IPsec tunnel interface to the network core";
        unit 0 {
            family inet {
                address 10.10.1.1/30;
            }
        }
    }
    sp-3/1/0 {
        description "AS PIC interface";
        unit 0 {
            family inet {
                address 10.7.7.7/32;
            }
        }
        unit 1 {
            family inet;
            service-domain inside;
        }
        unit 2 {
            family inet;
            service-domain outside;
        }
    }
}
```

```
policy-options {
    policy-statement vpn-export-policy {
        then {
            community add community-name;
            accept;
        }
    }
    policy-statement vpn-import-policy {
        term term-name {
            from community community-name;
            then accept;
        }
    }
    community community-name members target:100:20;
}
routing-instances {
    vrf {
        instance-type vrf;
        interface sp-3/1/0.1; # Inside sp interface.
        interface so-0/0/0.0; # Interface that connects to the CE router.
        route-distinguisher route-distinguisher;
        vrf-import vpn-import-policy;
        vrf-export vpn-export-policy;
        routing-options {
            static {
                route ip-address/prefix next-hop so-0/0/0.0; # Routes for the CE router.
                route ip-address/prefix next-hop sp-3/1/0.1; # Routes for IPsec.
            }
        }
    }
}
services {
    service-set service-set-name {
        next-hop-service {
            inside-service-interface sp-3/1/0.1;
            outside-service-interface sp-3/1/0.2;
        }
        ipsec-vpn-options {
            local-gateway 10.10.1.1;
        }
        ipsec-vpn-rules rule-name;
    }
    ipsec-vpn {
        rule rule-name {
            term term-name {
                from {
                    source-address {
                        source-ip-address;
                    }
                }
                then {
                    remote-gateway 10.10.1.2;
                    dynamic {
                        ike-policy ike-policy-name;
                    }
                }
            }
            match-direction direction;
        }
        ike {
            policy ike-policy-name {
                pre-shared-key ascii-text preshared-key;
            }
        }
    }
}
```

For more information on VRF routing instances, see the *Junos VPNs Configuration Guide*. For more information on next-hop service sets, see the *Junos Services Interfaces Configuration Guide*.

Securing BGP Sessions with IPsec Transport Mode

For the ES PIC, you can use IPsec to secure BGP sessions between Routing Engines in M Series and T Series platforms. To configure, create a transport mode security association and apply the SA to the BGP configuration by including the `ipsec-sa` statement at the `[edit protocols bgp group` *`group-name`*`]` hierarchy level.

```
[edit]
protocols {
    bgp {
        group group-name {
            local-address ip-address;
            export export-policy;
```

```
              peer-as as-number;
              ipsec-sa sa-name;
              neighbor peer-ip-address;
          }
      }
  }
```

RELATED DOCUMENTATION

IPSec Modes | 72

Securing OSPFv2 Networks with IPsec Transport Mode

By default, you can configure MD5 or simple text password-based authentication over OSPFv2 links. In addition to these basic authentications, the Junos OS supports OSPFv2 with a security authentication header (AH), Encapsulating Security Payload (ESP), or an IPsec protocol bundle that supports both AH and ESP. You can configure IPsec over OSPFv2 using transport mode security associations on physical, sham, or virtual links.

Because the Junos OS supports only bidirectional security associations over OSPFv2, OSPFv2 peers must be configured with the same IPsec security association. Configuring OSPFv2 peers with different security associations or with dynamic IKE will prevent adjacencies from being established. In addition, you must configure identical security associations for sham links with the same remote endpoint address, for virtual links with the same remote endpoint address, for all neighbors on OSPF nonbroadcast multiaccess (NBMA) or point-to-multipoint links, and for every subnet that is part of a broadcast link.

To create a manual bidirectional security association, include the `security-association` *`security-association-name`* statement at the [**edit security ipsec**] hierarchy level:

```
[edit]
security {
    ipsec {
        security-association security-association name {
            mode transport;
            manual {
                direction bidirectional {
                    protocol (ah | esp | bundle);
                    spi spi--value;
                    authentication {
                        algorithm (hmac-md5-96 | hmac-sha1-96);
                        key (ascii-text key | hexadecimal key);
                    }
                }
            }
        }
    }
}
```

To configure IPsec on an OSPFv2 interface, create a transport mode security association and include the `ipsec-sa name` statement at the [**edit protocols ospf area *area-id***] hierarchy level:

```
[edit]
protocols {
    ospf {
        area area-id {
            interface interface-name {
                ipsec-sa sa-name;
            }
            virtual-link neighbor-id a.b.c.d transit-area x.x.x.x {
                ipsec-sa sa-name;
            }
            sham-link-remote {
                ipsec-sa sa-name;
            }
        }
    }
}
```

To verify your configuration, enter the `show ospf interface detail` command. This command gives detailed information about the **ospfv2** interface and displays the interface's security association at the bottom of the output. In the example below, the security association configured on this router is **sa1**.

```
user@router> show ospf interface detail
Interface              State     Area            DR ID           BDR ID Nbrs
fe-0/0/1.0             BDR      0.0.0.0         192.168.37.12   10.255.245.215 1
Type LAN, address 192.168.37.11, Mask 255.255.255.248, MTU 4460, Cost 40
DR addr 192.168.37.12, BDR addr 192.168.37.11, Adj count 1, Priority 128
Hello 10, Dead 40, ReXmit 5, Not Stub
```

```
t1-0/2/1.0             PtToPt   0.0.0.0         0.0.0.0         0.0.0.0 0
Type P2P, Address 0.0.0.0, Mask 0.0.0.0, MTU 1500, Cost 2604
Adj count 0
Hello 10, Dead 40, ReXmit 5, Not Stub
Auth type: MD5, Active key ID 3, Start time 2002 Nov 19 10:00:00 PST
IPsec SA Name: sa1
```

RELATED DOCUMENTATION

IPSec Modes | **72**

CHAPTER 7

Configuring IPsec Dynamic Endpoints

IN THIS CHAPTER

Configuring an IKE Access Profile for IPsec Dynamic Endpoint Tunnels

You can configure only one tunnel profile per service set for all dynamic peers. The configured preshared key in the profile is used for IKE authentication of all dynamic peers terminating in that service set.

The IKE tunnel profile specifies all the information needed to complete the IKE negotiation. For more information on access profiles, see the *Junos System Basics Configuration Guide*.

```
[edit access]
profile profile-name {
    client * {
        ike {
            allowed-proxy-pair {
                remote remote-proxy-address local local-proxy-address;
            }
            pre-shared-key ([ ascii-text key-string ] | [hexadecimal key-string ]);
            interface-id string-value;
            ipsec-policy ipsec-policy;
        }
    }
}
```

NOTE: For dynamic peers, the Junos OS supports only IKE **main** mode with the preshared key method of authentication. In this mode, an IPv4 or IPv6 address is used to identify a tunnel peer to get the preshared key information. The **client** value * (wildcard) means that the configuration within this profile is valid for all dynamic peers terminating within the service set accessing this profile.

The following statements are the parts of the IKE profile:

- **allowed-proxy-pair**—During phase 2 IKE negotiation, the remote peer supplies its network address (**remote**) and its peer's network address (**local**). Since multiple dynamic tunnels are authenticated through the same mechanism, this statement must include the list of possible combinations. If the dynamic peer does not present a valid combination, the phase 2 IKE negotiation fails.

  By default, remote 0.0.0.0/0 local 0.0.0.0/0 is used if no values are configured.

- **pre-shared-key**—Mandatory key used to authenticate the dynamic peer during IKE phase 1 negotiation. This key must be configured on both ends of the tunnel and distributed through an out-of-band secure mechanism. You can configure the key value either in **hexadecimal** or **ascii-text** format.

- **interface-id**—Interface identifier, a mandatory attribute used to derive the logical service interface information for the session.

- **ipsec-policy**—Name of the IPsec policy that defines the IPsec policy information for the session. You define the IPsec policy at the `[edit services ipsec-vpn ipsec policy` *`policy-name`*`]` hierarchy level. If no policy is set, any policy proposed by the dynamic peer is accepted.

Configuring the Service Set for IPsec Dynamic Endpoint Tunnels

To complete a dynamic endpoint tunnel configuration, you need to reference the IKE access profile configured at the [**edit access**] hierarchy level in the service set. To do this, include the `ike-access-profile` statement at the [**edit services service-set** ***name*** **ipsec-vpn-options**] hierarchy level:

```
[edit services]
service-set name {
    next-hop-service {
        inside-service-interface interface-name;
        outside-service-interface interface-name;
    }
    ipsec-vpn-options {
        local-gateway address;
        ike-access-profile profile-name;
    }
}
```

You can reference only one access profile in each service set. This profile is used to negotiate IKE and IPsec security associations with dynamic peers only.

NOTE: If you configure an IKE access profile in a service set, no other service set can share the same **local-gateway** address.

Configuring the Interface Identifier for IPsec Dynamic Endpoint Tunnels

You can configure an interface identifier for a group of dynamic peers, which specifies which adaptive services logical interface(s) take part in the dynamic IPsec negotiation. By assigning the same interface identifier to multiple logical interfaces, you can create a pool of interfaces for this purpose. To configure, include the `ipsec-interface-id` statement at the [**edit interfaces** ***interface-name***] hierarchy level:

```
[edit interfaces sp-fpc/pic/port]
unit logical-unit-number {
    dial-options {
        ipsec-interface-id identifier;
        (shared | dedicated);
    }
}
```

Specifying the interface identifier in the `dial-options` statement makes this logical interface part of the pool identified by the IPsec interface identifier.

NOTE: Only one interface identifier can be specified at a time. You can include the `ipsec-interface-id` statement or the `l2tp-interface-id` statement, but not both simultaneously.

The `shared` statement enables one logical interface to be shared across multiple tunnels. The `dedicated` statement specifies that the logical interface is associated with a single tunnel, which is necessary when you are configuring an IPsec link-type tunnel. You must include the `dedicated` statement when you specify an **ipsec-interface-id** value.

Configuring an IKE Access Profile for IPsec Dynamic Endpoint Tunnels

IN THIS SECTION

Configuring an IKE Access Profile for IPsec Dynamic Endpoint Tunnels

You can configure only one tunnel profile per service set for all dynamic peers. The configured preshared key in the profile is used for IKE authentication of all dynamic peers terminating in that service set.

The IKE tunnel profile specifies all the information needed to complete the IKE negotiation. For more information on access profiles, see the *Junos System Basics Configuration Guide*.

```
[edit access]
profile profile-name {
    client * {
        ike {
            allowed-proxy-pair {
                remote remote-proxy-address local local-proxy-address;
            }
            pre-shared-key ([ ascii-text key-string ] | [hexadecimal key-string ]);
            interface-id string-value;
            ipsec-policy ipsec-policy;
        }
    }
}
```

NOTE: For dynamic peers, the Junos OS supports only IKE **main** mode with the preshared key method of authentication. In this mode, an IPv4 or IPv6 address is used to identify a tunnel peer to get the preshared key information. The **client** value ***** (wildcard) means that the configuration within this profile is valid for all dynamic peers terminating within the service set accessing this profile.

The following statements are the parts of the IKE profile:

- **allowed-proxy-pair**—During phase 2 IKE negotiation, the remote peer supplies its network address (**remote**) and its peer’s network address (**local**). Since multiple dynamic tunnels are authenticated through the same mechanism, this statement must include the list of possible combinations. If the dynamic peer does not present a valid combination, the phase 2 IKE negotiation fails.

  By default, remote 0.0.0.0/0 local 0.0.0.0/0 is used if no values are configured.

- **pre-shared-key**—Mandatory key used to authenticate the dynamic peer during IKE phase 1 negotiation. This key must be configured on both ends of the tunnel and distributed through an out-of-band secure mechanism. You can configure the key value either in **hexadecimal** or **ascii-text** format.

- **interface-id**—Interface identifier, a mandatory attribute used to derive the logical service interface information for the session.

- **ipsec-policy**—Name of the IPsec policy that defines the IPsec policy information for the session. You define the IPsec policy at the `[edit services ipsec-vpn ipsec policy` *`policy-name`*`]` hierarchy level. If no policy is set, any policy proposed by the dynamic peer is accepted.

Configuring the Service Set for IPsec Dynamic Endpoint Tunnels

To complete a dynamic endpoint tunnel configuration, you need to reference the IKE access profile configured at the [**edit access**] hierarchy level in the service set. To do this, include the `ike-access-profile` statement at the [**edit services service-set *name* ipsec-vpn-options**] hierarchy level:

```
[edit services]
service-set name {
    next-hop-service {
        inside-service-interface interface-name;
        outside-service-interface interface-name;
    }
    ipsec-vpn-options {
        local-gateway address;
        ike-access-profile profile-name;
    }
}
```

You can reference only one access profile in each service set. This profile is used to negotiate IKE and IPsec security associations with dynamic peers only.

NOTE: If you configure an IKE access profile in a service set, no other service set can share the same **local-gateway** address.

Configuring the Interface Identifier for IPsec Dynamic Endpoint Tunnels

You can configure an interface identifier for a group of dynamic peers, which specifies which adaptive services logical interface(s) take part in the dynamic IPsec negotiation. By assigning the same interface identifier to multiple logical interfaces, you can create a pool of interfaces for this purpose. To configure, include the `ipsec-interface-id` statement at the [**edit interfaces *interface-name***] hierarchy level:

```
[edit interfaces sp-fpc/pic/port]
unit logical-unit-number {
    dial-options {
        ipsec-interface-id identifier;
        (shared | dedicated);
    }
}
```

Specifying the interface identifier in the `dial-options` statement makes this logical interface part of the pool identified by the IPsec interface identifier.

NOTE: Only one interface identifier can be specified at a time. You can include the `ipsec-interface-id` statement or the `l2tp-interface-id` statement, but not both simultaneously.

The `shared` statement enables one logical interface to be shared across multiple tunnels. The `dedicated` statement specifies that the logical interface is associated with a single tunnel, which is necessary when you are configuring an IPsec link-type tunnel. You must include the `dedicated` statement when you specify an **ipsec-interface-id** value.

Configuring Multiple Routed Tunnels in a Single Next-Hop Service Set

You can optionally configure several routed IPSec tunnels within a single next-hop service set. To do so, start by establishing multiple services interfaces as inside interfaces by including the **service-domain inside** statement at the **[edit interfaces sp-*fpc/pic/port* unit *logical-unit-number*]** hierarchy level. Then, include the `ipsec-inside-interface` statement at the `[edit services ipsec-vpn rule` *`rule-name`* `term` *`term-name`* `from]` hierarchy level.

NOTE: The full IPsec and IKE proposals and policies are not shown in the following example for the sake of brevity.

```
[edit]
interfaces {
    sp-3/3/0 {
        unit 3 {
            family inet;
            service-domain inside;
        }
        unit 4 {
            family inet;
            service-domain outside;
        }
        unit 5 {
            family inet;
            service-domain inside;
        }
    }
}
services {
    service-set link_type_ss_1 {
        next-hop-service {
            inside-service-interface sp-3/3/0.3;
            outside-service-interface sp-3/3/0.4;
        }
        ipsec-vpn-options {
            local-gateway 10.8.7.2;
        }
        ipsec-vpn-rules link_rule_1;
    }
    ipsec-vpn {
        rule link_rule_1 {
            term 1 {
                from {
                    ipsec-inside-interface sp-3/3/0.3;
                }
                then {
                    remote-gateway 10.10.7.3;
                    backup-remote-gateway 10.8.7.1;
                    dynamic {
                        ike-policy main_mode_ike_policy;
                        ipsec-policy dynamic_ipsec_policy;
                    }
                }
            }
            term 2 {
                from {
                    ipsec-inside-interface sp-3/3/0.5;
                }
                then {
                    remote-gateway 10.12.7.5;
                    dynamic {
                        ike-policy main_mode_ike_policy;
                        ipsec-policy dynamic_ipsec_policy;
                    }
                }
            }
            match-direction input;
        }
    }
}
```

To confirm that your configuration is working, issue the `show services ipsec-vpn ipsec security-associations` command. Notice that each IPsec inside interface that you assigned to each IPsec tunnel is included in the output of this command.

```
user@router> show services ipsec-vpn ipsec security-associations
Service set: link_type_ss_1

  Rule: link_rule_1, Term: 1, Tunnel index: 1
  Local gateway: 10.8.7.2, Remote gateway: 10.8.7.1
  IPSec inside interface: sp-3/3/0.3
    Direction SPI         AUX-SPI     Mode       Type     Protocol
    inbound   3216392497  0           tunnel     dynamic  ESP
    outbound  398917249   0           tunnel     dynamic  ESP

  Rule: link_rule_1, Term: 2, Tunnel index: 2
  Local gateway: 10.8.7.2, Remote gateway: 10.12.7.5
  IPSec inside interface: sp-3/3/0.5
    Direction SPI         AUX-SPI     Mode       Type     Protocol
    inbound   762146783   0           tunnel     dynamic  ESP
    outbound  319191515   0           tunnel     dynamic  ESP
```

SEE ALSO

Configuring IKE Dynamic SAs | 77

CHAPTER 8

Additional ES and AS PIC Configuration Examples

IN THIS CHAPTER

Example: ES PIC Manual SA Configuration

IN THIS SECTION

Figure 4: ES PIC Manual SA Topology Diagram

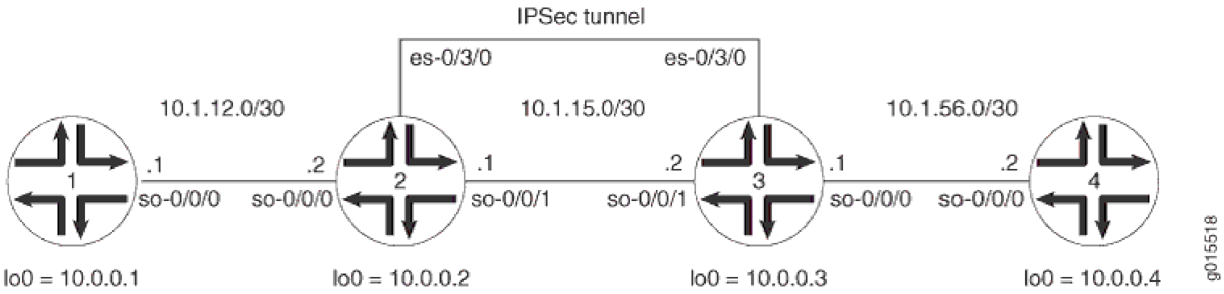


Figure 4 on page 107 shows an IPSec topology containing a group of four routers. Routers 2 and 3 establish an IPSec tunnel using an ES PIC and manual SA settings. Routers 1 and 4 provide basic connectivity and are used to verify that the IPSec tunnel is operational.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R2 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.1;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

On Router 2, enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional manual SA called **sa-manual** at the `[edit security ipsec security-association]` hierarchy level. Use AH for the protocol, **400** for the SPI, HMAC-MD5-96 for authentication, and a 32-bit hexadecimal authentication key for the MD5 authentication key. (For more information about key length, see "

Authentication and Encryption Key Lengths" on page 36.) Because you are using AH, there is no need to configure encryption.

To direct traffic into the ES PIC and the IPSec tunnel, create two firewall filters. The **es-traffic** filter matches inbound traffic from Router 1 destined for Router 4, whereas the **es-return** filter matches the return path from Router 4 to Router 1. Apply the **es-traffic** filter to the **so-0/0/0** interface; then apply both the **es-return** filter and the **sa-manual** SA to the **es-0/3/0** interface.

Router 2

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R1 so-0/0/0";
        unit 0 {
            family inet {
                filter {
                                           input es-traffic; # Apply a filter that sends traffic
to the IPSec tunnel here.
                }
                address 10.1.12.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R3 so-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.1/30;
            }
        }
    }
    es-0/3/0 {
        unit 0 {
                                 tunnel { # Specify the IPSec tunnel endpoints here.
                source 10.1.15.1;
                destination 10.1.15.2;
            }
            family inet {
                                     ipsec-sa sa-manual; # Apply the manual SA here.
                filter {
                                           input es-return; # Apply the filter that matches
return IPSec traffic here.
```

```
                }
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.2/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.2;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface so-0/0/1.0;
            interface lo0.0;
        }
    }
}
security {
    ipsec {
                           security-association sa-manual { # Define the manual SA
specifications here.
                               mode tunnel;
                                 manual {
                                       direction bidirectional {
                                           protocol ah;
                                           spi 400;
                                             authentication {
                                                 algorithm hmac-md5-96;
                                                 key hexadecimal "$ABC123”;
                    }
                }
            }
        }
    }
}
```

```
# The 32-bit unencrypted hexadecimal key is abcdef01abcdef01abcdef01abcdef01.
firewall {
                     filter es-traffic { # Define a filter that sends traffic to the IPSec
tunnel here.
        term to-es {
            from {
                source-address {
                    10.1.12.0/24;
                }
                destination-address {
                    10.1.56.0/24;
                }
            }
            then {
                count ipsec-tunnel;
                ipsec-sa sa-manual;
            }
        }
        term other {
            then accept;
        }
    }
                     filter es-return { # Define a filter that matches return IPSec traffic here.
        term return {
            from {
                source-address {
                    10.1.56.0/24;
                }
                destination-address {
                    10.1.12.0/24;
                }
            }
            then accept;
        }
    }
}
```

On Router 3, enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional manual SA called **sa-manual** at the **[edit security ipsec security-association]** hierarchy level. Use the exact same specifications that you used for the SA on Router 2: AH for the protocol, **400** for the SPI, HMAC-MD5-96 for authentication, and a 32-bit hexadecimal authentication key of **abcdef01abcdef01abcdef01abcdef01** for the MD5 authentication key. (For more information about authentication key length, see " Authentication and Encryption Key Lengths" on page 36.) Because you are using AH, there is no need to configure an encryption algorithm.

To direct traffic into the ES PIC and the IPSec tunnel, create two firewall filters. The **es-traffic** filter matches inbound traffic from Router 4 destined for Router 1, whereas the **es-return** filter matches the return path from Router 1 to Router 4. Apply the **es-traffic** filter to the **so-0/0/0** interface; then apply both the **es-return** filter and the **sa-manual** SA to the **es-0/3/0** interface.

Router 3

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R4 so-0/0/0";
        unit 0 {
            family inet {
                filter {
                                           input es-traffic; # Apply a filter that sends traffic
to the IPSec tunnel here.
                }
                address 10.1.56.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R2 so-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.2/30;
            }
        }
    }
    es-0/3/0 {
        unit 0 {
                                 tunnel { # Specify the IPSec tunnel endpoints here.
                                     source 10.1.15.2;
                destination 10.1.15.1;
            }
            family inet {
                                     ipsec-sa sa-manual; # Apply the manual SA here.
                filter {
                                           input es-return; # Apply the filter that matches
return IPSec traffic here.
                }
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.3/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.3;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface so-0/0/1.0;
            interface lo0.0;
        }
    }
}
security {
    ipsec {
                           security-association sa-manual { # Define the manual SA
specifications here.
                               mode tunnel;
                                 manual {
                                       direction bidirectional {
                                           protocol ah;
                                           spi 400;
                                             authentication {
                                                 algorithm hmac-md5-96;
                                                 key hexadecimal "$ABC123”;
                    }
                }
            }
        }
    }
}
```

```
## The 32-bit unencrypted hexadecimal key is abcdef01abcdef01abcdef01abcdef01.
firewall {
                     filter es-traffic { # Define a filter that sends traffic to the IPSec
tunnel here.
        term to-es {
            from {
                source-address {
                    10.1.56.0/24;
                }
                destination-address {
                    10.1.12.0/24;
                }
            }
            then {
                count ipsec-tunnel;
                ipsec-sa sa-manual;
            }
        }
        term other {
            then accept;
        }
    }
                     filter es-return { # Define a filter that matches return IPSec traffic here.
        term return {
            from {
                source-address {
                    10.1.12.0/24;
                }
                destination-address {
                    10.1.56.0/24;
                }
            }
            then accept;
        }
    }
}
```

On Router 4, provide basic OSPF connectivity to Router 3.

Router 4

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R3 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.56.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.4/32;
            }
        }
    }
}
routing-options {
        router-id 10.0.0.4;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.ping
        }
    }
}
```

Verifying Your Work

To verify proper operation of a manual IPSec SA on the ES PIC, use the following commands:

- ping

- `show ipsec security-associations (detail)`
- **traceroute**

The following sections show the output of these commands used with the configuration example:

Router 1

On Router 1, issue a `ping` command to the **so-0/0/0** interface of Router 4 to send traffic across the IPsec tunnel.

```
user@R1> ping 10.1.56.2
PING 10.1.56.2 (10.1.56.2): 56 data bytes
64 bytes from 10.1.56.2: icmp_seq=0 ttl=253 time=0.939 ms
64 bytes from 10.1.56.2: icmp_seq=1 ttl=253 time=0.886 ms
64 bytes from 10.1.56.2: icmp_seq=2 ttl=253 time=0.826 ms
^C
--- 10.1.56.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.826/0.884/0.939/0.046 ms
```

You can also issue the `traceroute` command to verify that traffic to **10.1.56.2** travels over the IPsec tunnel between Router 2 and Router 3. Notice that the second hop does not reference **10.1.15.2**—the physical interface on Router 3. Instead, the loopback address of **10.0.0.3** on Router 3 appears as the second hop. This indicates that the IPSec tunnel is operating correctly.

```
user@R1> traceroute 10.1.56.2
traceroute to 10.1.56.2 (10.1.56.2), 30 hops max, 40 byte packets
 1  10.1.12.1 (10.1.12.1)  0.655 ms  0.549 ms  0.508 ms
 2  10.0.0.3 (10.0.0.3)  0.833 ms  0.786 ms  0.757 ms
 3  10.1.56.2 (10.1.56.2)  0.808 ms  0.741 ms  0.716 ms
```

Router 2

Another way to verify that matched traffic is being diverted to the bidirectional IPsec tunnel is to view the firewall filter counter. After you issue the `ping` command from Router 1 (three packets), the **es-traffic** firewall filter counter looks like this:

```
user@R2> show firewall filter es-traffic
Filter: es-traffic
```

```
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          252                    3
```

After you issue the `ping` command from both Router 1 (three packets) and Router 4 (two packets), the **es-traffic** firewall filter counter looks like this:

```
user@R2> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          420                    5
```

To verify that the IPsec security association is active, issue the `show ipsec security-associations detail` command. Notice that the SA contains the settings you specified, such as AH for the protocol and HMAC-MD5-96 for the authentication algorithm.

```
user@R2> show ipsec security-associations detail
Security association: sa-manual, Interface family: Up

  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  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)

      Direction: inbound, SPI: 400, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: AH, Authentication: hmac-md5-96, Encryption: None
    Anti-replay service: Disabled

    Direction: outbound, SPI: 400, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: AH, Authentication: hmac-md5-96, Encryption: None
    Anti-replay service: Disabled
```

Router 3

View the firewall filter counter to continue verifying that matched traffic is being diverted to the bidirectional IPsec tunnel. After you issue the `ping` command from Router 1 (three packets), the **es-traffic** firewall filter counter looks like this:

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                 Bytes              Packets
ipsec-tunnel                                           252                    3
```

After you issue the `ping` command from both Router 1 (three packets) and Router 4 (two packets), the **es-traffic** firewall filter counter looks like this:

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                 Bytes              Packets
ipsec-tunnel                                           420                    5
```

To verify that the IPsec security association is active, issue the `show ipsec security-associations detail` command. Notice that the SA on Router 3 contains the same settings you specified on Router 2.

```
user@R3> show ipsec security-associations detail
Security association: sa-manual, Interface family: Up

  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  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)

    Direction: inbound, SPI: 400, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: AH, Authentication: hmac-md5-96, Encryption: None
    Anti-replay service: Disabled

    Direction: outbound, SPI: 400, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: AH, Authentication: hmac-md5-96, Encryption: None
    Anti-replay service: Disabled
```

Router 4

On Router 4, issue a `ping` command to the **so-0/0/0** interface of Router 1 to send traffic across the IPsec tunnel.

```
user@R4> ping 10.1.12.2
PING 10.1.12.2 (10.1.12.2): 56 data bytes
64 bytes from 10.1.12.2: icmp_seq=0 ttl=253 time=0.937 ms
64 bytes from 10.1.12.2: icmp_seq=1 ttl=253 time=0.872 ms
^C
--- 10.1.12.2 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.872/0.905/0.937/0.032 ms
```

You can also issue the `traceroute` command to verify that traffic to **10.1.12.2** travels over the IPsec tunnel between Router 3 and Router 2. Notice that the second hop does not reference **10.1.15.1**—the physical interface on Router 2. Instead, the loopback address of **10.0.0.2** on Router 2 appears as the second hop. This indicates that the IPSec tunnel is operating correctly.

```
user@R4> traceroute 10.1.12.2
traceroute to 10.1.12.2 (10.1.12.2), 30 hops max, 40 byte packets
 1  10.1.56.1 (10.1.56.1)  0.670 ms  0.589 ms  0.548 ms
 2  10.0.0.2 (10.0.0.2)  0.815 ms  0.791 ms  0.763 ms
 3  10.1.12.2 (10.1.12.2)  0.798 ms  0.741 ms  0.714 ms
```

Example: AS PIC Manual SA Configuration

IN THIS SECTION

Figure 5: AS PIC Manual SA Topology Diagram

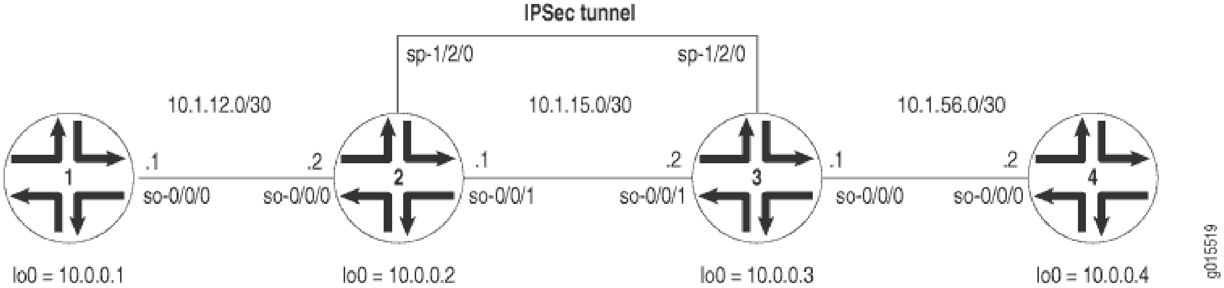


Figure 5 on page 120 shows a similar IPSec topology to the one used in the ES PIC manual SA example. The difference is that Routers 2 and 3 establish an IPSec tunnel using an AS PIC and use slightly modified manual SA settings. Routers 1 and 4 again provide basic connectivity and are used to verify that the IPSec tunnel is operational.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R2 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.1;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

On Router 2, enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional manual SA in a rule called rule-manual-SA-BiEspshades at the `[edit ipsec-vpn rule]` hierarchy level. Reference this rule in a service set called service-set-manual-BiEspshades at the `[edit services service-set]` hierarchy level.

Configure all specifications for your manual SA. Use ESP for the protocol, **261** for the SPI, HMAC-SHA1-96 for authentication, DES-CBC for encryption, a 20-bit ASCII authentication key for the SHA-1 authentication key, and an 8-bit ASCII encryption key for the DES-CBC authentication key. (For more information about key lengths, see " Authentication and Encryption Key Lengths" on page 36.)

To direct traffic into the AS PIC and the IPSec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPSec inside interface into the OSPF configuration.

Router 2

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R1 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R3 so-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.1/30;
            }
        }
    }
    sp-1/2/0 {
        services-options {
            syslog {
                host local {
                    services info;
                }
            }
        }
        unit 0 {
            family inet {
            }
                                 unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
                family inet;
                service-domain inside;
            }
                                 unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
                family inet;
                service-domain outside;
            }
        }
        lo0 {
            unit 0 {
                family inet {
                    address 10.0.0.2/32;
                }
            }
        }
    }
    routing-options {
        router-id 10.0.0.2;
    }
    protocols {
        ospf {
            area 0.0.0.0 {
                interface so-0/0/0.0;
                interface lo0.0;
                                     interface sp-1/2/0.1; # This sends OSPF traffic over the
IPSec tunnel.
            }
        }
    }
    services {
                           service-set service-set-manual-BiEspshades { # Define your service
set here.
                                 next-hop-service { # Required for dynamic routing protocols
such as OSPF.
                inside-service-interface sp-1/2/0.1;
                outside-service-interface sp-1/2/0.2;
            }
            ipsec-vpn-options {
                                     local-gateway 10.1.15.1; # Specify the local IP address of
the IPSec tunnel.
            }
                               ipsec-vpn-rules rule-manual-SA-BiEspshades; # Reference the IPSec
rule here.
        }
        ipsec-vpn {
                                 rule rule-manual-SA-BiEspshades { # Define your IPSec VPN rule
here.
                term term-manual-SA-BiEspshades {
                    then {
                                                 remote-gateway 10.1.15.2; # The remote IP
address of the IPSec tunnel.
                                                   manual { # Define the manual SA
specifications here.
                                                         direction bidirectional
{
                                protocol esp;
                                spi 261;
                                authentication {
                                    algorithm hmac-sha1-96;
                                    key ascii-text "$ABC123";
                                    ## The unencrypted key is juniperjuniperjunipe (20
characters for HMAC-SHA-1-96).
                                }
                                                               encryption
{
                                    algorithm des-cbc;
                                    key ascii-text "$ABC123";
                                    ## The unencrypted key is juniperj (8 characters for DES-
CBC).
                                }
                            }
                        }
                    }
                }
                                     match-direction input; # Correct match direction for next-
hop service sets.
            }
        }
```

```
    }
    }
security {
    pki {
        auto-re-enrollment {
            certificate-id certificate-name {
                ca-profile ca-profile-name;
                challenge-password password;
                re-enroll-trigger-time-percentage percentage; #Percentage of validity-period
# (specified in certificate) when automatic
# reenrollment should be initiated.
                re-generate-keypair;
                validity-period number-of-days;
            }
        }
    }
```

On Router 3, enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional manual SA in a rule called rule-manual-SA-BiEspshades at the `[edit ipsec-vpn rule]` hierarchy level. Reference this rule in a service set called service-set-manual-BiEspshades at the `[edit services service-set]` hierarchy level.

Configure the same specifications for your manual SA that you specified on Router 2. Use ESP for the protocol, **261** for the SPI, HMAC-SHA1-96 for authentication, DES-CBC for encryption, a 20-bit ASCII authentication key for the SHA-1 authentication key, and an 8-bit ASCII encryption key for the DES-CBC authentication key. (For more information about key lengths, see " Authentication and Encryption Key Lengths" on page 36.)

To direct traffic into the AS PIC and the IPSec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPSec inside interface into the OSPF configuration.

Router 3

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R4 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.56.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R2 so-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.2/30;
            }
        }
    }
    sp-1/2/0 {
        services-options {
            syslog {
                host local {
                    services info;
                }
            }
        }
        unit 0 {
            family inet {
            }
                                 unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
                family inet;
                service-domain inside;
            }
                                 unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
                family inet;
                service-domain outside;
            }
        }
        lo0 {
            unit 0 {
                family inet {
                    address 10.0.0.3/32;
                }
            }
        }
    }
    routing-options {
        router-id 10.0.0.3;
    }
    protocols {
        ospf {
            area 0.0.0.0 {
                interface so-0/0/0.0;
                interface lo0.0;
                                     interface sp-1/2/0.1; # This sends OSPF traffic over the
IPSec tunnel.
            }
        }
    }
    services {
                           service-set service-set-manual-BiEspshades { # Define your service
set here.
                                 next-hop-service { # Required for dynamic routing protocols
such as OSPF.
                inside-service-interface sp-1/2/0.1;
                outside-service-interface sp-1/2/0.2;
            }
            ipsec-vpn-options {
                                     local-gateway 10.1.15.2; # Specify the local IP address of
the IPSec tunnel.
            }
                               ipsec-vpn-rules rule-manual-SA-BiEspshades; # Reference the IPSec
rule here.
        }
        ipsec-vpn {
                                 rule rule-manual-SA-BiEspshades { # Define your IPSec VPN rule
here.
                term term-manual-SA-BiEspshades {
                    then {
                                                 remote-gateway 10.1.15.1; # The remote IP
address of the IPSec tunnel.
                                                   manual { # Define the manual SA
specifications here.
                                                         direction bidirectional
{
                                onprotocol esp;
                                spi 261;
                                authentication {
                                    algorithm hmac-sha1-96;
                                    key ascii-text "$ABC123";
                                    ## The unencrypted key is juniperjuniperjunipe (20
characters for HMAC-SHA-1-96).
                                }
                                                               encryption
{
                                    algorithm des-cbc;
                                    key ascii-text "$ABC123";
                                    ## The unencrypted key is juniperj (8 characters for DES-
CBC).
                                }
                            }
                        }
                    }
                }
                                    match-direction input; # Specify in which direction the
rule should match.
            }
        }
    }
}
```

On Router 4, provide basic OSPF connectivity to Router 3.

Router 4

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R3 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.56.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.4/32;
            }
        }
    }
}
routing-options {
        router-id 10.0.0.4;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

Verifying Your Work

To verify proper operation of a manual IPSec SA on the AS PIC, use the following commands:

- **ping**
- `show services ipsec-vpn ipsec security-associations` (detail)
- `show services ipsec-vpn ipsec statistics`

The following sections show the output of these commands used with the configuration example:

Router 1

On Router 1, issue a `ping` command to the **lo0** interface on Router 4 to send traffic across the IPsec tunnel.

```
user@R1> ping 10.0.0.4
PING 10.0.0.4 (10.0.0.4): 56 data bytes
64 bytes from 10.0.0.4: icmp_seq=0 ttl=254 time=1.375 ms
64 bytes from 10.0.0.4: icmp_seq=1 ttl=254 time=18.375 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=254 time=1.120 ms
^C
--- 10.0.0.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.120/6.957/18.375/8.075 ms
```

Router 2

To verify that the IPSec security association is active, issue the **show services ipsec-vpn ipsec security-associations detail** command. Notice that the SA contains the settings you specified, such as ESP for the protocol and HMAC-SHA1-96 for the authentication algorithm.

```
user@R2> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-manual-BiEspshades
  Rule: rule-manual-SA-BiEspshades, Term: term-manual-SA-BiEspshades,
  Tunnel index: 1
  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  Local identity: ipv4_subnet(any:0,[0..7]=10.0.0.0/8)
  Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)

    Direction: inbound, SPI: 261, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
    Anti-replay service: Disabled

    Direction: outbound, SPI: 261, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
    Anti-replay service: Disabled
```

To verify that traffic is traveling over the bidirectional IPsec tunnel, issue the `show services ipsec-vpn statistics` command:

```
user@R2> show services ipsec-vpn ipsec statistics

PIC: sp-1/2/0, Service set: service-set-manual-BiEspshades

ESP Statistics:
  Encrypted bytes:             1616
  Decrypted bytes:             1560
  Encrypted packets:             20
  Decrypted packets:             19
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
```

Router 3

To verify that the IPsec security association is active, issue the `show services ipsec-vpn ipsec security-associations detail` command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

```
user@R3> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-manual-BiEspshades
  Rule: rule-manual-SA-BiEspshades, Term: term-manual-SA-BiEspshades,
  Tunnel index: 1
  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  Local identity: ipv4_subnet(any:0,[0..7]=10.0.0.0/8)
  Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)

    Direction: inbound, SPI: 261, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
    Anti-replay service: Disabled

    Direction: outbound, SPI: 261, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
    Anti-replay service: Disabled
```

To verify that traffic is traveling over the bidirectional IPsec tunnel, issue the `show services ipsec-vpn statistics` command:

```
user@R3> show services ipsec-vpn ipsec statistics
PIC: sp-1/2/0, Service set: service-set-manual-BiEspshades
ESP Statistics:
  Encrypted bytes:             1560
  Decrypted bytes:             1616
  Encrypted packets:             19
  Decrypted packets:             20
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
```

Example: ES PIC IKE Dynamic SA Configuration

IN THIS SECTION

Figure 6: ES PIC IKE Dynamic SA Topology Diagram

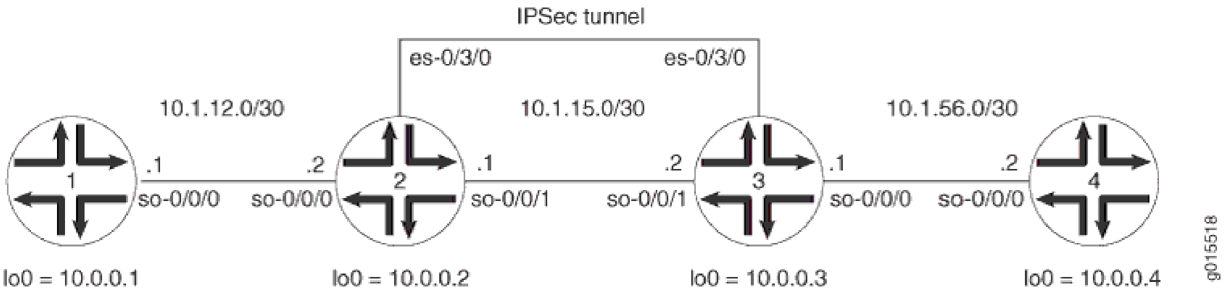


Figure 6 on page 131 shows the same IPSec topology as seen in the ES PIC manual SA example. However, this time the configuration requires Routers 2 and 3 to establish an IPSec tunnel using an IKE dynamic SA, enhanced authentication, and stronger encryption. Routers 1 and 4 continue to provide basic connectivity and are used to verify that the IPSec tunnel is operational.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R2 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.1;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

On Router 2, enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional IKE dynamic SA called **sa-dynamic** at the `[edit security ipsec security-association]` hierarchy level. For your IKE policy and proposal, use preshared keys for the authentication method, SHA-1 for the authentication algorithm, 3DES-CBC for encryption, group 2 for the Diffie-Hellman group, main mode, 3600 seconds for the lifetime, and a preshared key of **juniper** for the initial IKE negotiation. For your IPSec policy and proposal, use ESP for the protocol, HMAC-SHA1-96 for authentication, 3DES-CBC for encryption, 28800 seconds for the lifetime, and group 2 for the PFS group.

To direct traffic into the ES PIC and the IPSec tunnel, create two firewall filters. The **es-traffic** filter matches inbound traffic from Router 1 destined for Router 4, whereas the **es-return** filter matches the return path from Router 4 to Router 1. Apply the **es-traffic** filter to the **so-0/0/0** interface, and then apply both the **es-return** filter and the **sa-dynamic** SA to the **es-0/3/0** interface.

Router 2

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R1 so-0/0/0";
        unit 0 {
            family inet {
                filter {
                                           input es-traffic; # Apply a filter that sends traffic
to the IPSec tunnel here.
                }
                address 10.1.12.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R3 so-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.1/30;
            }
        }
    }
    es-0/3/0 {
        unit 0 {
                                 tunnel { # Specify the IPSec tunnel endpoints here.
                source 10.1.15.1;
                destination 10.1.15.2;
            }
            family inet {
                                     ipsec-sa sa-dynamic; # Apply the dynamic SA here.
                filter {
                                           input es-return; # Apply the filter that matches
return IPSec traffic here.
                }
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.2/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.2;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface so-0/0/1.0;
            interface lo0.0;
        }
    }
}
security {
    ipsec {
                           proposal es-ipsec-proposal { # Define your IPSec proposal
specifications here.
            protocol esp;
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 28800;
        }
                           policy es-ipsec-policy { # Define your IPSec policy specifications
here.
            perfect-forward-secrecy {
                keys group2;
            }
                               proposals es-ipsec-proposal; # Reference the IPSec proposal here.
        }
                           security-association sa-dynamic { # Define your dynamic SA here.
                               mode tunnel;
                                 dynamic {
                                     ipsec-policy es-ipsec-policy; # Reference the IPSec policy
here.
            }
        }
    }
    ike {
                           proposal es-ike-proposal { # Define your IKE proposal specifications
```

```
here.
            authentication-method pre-shared-keys;
            dh-group group2;
            authentication-algorithm sha1;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 3600;
        }
                           policy 10.1.15.2 { # Define your IKE policy specifications here.
            mode main;
                               proposals es-ike-proposal; # Reference the IKE proposal here.
                               pre-shared-key ascii-text "$ABC123";
            ## The unencrypted preshared key for this example is juniper.
        }
    }
}
firewall {
                     filter es-traffic { # Define a filter that sends traffic to the IPSec
tunnel here.
        term to-es {
            from {
                source-address {
                    10.1.12.0/24;
                }
                destination-address {
                    10.1.56.0/24;
                }
            }
            then {
                count ipsec-tunnel;
                ipsec-sa sa-dynamic;
            }
        }
        term other {
            then accept;
        }
    }
                     filter es-return { # Define a filter that matches return IPSec traffic here.
        term return {
            from {
                source-address {
                    10.1.56.0/24;
                }
                destination-address {
                    10.1.12.0/24;
                }
            }
            then accept;
        }
    }
}
```

On Router 3, enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional IKE dynamic SA called **sa-dynamic** at the `[edit security ipsec security-association]` hierarchy level. Use the same policies and proposals that you used on Router 2.

For your IKE policy and proposal, use preshared keys for the authentication method, SHA-1 for the authentication algorithm, 3DES-CBC for encryption, group 2 for the Diffie-Hellman group, main mode, 3600 seconds for the lifetime, and a preshared key of **juniper** for the initial IKE negotiation. For your IPSec policy and proposal, use ESP for the protocol, HMAC-SHA1-96 for authentication, 3DES-CBC for encryption, 28800 seconds for the lifetime, and group 2 for the PFS group.

To direct traffic into the ES PIC and the IPSec tunnel, create two firewall filters. The **es-traffic** filter matches inbound traffic from Router 4 destined for Router 1, whereas the **es-return** filter matches the return path from Router 1 to Router 4. Apply the **es-traffic** filter to the **so-0/0/0** interface; then apply both the **es-return** filter and the **sa-dynamic** SA to the **es-0/3/0** interface.

Router 3

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R4 so-0/0/0";
        unit 0 {
            family inet {
                filter {
                                           input es-traffic; # Apply a filter that sends traffic
to the IPSec tunnel here.
                }
                address 10.1.56.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R2 so-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.2/30;
            }
        }
    }
    es-0/3/0 {
        unit 0 {
                                 tunnel { # Specify the IPSec tunnel endpoints here.
                source 10.1.15.2;
                destination 10.1.15.1;
            }
            family inet {
                                     ipsec-sa sa-dynamic; # Apply the dynamic SA here.
                filter {
                                           input es-return; # Apply the filter that matches
return IPSec traffic here.
                }
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.3/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.3;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface so-0/0/1.0;
            interface lo0.0;
        }
    }
}
security {
    ipsec {
```

```
                           proposal es-ipsec-proposal { # Define your IPSec proposal
specifications here.
            protocol esp;
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 28800;
        }
                           policy es-ipsec-policy { # Define your IPSec policy specifications
here.
            perfect-forward-secrecy {
                keys group2;
            }
                               proposals es-ipsec-proposal; # Reference the IPSec proposal here.
        }
                           security-association sa-dynamic { # Define your dynamic SA here.
                               mode tunnel;
                                 dynamic {
                                     ipsec-policy es-ipsec-policy; # Reference the IPSec policy
here.
            }
        }
    }
    ike {
                           proposal es-ike-proposal { # Define your IKE proposal specifications
here.
            authentication-method pre-shared-keys;
            dh-group group2;
            authentication-algorithm sha1;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 3600;
        }
                           policy 10.1.15.1 { # Define your IKE policy specifications here.
            mode main;
                               proposals es-ike-proposal; # Reference the IKE proposal here.
                               pre-shared-key ascii-text "$ABC123";
            ## The unencrypted preshared key for this example is juniper.
        }
    }
}
firewall {
                     filter es-traffic { # Define a filter that sends traffic to the IPSec
tunnel here.
        term to-es {
            from {
                source-address {
                    10.1.56.0/24;
                }
                destination-address {
                    10.1.12.0/24;
                }
            }
            then {
                count ipsec-tunnel;
                ipsec-sa sa-dynamic;
            }
        }
        term other {
            then accept;
        }
    }
                     filter es-return { # Define a filter that matches return IPSec traffic here.
        term return {
            from {
                source-address {
                    10.1.12.0/24;
                }
                destination-address {
                    10.1.56.0/24;
                }
            }
            then accept;
        }
    }
}
```

On Router 4, provide basic OSPF connectivity to Router 3.

Router 4

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R3 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.56.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.4/32;
            }
        }
    }
}
routing-options {
        router-id 10.0.0.4;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

Verifying Your Work

To verify proper operation of an IKE-based dynamic SA on the ES PIC, use the following commands:

- **ping**
- `show ike security-associations (detail)`
- `show ipsec security-associations (detail)`
- **traceroute**

The following sections show the output of these commands used with the configuration example:

Router 1

On Router 1, issue a `ping` command to the **so-0/0/0** interface of Router 4 to send traffic across the IPsec tunnel.

```
user@R1> ping 10.1.56.2
PING 10.1.56.2 (10.1.56.2): 56 data bytes
64 bytes from 10.1.56.2: icmp_seq=0 ttl=253 time=0.917 ms
64 bytes from 10.1.56.2: icmp_seq=1 ttl=253 time=0.881 ms
64 bytes from 10.1.56.2: icmp_seq=2 ttl=253 time=0.897 ms
64 bytes from 10.1.56.2: icmp_seq=3 ttl=253 time=0.871 ms
64 bytes from 10.1.56.2: icmp_seq=4 ttl=253 time=0.890 ms
64 bytes from 10.1.56.2: icmp_seq=5 ttl=253 time=0.858 ms
64 bytes from 10.1.56.2: icmp_seq=6 ttl=253 time=0.904 ms
^C
--- 10.1.56.2 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.858/0.888/0.917/0.019 ms
```

You can also issue the `traceroute` command to verify that traffic to **10.1.56.2** travels over the IPsec tunnel between Router 2 and Router 3. Notice that the second hop does not reference **10.1.15.2**—the physical interface on Router 3. Instead, the loopback address of **10.0.0.3** on Router 3 appears as the second hop. This indicates that the IPSec tunnel is operating correctly.

```
user@R1> traceroute 10.1.56.2
traceroute to 10.1.56.2 (10.1.56.2), 30 hops max, 40 byte packets
 1  10.1.12.1 (10.1.12.1)  0.655 ms  0.549 ms  0.508 ms
 2  10.0.0.3 (10.0.0.3)  0.833 ms  0.786 ms  0.757 ms
```

3 10.1.56.2 (10.1.56.2) 0.808 ms 0.741 ms 0.716 ms

Router 2

Another way to verify that matched traffic is being diverted to the bidirectional IPsec tunnel is to view the firewall filter counter. After you issue the `ping` command from Router 1 (seven packets), the **es-traffic** firewall filter counter looks like this:

```
user@R2> show firewall filter es-traffic
Filter: es-traffic
Counters:
```

```
Name                                               Bytes              Packets
ipsec-tunnel                                         588                    7
```

After you issue the `ping` command from both Router 1 (seven packets) and Router 4 (five packets), the **es-traffic** firewall filter counter looks like this:

```
user@R2> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                               Bytes              Packets
ipsec-tunnel                                        1008                   12
```

To verify that the IKE SA negotiation between Routers 2 and 3 is successful, issue the `show ike security-associations detail` command. Notice that the SA contains the settings you specified, such as SHA-1 for the authentication algorithm and 3DES-CBC for the encryption algorithm.

```
user@R2> show ike security-associations detail
IKE peer 10.1.15.2
  Role: Initiator, State: Matured
  Initiator cookie: b5dbdfe2f9000000, Responder cookie: a24c868410000041
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.1.15.1:500, Remote: 10.1.15.2:500
  Lifetime: Expires in 401 seconds
  Algorithms:
   Authentication        : sha1
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
  Traffic statistics:
   Input  bytes  :                 1736
   Output bytes  :                 2652
   Input  packets:                    9
   Output packets:                   15
  Flags: Caller notification sent
  IPSec security associations: 3 created, 0 deleted
  Phase 2 negotiations in progress: 0
```

To verify that the IPsec security association is active, issue the `show ipsec security-associations detail` command. Notice that the SA contains the settings you specified, such as ESP for the protocol, HMAC-SHA1-96 for the authentication algorithm, and 3DES-CBC for the encryption algorithm.

```
user@R2> show ipsec security-associations detail
Security association: sa-dynamic, Interface family: Up
  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
    Direction: inbound, SPI: 2133029543, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26212 seconds
    Hard lifetime: Expires in 26347 seconds
    Anti-replay service: Disabled
    Direction: outbound, SPI: 1759450863, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26212 seconds
    Hard lifetime: Expires in 26347 seconds
    Anti-replay service: Disabled
```

Router 3

View the firewall filter counter to continue verifying that matched traffic is being diverted to the bidirectional IPsec tunnel. After you issue the `ping` command from Router 1 (seven packets), the **es-traffic** firewall filter counter looks like this:

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          588                    7
```

After you issue the `ping` command from both Router 1 (seven packets) and Router 4 (five packets), the **es-traffic** firewall filter counter looks like this:

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
```

```
Name                                                 Bytes              Packets
ipsec-tunnel                                          1008                   12
```

To verify the success of the IKE security association, issue the `show ike security-associations detail` command. Notice that the SA on Router 3 contains the same settings you specified on Router 2.

```
user@R3> show ike security-associations detail
IKE peer 10.1.15.1
  Role: Responder, State: Matured
  Initiator cookie: b5dbdfe2f9000000, Responder cookie: a24c868410000041
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.1.15.2:500, Remote: 10.1.15.1:500
  Lifetime: Expires in 564 seconds
  Algorithms:
   Authentication        : sha1
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
  Traffic statistics:
   Input  bytes  :                 2652
   Output bytes  :                 1856
   Input  packets:                   15
   Output packets:                   10
  Flags: Caller notification sent
  IPSec security associations: 3 created, 4 deleted
  Phase 2 negotiations in progress: 0
```

To verify that the IPsec security association is active, issue the `show ipsec security-associations detail` command. Notice that the SA on Router 3 contains the same settings you specified on Router 2.

```
user@R3> show ipsec security-associations detail
Security association: sa-dynamic, Interface family: Up
  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
    Direction: inbound, SPI: 1759450863, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26427 seconds
    Hard lifetime: Expires in 26517 seconds
    Anti-replay service: Disabled
    Direction: outbound, SPI: 2133029543, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26427 seconds
    Hard lifetime: Expires in 26517 seconds
    Anti-replay service: Disabled
```

Router 4

On Router 4, issue a `ping` command to the **so-0/0/0** interface of Router 1 to send traffic across the IPsec tunnel.

```
user@R4> ping 10.1.12.2
user@R4> ping 10.1.12.2
PING 10.1.12.2 (10.1.12.2): 56 data bytes
64 bytes from 10.1.12.2: icmp_seq=0 ttl=253 time=13.528 ms
64 bytes from 10.1.12.2: icmp_seq=1 ttl=253 time=0.873 ms
64 bytes from 10.1.12.2: icmp_seq=2 ttl=253 time=32.145 ms
64 bytes from 10.1.12.2: icmp_seq=3 ttl=253 time=0.921 ms
64 bytes from 10.1.12.2: icmp_seq=4 ttl=253 time=0.899 ms
^C
--- 10.1.12.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.873/9.673/32.145/12.255 ms
```

You can also issue the `traceroute` command to verify that traffic to **10.1.12.2** travels over the IPsec tunnel between Router 3 and Router 2. Notice that the second hop does not reference **10.1.15.1**—the physical interface on Router 2. Instead, the loopback address of **10.0.0.2** on Router 2 appears as the second hop. This indicates that the IPSec tunnel is operating correctly.

```
user@R4> traceroute 10.1.12.2
traceroute to 10.1.12.2 (10.1.12.2), 30 hops max, 40 byte packets
 1  10.1.56.1 (10.1.56.1)  0.681 ms  0.624 ms  0.547 ms
 2  10.0.0.2 (10.0.0.2)  0.800 ms  0.770 ms  0.737 ms
 3  10.1.12.2 (10.1.12.2)  0.793 ms  0.742 ms  0.716 ms
```

Example: AS PIC IKE Dynamic SA Configuration

Figure 7: AS PIC IKE Dynamic SA Topology Diagram

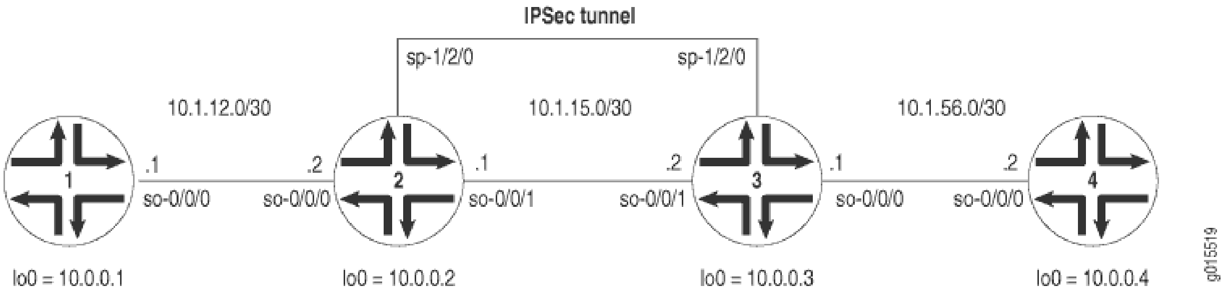


Figure 7 on page 146 shows the same IPSec topology as seen in the AS PIC manual SA example. However, this configuration requires Routers 2 and 3 to establish an IPSec tunnel using an IKE dynamic SA, enhanced authentication, and stronger encryption. Routers 1 and 4 continue to provide basic connectivity and are used to verify that the IPSec tunnel is operational.

NOTE: When you do not specify an IKE proposal, an IPSec proposal, and an IPSec policy on an AS PIC, the Junos OS defaults to the highest level of encryption and authentication. As a result, the default authentication protocol is ESP, the default authentication mode is HMAC-SHA1-96, and the default encryption mode is 3DES-CBC. For more information about default IKE and IPSec policies and proposals on the AS PIC, see "Configuring IKE Dynamic SAs" on page 77.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R2 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.1;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

On Router 2, enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional IKE dynamic SA in a rule called **rule-ike** at the `[edit ipsec-vpn rule]` hierarchy level. Reference this rule in a service set called **service-set-dynamic-BiEspsha3des** at the `[edit services service-set]` hierarchy level.

Using default values in the AS PIC, you do not need to specify an IPsec proposal, IPsec policy, or IKE proposal. However, you do need to configure a preshared key in an IKE policy with the `pre-shared-key` statement at the `[edit services ipsec-vpn ike policy` *`policy-name`*`]` hierarchy level. (For more information about default IKE and IPsec policies and proposals on the AS PIC, see "Configuring IKE Dynamic SAs" on page 77.)

To direct traffic into the AS PIC and the IPSec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPSec inside interface into the OSPF configuration.

Router 2

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R1 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R3 so-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.1/30;
            }
        }
    }
    sp-1/2/0 {
        services-options {
            syslog {
                host local {
                    services info;
                }
            }
        }
        unit 0 {
            family inet {
            }
                                     unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
                family inet;
                service-domain inside;
            }
                                     unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
                family inet;
                service-domain outside;
            }
        }
        lo0 {
            unit 0 {
                family inet {
                    address 10.0.0.2/32;
                }
            }
        }
    }
    routing-options {
        router-id 10.0.0.2;
    }
    protocols {
        ospf {
            area 0.0.0.0 {
                interface so-0/0/0.0;
                interface lo0.0;
                                     interface sp-1/2/0.1; # This sends OSPF traffic over the
IPSec tunnel.
            }
        }
    }
    services {
                           service-set service-set-dynamic-BiEspsha3des { # Define your service
set here.
                                 next-hop-service { # Required for dynamic routing protocols
such as OSPF.
                inside-service-interface sp-1/2/0.1;
                outside-service-interface sp-1/2/0.2;
            }
            ipsec-vpn-options {
                                     local-gateway 10.1.15.1; # Specify the local IP address of
the IPSec tunnel.
            }
                               ipsec-vpn-rules rule-ike; # Reference your IPSec VPN rule here.
        }
        ipsec-vpn {
                                 rule rule-ike { # Define your IPSec VPN rule here.
                term term-ike {
                    then {
                                                 remote-gateway 10.1.15.2; # The remote IP
address of the IPSec tunnel.
                                                   dynamic { # This creates a dynamic SA.
                                                       ike-policy ike-policy-preshared; #
Reference your IKE policy here.
                        }
                    }
                }
                                     match-direction input; # Specify in which direction the
rule should match.
            }
            ike {
```

```
                                       policy ike-policy-preshared { # Define your IKE policy
specifications here.
                                           pre-shared-key ascii-text
"$ABC123";
                    ## The unencrypted preshared key for this example is juniper.
                }
            }
        }
    }
}
```

On Router 3, enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional IKE dynamic SA in a rule called **rule-ike** at the `[edit ipsec-vpn rule]` hierarchy level. Reference this rule in a service set called **service-set-dynamic-BiEspsha3des** at the `[edit services service-set]` hierarchy level.

Again, use the same default policies and proposals that you used on Router 2. However, remember to configure a preshared key in an IKE policy with the **pre-shared-key** statement at the **[edit services ipsec-vpn ike policy *policy-name*]** hierarchy level. The key must match the one you specified on Router 2. (For more information about default IKE and IPSec policies and proposals on the AS PIC, see "Configuring IKE Dynamic SAs" on page 77.)

To direct traffic into the AS PIC and the IPSec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPSec inside interface into the OSPF configuration.

Router 3

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R4 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.56.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R2 so-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.2/30;
            }
        }
    }
    sp-1/2/0 {
        services-options {
            syslog {
                host local {
                    services info;
                }
            }
        }
        unit 0 {
            family inet {
            }
                                 unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
                family inet;
                service-domain inside;
            }
                                 unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
                family inet;
                service-domain outside;
            }
        }
        lo0 {
            unit 0 {
                family inet {
                    address 10.0.0.3/32;
                }
            }
        }
    }
    routing-options {
        router-id 10.0.0.3;
    }
    protocols {
        ospf {
            area 0.0.0.0 {
                interface so-0/0/0.0;
                interface lo0.0;
                                     interface sp-1/2/0.1; # This sends OSPF traffic over the
IPSec tunnel.
            }
        }
    }
    services {
                           service-set service-set-dynamic-BiEspsha3des { # Define your service
set here.
                                 next-hop-service { # Required for dynamic routing protocols
such as OSPF.
                inside-service-interface sp-1/2/0.1;
                outside-service-interface sp-1/2/0.2;
            }
            ipsec-vpn-options {
                                     local-gateway 10.1.15.2; # Specify the local IP address of
the IPSec tunnel.
            }
                               ipsec-vpn-rules rule-ike; # Reference your IPSec VPN rule here.
        }
        ipsec-vpn {
                                 rule rule-ike { # Define your IPSec VPN rule here.
                term term-ike {
                    then {
                                                 remote-gateway 10.1.15.1; # The remote IP
address of the IPSec tunnel.
                                                   dynamic { # This creates a dynamic SA.
                                                       ike-policy ike-policy-preshared; #
Reference your IKE policy here.
                        }
                    }
                }
                                     match-direction input; # Specify in which direction the
rule should match.
            }
            ike {
                                       policy ike-policy-preshared { # Define your IKE policy
specifications here.
                                           pre-shared-key ascii-text
"$ABC123";
                    ## The unencrypted preshared key for this example is juniper.
                }
            }
        }
    }
}
```

On Router 4, provide basic OSPF connectivity to Router 3.

Router 4

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R3 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.56.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.4/32;
            }
        }
    }
}
routing-options {
        router-id 10.0.0.4;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

Verifying Your Work

To verify proper operation of an IKE-based dynamic SA on the AS PIC, use the following commands:

- **ping**
- `show services ipsec-vpn ike security-associations (detail)`
- `show services ipsec-vpn ipsec security-associations (detail)`

- `show services ipsec-vpn ipsec statistics`
- **traceroute**

The following sections show the output of these commands used with the configuration example:

Router 1

On Router 1, issue a **ping** command to the **so-0/0/0** interface on Router 4 to send traffic across the IPSec tunnel.

```
user@R1> ping 10.1.56.2
PING 10.1.56.2 (10.1.56.2): 56 data bytes
64 bytes from 10.1.56.2: icmp_seq=0 ttl=254 time=1.351 ms
64 bytes from 10.1.56.2: icmp_seq=1 ttl=254 time=1.187 ms
64 bytes from 10.1.56.2: icmp_seq=2 ttl=254 time=1.172 ms
64 bytes from 10.1.56.2: icmp_seq=3 ttl=254 time=1.154 ms
64 bytes from 10.1.56.2: icmp_seq=4 ttl=254 time=1.156 ms
^C
--- 10.1.56.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.154/1.204/1.351/0.074 ms
```

Router 2

To verify that the IKE SA negotiation is successful, issue the `show services ipsec-vpn ike security-associations` command.

```
user@R2> show services ipsec-vpn ike security-associations
Remote Address  State         Initiator cookie  Responder cookie  Exchange type
10.1.15.2       Matured       03075bd3a0000003  4bff26a5c7000003  Main
```

To verify that the IPsec security association is active, issue the `show services ipsec-vpn ipsec security-associations detail` command. Notice that the SA contains the default settings inherent in the AS PIC, such as ESP for the protocol and HMAC-SHA1-96 for the authentication algorithm.

```
user@R2> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-dynamic-BiEspsha3des
  Rule: rule-ike, Term: term-ike, Tunnel index: 1
  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
    Direction: inbound, SPI: 2666326758, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26863 seconds
    Hard lifetime: Expires in 26998 seconds
    Anti-replay service: Enabled, Replay window size: 64
    Direction: outbound, SPI: 684772754, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26863 seconds
    Hard lifetime: Expires in 26998 seconds
    Anti-replay service: Enabled, Replay window size: 64
```

To verify that traffic is traveling over the bidirectional IPSec tunnel, issue the **show services ipsec-vpn statistics** command:

```
user@R2> show services ipsec-vpn ipsec statistics
PIC: sp-1/2/0, Service set: service-set-dynamic-BiEspsha3des
ESP Statistics:
  Encrypted bytes:             2248
  Decrypted bytes:             2120
  Encrypted packets:             27
  Decrypted packets:             25
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

Router 3

To verify that the IKE SA negotiation is successful, issue the `show services ipsec-vpn ike security-associations` command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

```
user@R3> show services ipsec-vpn ike security-associations
Remote Address  State         Initiator cookie  Responder cookie  Exchange type
10.1.15.1       Matured       03075bd3a0000003  4bff26a5c7000003  Main
```

To verify that the IPsec SA is active, issue the `show services ipsec-vpn ipsec security-associations detail` command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

```
user@R3> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-dynamic-BiEspsha3des
  Rule: rule-ike, Term: term-ike, Tunnel index: 1
  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
    Direction: inbound, SPI: 684772754, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26598 seconds
    Hard lifetime: Expires in 26688 seconds
    Anti-replay service: Enabled, Replay window size: 64
    Direction: outbound, SPI: 2666326758, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26598 seconds
    Hard lifetime: Expires in 26688 seconds
    Anti-replay service: Enabled, Replay window size: 64
```

To verify that traffic is traveling over the bidirectional IPsec tunnel, issue the `show services ipsec-vpn statistics` command:

```
user@R3> show services ipsec-vpn ipsec statistics
PIC: sp-1/2/0, Service set: service-set-dynamic-BiEspsha3des
ESP Statistics:
  Encrypted bytes:             2120
  Decrypted bytes:           2248
  Encrypted packets:           25
  Decrypted packets:           27
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

Router 4

On Router 4, issue a `ping` command to the **so-0/0/0** interface on Router 1 to send traffic across the IPsec tunnel.

```
user@R4> ping 10.1.12.2
PING 10.1.12.2 (10.1.12.2): 56 data bytes
64 bytes from 10.1.12.2: icmp_seq=0 ttl=254 time=1.350 ms
64 bytes from 10.1.12.2: icmp_seq=1 ttl=254 time=1.161 ms
64 bytes from 10.1.12.2: icmp_seq=2 ttl=254 time=1.124 ms
64 bytes from 10.1.12.2: icmp_seq=3 ttl=254 time=1.142 ms
64 bytes from 10.1.12.2: icmp_seq=4 ttl=254 time=1.139 ms
64 bytes from 10.1.12.2: icmp_seq=5 ttl=254 time=1.116 ms
^C
--- 10.1.12.2 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.116/1.172/1.350/0.081 ms
```

The final way you can confirm that traffic travels over the IPsec tunnel is by issuing the `traceroute` command to the **so-0/0/0** interface on Router 1. Notice that the physical interface between Routers 2 and 3 is not referenced in the path; traffic enters the IPSec tunnel through the adaptive services IPSec inside interface on Router 3, passes through the loopback interface on Router 2, and ends at the **so-0/0/0** interface on Router 1.

```
user@R4> traceroute 10.1.12.2
traceroute to 10.1.12.2 (10.1.12.2), 30 hops max, 40 byte packets
 1  10.1.15.2 (10.1.15.2)  0.987 ms  0.630 ms  0.563 ms
```

```
  2  10.0.0.2 (10.0.0.2)  1.194 ms  1.058 ms  1.033 ms
  3  10.1.12.2 (10.1.12.2)  1.073 ms  0.949 ms  0.932 ms
```

Example: IKE Dynamic SA Between an AS PIC and an ES PIC Configuration

IN THIS SECTION

Figure 8: AS PIC to ES PIC IKE Dynamic SA Topology Diagram

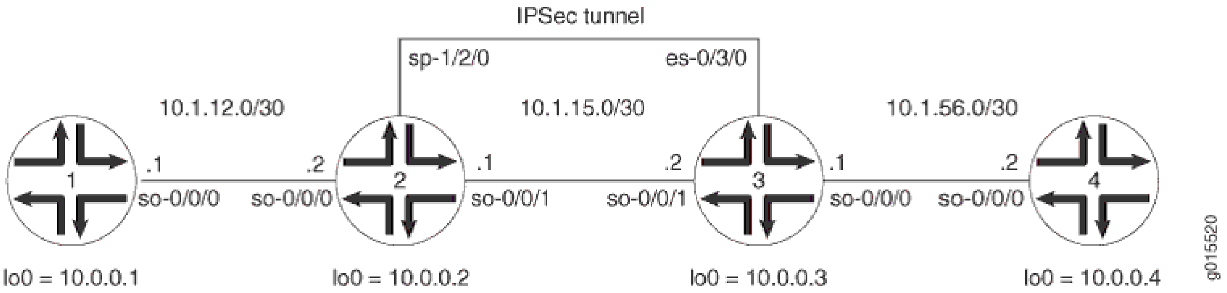


Figure 8 on page 158 shows a hybrid configuration that allows you to create an IPSec tunnel between the AS PIC and the ES PIC. Router 2 contains an AS PIC at **sp-1/2/0** and Router 3 has an ES PIC at **es-0/3/0**. To establish an IPSec tunnel using an IKE dynamic SA, the key is to learn the default IKE SA and IPSec SA settings built into the AS PIC and configure them explicitly on the ES PIC. Routers 1 and 4 again provide basic connectivity and are used to verify that the IPSec tunnel is operational.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R2 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.1;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

On Router 2, enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional IKE dynamic SA in a rule called **rule-ike** at the **[edit ipsec-vpn rule]** hierarchy level. Reference this rule in a service set called **service-set-dynamic-BiEspsha3des** at the **[edit services service-set]** hierarchy level.

Using default values in the AS PIC, you do not need to specify an IPSec proposal, IPSec policy, or IKE proposal. However, you do need to configure a preshared key in an IKE policy with the **pre-shared-key** statement at the **[edit services ipsec-vpn ike policy *policy-name*]** hierarchy level. (For more information about default IKE and IPSec policies and proposals on the AS PIC, see "Configuring IKE Dynamic SAs" on page 77.)

To direct traffic into the AS PIC and the IPSec tunnel, include match conditions in the **rule-ike** IPSec VPN rule to match inbound traffic from Router 1 that is destined for Router 4. Because the rule is already referenced by the service set, apply the service set to the **so-0/0/1** interface. To count the amount of traffic that enters the IPsec tunnel, configure a firewall filter called **ipsec-tunnel** and apply it to the **sp-1/2/0** interface.

Router 2

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R1 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R3 so-0/0/1";
        unit 0 {
            family inet {
                                       service { # Apply the service set here.
                                             input {
                        service-set service-set-dynamic-BiEspsha3des;
                    }
                                             output {
                        service-set service-set-dynamic-BiEspsha3des;
                    }
                }
                address 10.1.15.1/30;
            }
        }
    }
    sp-1/2/0 {
        services-options {
            syslog {
                host local {
                    services info;
                }
            }
        }
        unit 0 {
            family inet {
                filter {
                                           input ipsec-tunnel; # Apply the firewall filter with
the counter here.
                }
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.2/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.2;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface so-0/0/1.0;
            interface lo0.0;
        }
    }
}
firewall {
                     filter ipsec-tunnel { # Configure a firewall filter to count IPsec traffic
here.
        term 1 {
            then {
                count ipsec-tunnel;
                accept;
            }
        }
    }
}
services {
                     service-set service-set-dynamic-BiEspsha3des { # Define your service set
here.
        interface-service {
                               service-interface sp-1/2/0; # Specify an interface to process
IPsec.
        }
        ipsec-vpn-options {
                               local-gateway 10.1.15.1; # Specify the local IP address of the
```

```
IPsec tunnel.
        }
                         ipsec-vpn-rules rule-ike; # Reference your IPsec VPN rule here.
    }
    ipsec-vpn {
                           rule rule-ike { # Define your IPsec VPN rule here.
            term term-ike {
                from {
                    source-address {
                        10.1.12.0/24;
                    }
                    destination-address {
                        10.1.56.0/24;
                    }
                }
                then {
                                           remote-gateway 10.1.15.2; # The remote IP address of
the IPsec tunnel.
                                             dynamic { # This creates a dynamic SA.
                                                 ike-policy ike-policy-preshared; # Reference
your IKE proposal here.
                    }
                }
            }
                               match-direction output; # Specify in which direction the rule
should match.
        }
        ike {
                                 policy ike-policy-preshared { # Define your IKE policy
specifications here.
                                     pre-shared-key ascii-text "$ABC123";
                ## The unencrypted preshared key for this example is juniper.
            }
        }
    }
}
```

Router 2

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R1 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R3 so-0/0/1";
        unit 0 {
            family inet {
                                       service { # Apply the service set here.
                                             input {
                        service-set service-set-dynamic-BiEspsha3des;
                    }
                                             output {
                        service-set service-set-dynamic-BiEspsha3des;
                    }
                }
                address 10.1.15.1/30;
            }
        }
    }
    sp-1/2/0 {
        services-options {
            syslog {
                host local {
                    services info;
                }
            }
        }
        unit 0 {
            family inet {
                filter {
                                           input ipsec-tunnel; # Apply the firewall filter with
the counter here.
                }
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.2/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.2;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface so-0/0/1.0;
            interface lo0.0;
        }
    }
}
firewall {
                     filter ipsec-tunnel { # Configure a firewall filter to count IPSec traffic
here.
        term 1 {
            then {
                count ipsec-tunnel;
                accept;
            }
        }
    }
}
services {
                     service-set service-set-dynamic-BiEspsha3des { # Define your service set
here.
        interface-service {
                               service-interface sp-1/2/0; # Specify an interface to process
IPSec.
        }
        ipsec-vpn-options {
                               local-gateway 10.1.15.1; # Specify the local IP address of the
IPSec tunnel.
        }
                         ipsec-vpn-rules rule-ike; # Reference your IPSec VPN rule here.
    }
    ipsec-vpn {
                           rule rule-ike { # Define your IPSec VPN rule here.
            term term-ike {
                from {
                    source-address {
                        10.1.12.0/24;
                    }
                    destination-address {
                        10.1.56.0/24;
                    }
                }
                then {
                                           remote-gateway 10.1.15.2; # The remote IP address of
the IPSec tunnel.
                                             dynamic { # This creates a dynamic SA.
                                                 ike-policy ike-policy-preshared; # Reference
your IKE proposal here.
                    }
                }
            }
                               match-direction output; # Specify in which direction the rule
should match.
        }
        ike {
                                 policy ike-policy-preshared { # Define your IKE policy
specifications here.
                                     pre-shared-key ascii-text "$ABC123";
                ## The unencrypted preshared key for this example is juniper.
            }
        }
    }
}
```

On Router 3, enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional IKE dynamic SA called **sa-dynamic** at the **[edit security ipsec security-association]** hierarchy level. To allow the ES PIC to communicate with the IKE dynamic SA established on Router 2, you must explicitly configure the same policies and proposals on the ES PIC that are available by default on the AS PIC. (For more information about default IKE and IPSec policies and proposals on the AS PIC, see Configuring IKE Dynamic SAs.)

For your IKE policy and proposal, use preshared keys for the authentication method, SHA-1 for the authentication algorithm, 3DES-CBC for encryption, group 2 for the Diffie-Hellman group, main mode, 3600 seconds for the lifetime, and a preshared key of **juniper** for the initial IKE negotiation. For your IPSec policy and proposal, use ESP for the protocol, HMAC-SHA1-96 for authentication, 3DES-CBC for encryption, 28800 seconds for the lifetime, and group 2 for the PFS group.

To direct traffic into the ES PIC and the IPSec tunnel, create two firewall filters. The **es-traffic** filter matches inbound traffic from Router 4 destined for Router 1, whereas the **es-return** filter matches the return path from Router 1 to Router 4. Apply the **es-traffic** filter to the **so-0/0/0** interface; then apply both the **es-return** filter and the **sa-dynamic SA** to the **es-0/3/0** interface.

Router 3

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R4 so-0/0/0";
        unit 0 {
            family inet {
                filter {
                                           input es-traffic; # Apply a filter that sends traffic
to the IPSec tunnel here.
                }
                address 10.1.56.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R2 so-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.2/30;
            }
        }
    }
    es-0/3/0 {
        unit 0 {
                                 tunnel { # Specify the IPSec tunnel endpoints here.
                source 10.1.15.2;
                destination 10.1.15.1;
            }
            family inet {
                                     ipsec-sa sa-dynamic; # Apply the dynamic SA here.
                filter {
                                           input es-return; # Apply the filter that matches
return IPSec traffic here.
                }
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.3/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.3;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface so-0/0/1.0;
            interface lo0.0;
        }
    }
}
security {
    ipsec {
                           proposal es-ipsec-proposal { # Define your IPSec proposal
specifications here.
            protocol esp;
            authentication-algorithm hmac-sha1-96;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 28800;
        }
                           policy es-ipsec-policy { # Define your IPSec policy specifications
here.
            perfect-forward-secrecy {
                keys group2;
            }
                               proposals es-ipsec-proposal; # Reference the IPSec proposal here.
        }
                           security-association sa-dynamic { # Define your dynamic SA here.
                               mode tunnel;


            dynamic {
```

```
            ipsec-policy es-ipsec-policy; # Reference the IPSec policy here.
            }
        }
    }
    ike {
                           proposal es-ike-proposal { # Define your IKE proposal specifications
here.
            authentication-method pre-shared-keys;
            dh-group group2;
            authentication-algorithm sha1;
            encryption-algorithm 3des-cbc;
            lifetime-seconds 3600;
        }
                           policy 10.1.15.1 { # Define your IKE policy specifications here.
            mode main;
                               proposals es-ike-proposal; # Reference the IKE proposal here.
                               pre-shared-key ascii-text "$ABC123";
            ## The unencrypted preshared key for this example is juniper.
        }
    }
}
firewall {
                     filter es-traffic { # Define a filter that sends traffic to the IPSec
tunnel here.
        term to-es {
            from {
                source-address {
                    10.1.56.0/24;
                }
                destination-address {
                    10.1.12.0/24;
                }
            }
            then {
                count ipsec-tunnel;
                ipsec-sa sa-dynamic;
            }
        }
        term other {
            then accept;
        }
    }
                     filter es-return { # Define a filter that matches return IPSec traffic here.
        term return {
            from {
                source-address {
                    10.1.12.0/24;
                }
                destination-address {
                    10.1.56.0/24;
                }
            }
            then accept;
        }
    }
}
```

Router 4

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R3 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.56.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.4/32;
            }
        }
    }
}
routing-options {
        router-id 10.0.0.4;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

On Router 3, enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional IKE dynamic SA called **sa-dynamic** at the **[edit security ipsec security-association]** hierarchy level. To allow the ES PIC to communicate with the IKE dynamic SA established on Router 2, you must explicitly configure the same policies and proposals on the ES PIC that are available by default on the AS PIC. (For more information about default IKE and IPSec policies and proposals on the AS PIC, see Configuring IKE Dynamic SAs.)

For your IKE policy and proposal, use preshared keys for the authentication method, SHA-1 for the authentication algorithm, 3DES-CBC for encryption, group 2 for the Diffie-Hellman group, main mode, 3600 seconds for the lifetime, and a preshared key of **juniper** for the initial IKE negotiation. For your IPSec policy and proposal, use ESP for the protocol, HMAC-SHA1-96 for authentication, 3DES-CBC for encryption, 28800 seconds for the lifetime, and group 2 for the PFS group.

To direct traffic into the ES PIC and the IPSec tunnel, create two firewall filters. The **es-traffic** filter matches inbound traffic from Router 4 destined for Router 1, whereas the **es-return** filter matches the return path from Router 1 to Router 4. Apply the **es-traffic** filter to the **so-0/0/0** interface; then apply both the **es-return** filter and the **sa-dynamic SA** to the **es-0/3/0** interface.

Router 3

```
user@R1> ping 10.1.56.2
PING 10.1.56.2 (10.1.56.2): 56 data bytes
64 bytes from 10.1.56.2: icmp_seq=0 ttl=253 time=1.172 ms
64 bytes from 10.1.56.2: icmp_seq=1 ttl=253 time=1.020 ms
64 bytes from 10.1.56.2: icmp_seq=2 ttl=253 time=0.998 ms
64 bytes from 10.1.56.2: icmp_seq=3 ttl=253 time=1.037 ms
^C
--- 10.1.56.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.998/1.057/1.172/0.068 ms
```

Router 4

```
user@R1> traceroute 10.1.56.2
traceroute to 10.1.56.2 (10.1.56.2), 30 hops max, 40 byte packets
```

```
 1  * * *
 2  10.1.56.2 (10.1.56.2)  1.045 ms  0.915 ms  0.850 ms
```

Verifying Your Work

To verify proper operation of an IKE-based dynamic SA on the AS PIC, use the following commands:

- **ping**
- `show services ipsec-vpn ike security-associations (detail)`
- `show services ipsec-vpn ipsec security-associations (detail)`
- **traceroute**

To verify proper operation of an IKE-based dynamic SA on the ES PIC, use the following commands:

- **ping**
- `show ike security-associations (detail)`
- `show ipsec security-associations (detail)`
- **traceroute**

The following sections show the output of these commands used with the configuration example:

Router 1

On Router 1, issue a **ping** command to the **so-0/0/0** interface of Router 4 to send traffic across the IPSec tunnel.

```
user@R2> show firewall filter ipsec-tunnel
Filter: ipsec-tunnel
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                            0                    0
```

You can also issue the `traceroute` command to verify that traffic to **10.1.56.2** travels over the IPsec tunnel between Router 2 and Router 3. Notice that the traced path does not reference **10.1.15.2**—the physical interface on Router 3. Instead, traffic arriving at Router 2 is immediately filtered into the IPSec tunnel and the path is listed as unknown with the *** notation. This indicates that the IPSec tunnel is operating correctly.

```
user@R2> show firewall filter ipsec-tunnel
Filter: ipsec-tunnel
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          336                    4
```

Router 2

One way to verify that matched traffic is being diverted to the bidirectional IPSec tunnel is to view the firewall filter counter. Before any traffic flows, the **ipsec-tunnel** firewall filter counter looks like this:

```
user@R2> show firewall filter ipsec-tunnel
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          840                   10
```

After you issue the **ping** command from Router 1 (four packets) to **10.1.56.2**, the **ipsec-tunnel** firewall filter counter looks like this:

```
user@R2> show services ipsec-vpn ike security-associations detail
IKE peer 10.1.15.2
  Role: Responder, State: Matured
  Initiator cookie: c8e1e4c0da000040, Responder cookie: 4fbaa5184e000044
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.1.15.1:500, Remote: 10.1.15.2:500
  Lifetime: Expires in 3535 seconds
  Algorithms:
   Authentication        : sha1
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
  Traffic statistics:
   Input  bytes  :                  840
   Output bytes  :                  756
   Input  packets:                    5
   Output packets:                    4
  Flags: Caller notification sent
  IPSec security associations: 1 created, 0 deleted
  Phase 2 negotiations in progress: 0
```

After you issue the `ping` command from both Router 1 to **10.1.56.2** (four packets) and from Router 4 to **10.1.12.2** (six packets), the **ipsec-tunnel** firewall filter counter looks like this:

```
user@R2> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-dynamic-BiEspsha3des
  Rule: rule-ike, Term: term-ike, Tunnel index: 1
  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
    Direction: inbound, SPI: 407204513, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 24546 seconds
    Hard lifetime: Expires in 24636 seconds
    Anti-replay service: Disabled
    Direction: outbound, SPI: 2957235894, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 24546 seconds
    Hard lifetime: Expires in 24636 seconds
    Anti-replay service: Disabled
```

To verify that the IKE SA negotiation is successful, issue the `show services ipsec-vpn ike security-associations detail` command. Notice that the SA contains the default IKE settings inherent in the AS PIC, such as SHA-1 for the authentication algorithm and 3DES-CBC for the encryption algorithm.

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                    Bytes              Packets
ipsec-tunnel                                              336                    4
```

To verify that the IPsec security association is active, issue the `show services ipsec-vpn ipsec security-associations detail` command. Notice that the SA contains the default settings inherent in the AS PIC, such as ESP for the protocol and HMAC-SHA1-96 for the authentication algorithm.

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          840                   10
```

Router 3

View the firewall filter counter to continue verifying that matched traffic is being diverted to the bidirectional IPsec tunnel. After you issue the `ping` command from Router 1 (four packets), the **es-traffic** firewall filter counter looks like this:

```
user@R3> show ike security-associations detail
IKE peer 10.1.15.1
  Role: Initiator, State: Matured
  Initiator cookie: c8e1e4c0da000040, Responder cookie: 4fbaa5184e000044
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.1.15.2:500, Remote: 10.1.15.1:500
  Lifetime: Expires in 3441 seconds
  Algorithms:
   Authentication        : sha1
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
  Traffic statistics:
   Input  bytes  :                  756
   Output bytes  :                  840
   Input  packets:                    4
   Output packets:                    5
  Flags: Caller notification sent
  IPSec security associations: 1 created, 0 deleted
  Phase 2 negotiations in progress: 0
```

After you issue the `ping` command from both Router 1 (four packets) and Router 4 (six packets), the **es-traffic** firewall filter counter looks like this:

```
user@R3> show ipsec security-associations detail
Security association: sa-dynamic, Interface family: Up
  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
    Direction: inbound, SPI: 2957235894, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 28555 seconds
    Hard lifetime: Expires in 28690 seconds
    Anti-replay service: Disabled
    Direction: outbound, SPI: 407204513, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 28555 seconds
    Hard lifetime: Expires in 28690 seconds
    Anti-replay service: Disabled
```

To verify the success of the IKE security association on the ES PIC, issue the `show ike security-associations detail` command. Notice that the IKE SA on Router 3 contains the same settings you specified on Router 2.

```
user@R4> ping 10.1.12.2
PING 10.1.12.2 (10.1.12.2): 56 data bytes
64 bytes from 10.1.12.2: icmp_seq=0 ttl=254 time=1.350 ms
64 bytes from 10.1.12.2: icmp_seq=1 ttl=254 time=1.161 ms
64 bytes from 10.1.12.2: icmp_seq=2 ttl=254 time=1.124 ms
64 bytes from 10.1.12.2: icmp_seq=3 ttl=254 time=1.142 ms
64 bytes from 10.1.12.2: icmp_seq=4 ttl=254 time=1.139 ms
64 bytes from 10.1.12.2: icmp_seq=5 ttl=254 time=1.116 ms
^C
--- 10.1.12.2 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.116/1.172/1.350/0.081 ms
```

To verify that the IPsec security association is active, issue the `show ipsec security-associations detail` command. Notice that the IPsec SA on Router 3 contains the same settings you specified on Router 2.

```
user@R4> traceroute 10.1.12.2
traceroute to 10.1.12.2 (10.1.12.2), 30 hops max, 40 byte packets
 1  10.1.56.1 (10.1.56.1)  3.561 ms  0.613 ms  0.558 ms
 2  * * *
 3  10.1.12.2 (10.1.12.2)  1.073 ms  0.862 ms  0.818 ms
```

Router 4

On Router 4, issue a `ping` command to the **so-0/0/0** interface on Router 1 to send traffic across the IPsec tunnel.

Again, the `traceroute` command verifies that traffic to **10.1.12.2** travels over the IPsec tunnel between Router 3 and Router 2. Notice that the second hop does not reference **10.1.15.1**—the physical interface on Router 2. Instead, the second hop is listed as unknown with the *** notation. This indicates that the IPsec tunnel is operating correctly.

Example: AS PIC IKE Dynamic SA with Digital Certificates Configuration

IN THIS SECTION

Figure 9: AS PIC IKE Dynamic SA Topology Diagram

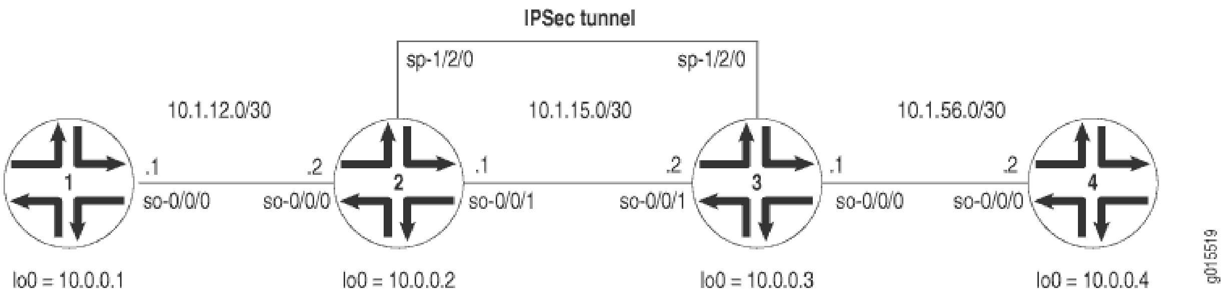


Figure 9 on page 176 shows the same IPSec topology as the AS PIC dynamic SA example on "Example: AS PIC IKE Dynamic SA Configuration" on page 146. However, this configuration requires Routers 2 and 3 to establish an IKE-based IPSec tunnel by using digital certificates in place of preshared keys. Routers 1 and 4 continue to provide basic connectivity and are used to verify that the IPSec tunnel is operational.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R2 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.1;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

On Router 2, you must request a CA certificate, create a local certificate, and load these digital certificates into the router before you can reference them in your IPSec configuration. To begin, configure an IPSec profile by specifying the trusted CA and URL of the CA server that handles CA certificate processing:

```
[edit]
security {
    pki {
        ca-profile entrust {
            ca-identity entrust;
            enrollment {
                url http://ca-1.example.com/cgi-bin/pkiclient.exe;
            }
        }
    }
}
```

Certificate revocation list (CRL) verification is enabled by default. You can optionally specify the Lightweight Access Directory (LDAP) server where the CA stores the CRL. The certificate typically includes a certificate distribution point (CDP), which contains information about how to retrieve the CRL for the certificate. The router uses this information to download the CRL automatically. In this example, the LDAP URL is specified, which overrides the location provided in the certificate:

```
[edit]
security pki ca-profile entrust {
    revocation-check {
        crl {
            url ldap://10.157.90.185/o=juniper,c=uscertificateRevocationListbase;
        }
    }
}
```

After you configure the CA profile, you can request a CA certificate from the trusted CA. In this example, the certificate is enrolled online and installed into the router automatically.

```
user@R2> request security pki ca-certificate enroll ca-profile entrust
Received following certificates:
  Certificate: C=us, O=juniper
    Fingerprint: 00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10
  Certificate: C=us, O=juniper, CN=First Officer
    Fingerprint: bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17
  Certificate: C=us, O=juniper, CN=First Officer
```

```
    Fingerprint: 46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f
Do you want to load the above CA certificate ? [yes,no] (no) yes
```

NOTE: If you obtain the CA certificate directly from the CA (for example, as an e-mail attachment or website download), you can install it with the `request security pki ca-certificate load` command.

Next, you must generate a private/public key pair before you can create a local certificate.

```
user@R2> request security pki generate-key-pair certificate-id local-entrust2
Generated key pair local-entrust2, key size 1024 bits
```

When the key pair is available, generate a local certificate request and send it to the CA for processing.

```
user@R2> request security pki generate-certificate-request
certificate-id local-entrust2 domain-name router2.example.com
filename entrust-req2 subject cn=router2.example.com
Generated certificate request
-----BEGIN CERTIFICATE REQUEST-----
MIIBoTCCAQoCAQAwGjEYMBYGA1UEAxMPdHAxLmp1bmlwZXIubmV0MIGfMA0GCSqG
SIb3DQEBAQUAA4GNADCBiQKBgQCiUFklQws1Ud+AqN5DDxRs2kVyKEhh9qoVFnz+
Hz4c9vsy3B8ElwTJlkmIt2cB3yifB6zePd+6WYpf57Crwre7YqPkiXM31F6z3YjX
H+1BPNbCxNWYvyrnSyVYDbFj8o0Xyqog8ACDfVL2JBWrPNBYy7imq/K9soDBbAs6
5hZqqwIDAQABoEcwRQYJKoZIhvcNAQkOMTgwNjAOBgNVHQ8BAf8EBAMCB4AwJAYD
VR0RAQH/BBowGIIWdHAxLmVuZ2xhYi5qdW5pcGVyLm5ldDANBgkqhkiG9w0BAQQF
AAOBgQBc2rq1v5SOQXH7LCb/FdqAL8ZM6GoaN5d6cGwq4bB6a7UQFgtoH406gQ3G
3iH0Zfz4xMIBpJYuGd1dkqgvcDoH3AgTsLkfn7Wi3x5H2qeQVs9bvL4P5nvEZLND
EIMUHwteolZCiZ70fO9Fer9cXWHSQs1UtXtgPqQJy2xIeImLgw==
-----END CERTIFICATE REQUEST-----
Fingerprint:
0d:90:b8:d2:56:74:fc:84:59:62:b9:78:71:9c:e4:9c:54:ba:16:97 (sha1)
1b:08:d4:f7:90:f1:c4:39:08:c9:de:76:00:86:62:b8 (md5)
```

NOTE: You can request the creation and installation of a local certificate online with the `request security pki local-certificate enroll` command. For more information, see

Generating and Enrolling a Local Digital Certificate or the *Junos System Basics and Services Command Reference*.

The trusted CA digitally signs the local certificate and returns it to you. Copy the certificate file into the router and load the certificate.

```
user@R2> request security pki local-certificate load filename /tmp/router2-cert
certificate-id local-entrust2
Local certificate local-entrust2 loaded successfully
```

NOTE: The name of the file sent to you by the CA might not match the name of the certificate identifier. However, the **certificate-id** name must always match the name of the key pair you generated for the router.

After the local and CA certificates have been loaded, you can reference them in your IPSec configuration.

Using default values in the AS PIC, you do not need to configure an IPSec proposal or IPSec policy. However, you must configure an IKE proposal that specifies the use of digital certificates, reference the IKE proposal and local certificate in an IKE policy, and apply the CA profile to the service set. To enable an IKE proposal for digital certificates, include the `rsa-signatures` statement at the `[edit services ipsec-vpn ike proposal` *`proposal-name`* `authentication-method]` hierarchy level. To reference the local certificate in the IKE policy, include the `local-certificate` statement at the `[edit services ipsec-vpn ike policy` *`policy-name`*`]` hierarchy level. To identify the CA or RA in the service set, include the `trusted-ca` statement at the `[edit services service-set` *`service-set-name`* `ipsec-vpn-options]` hierarchy level.

NOTE: For more information about default IKE and IPSec policies and proposals on the AS PIC, see "Configuring IKE Dynamic SAs" on page 77.

Optionally, you can configure automatic reenrollment of the certificate with the `auto-re-enrollment` statement at the [**edit security pki**] hierarchy level.

The remaining configuration components of your IKE-based IPSec tunnel are the same as when you use preshared keys. Enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional IKE dynamic SA in a rule called **rule-ike** at the `[edit ipsec-vpn rule]` hierarchy level. Reference this rule in a service set called **service-set-dynamic-BiEspsha3des** at the `[edit services service-set]` hierarchy level.

To direct traffic into the AS PIC and the IPsec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPSec inside interface into the OSPF configuration.

Router 2

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R1 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R3 so-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.1/30;
            }
        }
    }
    sp-1/2/0 {
        unit 0 {
            family inet;
        }
                           unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
            family inet;
            service-domain inside;
        }
                           unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
            family inet;
            service-domain outside;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.2/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.2;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
                               interface sp-1/2/0.1; # This sends OSPF traffic over the IPSec
tunnel.
            interface lo0.0;
        }
    }
}
               security { # Configure CA profiles here, including the URLs used to reach the CAs.
    pki {
        ca-profile entrust {
            ca-identity entrust;
            enrollment {
                url http://ca-1.example.com/cgi-bin/pkiclient.exe;
            }
                                 revocation-check {
                                       crl {
                      url ldap://10.157.90.185/o=juniper,c=uscertificateRevocationListbase;
                    # Specify the URL of the LDAP server where the CA stores the CRL.
                }
            }
        }
        ca-profile microsoft {
            ca-identity microsoft;
            enrollment {
                url http://192.168.11.78:80/certsrv/mscep/mscep.dll;
            }
        }
        ca-profile verisign {
            ca-identity verisign;
            enrollment {
                url http://pilotonsiteipsec.verisign.com/cgi-bin/pkiclient.exe;
            }
        }
    }
}
```

```
services {
                     service-set service-set-dynamic-BiEspsha3des { # Define your service set
here.
                           next-hop-service { # Required for dynamic routing protocols such as
OSPF.
            inside-service-interface sp-1/2/0.1;
            outside-service-interface sp-1/2/0.2;
        }
        ipsec-vpn-options {
                               trusted-ca entrust; # Reference the CA profile here.
                               local-gateway 10.1.15.1; # Specify the local IP address of the
IPSec tunnel.
        }
                         ipsec-vpn-rules rule-ike; # Reference your IPSec VPN rule here.
    }
    ipsec-vpn {
                           rule rule-ike { # Define your IPSec VPN rule here.
            term term-ike {
                then {
                                           remote-gateway 10.1.15.2; # The remote IP address of
the IPSec tunnel.
                                             dynamic { # This creates a dynamic SA.
                                                 ike-policy ike-digital-certificates; #
Reference your IKE policy here.
                    }
                }
            }
                               match-direction input; # Specify in which direction the rule
should match.
        }
        ike {
            proposal ike-proposal {
                                     authentication-method rsa-signatures; # Uses digital
certificates
            }
            policy ike-digital-certificates {
                                     proposals ike-proposal; # Apply the IKE proposal here.
                                     local-id fqdn router2.example.com; # Provide an identifier
for the local router.
                                     local-certificate local-entrust2; # Reference the local
certificate here.
                                     remote-id fqdn router3.example.com; # Provide an ID for the
remote router.
```

```
            }
        }
        establish-tunnels immediately;
    }
}
```

On Router 3, you must repeat the digital certificate procedures you performed on Router 2. If the IPSec peers do not have a symmetrical configuration containing all the necessary components, they cannot establish a peering relationship.

You need to request a CA certificate, create a local certificate, load these digital certificates into the router, and reference them in your IPSec configuration. Begin by configuring an IPSec CA profile. Include the **ca-profile** statement at the **[edit security pki]** hierarchy level and specify the trusted CA and URL of the CA server that handles CA certificate processing. Include the CRL statements found on Router 2 to complete your CA profile on Router 3.

After you configure the CA profile, request a CA certificate from the trusted CA. In this example, the certificate is enrolled online and installed into the router automatically.

```
user@R3> request security pki ca-certificate enroll ca-profile entrust
Received following certificates:
  Certificate: C=us, O=juniper
    Fingerprint: 00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10
  Certificate: C=us, O=juniper, CN=First Officer
    Fingerprint: bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17
  Certificate: C=us, O=juniper, CN=First Officer
    Fingerprint: 46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f
Do you want to load the above CA certificate ? [yes,no] (no) yes
```

NOTE: If you obtain the CA certificate directly from the CA (for example, as an e-mail attachment or website download), you can install it with the `request security pki ca-certificate load` command.

Next, generate a private/public key pair.

```
user@R3> request security pki generate-key-pair certificate-id local-entrust3
Generated key pair local-entrust3, key size 1024 bits
```

When the key pair is available, you can generate a local certificate request and send it to the CA for processing.

```
user@R3> request security pki generate-certificate-request
certificate-id local-entrust3 domain-name router3.example.com
filename entrust-req3 subject cn=router3.example.com
Generated certificate request
-----BEGIN CERTIFICATE REQUEST-----
MIIB8jCCAVsCAQAwZTEYMBYGA1UEAxMPdHA1Lmp1bmlwZXIubmV0MRQwEgYDVQQL
EwtFbmdpbmVlcmluZzEQMA4GA1UEChMHSnVuaXBlcjETMBEGA1UECBMKQ2FsaWZv
cm5pYTEMMAoGA1UEBhMDVVNBMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCg
Wjo50w8jrnphs0sRFvqQMwC6PlYa65thrJ8nHZ2qgYgRbSrO8hdODhvU6/5VuD2/
zBtgV5ZSAOlyV6DXqlbVj/2XirQAJMRCr1eYu6DhYRBMNq/UaQv4Z8Sse1EJv+uR
HTNbD7xlwpw2zwzltRuGFtFr/FrGB0hF7IE+Xm5e2wIDAQABoE0wSwYJKoZIhvcN
AQkOMT4wPDAOBgNVHQ8BAf8EBAMCB4AwKgYDVR0RAQH/BCAwHocEwKhGk4IWdHA1
LmVuZ2xhYi5qdW5pcGVyLm5ldDANBgkqhkiG9w0BAQQFAAOBgQBbiJ+ZCeQ59/eY
4Rd6awIpJFTz0svRZLxxjFWogusVTmaD2dsqFBqftS1eJBdeiueRcYMF9vOn0GKm
FNfouegwei5+vzdNmNo55eIb3rs4pP62q0W5CUgmbHrjtp3lyJsvuOxTTcPNY8zw
b6GyM2Hdkk3Vh2ReX11tQUSqYujTjw==
-----END CERTIFICATE REQUEST-----
Fingerprint:
7c:e8:f9:45:93:8d:a3:92:7f:18:29:02:f1:c8:e2:85:3d:ad:df:1f (sha1)
00:4e:df:a0:6b:ad:8c:50:da:7c:a1:cf:5d:37:b0:ea (md5)
```

The trusted CA digitally signs the local certificate and returns it to you. Copy the certificate file into the router and load the certificate.

```
user@R3> request security pki local-certificate load filename /tmp/router3-cert certificate-id
local-entrust3
Local certificate local-entrust3 loaded successfully
```

After the local and CA certificates have been loaded, you can reference them in your IPSec configuration. Using default values in the AS PIC, you do not need to configure an IPSec proposal or IPSec policy. However, you must configure an IKE proposal that uses digital certificates, reference the IKE proposal and local certificate in an IKE policy, and apply the CA profile to the service set. To enable the IKE proposal for digital certificates, include the `rsa-signatures` statement at the `[edit services ipsec-vpn ike proposal` *proposal-name* `authentication-method]` hierarchy level. To reference the local certificate in the IKE policy, include the `local-certificate` statement at the `[edit services ipsec-vpn ike policy` *policy-name*`]` hierarchy level. To identify the CA or RA in the service set, include the `trusted-ca` statement at the `[edit services service-set` *service-set-name* `ipsec-vpn-options]` hierarchy level.

The remaining configuration components of your IKE-based IPsec tunnel are the same as when you use preshared keys. Enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional IKE dynamic SA in a rule called **rule-ike** at the `[edit ipsec-vpn rule]` hierarchy level. Reference this rule in a service set called **service-set-dynamic-BiEspsha3des** at the `[edit services service-set]` hierarchy level.

To direct traffic into the AS PIC and the IPsec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPsec inside interface into the OSPF configuration.

Router 3

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R4 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.56.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R2 so-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.2/30;
            }
        }
    }
    sp-1/2/0 {
        unit 0 {
            family inet;
        }
                           unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
            family inet;
            service-domain inside;
        }
                           unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
            family inet;
            service-domain outside;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.3/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.3;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
                               interface sp-1/2/0.1; # This sends OSPF traffic over the IPSec
tunnel.
            interface lo0.0;
        }
    }
}
               security { # Configure CA profiles here, including the URLs used to reach the CAs.
    pki {
        ca-profile entrust {
            ca-identity entrust;
            enrollment {
                url http://ca-1.example.com/cgi-bin/pkiclient.exe;
            }
                                 revocation-check {
                                       crl {
                                           url                       ldap://10.157.90.185/
o=juniper,c=uscertificateRevocationListbase;
                    # Specify the URL of the LDAP server where the CA stores the CRL.
                }
            }
        }
        ca-profile microsoft {
            ca-identity microsoft;
            enrollment {
                url http://192.168.11.78:80/certsrv/mscep/mscep.dll;
            }
        }
        ca-profile verisign {
            ca-identity verisign;
            enrollment {
                url http://pilotonsiteipsec.verisign.com/cgi-bin/pkiclient.exe;
            }
        }
    }
}
services {
                     service-set service-set-dynamic-BiEspsha3des { # Define your service set
here.
                           next-hop-service { # Required for dynamic routing protocols such as
OSPF.
            inside-service-interface sp-1/2/0.1;
            outside-service-interface sp-1/2/0.2;
        }
        ipsec-vpn-options {
                               trusted-ca entrust; # Reference the CA profile here.
                               local-gateway 10.1.15.2; # Specify the local IP address of the
IPSec tunnel.
        }
                         ipsec-vpn-rules rule-ike; # Reference your IPSec VPN rule here.
    }
    ipsec-vpn {
                           rule rule-ike { # Define your IPSec VPN rule here.
            term term-ike {
                then {
                                           remote-gateway 10.1.15.1; # The remote IP address of
the IPSec tunnel.
                                             dynamic { # This creates a dynamic SA.
                                                 ike-policy ike-digital-certificates; #
Reference your IKE policy here.
                    }
                }
            }
                               match-direction input; # Specify in which direction the rule
should match.
        }
        ike {
            proposal ike-proposal {
                                     authentication-method rsa-signatures; # Uses digital
certificates
            }
            policy ike-digital-certificates {
                                     proposals ike-proposal; # Apply the IKE proposal here.
                                     local-id fqdn router3.example.com; # Provide an identifier 
for the local router.
                                     local-certificate local-entrust3; # Reference the local 
certificate here.
                                     remote-id fqdn router2.example.com; # Provide an ID for the 
remote router.
            }
        }
        establish-tunnels immediately;
    }
}
```

On Router 4, provide basic OSPF connectivity to Router 3.

Router 4

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R3 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.56.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.4/32;
            }
        }
    }
}
routing-options {
        router-id 10.0.0.4;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

Verifying Your Work

To verify proper operation of an IKE-based dynamic SA on the AS PIC, use the following commands:

- **ping**
- `show services ipsec-vpn certificates (detail)`
- `show services ipsec-vpn ike security-associations (detail)`
- `show services ipsec-vpn ipsec security-associations (detail)`
- `show services ipsec-vpn ipsec statistics`
- **traceroute**

To verify and manage digital certificates in your router, use the following commands:

- `show security pki ca-certificate (detail)`
- `show security pki certificate-request (detail)`
- `show security pki local-certificate (detail)`

The following sections show the output of these commands used with the configuration example:

Router 1

On Router 1, issue a **ping** command to the **so-0/0/0** interface on Router 4 to send traffic across the IPSec tunnel.

```
user@R1> ping 10.1.56.2
PING 10.1.56.2 (10.1.56.2): 56 data bytes
64 bytes from 10.1.56.2: icmp_seq=0 ttl=254 time=1.351 ms
64 bytes from 10.1.56.2: icmp_seq=1 ttl=254 time=1.187 ms
64 bytes from 10.1.56.2: icmp_seq=2 ttl=254 time=1.172 ms
64 bytes from 10.1.56.2: icmp_seq=3 ttl=254 time=1.154 ms
64 bytes from 10.1.56.2: icmp_seq=4 ttl=254 time=1.156 ms
^C
--- 10.1.56.2 ping statistics ---
```

```
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.154/1.204/1.351/0.074 ms
```

If you ping the loopback address of Router 4, the operation succeeds because the address is part of the OSPF network configured on Router 4.

```
user@R1> ping 10.0.0.4
PING 10.0.0.4 (10.0.0.4): 56 data bytes
64 bytes from 10.0.0.4: icmp_seq=0 ttl=62 time=1.318 ms
64 bytes from 10.0.0.4: icmp_seq=1 ttl=62 time=1.084 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=62 time=3.260 ms
^C
--- 10.0.0.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.084/1.887/3.260/0.975 ms
```

Router 2

To verify that matched traffic is being diverted to the bidirectional IPSec tunnel, view the IPSec statistics:

```
user@R2> show services ipsec-vpn ipsec statistics

PIC: sp-1/2/0, Service set: service-set-dynamic-BiEspsha3des

ESP Statistics:
  Encrypted bytes:           162056
  Decrypted bytes:           161896
  Encrypted packets:           2215
  Decrypted packets:           2216
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
```

To verify that the IKE SA negotiation is successful, issue the `show services ipsec-vpn ike security-associations` command:

```
user@R2> show services ipsec-vpn ike security-associations
Remote Address  State         Initiator cookie  Responder cookie  Exchange type
10.1.15.2       Matured       d82610c59114fd37  ec4391f76783ef28  Main
```

To verify that the IPsec security association is active, issue the `show services ipsec-vpn ipsec security-associations detail` command. Notice that the SA contains the default settings inherent in the AS PIC, such as ESP for the protocol and HMAC-SHA1-96 for the authentication algorithm.

```
user@R2> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-dynamic-BiEspsha3des

  Rule: rule-ike, Term: term-ike, Tunnel index: 1
  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  IPSec inside interface: sp-1/2/0.1
  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)

    Direction: inbound, SPI: 857451461, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 9052 seconds
    Hard lifetime: Expires in 9187 seconds
    Anti-replay service: Enabled, Replay window size: 64

    Direction: outbound, SPI: 1272330309, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 9052 seconds
    Hard lifetime: Expires in 9187 seconds
    Anti-replay service: Enabled, Replay window size: 64
```

To display the digital certificates that are used to establish the IPSec tunnel, issue the **show services ipsec-vpn certificates** command:

```
user@R2> show services ipsec-vpn certificates
Service set: service-set-dynamic-BiEspsha3des, Total entries: 3
  Certificate cache entry: 3
```

```
    Flags: Non-root Trusted
    Issued to: router3.example.com, Issued by: juniper
    Alternate subject: router3.example.com
    Validity:
      Not before: 2005 Nov 21st, 23:33:58 GMT
      Not after: 2008 Nov 22nd, 00:03:58 GMT

  Certificate cache entry: 2
    Flags: Non-root Trusted
    Issued to: router2.example.com, Issued by: juniper
    Alternate subject: router2.example.com
    Validity:
      Not before: 2005 Nov 21st, 23:28:22 GMT
      Not after: 2008 Nov 21st, 23:58:22 GMT

  Certificate cache entry: 1
    Flags: Root Trusted
    Issued to: juniper, Issued by: juniper
    Validity:
      Not before: 2005 Oct 18th, 23:54:22 GMT
      Not after: 2025 Oct 19th, 00:24:22 GMT
```

To display the CA certificate, issue the `show security pki ca-certificate detail` command. Notice that there are three separate certificates: one for certificate signing, one for key encipherment, and one for the CA's digital signature.

```
user@R2> show security pki ca-certificate detail
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 9235
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us
  Validity:
    Not before: 2005 Oct 18th, 23:54:22 GMT
    Not after: 2025 Oct 19th, 00:24:22 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    cb:9e:2d:c0:70:f8:ea:3c:f2:b5:f0:02:48:87:dc:68:99:a3:57:4f
    0e:b9:98:0b:95:47:0d:1f:97:7c:53:17:dd:1a:f8:da:e5:08:d1:1c
    78:68:1f:2f:72:9f:a2:cf:81:e3:ce:c5:56:89:ce:f0:97:93:fa:36
    19:3e:18:7d:8c:9d:21:fe:1f:c3:87:8d:b3:5d:f3:03:66:9d:16:a7
    bf:18:3f:f0:7a:80:f0:62:50:43:83:4f:0e:d7:c6:42:48:c0:8a:b2
    c7:46:30:38:df:9b:dc:bc:b5:08:7a:f3:cd:64:db:2b:71:67:fe:d8
    04:47:08:07:de:17:23:13
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10 (sha1)
    71:6f:6a:76:17:9b:d6:2a:e7:5a:72:97:82:6d:26:86 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: CRL signing, Certificate signing

Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 925c
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us, Common name: First Officer
  Validity:
    Not before: 2005 Oct 18th, 23:55:59 GMT
    Not after: 2008 Oct 19th, 00:25:59 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    c0:a4:21:32:95:0a:cd:ec:12:03:d1:a2:89:71:8e:ce:4e:a6:f9:2f
    1a:9a:13:8c:f6:a0:3d:c9:bd:9d:c2:a0:41:77:99:1b:1e:ed:5b:80
    34:46:f8:5b:28:34:38:2e:91:7d:4e:ad:14:86:78:67:e7:02:1d:2e
    19:11:b7:fa:0d:ba:64:20:e1:28:4e:3e:bb:6e:64:dc:cd:b1:b4:7a
    ca:8f:47:dd:40:69:c2:35:95:ce:b8:85:56:d7:0f:2d:04:4d:5d:d8
    42:e1:4f:6b:bf:38:c0:45:1e:9e:f0:b4:7f:74:6f:e9:70:fd:4a:78
    da:eb:10:27:bd:46:34:33
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17 (sha1)
    23:79:40:c9:6d:a6:f0:ca:e0:13:30:d4:29:6f:86:79 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: Key encipherment
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 925b
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us, Common name: First Officer
  Validity:
    Not before: 2005 Oct 18th, 23:55:59 GMT
    Not after: 2008 Oct 19th, 00:25:59 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    ea:75:c4:f3:58:08:ea:65:5c:7e:b3:de:63:0a:cf:cf:ec:9a:82:e2
    d7:e8:b9:2f:bd:4b:cd:86:2f:f1:dd:d8:a2:95:af:ab:51:a5:49:4e
    00:10:c6:25:ff:b5:49:6a:99:64:74:69:e5:8c:23:5b:b4:70:62:8e
    e4:f9:a2:28:d4:54:e2:0b:1f:50:a2:92:cf:6c:8f:ae:10:d4:69:3c
    90:e2:1f:04:ea:ac:05:9b:3a:93:74:d0:59:24:e9:d2:9d:c2:ef:22
    b9:32:c7:2c:29:4f:91:cb:5a:26:fe:1d:c0:36:dc:f4:9c:8b:f5:26
    af:44:bf:53:aa:d4:5f:67
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f (sha1)
    ee:cc:c7:f4:5d:ac:65:33:0a:55:db:59:72:2c:dd:16 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: Digital signature
```

To display the local certificate request, issue the `show security pki certificate-request` command:

```
user@R2> show security pki certificate-request
Certificate identifier: local-entrust2
  Issued to: router2.example.com
  Public key algorithm: rsaEncryption(1024 bits)
  Public key verification status: Passed
```

To display the local certificate, issue the `show security pki local-certificate` command:

```
user@R2> show security pki local-certificate
Certificate identifier: local-entrust2
  Issued to: router2.example.com, Issued by: juniper
  Validity:
    Not before: 2005 Nov 21st, 23:28:22 GMT
    Not after: 2008 Nov 21st, 23:58:22 GMT
  Public key algorithm: rsaEncryption(1024 bits)
  Public key verification status: Passed
```

Router 3

To verify that matched traffic is being diverted to the bidirectional IPSec tunnel, view the IPSec statistics:

```
user@R3> show services ipsec-vpn ipsec statistics

PIC: sp-1/2/0, Service set: service-set-dynamic-BiEspsha3des

ESP Statistics:
  Encrypted bytes:           161896
  Decrypted bytes:           162056
  Encrypted packets:           2216
  Decrypted packets:           2215
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
```

To verify that the IKE SA negotiation is successful, issue the `show services ipsec-vpn ike security-associations` command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

```
user@R3> show services ipsec-vpn ike security-associations
Remote Address  State         Initiator cookie  Responder cookie  Exchange type
10.1.15.1       Matured       d82610c59114fd37  ec4391f76783ef28  Main
```

To verify that the IPsec SA is active, issue the `show services ipsec-vpn ipsec security-associations detail` command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

```
user@R3> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-dynamic-BiEspsha3des

  Rule: rule-ike, Term: term-ike, Tunnel index: 1
  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  IPSec inside interface: sp-1/2/0.1
  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)

    Direction: inbound, SPI: 1272330309, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 7219 seconds
    Hard lifetime: Expires in 7309 seconds
    Anti-replay service: Enabled, Replay window size: 64

    Direction: outbound, SPI: 857451461, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 7219 seconds
    Hard lifetime: Expires in 7309 seconds
    Anti-replay service: Enabled, Replay window size: 64
```

To display the digital certificates that are used to establish the IPsec tunnel, issue the `show services ipsec-vpn certificates` command:

```
user@R3> show services ipsec-vpn certificates
Service set: service-set-dynamic-BiEspsha3des, Total entries: 3
  Certificate cache entry: 3
    Flags: Non-root Trusted
    Issued to: router3.example.com, Issued by: juniper
    Alternate subject: router3.example.com
    Validity:
      Not before: 2005 Nov 21st, 23:33:58 GMT
      Not after: 2008 Nov 22nd, 00:03:58 GMT

  Certificate cache entry: 2
    Flags: Non-root Trusted
    Issued to: router2.example.com, Issued by: juniper
    Alternate subject: router2.example.com
    Validity:
      Not before: 2005 Nov 21st, 23:28:22 GMT
      Not after: 2008 Nov 21st, 23:58:22 GMT

  Certificate cache entry: 1
    Flags: Root Trusted
    Issued to: juniper, Issued by: juniper
    Validity:
      Not before: 2005 Oct 18th, 23:54:22 GMT
      Not after: 2025 Oct 19th, 00:24:22 GMT
```

To display the CA certificate, issue the `show security pki ca-certificate detail` command. Notice that there are three separate certificates: one for certificate signing, one for key encipherment, and one for the CA's digital signature.

```
user@R3> show security pki ca-certificate detail
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 9235
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us
  Validity:
    Not before: 2005 Oct 18th, 23:54:22 GMT
    Not after: 2025 Oct 19th, 00:24:22 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    cb:9e:2d:c0:70:f8:ea:3c:f2:b5:f0:02:48:87:dc:68:99:a3:57:4f
    0e:b9:98:0b:95:47:0d:1f:97:7c:53:17:dd:1a:f8:da:e5:08:d1:1c
    78:68:1f:2f:72:9f:a2:cf:81:e3:ce:c5:56:89:ce:f0:97:93:fa:36
    19:3e:18:7d:8c:9d:21:fe:1f:c3:87:8d:b3:5d:f3:03:66:9d:16:a7
    bf:18:3f:f0:7a:80:f0:62:50:43:83:4f:0e:d7:c6:42:48:c0:8a:b2
    c7:46:30:38:df:9b:dc:bc:b5:08:7a:f3:cd:64:db:2b:71:67:fe:d8
    04:47:08:07:de:17:23:13
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10 (sha1)
    71:6f:6a:76:17:9b:d6:2a:e7:5a:72:97:82:6d:26:86 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: CRL signing, Certificate signing
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 925c
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us, Common name: First Officer
  Validity:
    Not before: 2005 Oct 18th, 23:55:59 GMT
    Not after: 2008 Oct 19th, 00:25:59 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    c0:a4:21:32:95:0a:cd:ec:12:03:d1:a2:89:71:8e:ce:4e:a6:f9:2f
    1a:9a:13:8c:f6:a0:3d:c9:bd:9d:c2:a0:41:77:99:1b:1e:ed:5b:80
    34:46:f8:5b:28:34:38:2e:91:7d:4e:ad:14:86:78:67:e7:02:1d:2e
    19:11:b7:fa:0d:ba:64:20:e1:28:4e:3e:bb:6e:64:dc:cd:b1:b4:7a
    ca:8f:47:dd:40:69:c2:35:95:ce:b8:85:56:d7:0f:2d:04:4d:5d:d8
    42:e1:4f:6b:bf:38:c0:45:1e:9e:f0:b4:7f:74:6f:e9:70:fd:4a:78
    da:eb:10:27:bd:46:34:33
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17 (sha1)
    23:79:40:c9:6d:a6:f0:ca:e0:13:30:d4:29:6f:86:79 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: Key encipherment
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 925b
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us, Common name: First Officer
  Validity:
    Not before: 2005 Oct 18th, 23:55:59 GMT
    Not after: 2008 Oct 19th, 00:25:59 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    ea:75:c4:f3:58:08:ea:65:5c:7e:b3:de:63:0a:cf:cf:ec:9a:82:e2
    d7:e8:b9:2f:bd:4b:cd:86:2f:f1:dd:d8:a2:95:af:ab:51:a5:49:4e
    00:10:c6:25:ff:b5:49:6a:99:64:74:69:e5:8c:23:5b:b4:70:62:8e
    e4:f9:a2:28:d4:54:e2:0b:1f:50:a2:92:cf:6c:8f:ae:10:d4:69:3c
    90:e2:1f:04:ea:ac:05:9b:3a:93:74:d0:59:24:e9:d2:9d:c2:ef:22
    b9:32:c7:2c:29:4f:91:cb:5a:26:fe:1d:c0:36:dc:f4:9c:8b:f5:26
    af:44:bf:53:aa:d4:5f:67
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f (sha1)
    ee:cc:c7:f4:5d:ac:65:33:0a:55:db:59:72:2c:dd:16 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: Digital signature
```

To display the local certificate request, issue the **show security pki certificate-request** command:

```
user@R3> show security pki certificate-request
Certificate identifier: local-entrust3
  Issued to: router3.example.com
  Public key algorithm: rsaEncryption(1024 bits)
  Public key verification status: Passed
```

To display the local certificate, issue the `show security pki local-certificate` command:

```
user@R3> show security pki local-certificate
Certificate identifier: local-entrust3
  Issued to: router3.example.com, Issued by: juniper
  Validity:
    Not before: 2005 Nov 21st, 23:33:58 GMT
    Not after: 2008 Nov 22nd, 00:03:58 GMT
  Public key algorithm: rsaEncryption(1024 bits)
  Public key verification status: Passed
```

Router 4

On Router 4, issue a `ping` command to the **so-0/0/0** interface on Router 1 to send traffic across the IPsec tunnel.

```
user@R4> ping 10.1.12.2
PING 10.1.12.2 (10.1.12.2): 56 data bytes
64 bytes from 10.1.12.2: icmp_seq=0 ttl=254 time=1.350 ms
64 bytes from 10.1.12.2: icmp_seq=1 ttl=254 time=1.161 ms
64 bytes from 10.1.12.2: icmp_seq=2 ttl=254 time=1.124 ms
64 bytes from 10.1.12.2: icmp_seq=5 ttl=254 time=1.116 ms
^C
--- 10.1.12.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.116/1.172/1.350/0.081 ms
```

The final way you can confirm that traffic travels over the IPsec tunnel is by issuing the `traceroute` command to the **so-0/0/0** interface on Router 1. Notice that the physical interface between Routers 2 and 3 is not referenced in the path; traffic enters the IPSec tunnel through the adaptive services IPSec inside interface on Router 3, passes through the loopback interface on Router 2, and ends at the **so-0/0/0** interface on Router 1.

```
user@R4> traceroute 10.1.12.2
traceroute to 10.1.12.2 (10.1.12.2), 30 hops max, 40 byte packets
 1  10.1.15.2 (10.1.15.2)  0.987 ms  0.630 ms  0.563 ms
 2  10.0.0.2 (10.0.0.2)  1.194 ms  1.058 ms  1.033 ms
 3  10.1.12.2 (10.1.12.2)  1.073 ms  0.949 ms  0.932 ms
```

For additional information on using digital certificates, see the *Junos Services Interfaces Configuration Guide* and the *Junos System Basics and Services Command Reference*.

Example: Dynamic Endpoint Tunneling Configuration

IN THIS SECTION

Figure 10: IPSec Dynamic Endpoint Tunneling Topology Diagram

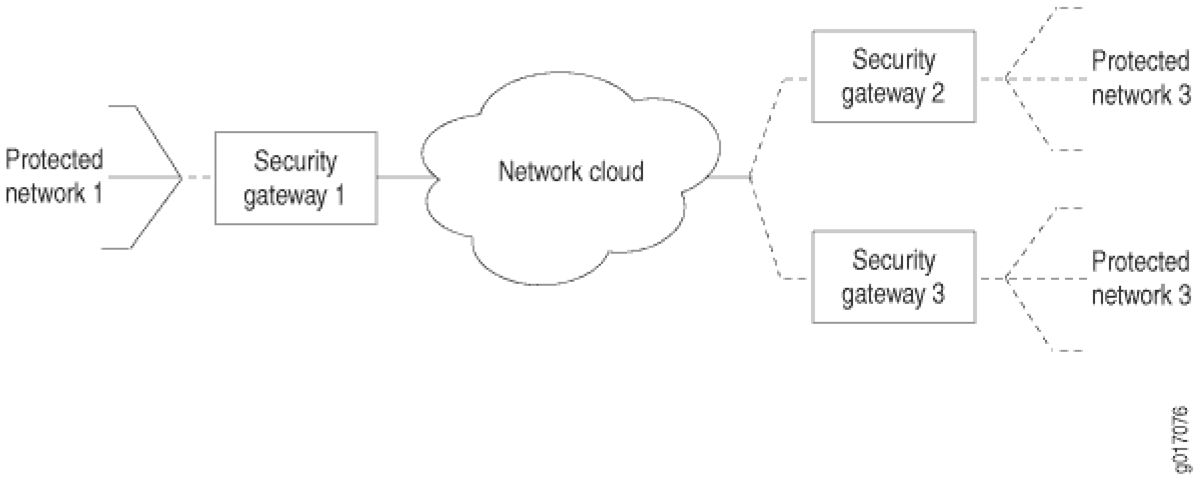


Figure 10 on page 201 shows a local network N-1 located behind security gateway SG-1. SG-1 is a Juniper Networks router terminating dynamic peer endpoints. The tunnel termination address on SG-1 is **10.7.7.2** and the local network address is **172.16.1.0/24**.

A remote peer router obtains addresses from an ISP pool and runs RFC-compliant IKE. Remote network N-2 has address **172.16.2.0/24** and is located behind security gateway SG-2 with tunnel termination address **10.7.7.1**.

On Router SG-1, configure an IKE access profile to accept proposals from SG-2. Apply the interface identifier from the access profile to the inside services interface and apply the IKE access profile itself to the IPSec next-hop style service set.

Router SG-1

```
[edit]
access {
    profile ike_access {
                           client * { # Accepts proposals from specified peers that use the
preshared key.
            ike {
                allowed-proxy-pair local 10.255.14.63/32 remote 10.255.14.64/32;
                pre-shared-key ascii-text "$ABC123"; # SECRET-DATA
                                     interface-id test_id; # Apply this ID to the inside
services interfaces.
            }
        }
    }
}
interfaces {
    fe-0/0/0 {
        description "Connection to the local network";
        unit 0 {
            family inet {
                address 172.16.1.1/24;
            }
        }
    }
    so-1/0/0 {
        description "Connection to SG-2";
        no-keepalives;
        encapsulation cisco-hdlc;
        unit 0 {
            family inet {
                address 10.7.7.2/30;
            }
        }
    }
    sp-3/3/0 {
        unit 0 {
            family inet;
        }
        unit 3 {
            dial-options {
                                     ipsec-interface-id test_id; # Accepts dynamic endpoint
tunnels.
                shared;
            }
            service-domain inside;
        }
        unit 4 {
            family inet;
            service-domain outside;
        }
    }
}
services {
                     service-set dynamic_nh_ss { # Create a next-hop service set
        next-hop-service { # for the dynamic endpoint tunnels.
            inside-service-interface sp-3/3/0.3;
            outside-service-interface sp-3/3/0.4;
        }
        ipsec-vpn-options {
            local-gateway 10.7.7.2;
                               ike-access-profile ike_access; # Apply the IKE access profile
here.
        }
    }
}
```

Verifying Your Work

To verify proper operation of a dynamic endpoint tunnel configured on the AS PIC, use the following command:

```
show services ipsec-vpn ipsec security-associations (detail)
```

The following section shows output from this command used with the configuration example. The dynamically created rule **_junos_** appears in the output, as well as the establishment of the inbound and outbound dynamically created tunnels.

```
user@router> show services ipsec-vpn ipsec security-associations detail
Service set: dynamic_nh_ss

  Rule:  _junos_ , Term: tunnel4, Tunnel index: 4
  Local gateway: 10.7.7.2, Remote gateway: 10.7.7.1
  Local identity: ipv4(any:0,[0..3]=10.255.14.63)
  Remote identity: ipv4(any:0,[0..3]=10.255.14.64)

    Direction: inbound , SPI: 428111023, AUX-SPI: 0
     Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 27660 seconds
    Hard lifetime: Expires in 27750 seconds
    Anti-replay service: Enabled, Replay window size: 64

    Direction: outbound , SPI: 4035429231, AUX-SPI: 0
     Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 27660 seconds
    Hard lifetime: Expires in 27750 seconds
    Anti-replay service: Enabled, Replay window size: 64
```

3

PART

Digital Certificates

CHAPTER 9

Configuring Digital Certificates

IN THIS CHAPTER

Public Key Cryptography

IN THIS SECTION

Understanding Public Key Cryptography on Switches

Cryptography describes the techniques related to the following aspects of information security:

- Privacy or confidentiality
- Integrity of data
- Authentication
- Nonrepudiation or nonrepudiation of origin—Nonrepudiation of origin means that signers cannot claim that they did not sign a message while claiming that their private key remains secret. In some nonrepudiation schemes used in digital signatures, a timestamp is attached to the digital signature, so that even if the private key is exposed, the signature remains valid. Public and private keys are described in the following text.

In practice, cryptographic methods protect the data transferred from one system to another over public networks by encrypting the data using an encryption key. Public key cryptography (PKC), which is used on Juniper Networks EX Series Ethernet Switches, uses a pair of encryption keys: a public key and a private key. The public and private keys are created simultaneously using the same encryption algorithm. The private key is held by a user secretly and the public key is published. Data encrypted with a public key can be decrypted only with the corresponding private key and vice versa. When you generate a public/private key pair, the switch automatically saves the key pair in a file in the certificate store, from which 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), Juniper Networks Junos operating system (Junos OS) supports RSA only.

To configure, monitor, and manage the public key infrastructure (PKI) on Juniper Networks devices, see Public Key Infrastructure User Guide.

Configuring RADIUS over TLS (RADSEC)

IN THIS SECTION

RADIUS over TLS is designed to provide secure communication of RADIUS requests using the Transport Secure Layer (TLS) protocol. RADIUS over TLS, also known as RADSEC, redirects regular RADIUS traffic to remote RADIUS servers connected over TLS. RADSEC allows RADIUS authentication, authorization, and accounting data to be passed safely across untrusted networks.

Configuring RADSEC over TLS Using an Existing Certificate

IN THIS SECTION

This section explains how to configure RADIUS over TLS (RADSEC) using an existing certificate.

Configure the RADSEC server

Use the following configuration example to add a RADIUS server named `RADSEC`.

```
admin@t327-dut1.cond# configure authority radius-server radsec
admin@t327-dut1.cond (radius-server[name=radsec])# address 172.18.5.224
admin@t327-dut1.cond (radius-server[name=radsec])# port 2083
admin@t327-dut1.cond (radius-server[name=radsec])# protocol tls
admin@t327-dut1.cond (radius-server[name=radsec])# account-creation manual
admin@t327-dut1.cond (radius-server[name=radsec])# ocsp strict
admin@t327-dut1.cond (radius-server[name=radsec])# server-name t327-dut1.openstacklocal
admin@t327-dut1.cond (radius-server[name=radsec])# top
```

Configure the Trusted CA Certificate

The trusted CA certificate is necessary to validate the incoming client certificate. Certificates are pasted in as a multi-line config. Create a certificate root named `ca_root` and paste the certificate file content into the command:

```
admin@conductor-node-1.Conductor# config authority trusted-ca-certificate ca_root
admin@conductor-node-1.Conductor (trusted-ca-certificate[name=ca_root])# content
Enter plain for content (Press CTRL-D to finish):
<paste-cert-file-content-here>
```

NOTE: The `trusted-ca-certificate` is a list and may contain different CA roots used for different certificates. In that case, naming them all `ca_root` would not be suitable. In that case, choose a name that is meaningful to the user and CA, eg: `globalsign_root`

Configure a Client Certificate to be used for the RADIUS client

Repeat the previous certificate to create a client certificate named `radsec`

```
admin@conductor-node-1.Conductor# config authority client-certificate radsec
admin@conductor-node-1.Conductor (client-certificate[name=radsec])# content
Enter plain for content (Press CTRL-D to finish):
<paste-cert-file-content-here>
```

Configure the RADIUS server at the Authority level to use the configured client certificate

Associate the previously configured `radsec` client certificate to the radius server running on a specified node.

```
configure authority router cond node t327-dut1 radius client-certificate-name radsec
```

Note that the client certificate selected should match the appropriate IP/hostname of the node as seen from the RADIUS server.

`validate` and `commit` the changes.

Create a RADIUS User

Create a remotely authenticated RADIUS user. In this example, we create a user called `test1`

```
*admin@conductor-node-1.Conductor# create user test1
Full Name: test1
Authentication Type (remote or local): remote
Roles (space separated): admin
Enabled (true or false): true
Account 'test1' successfully created
```

When the user logs into the node `t327-dut1` via ssh, the authentication request is sent via RADSEC to the server `172.18.5.224` and the user is authenticated.

RADSEC Configuration - Generate Certificate

Use the following examples to generate a client certificate for use on the device.

RADSEC Configuration - Generate Certificate

IN THIS SECTION

Generate the Signing Request

Use the `create certificate request client` command to generate the signing request.

```
admin@conductor-node-1.Conductor# create certificate request client radsec
Country name (2 letter code): US
State or province name (full name): MA
Locality name (eg: city): Westford
Organization name (eg: company): Juniper
Organization unit (eg: engineering):
Common name: dut1
Email address:
Subject Alternative Name - DNS (fully qualified domain name):
Subject Alternative Name - IP Address:
% Error: Could not create request: Subject Alternative Name (DNS or IP address) is required
admin@conductor-node-1.Conductor# create certificate request client radsec
Country name (2 letter code): US
State or province name (full name): MA
Locality name (eg: city): Westford
Organization name (eg: company): Juniper
Organization unit (eg: engineering):
Common name: dut1
Email address:
Subject Alternative Name - DNS (fully qualified domain name): dut1
Subject Alternative Name - IP Address: 10.27.32.203
```

```
Request successfully generated:

-----BEGIN CERTIFICATE REQUEST-----
MIIC1jCCAb4CAQAwTjENMAsGA1UEAwwEZHV0MTELMAkGA1UEBhMCVVMxETAPBgNV
BAcMCFdlc3Rmb3JkMRAwDgYDVQQKDAdKdW5pcGVyMQswCQYDVQQIDAJNQTCCASIw
DQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAJ8WwHXP/z49sFsxpN5L9THO5y8N
f/as8Nn6XUyG86YyxcR5IYL5gKR5//EunoVjLAUCHgBqxwaUa3enhNEQS97N4Bcs
E7YygMkI7oAnHCioslB+x2Am/xKPRosh3s50fIN3mY409/byMGipfGcyNlMT8XbS
XF/zmGBI1/4aRbeqL5VMDPO+9DNRxXMgqBs2y48WanGvZeZTP5B/sSczlhOSxHnu
DxNYQ7+rZs9NpKzktCXOSA8nsz
.
.
.
wp4dOHuKsnf+ZsfNK4AGUYdh3qEa1/xJxyug1R3AGjItbkUzbJpR6hp7B0YYWV87
QALMf6F0SKBDXg++
-----END CERTIFICATE REQUEST-----
```

Configure the Trusted CA Certificate

The trusted CA certificate is necessary to validate the incoming client certificate. Certificates are pasted in, as a multi-line configuration. Create a root certificate named `ca_root` and paste the certificate file content into the command.

```
admin@conductor-node-1.Conductor# configure authority trusted-ca-certificate ca_root
*admin@conductor-node-1.Conductor (trusted-ca-certificate[name=ca_root])# content
Enter plain for content (Press CTRL-D to finish):
-----BEGIN PRIVATE KEY-----
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQCqfzVmeFPMA+Jc
53MlVF3LoYZAkqh1Dz3+HFnegcAU3/tCGSdfJad/PeF5KEQDDnF0vc9XbfS2/wJC
wHAt15TH3iarSPE3dV3L0c1tyOFaMUNLAd3nsPArR0w/1YAfr1cAN0rEUZ4WmkZK
vyFx6AsuVm5MpXR4z7U4j955sqRkWsi3I1hLtMPzuWEJA/AbpTCxb1k2xJDQWira
/NALlz6NPVRcngBt56ZDhMNmy/g2zGEcmitEqMUOS7apvRk6hZK94dfjSQe4iEpX
Sdd6vvZxdrWGV10lmDDH0SPtmGBE+34r1UNIbp/XVRh6KxiNcjFVNBwlwqATmTYh
xkXAPw1pAgMBAAECggEACZ3YNLnnvBOiAmx5larvCWvIZz7+am/cJseRmBfIbkT9
5ooFqvu0OVyTqaJIR8XaR2PnXH6StXmntnqDpHWQTqUvlbGANIqWsyiig26zFCEu
IAXwr0TKRERzKAWT4lwmOAGi4LuQa6Ty/wdNyx9z9f6hBQi2C5Rnm9OdkE6vsAtJ
NbNcsV+bvedfLoJqG1MM3sh3LT3RAltaM0ntw3PdFiMVcQIJgGr85nVJcg4SCUkh
JKlfUE83IqkwAd1V0jn/2yopCmQBLrpyqlRu2MmwFiIS+IUcoReemNK8mlfd8hbR
.
.
```

```
.
2P6CP4iOY1EjsxNssrLJKkxXdagYeZo5X2KOIqZ8FeVli4BM0mqX96UPN2zV3dNP
eN1DF6VSLghh30ITUauYdQ++
-----END PRIVATE KEY-----

-----BEGIN CERTIFICATE-----
MIIDlDCCAnygAwIBAgIVAJHxzhL42q7io2PBDPR+TCeBsyQgMA0GCSqGSIb3DQEB
CwUAMFExCzAJBgNVBAYTAlVTMRYwFAYDVQQIDA1NYXNzYWNodXNldHRzMREwDwYD
VQQKDAhUZXN0IEluYzEXMBUGA1UEAwwOY2EuZXhhbXBsZS5jb20wHhcNMjQxMDIy
MTYzODI1WhcNMjUxMDIyMTYzODI1WjBRMQswCQYDVQQGEwJVUzEWMBQGA1UECAwN
TWFzc2FjaHVzZXR0czERMA8GA1UECgwIVGVzdCBJbmMxFzAVBgNVBAMMDmNhLmV4
YW1wbGUuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqn81ZnhT
zAPiXOdzJVRdy6GGQJKodQ89/hxZ3oHAFN/7QhknXyWnfz3heShEAw5xdL3PV230
tv8CQpHKHjWWQzG1MM3sh3LT3RAltaM0NT6shNXE3va46f3zotWBd6PK9jC/Tpme
.
.
.
qynFiqlV0UDGgH+e8hCp41Seva5vBGYvwMVHPU80rhoAsTh1BNpM1r9xbvDQs5ui
3QyeFCt/O0A=
-----END CERTIFICATE-----
```

Import the Client Certificate

After the certificate is signed and returned, it is imported into the SSR for use by the client using the `import certificate client` command. It is validated against any trusted certificates entered using `trusted-ca-certificate`. See the following example that shows a valid self-signed certificate being imported.

```
admin@conductor-node-1.Conductor# import certificate client radsec
Enter the end point certificate in PEM format (Press CTRL-D to finish):
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQDFrn/2q4mijt14
gjmN2agDfu6sykg4OJ2NDy4IRrBYilExRJHllAndtc04rp7EQ544Z+/J/dNJrmXK
GnHvm/Rg0UdKnbFrw5aentpx3rFefdaf8nlJLW5rFH1wxDqUhE+y5q+s+8k3ESt0
9L/26OxTQP11t5Vh/BEkK5iVHLDBGyHntUvEnM5tFWL7+NvefhuZ6McvY7GPDR8c
bkuNHXlv9laeXQlI6IiiYum8waQDnJBGEx2wPTUguZJWP0YgxLinKiCDIINNEf+Y
dGqxf7I/h01yH4nDGR3nad30fAN+10chzjMHYhmpPVR0K9IAPbyGucK0aOriJqZ5
91wL39G5AgMBAAECggEAE2/xDSQYyG8bv7muRxBbwNw+Q6cwKrcGZtRTRmUM+ee/
zAReBCDmR3KU1zn0SoALkqhFn6rhl6EaSSEIivLeuJZbWC7hPyNgMACWohOvhQcC
.
.
.
```

```
WiYWxHz5Q4wUxV5uTJR3Jq5rzcHr1shyVDT+aFf9tyNdcLFfbziZ1y/EfAPkOOoH
jLD4SXCWbmRxHYVMn3yhqK4=
-----END PRIVATE KEY-----

-----BEGIN CERTIFICATE-----
MIIDpDCCAoygAwIBAgIVAL1k460IeyrQWoU82ZVHZ2asUrTuMA0GCSqGSIb3DQEB
CwUAMFExCzAJBgNVBAYTAlVTMRYwFAYDVQQIDA1NYXNzYWNodXNldHRzMREwDwYD
VQQKDAhUZXN0IEluYzEXMBUGA1UEAwwOY2EuZXhhbXBsZS5jb20wHhcNMjQxMDIy
MTYzODI4WhcNMjUwMTIwMTYzODI4WjBVMQswCQYDVQQGEwJVUzEWMBQGA1UECAwN
TWFzc2FjaHVzZXR0czERMA8GA1UECgwIVGVzdCBJbmMxGzAZBgNVBAMMEmNsaWVu
dC5leGFtcGxlLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAMWu
f/ariaKO3XiCOY3ZqAN+7qzKSDg4nY0PLghGsFiKUTFEkeWUCd21zTiunsRDnjhn
78n900muZcoace+b9GDRR0qdsWvDlp6e2nHesV591p/yeUktbmsUfXDEOpSET7Lm
r6z7yTcRK3T0v/bo7FNA/XW3lWH8ESQrmJUcsMEbIee1S8Sczm0VYvv4295+G5no
xy9jsY8NHxxuS40deW/2Vp5dCUjoiKJi6bzBpAOckEYTHbA9NSC5klY/RiDEuKcq
IIMgg00R/5h0arF/sj/fL0cKofSeAgu11z1891d1sc0OMwdiGak9VHQr0gA9vIa5
.
.
.
9cgLsL60tukLdwxH5S6gAw/MSm6ABYjdv
-----END CERTIFICATE-----



/usr/lib/128technology/unzip/pcli/runfiles/pypi__36__cryptography_40_0_2/cryptography/x509/
base.py:576: CryptographyDeprecationWarning: Parsed a negative serial number, which is
disallowed by RFC 5280.
  return rust_x509.load_pem_x509_certificates(data)
✓ Importing...
Certificate imported successfully
Would you like to add the certificate to your configuration? [y/N]: y
Which router is this certificate for? (Select all if it applies to the entire authority) [all]:
all
% Warning:
1. certificate contains the following issues: does not have the extendKeyUsage extension


    config
        authority
            client-certificate radius
                content

2. certificate contains the following issues: does not have the extendKeyUsage extension
```

```
    config
        authority
            client-certificate conductor-radius
                content

Certificate imported successfully
Would you like to clean up the temporary certificate and key files? [Y/n]: Y
```

Configure the Device to Accept the Client Certificate

Use this command to configure your device to accept the certificate.

```
configure authority router ComboWest node combo-west radius client-certificate-name radsec
```

IKE Policy for Digital Certificates on an ES PIC

IN THIS SECTION

Configuring an IKE Policy for Digital Certificates for an ES PIC

IN THIS SECTION

An *IKE* policy for *digital certificate*s defines a combination of security parameters (IKE proposals) to be used during IKE negotiation. It defines a peer address and the proposals needed for that connection. During the IKE negotiation, IKE looks for an IKE policy that is the same on both peers. The peer that initiates the negotiation sends all its policies to the remote peer, and the remote peer tries to find a match.

To configure an IKE policy for digital certificates for an ES *PIC*, include the following statements at the `[edit security ike policy ike-peer-address]` hierarchy level:

```
[edit security ike]
policy ike-peer-address{
    encoding (binary | pem);
    identity identity-name;
    local-certificate certificate-filename;
    local-key-pair private-public-key-file;
}
```

Tasks for configuring an IKE policy for digital certificates are:

Configuring the Type of Encoding Your CA Supports

By default, the encoding is set to binary. Encoding specifies the file format used for the `local-certificate` and `local-key-pair` statements. By default, the binary (distinguished encoding rules) format is enabled. *PEM* is an ASCII base 64 encoded format. Check with your CA to determine which file formats it supports.

To configure the file format that your CA supports, include the `encoding` statement and specify a binary or PEM format at the `[edit security ike policy ike-peer-address]` hierarchy level:

```
[edit security ike policy ike-peer-address ]
encoding (binary | pem);
```

Configuring the Identity to Define the Remote Certificate Name

To define the remote certificate name, include the `identity` statement at the `[edit security ike policy ike-peer-address]` hierarchy level:

```
[edit security ike policy ike-peer-address]
identity identity-name;
```

identity-name defines the identity of the remote certificate name if the identity cannot be learned through IKE (ID payload or IP address).

Specifying the Certificate Filename

To configure the certificate filename from which to read the local certificate, include the `local-certificate` statement at the `[edit security ike policy ike-peer-address]` hierarchy level:

```
[edit security ike policy ike-peer-address]
local-certificate certificate-filename;
```

certificate-filename specifies the file from which to read the local certificate.

Specifying the Private and Public Key File

To specify the filename from which to read the public and private key, include the `local key-pair` statement at the `[edit security ike policy ike-peer-address]` hierarchy level:

```
[edit security ike policy ike-peer-address ]
local-key-pair private-public-key-file;
```

private-public-key-file specifies the file from which to read the pair key.

SEE ALSO

> Digital Certificates Overview

Obtaining a Signed Certificate from the CA for an ES PIC

To obtain a signed certificate from the *CA*, issue the following command:

```
user@host> request security certificate enroll filename filename subject c=us,o=x alternative-
subject certificate-ip-address certification-authority certificate-authority key-file key-file-
name domain-name domain-name
```

The results are saved in a specified file to the **/var/etc/ikecert** directory.

The following example shows how to obtain a CA signed certificate by referencing the configured `certification-authority` statement `local` . This statement is referenced by the `request security certificate` enroll filename *filename* subject *subject* alternative-subject *alternative-subject* certification-authority *certification-authority* command.

```
[edit]
security {
    certificates {
        certification-authority local {
            ca-name xyz.company.com;
            file l;
            enrollment-url "http://www.xyzcompany.com";
        }
    }
}
```

To obtain a signed certificate from the CA, issue the following command:

```
user@host> request security certificate enroll filename I subject c=uk,o=london alternative-
subject 10.50.1.4 certification-authority verisign key-file host-1.prv domain-name
host.xyzcompany.com
CA name: xyz.company.com CA file: ca_verisign
local pub/private key pair: host.prv
subject: c=uk,o=london domain name: host.example.com
alternative subject: 10.50.1.4
Encoding: binary
Certificate enrollment has started. To see the certificate enrollment status, check the key
management process (kmd) log file at /var/log/kmd. <--------------
```

For information about how to use the operational mode commands to obtain a signed certificate, see the CLI Explorer.

Another way to obtain a signed certificate from the CA is to reference the configured statements such as the URL, CA name, and CA certificate file by means of the `certification-authority` statement:

```
user@host> request security certificate enroll filename m subject c=us ,o=x alternative-subject
192.0.2.1 certification-authority local key-file y domain-name abc.company.com
```

SEE ALSO

Digital Certificates Overview

Associating the Configured Security Association with a Logical Interface

Configuring the ES *PIC* associates the configured *SA* with a logical interface. This configuration defines the *tunnel* itself (logical subunit, tunnel addresses, maximum transmission unit [*MTU*], optional interface addresses, and the name of the SA to apply to traffic).

The addresses configured as the tunnel source and destination are the addresses in the outer IP header of the tunnel.

NOTE: The tunnel source address must be configured locally on the router, and the tunnel destination address must be a valid address for the security gateway terminating the tunnel.

The M5, M10, M20, and M40 routers support the ES PIC.

The SA must be a valid tunnel-mode SA. The interface address and destination address listed are optional. The destination address allows the user to configure a static route to encrypt traffic. If a static route uses that destination address as the next hop, traffic is forwarded through the portion of the tunnel in which encryption occurs.

The following example shows how to configure an *IPsec* tunnel as a logical interface on the ES PIC. The logical interface specifies the tunnel through which the encrypted traffic travels. The **ipsec-sa** statement associates the security profile with the interface.

```
[edit interfaces]
es-0/0/0 {
    unit 0 {
        tunnel {
            source tunnel 10.5.5.5;                      # tunnel source address
            destination 10.6.6.6;                          # tunnel destination address
        }
        family inet {
            ipsec-sa ipsec-sa; # name of security association to apply to packet
            address 10.1.1.8/32 { # local interface address inside local VPN
            destination 10.2.2.254; # destination address inside remote VPN
        }
    }
}
```

SEE ALSO

Configuring Security Associations for IPsec on an ES PIC | 44

IPsec Tunnel Traffic Configuration

IN THIS SECTION

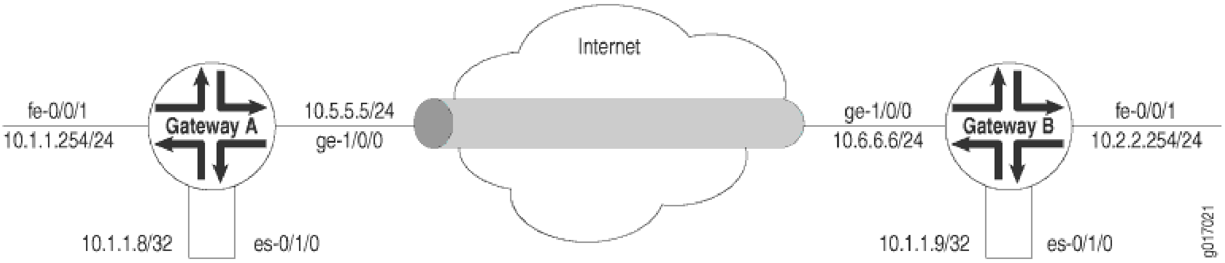
IPsec Tunnel Traffic Configuration Overview

Traffic configuration defines the traffic that must flow through the *IPsec tunnel*. You configure outbound and inbound firewall filters, which identify and direct traffic to be encrypted and confirm that decrypted traffic parameters match those defined for the given tunnel. The outbound filter is applied to the LAN or WAN interface for the incoming traffic you want to encrypt off of that *LAN* or *WAN*. The inbound filter is applied to the ES PIC to check the policy for traffic coming in from the remote host. Because of the complexity of configuring a router to forward packets, no automatic checking is done to ensure that the configuration is correct. Make sure that you configure the router very carefully.

NOTE: The valid firewall filters statements for IPsec are **destination-port**, **source-port**, **protocol**, **destination-address**, and **source-address**.

In Figure 11 on page 220, Gateway A protects the network **10.1.1.0/24**, and Gateway B protects the network **10.2.2.0/24**. The gateways are connected by an IPsec tunnel.

Figure 11: Example: IPsec Tunnel Connecting Security Gateways



The SA and ES interfaces for Gateway A are configured as follows:

```
[edit security ipsec]
security-association manual-sa1 {
    manual {
        direction bidirectional {
            protocol esp;
            spi 2312;
            authentication {
                algorithm hmac-md5-96;
                key ascii-text 1234123412341234;
            }
            encryption {
                algorithm 3des-cbc;
                key ascii-text 123456789009876543211234;
            }
        }
    }
}
[edit interfaces es-0/1/0]
unit 0 {
    tunnel {
        source 10.5.5.5;
        destination 10.6.6.6;
    }
    family inet {
        ipsec-sa manual-sa1;
        address 10.1.1.8/32 {
            destination 10.1.1.9;
        }
    }
}
```

The SA and ES interfaces for Gateway B are configured as follows:

```
[edit security ipsec]
security-association manual-sa1 {
    manual {
        direction bidirectional {
            protocol esp;
            spi 2312;
            authentication {
                algorithm hmac-md5-96;
                key ascii-text 1234123412341234;
            }
            encryption {
                algorithm 3des-cbc;
                key ascii-text 123456789009876543211234;
            }
        }
    }
}
[edit interfaces es-0/1/0]
unit 0 {
    tunnel {
        source 10.6.6.6;
        destination 10.5.5.5;
    }
    family inet {
        ipsec-sa manual-sa1;
        address 10.1.1.9/32; {
            destination 10.1.1.8;
        }
    }
}
```

SEE ALSO

Example: Configuring an Outbound Traffic Filter | 222

Example: Applying an Outbound Traffic Filter | 223

Example: Configuring an Outbound Traffic Filter

*Firewall filter*s for outbound traffic direct the traffic through the desired *IPsec tunnel* and ensure that the tunneled traffic goes out the appropriate interface (see "IPsec Tunnel Traffic Configuration Overview" on page 219). Here, an outbound firewall filter is created on security Gateway A; it identifies the traffic to be encrypted and adds it to the input side of the interface that carries the internal VPN traffic:

```
[edit firewall]
filter ipsec-encrypt-policy-filter {
    term term1 {
        from {
            source-address {          # local network
            10.1.1.0/24;
        }
        destination-address {    # remote network
        10.2.2.0/24;
    }
}
then ipsec-sa manual-sa1;     # apply SA name to packet
term default {
    then accept;
}
```

NOTE: The source address, port, and protocol on the outbound traffic filter must match the destination address, port, and protocol on the inbound traffic filter. The destination address, port, and protocol on the outbound traffic filter must match the source address, port, and protocol on the inbound traffic filter.

SEE ALSO

Example: Applying an Outbound Traffic Filter | 223

IPsec Tunnel Traffic Configuration Overview | 219

Example: Applying an Outbound Traffic Filter

After you have configured the outbound firewall filter, you apply it:

```
[edit interfaces]
fe-0/0/1 {
    unit 0 {
        family inet {
            filter {
                input ipsec-encrypt-policy-filter;
            }
            address 10.1.1.254/24;
        }
    }
}
```

The outbound filter is applied on the Fast Ethernet interface at the `[edit interfaces fe-0/0/1 unit 0 family inet]` hierarchy level. Any packet matching the IPsec action term (`term 1`) on the input filter (`ipsec-encrypt-policy-filter`), configured on the Fast Ethernet interface, is directed to the ES PIC interface at the `[edit interfaces es-0/1/0 unit 0 family inet]` hierarchy level. If a packet arrives from the source address 10.1.1.0/24 and goes to the destination address 10.2.2.0/24, the Packet Forwarding Engine directs the packet to the ES PIC interface, which is configured with the `manual-sa1` SA. The ES PIC receives the packet, applies the `manual-sa1` SA, and sends the packet through the tunnel.

The router must have a route to the tunnel endpoint; add a static route if necessary.

SEE ALSO

IPsec Tunnel Traffic Configuration Overview | 219

Example: Configuring an Inbound Traffic Filter for a Policy Check

IN THIS SECTION

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

Here, an inbound firewall filter, which performs the final IPsec policy check, is created on security Gateway A. This check ensures that only packets that match the traffic configured for this tunnel are accepted. This filter is configured via the CLI interface at the `[edit firewall family inet]` hierarchy level.

Configuration

IN THIS SECTION

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 this example, perform the following tasks:

CLI Quick Configuration

To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the `[edit]` hierarchy level.

```
[edit]
set firewall family inet filter ipsec-decrypt-policy-filter term term1 from source-address
10.2.2.0/24
set firewall family inet filter ipsec-decrypt-policy-filter term term1 from destination-address
10.1.1.0/24
set firewall family inet filter ipsec-decrypt-policy-filter term term1 then accept
commit
```

Configuring the firewall filter

Step-by-Step Procedure

To configure the firewall filter, `ipsec-decrypt-policy-filter` that catches traffic from the remote `10.2.2.0/24` netowrk that is destined for the local `10.1.1.0/24` network:

1. Create the firewall filter:

```
[edit]
user@host# edit firewall family inet filter ipsec-decrypt-policy-filter
```

2. Configure matching for source and destination addresses:

```
[edit firewall family inet filter ipsec-decrypt-policy-filter]
user@host# set term term1 from source-address 10.2.2.0/24
user@host# set term term1 from destination-address 10.1.1.0/24
```

3. Configure the filter to accept the matched traffic:

```
[edit firewall family inet filter ipsec-decrypt-policy-filter]
user@host# set term term1 then accept
```

NOTE: The accept statement within the `term` *`term1`* is for this filter only. Traffic that does not match this filter term will be dropped by the default firewall action.

4. Confirm your candidate firewall configuration by issuing the `show` configuration command at the `[edit firewall family inet]` hierarchy level

```
[edit firewall family inet]
user@host# show
filter ipsec-decrypt-policy-filter {
    term term1 {                                  # perform policy check
        from {
            source-address {                      # remote network
                10.2.2.0/24;
            }
            destination-address {                 # local network
```

```
                  10.1.1.0/24;
              }
          }
      then accept;
      }
  }
```

If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

5. If you are done configuring the device, commit your candidate configuration.

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

To implement this filter, you apply it as an input filter to the `es-0/1/0` logical interface of Gateway A. See Example: Applying an Inbound Traffic Filter to an ES PIC for a Policy Check for details.

SEE ALSO

IPsec Tunnel Traffic Configuration Overview | 219
Example: Applying an Inbound Traffic Filter to an ES PIC for a Policy Check | 226

Example: Applying an Inbound Traffic Filter to an ES PIC for a Policy Check

After you create the inbound firewall filter, apply it to the ES PIC. Here, the inbound firewall filter (`ipsec-decrypt-policy-filter`) is applied on the decrypted packet to perform the final policy check. The IPsec `manual-sa1` SA is referenced at the `[edit interfaces es-1/2/0 unit 0 family inet]` hierarchy level and decrypts the incoming packet.

```
[edit interfaces]
es-1/2/0 {
    unit 0 {
        tunnel {
            source 10.5.5.5;                         # tunnel source address
            destination 10.6.6.6;                  # tunnel destination address
        }
        family inet {
            filter {
                input ipsec-decrypt-policy-filter;
            }
        ipsec-sa manual-sa1; # SA name applied to packet
            address 10.1.1.8/32 { # local interface address inside local VPN
                destination 10.2.2.254; # destination address inside remote VPN
            }
    }
}
```

The Packet Forwarding Engine directs IPsec packets to the ES PIC. It uses the packet's SPI, protocol, and destination address to look up the SA configured on one of the ES interfaces. The IPsec `manual-sa1` SA is referenced at the `[edit interfaces es-1/2/0 unit 0 family inet]` hierarchy level and is used to decrypt the incoming packet. When the packets are processed (decrypted, authenticated, or both), the input firewall filter (`ipsec-decrypt-policy-filter`) is applied on the decrypted packet to perform the final policy check. Term1 defines the decrypted (and verified) traffic and performs the required policy check.

NOTE: The inbound traffic filter is applied after the ES PIC has processed the packet, so the decrypted traffic is defined as any traffic that the remote gateway is encrypting and sending to this router. IKE uses this filter to determine the policy required for a tunnel. This policy is used during the negotiation with the remote gateway to find the matching SA configuration.

SEE ALSO

IPsec Tunnel Traffic Configuration Overview | **219**

ES Tunnel Interface Configuration for a Layer 3 VPN

To configure an ES tunnel interface for a Layer 3 VPN, you need to configure an ES tunnel interface on the provider edge (PE) router and on the customer edge (CE) router. You also need to configure IPsec on the PE and CE routers.

SEE ALSO

IPsec Tunnel Traffic Configuration Overview | **219**

Tracing Operations for Security Services

IN THIS SECTION

Configuring Tracing Operations

To configure trace options for security services, specify flags using the `traceoptions` statement:

```
[edit security]
traceoptions {
    file filename <files number> <size size>;
    flag all;
    flag database;
    flag general;
    flag ike;
    flag parse;
    flag policy-manager;
    flag routing-socket;
    flag timer;
}
```

You can include these statements at the following hierarchy levels:

- `[edit security]`
- `[edit services ipsec-vpn]`

You can specify one or more of the following security tracing flags:

- `all`—Trace all security events
- `database`—Trace database events
- `general`—Trace general events
- `ike`—Trace IKE module processing
- `parse`—Trace configuration processing

- policy-manager—Trace policy manager processing
- routing-socket—Trace routing socket messages
- timer—Trace internal timer events

SEE ALSO

Configuring Tracing Operations for IPsec Events for Adaptive Services PICs | 229

Security Associations Overview | 63

Configuring Tracing Operations for IPsec Events for Adaptive Services PICs

To configure trace options to trace IPsec events for Adaptive Services PICs, include the following statements at the [edit services ipsec-vpn] hierarchy level:

```
[edit services ipsec-vpn]
traceoptions {
    file filename <files number> <size size>;
    flag all;
    flag database;
    flag general;
    flag ike;
    flag parse;
    flag policy-manager;
    flag routing-socket;
    flag timer;
}
```

Trace option output is recorded in the **/var/log/kmd** file.

You can specify one or more of the following security tracing flags:

- all—Trace all security events
- database—Trace database events
- general—Trace general events
- ike—Trace IKE module processing
- parse—Trace configuration processing
- policy-manager—Trace policy manager processing

- `routing-socket`—Trace routing socket messages

- `timer`—Trace internal timer events

SEE ALSO

> Configuring Tracing Operations | **228**

CHAPTER 10

Configuring SSH and SSL Router Access

IN THIS CHAPTER

Configure SSH Known Host Keys for Secure Copying of Data

IN THIS SECTION

Secure Shell (*SSH*) uses *encryption* algorithms to generate a host, server, and session key system that ensures secure data transfer. You can configure SSH host keys to support secure copy (*SCP*) as an alternative to *FTP* for the background transfer of data such as configuration archives and event logs. To configure SSH support for SCP, you must complete the following tasks:

- Specify SSH known hosts by including hostnames and host key information in the Routing Engine configuration hierarchy.
- Set an SCP URL to specify the host from which to receive data. Setting this attribute automatically retrieves SSH host key information from the SCP server.
- Verify that the host key is authentic.

- Accept the secure connection. Accepting this connection automatically stores host key information in the local host key database. Storing host key information in the configuration hierarchy automates the secure handshake and allows background data transfer using SCP.

Tasks to configure SSH host keys for secure copying of data are:

Configure SSH Known Hosts

To configure SSH known hosts, include the `host` statement, and specify hostname and host key options for trusted servers at the `[edit security ssh-known-hosts]` hierarchy level:

```
[edit security ssh-known-hosts]
host corporate-archive-server  {
    dsa-key key;
}
host archive-server-url {
    rsa-key key;
}
host server-with-ssh-version-1 {
    rsa1-key key;
}
```

Host keys are one of the following:

- `dsa-key` *key*—Base64 encoded Digital Signature Algorithm (DSA) key for SSH version 2.

- `ecdsa-sha2-nistp256-key` *key*—Base64 encoded ECDSA-SHA2-NIST256 key.

- `ecdsa-sha2-nistp384-key` *key*—Base64 encoded ECDSA-SHA2-NIST384 key.

- `ecdsa-sha2-nistp521-key` *key*—Base64 encoded ECDSA-SHA2-NIST521 key.

- `ed25519-key` *key*—Base64 encoded ED25519 key.

- `rsa-key` *key*—Base64 encoded public key algorithm that supports encryption and digital signatures for SSH version 1 and SSH version 2.

- `rsa1-key` *key*—Base64 encoded RSA public key algorithm, which supports encryption and digital signatures for SSH version 1.

Configure Support for SCP File Transfer

To configure a known host to support background SCP file transfers, include the `archive-sites` statement at the `[edit system archival configuration]` hierarchy level.

```
[edit system archival configuration]
archive-sites {
    scp://username<:password>@host<:port>/url-path;
}
```

NOTE: When specifying a URL in a Junos OS statement using an IPv6 host address, you must enclose the entire URL in quotation marks (" ") and enclose the IPv6 host address in brackets ([ ]). For example, **“scp://*username*<:*password*>@[*host*]<:*port*>/*url-path*”**;

Setting the `archive-sites` statement to point to an SCP URL triggers automatic host key retrieval. At this point, Junos OS connects to the SCP host to fetch the SSH public key, displays the host key message digest or fingerprint as output to the console, and terminates the connection to the server.

```
user@host# set system archival configuration archive-sites “<scp-url-path>”
The authenticity of host <my-archive-server (<server-ip-address>)> can’t be established. RSA key
fingerprint is <ascii-text key>. Are you sure you want to continue connecting (yes/no)?
```

To verify that the host key is authentic, compare this fingerprint with a fingerprint that you obtain from the same host using a trusted source. If the fingerprints are identical, accept the host key by entering **yes** at the prompt. The host key information is then stored in the Routing Engine configuration and supports background data transfers using SCP.

Update SSH Host Key Information

IN THIS SECTION

Typically, SSH host key information is automatically retrieved when you set a URL attribute for SCP using the `archival configuration archive-sites` statement at the `[edit system]` hierarchy level. However, if you need to manually update the host key database, use one of the following methods.

Retrieve Host Key Information Manually

To manually retrieve SSH public host key information, configure the `fetch-from-server` option at the `[edit security ssh-known-hosts]` hierarchy level. You must to specify the host from which to retrieve the SSH public key.

```
user@host# set security ssh-known-hosts fetch-from-server <hostname>
```

Import Host Key Information from a File

To manually import SSH host key information from a **known_hosts** file, include the `load-key-file` option at the `[edit security ssh-known-hosts]` hierarchy level. You must specify the path to the file from which to import host key information.

```
user@host# set security ssh-known-hosts load-key-file /var/tmp/known-hosts
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
18.3R1	Starting in Junos OS Release 18.3R1, the `ssh-dss` and `ssh-dsa` hostkey algorithms are deprecated— rather than immediately removed—to provide backward compatibility and a chance to bring your configuration into compliance with the new configuration.

Importing SSL Certificates for Junos XML Protocol Support

NOTE: For FIPS mode, the digital security certificates must be compliant with the National Institute of Standards and Technology (NIST) SP 800-131A standard.

A Junos XML protocol client application can use one of four protocols to connect to the Junos XML protocol server on a router or switch: clear-text (a Junos XML protocol-specific protocol for sending unencrypted text over a TCP connection), *SSH*, *SSL*, or *Telnet*. For clients to use the SSL protocol, you must copy an X.509 authentication certificate onto the router or switch, as described in this topic. You must also include the `xnm-ssl` statement at the `[edit system services]` hierarchy level.

NOTE: The `xnm-ssl` statement does not apply to standard IPsec services.

After obtaining an X.509 authentication certificate and private key, copy it to the router or switch by including the `local` statement at the `[edit security certificates]` hierarchy level:

```
[edit security certificates]
local certificate-name {
    load-key-file (filename | url);
}
```

certificate-name is a name you choose to identify the certificate uniquely (for example, `Junos XML protocol-ssl-client-`*hostname*, where *hostname* is the computer where the client application runs).

filename is the pathname of the file on the local disk that contains the paired certificate and private key (assuming you have already used another method to copy them to the router's or switch's local disk).

url is the URL to the file that contains a paired certificate and private key (for instance, on the computer where the Junos XML protocol client application runs).

NOTE: The CLI expects the private key in the *URL-or-path* file to be unencrypted. If the key is encrypted, the CLI prompts you for the passphrase associated with it, decrypts it, and stores the unencrypted version.

The `load-key-file` statement acts as a directive that copies the contents of the certificate file into the configuration. When you view the configuration, the CLI displays the string of characters that constitute the private key and certificate, marking them as `SECRET-DATA`. The `load-key-file` keyword is not recorded in the configuration.

RELATED DOCUMENTATION

Configuring SSH Host Keys for Secure Copying of Data

Configuring clear-text or SSL Service for Junos XML Protocol Client Applications

Configuring IPsec for FIPS Mode

IN THIS SECTION

Configuring IPsec for Enabling Internal Communications Between Routing Engines for Junos OS in FIPS Mode

IN THIS SECTION

In a Junos OS in *FIPS mode* environment, routers with two Routing Engines must use *IPsec* for internal communication between the Routing Engines. You configure internal IPsec after you install the Junos OS in FIPS mode. You must be a *Crypto Officer* to configure internal IPsec.

NOTE: You cannot configure DES-based IPsec SAs in Junos OS in FIPS mode. The internal IPsec SAs use HMAC-SHA1-96 authentication and 3DES-CBC encryption.

Manual SAs require no negotiation. All values, including the keys, are static and specified in the configuration. Manual SAs statically define the SPI values, algorithms, and keys to be used, and require matching configurations on both ends of the tunnel. Each peer must have the same configured options for communication to take place.

NOTE: When the switch is in FIPS mode, you cannot use the `commit synchronize` command until you have established an IPsec SA on each Routing Engine.

As Crypto Officer, you configure an internal IPsec SA for communication between Routing Engines by creating an SA on each Routing Engine with the following statements at the `[security]` hierarchy level:

To configure internal IPsec, include the `security-association` statement at the `[security]` hierarchy level. You can configure parameters, such as the direction in which the manual IPsec SAs must be applied, the SPI value that uniquely identifies the SA to use at the receiving Routing Engine, and the IPsec key that defines the authentication and encryption keys for the manual IPsec SA.

```
[ security]
ipsec {
    internal {
        security-association {
            manual {
                direction (bidirectional | inbound | outbound) {
                    protocol esp;
                    spi spi-value;
                    encryption {
                        algorithm  (hmac-sha1-96 | hmac-sha2-256);
                        key (ascii-text ascii-text-string | hexadecimal hexadecimal-number);
                    }
                }
            }
        }
    }
}
```

Tasks for configuring internal IPsec for Junos-FIPS are the following. You can configure the direction in which the manual IPsec SAs must be applied, the SPI value that uniquely identifies the SA to use at the receiving Routing Engine, and the IPsec key that defines the authentication and encryption keys for the manual IPsec SA.

Configuring the SA Direction

To configure the IPsec SA direction in which manual SAs of the IPsec tunnels must be applied, include the `direction` statement at the `[security ipsec internal security-association manual]` hierarchy level:

```
direction (bidirectional | inbound | outbound);
```

The value can be one of the following:

- `bidirectional`—Apply the same *SA* values in both directions between Routing Engines.

- inbound—Apply these SA properties only to the inbound IPsec tunnel.
- outbound—Apply these SA properties only to the outbound IPsec tunnel.

If you do not configure the SA to be bidirectional, you must configure SA parameters for IPsec tunnels in both the inbound and outbound directions. The following example uses an inbound and outbound IPsec *tunnel*:

NOTE: We recommend that you do not use the IPsec keys as ASCII keys for Junos OS in FIPS mode. Instead, you must use the IPsec keys as hexadecimal keys for maximum key strength.

```
[security]
ipsec {
    internal {
        security-association {
            manual {
                direction inbound {
                    protocol esp;
                    spi 512;
                    encryption {
                        algorithm 3des-cbc;
                        key hexadecimal 309fc4be20f04e53e011b00744642d3fe66c2c7c;
                    }
                }
                direction outbound {
                    protocol esp;
                    spi 513;
                    encryption {
                        algorithm 3des-cbc;
                        key hexadecimal b0344c61d8db38535ca8afceaf0bf12b881dc200c9833da7;
                    }
                }
            }
        }
    }
}
```

Configuring the IPsec SPI

A security parameter index (*SPI*) is a 32-bit index that identifies a security context between a pair of Routing Engines. To configure the IPsec SPI value, include the `spi` statement at the `[security ipsec internal security-association manual direction]` hierarchy level:

```
spi value;
```

The value must be from 256 through 16,639.

Configuring the IPsec Key

NOTE: We recommend that you do not use the IPsec keys as ASCII keys for Junos OS in FIPS mode. Instead, you must use the IPsec keys as hexadecimal keys for maximum key strength.

The distribution and management of keys are critical to using VPNs successfully. You must configure the ASCII text key values for authentication and encryption. To configure the ASCII text key, include the `key` statement at the `[security ipsec internal security-association manual direction encryption]` hierarchy level:

```
key (ascii-text ascii-text-string | hexadecimal hexadecimal-string);
```

For this type of SA, both keys must be preshared hexadecimal values, and each requires a specific cryptographic algorithm:

- Authentication algorithm
  - HMAC-SHA1-96 (40 characters)
  - HMAC-SHA2-256 (64 characters)
- Encryption algorithm
  - 3DES-CBC (48 characters)

You must enter the key hexadecimal value twice and the strings entered must match, or the key will not be set. The hexadecimal key is never displayed in plain text. We recommend that you use the IPsec keys as hexadecimal keys for maximum key strength and not as ASCII keys for Junos OS in FIPS mode.

RELATED DOCUMENTATION

Example: Configuring Internal IPsec | **240**

Example: Configuring Internal IPsec

Configure a bidirectional IPsec SA with an SPI value of 512 and a key value conforming to the FIPS 140-2 rules:

```
[edit security]
ipsec {
    internal {
        security-association {
            manual {
                direction bidirectional {
                    protocol esp;
                    spi 512;
                    encryption {
                        algorithm 3des-cbc;
                        key ascii-text “$ABC123”;
                    }
                }
            }
        }
    }
}
```

SEE ALSO

Configuring IPsec for Enabling Internal Communications Between Routing Engines for Junos OS in FIPS Mode | **236**

4

PART

Trusted Platform Module

Trusted Platform Module Overview

SUMMARY

Learn about trusted platform module (TPM), use of TPM-based certificates and benefits.

IN THIS SECTION

A Trusted Platform Module (TPM) is a hardware component that ensures your device is running optimally. It serves as a secure storage mechanism for essential security artifacts such as cryptographic keys and digital certificates.

TPM-Based Certificates

IN THIS SECTION

Starting in Junos OS Release 24.2R1, you can use the TPM based certificate with SRX1600, SRX2300, SRX4120 and SRX4300 Series Firewalls.

The firewall uses the TPM-based certificate to ensure secure identification of the device. The firewall has burnt-in idev-id certificate built on TPM. The idev-id certificate provides the firewall's JNPR serial number and model, proving that the firewall was manufactured in a Juniper facility. Hence, TPM certificate is a secure way for a Juniper device to prove its identity.

Benefits of TPM-Based Certificates

- Provides trust. Helps to establish advanced security in an insecure digital world.
- Provides confidentiality. Data sent is encrypted and only visible to the server and client.

- Provides integrity. Ensures that the data has not been modified during the transfer.

How Does a Conventional SSL/TLS Certificate Work?

Secure Sockets Layer (SSL) is a protocol that allows encryption. It helps to secure and authenticate communications between a client and a server. It can also secure email, VoIP, and other communications over unsecured networks. SSL is also called as Transport Layer Security (TLS).

In unsecured HTTP connections, hackers can easily intercept messages between client and server. SSL certificates use a public/private keypair system to initiate the HTTPS protocol. Hence, SSL certificates enable secure connections for users and clients to connect. SSL/TLS works through:

- Secure communication that begins with a TLS handshake. The two communicating parties open a secure connection and exchange the public key.

- During the TLS handshake, the two parties generate session keys. The session keys encrypt and decrypt all communications after the TLS handshake.

- Different session keys encrypt communications in each new session.

- TLS ensures that the user on the server side, or the website the user is interacting with, is who they claim to be.

- TLS also ensures that data has not been altered, since a message authentication code (MAC) is included with transmissions.

When a signed SSL certificate secures a website, it proves that the organization has verified and authenticated its identity with the trusted third party. When the browser trusts the CA, the browser now trusts that organization’s identity too.

The easiest way to check if the website has an SSL installed is to see if the website URL starts with “HTTPS:”. If the website has an SSL certificate installed on the server, click the padlock icon in the address bar to view the certificate information.

How Firewall Manages the TPM-Based Certificates Using PKI

When you use applications such as the advanced anti-malware detection (AAMWD) using Juniper ATP Cloud, you can use the TPM-based certificate for attestation, allowing the applications to verify the legitimacy of your device. The firewall manages the TPM-based certificates using the PKID process. Note the following when using PKID process for TPM-based certificates:

- The firewall loads the TPM-based certificate using the PKID process during the device start and restart operations.

- The device loads the certificate and the private key handle against the TPM based certificate ID, referred as `idev-id` certificate ID, from your device's local certificate list. To view the TPM-based certificate ID, referred as `idev-id`, use the `show security pki node-local local-certificate certificate-id idev-id` command.

- You should not use the command `request security pki node-local local-certificate verify certificate-id idev-id` to verify the `idev-id` certificate ID. The verification doesn't go through as the CA certificate for the `idev-id` certificate ID is not available on your firewall. You'll notice an error message `local certificate verification can't performed for IDev-ID certificate as the CA cert for the same is not available` when you try to verify using the command.

How AAMWD and SSL/TLS Use TPM-Based Certificates

Applications such as the AAMW detection (AAMWD) using Juniper ATP Cloud on SRX Series Firewall must use the TPM burnt-in certificate for all its device identification and authentication instead of conventional certificates. This helps to establish a secure client authenticated SSL connection with the cloud server using the new TPM certificate. This applies to both control and data plane (SSL-I)/TLS connections established by AAMWD. The cloud server confirms the authenticity of TPM private key using TPM public key that is shared in SSL handshake. The device identification information is part of the shared TPM certificate.

When you configure SSL-Initiation (SSL-I) profile, there's a requirement on PKID side to load the TPM certificate and private key handle on start/restart against a certain certificate ID. You can use this certificate to configure the SSL-I profile. This certificate can be used by AAMWD for TLS connections. SSL-I need changes on data plane side to use TPM chip for sign/verify purpose for AAMWD TLS connections. SSL-I provide the SSL client functionality to ATP Cloud with client authentication. SSL-I mode needs to be supported with TPM certificate/private key. Earlier, SSL-I use the file system local certificate and private key.

Starting in Junos OS Release 24.2R1, you can use SSL-I in two modes:

- SSL-I with TPM certificate/keys
- SSL-I with file-system certificate/key

You can configure the `tpm` option using the `set services ssl initiation profile profile-name crypto-hardware-offload` command.

SEE ALSO

No Link Title

No Link Title

5

PART

MACsec

CHAPTER 11

Understanding MACsec

IN THIS CHAPTER

Understanding Media Access Control Security (MACsec)

IN THIS SECTION

Understanding Media Access Control Security (MACsec)

IN THIS SECTION

Media Access Control security (MACsec) provides point-to-point security on Ethernet links. MACsec is defined by IEEE standard 802.1AE. You can use MACsec in combination with other security protocols, such as IP Security (IPsec) and Secure Sockets Layer (SSL), to provide end-to-end network security.

MACsec is capable of identifying and preventing most security threats, including denial of service, intrusion, man-in-the-middle, masquerading, passive wiretapping, and playback attacks. MACsec secures an Ethernet link for almost all traffic, including frames from the Link Layer Discovery Protocol (LLDP), Link Aggregation Control Protocol (LACP), Dynamic Host Configuration Protocol (DHCP), Address Resolution Protocol (ARP), and other protocols that are not typically secured on an Ethernet link because of limitations with other security solutions.

Use Feature Explorer to confirm platform and release support for MACsec.

Review the "Platform-Specific MACsec Behavior" section for notes related to your platform.

How MACsec Works

When MACsec is enabled on a point-to-point Ethernet link, the link is secured after matching security keys are exchanged and verified between the interfaces at each end of the link. The key can be configured manually, or can be generated dynamically, depending on the security mode used to enable MACsec. For more information on MACsec security modes, see "MACsec Security Modes" on page 248.

MACsec uses a combination of data integrity checks and encryption to secure traffic traversing the link:

Data integrity
MACsec appends an 8-byte header and a 16-byte tail to all Ethernet frames traversing the MACsec-secured link. The header and tail are checked by the receiving interface to ensure that the data was not compromised while traversing the link. If the data integrity check detects anything irregular about the traffic, the traffic is dropped.

Encryption
Encryption ensures that the data in the Ethernet frame cannot be viewed by anybody monitoring traffic on the link. MACsec encryption is optional and user-configurable. You can enable MACsec to ensure the data integrity checks are performed while still sending unencrypted data “in the clear” over the MACsec-secured link, if desired.

> **NOTE**: When MACsec is enabled on a logical interface, VLAN tags are not encrypted. All the VLAN tags configured on the logical interface enabled for MACsec are sent in clear text.

Connectivity Associations

MACsec is configured in connectivity associations. A connectivity association is a set of MACsec attributes that interfaces use to create two secure channels, one for inbound traffic and one for outbound traffic. The secure channels are responsible for transmitting and receiving data on the MACsec-secured link.

The secure channels are automatically created. They do not have any user-configurable parameters. All configuration is done within the connectivity association but outside of the secure channels.

The connectivity association must be assigned to a MACsec-capable interface on each side of the point-to-point Ethernet link. If you want to enable MACsec on multiple Ethernet links, you must configure MACsec individually on each link. Other user-configurable parameters, such as MAC address or port, must also match on the interfaces on each side of the link to enable MACsec.

MACsec Security Modes

MACsec can be enabled using one of the following security modes:

- Static CAK mode
- Dynamic CAK mode

BEST PRACTICE: Static CAK mode is recommended for links connecting switches or routers. Static CAK mode ensures security by frequently refreshing to a new random security key and by sharing only the security key between the two devices on the MACsec-secured point-to-point link.

Static CAK Mode

When you enable MACsec using static CAK mode, two security keys—a connectivity association key (CAK) that secures control plane traffic and a randomly-generated secure association key (SAK) that secures data plane traffic—are used to secure the link. Both keys are regularly exchanged between both devices on each end of the point-to-point Ethernet link to ensure link security.

You initially establish a MACsec-secured link using a pre-shared key when you are using static CAK security mode to enable MACsec. A pre-shared key includes a connectivity association name (CKN) and its own CAK. The CKN and CAK are configured by the user in the connectivity association and must match on both ends of the link to initially enable MACsec.

Once matching pre-shared keys are successfully exchanged, the MACsec Key Agreement (MKA) protocol is enabled. The MKA protocol is responsible for maintaining MACsec on the link, and decides which switch on the point-to-point link becomes the key server. The key server then creates an SAK that is shared with the switch at the other end of the point-to-point link only, and that SAK is used to secure all data traffic traversing the link. The key server will continue to periodically create and share a randomly-created SAK over the point-to-point link for as long as MACsec is enabled.

NOTE: If the MACsec session is terminated due to a link failure, when the link is restored, the MKA key server elects a key server and generates a new SAK.

NOTE: The switches on each end of a MACsec-secured switch-to-switch link must either both be using Junos OS Release 14.1X53-D10 or later, or must both be using an earlier version of Junos, in order to establish a MACsec-secured connection when using static CAK security mode.

Dynamic CAK Mode

In dynamic CAK mode, the peer nodes on the MACsec link generate the security keys dynamically as part of the 802.1X authentication process. The peer nodes receive MACsec key attributes from the RADIUS server during authentication and use these attributes to dynamically generate the CAK and the CKN. Then they exchange the keys to create a MACsec-secured connection.

Dynamic CAK mode provides easier administration than static CAK mode, because the keys do not need to be configured manually. Also, the keys can be centrally-managed from the RADIUS server.

You can use dynamic CAK mode to secure a switch-to-host link or a link that connects switches or routers. On a switch-to-host link, the switch is the 802.1X authenticator and the host is the supplicant. On a link connecting switches or routers, the devices must act as both authenticator and supplicant so they can authenticate each other.

Dynamic CAK mode relies on certificate-based validation using Extensible Authentication Protocol-Transport Layer Security (EAP-TLS). The RADIUS server and switching devices must use EAP-TLS and public key infrastructure to support MACsec in dynamic CAK mode.

MACsec in a Virtual Chassis

MACsec can be configured on supported switch interfaces when those switches are configured in a *Virtual Chassis* or Virtual Chassis Fabric (VCF), including when MACsec-supported interfaces are on member switches in a mixed Virtual Chassis or VCF that includes switch interfaces that do not support MACsec. MACsec, however, cannot be enabled on Virtual Chassis ports (VCPs) to secure traffic travelling between member switches in a Virtual Chassis or VCF.

MACsec Limitations

- All types of Spanning Tree Protocol frames cannot currently be encrypted using MACsec.
- MACsec traffic drops are expected during GRES switchover.

Supported Cipher Suites for SRX Platforms

The following tables show cipher suite support on SRX platforms.

Table 6: Supported Ciphers (IKE) on SRX Platforms

Ciphers (IKE)	SRX1500	SRX1600	SRX2300, SRX4120	SRX4100, SRX4200 and SRX4300	SRX4600, and SRX4700	SRX5400, SRX5600, and SRX5800
AES-128-GCM	Yes	Yes	Yes	Yes	Yes	Yes
AES-192-GCM	No	No	No	No	No	No
AES-256-GCM	Yes	Yes	Yes	Yes	Yes	Yes

Table 7: Supported Ciphers (IPSec) on SRX Platforms

Ciphers (IPSec)	SRX1500	SRX1600	SRX2300, SRX4120	SRX4100, SRX4200 and SRX4300	SRX4600, and SRX4700	SRX5400, SRX5600, and SRX5800
AES-128-GCM	Yes	Yes	Yes	Yes	Yes	Yes
AES-192-GCM	Yes	Yes	Yes	Yes	Yes	Yes
AES-256-GCM	Yes	Yes	Yes	Yes	Yes	Yes

Platform-Specific MACsec Behavior

Use Feature Explorer to confirm platform and release support for specific features.

Use the following table to review platform-specific behavior for your platform.

Platform	Difference
- EX Series	EX Series device that supports MACsec might not work properly on PHY84756 1G SFP ports if auto negotiation is enabled and MACsec is configured on those ports. As a workaround, configure `no- auto-negotiation` on PHY84756 1G SFP ports before configuring MACsec on those ports.

SEE ALSO

Configuring MACsec | 259

cipher-suite

MACsec Licensing and Software Requirements

IN THIS SECTION

MACsec Feature Licenses

A feature license is required to configure MACsec on EX Series and QFX series switches, with the exception of the QFX10000-6C-DWDM and QFX10000-30C-M line cards. If the MACsec licence is not installed, MACsec functionality cannot be activated.

To purchase a feature license for MACsec, contact your Juniper Networks sales representative (https://www.juniper.net/us/en/contact-us/sales-offices). The Juniper sales representative will provide you with a feature license file and a license key. You will be asked to supply the chassis serial number of your switch; you can obtain the serial number by running the `show chassis hardware` command.

The MACsec feature license is an independent feature license. The enhanced feature licenses (EFLs) or advanced feature licenses (AFLs) that must be purchased to enable some features on EX Series or QFX Series switches cannot be purchased to enable MACsec.

For a Virtual Chassis deployment, two MACsec license keys are recommended for redundancy—one for the device in the primary role and the other for the device in the backup role. Two MACsec licenses may be required per Virtual Chassis Fabric (VCF) and per Virtual Chassis (VC), depending on model and configuration. See the licensing documents below for platform and feature specific details.

- *Licenses for EX Series*
- *Licenses for QFX Series*
- *Legacy Licenses for QFX5200-32C Switch*

A MACsec feature license is installed and maintained like any other switch license. See Managing Licenses for EX Series Switches (CLI Procedure) or Adding New Licenses (CLI Procedure) for more detailed information on configuring and managing your MACsec software license.

MACsec Software Requirements for MX Series Routers

Following are some of the key software requirements for MACsec on MX Series Routers:

> **NOTE**: A feature license is not required to configure MACsec on MX Series routers with the enhanced 20-port Gigabit Ethernet MIC (model number MIC-3D-20GE-SFP-E).

MACsec is supported on MX Series routers with MACsec-capable interfaces.

MACsec supports 128 and 256-bit cipher-suite with and without extended packet numbering (XPN).

MACsec supports MACsec Key Agreement (MKA) protocol with Static-CAK mode using preshared keys.

MACsec supports a single connectivity-association (CA) per physical port or physical interface.

Starting in Junos OS Release 20.3R1, you can configure Media Access Control Security (MACsec) at the logical interface level on the MPC7E-10G line card. This configuration enables multiple MACsec Key Agreement (MKA) sessions on a single physical port. VLAN tags are transmitted in clear text, which allows intermediate switches that are MACsec-unaware to switch the packets based on the VLAN tags.

Starting with Junos OS Release 15.1, MACsec is supported on member links of an aggregated Ethernet (`ae-`) interface bundle, and also regular interfaces that are not part of an interface bundle.

Starting with Junos OS Release 17.3R2, MACsec supports 256-bit cipher-suite GCM-AES-256 and GCM-AES-XPN-256 on MX10003 routers with the modular MIC (model number-JNP-MIC1-MACSEC).

Starting in Junos OS Release 18.4R2, the MIC-MACSEC-20GE MIC provides 256-bit cipher-suite GCM-AES-256 and GCM-AES-XPN-256. The MIC-MACSEC-20GE MIC supports MACsec on both twenty 1-

Gigabit Ethernet SFP ports and on two 10-Gigabit Ethernet SFP+ ports in the following hardware configurations:

- Installed directly on the MX80 and MX104 routers
- Installed on MPC1, MPC2, MPC3, MPC2E, MPC3E, MPC2E-NG, and MPC3E-NG line cards on the MX240, MX480, and MX960 routers

Refer *Interface Naming Conventions for MIC-MACSEC-20GE* and *Port Speed for Routing Devices* for more information.

MACsec Software Image Requirements for EX Series and QFX Series Switches

Junos OS Release 16.1 and Later

For Junos OS Release 16.1 and later, you must download the standard Junos image to enable MACsec. MACsec is not supported in the limited image.

The standard version of Junos OS software contains encryption and is, therefore, not available to customers in all geographies. The export and re-export of this Junos OS software is strictly controlled under United States export laws. The export, import, and use of this Junos OS software is also subject to controls imposed under the laws of other countries. If you have questions about acquiring this version of your Junos OS software, contact Juniper Networks Trade Compliance group at mailto:compliance_helpdesk@juniper.net.

Junos OS Releases Prior to 16.1

For releases prior to Junos OS Release 16.1, you must download the controlled version of your Junos OS software to enable MACsec. MACsec support is not available in the domestic version of Junos OS software in releases prior to Junos OS Release 16.1.

The controlled version of Junos OS software includes all features and functionality available in the domestic version of Junos OS, while also supporting MACsec. The domestic version of Junos OS software is shipped on all switches that support MACsec, so you must download and install a controlled version of Junos OS software for your switch before you can enable MACsec.

The controlled version of Junos OS software contains encryption and is, therefore, not available to customers in all geographies. The export and re-export of the controlled version of Junos OS software is strictly controlled under United States export laws. The export, import, and use of the controlled version of Junos OS software is also subject to controls imposed under the laws of other countries. If you have questions about acquiring the controlled version of your Junos OS software, contact Juniper Networks Trade Compliance group at mailto:compliance_helpdesk@juniper.net.

Acquiring and Downloading the Junos OS Software

You can identify whether a software package is the standard or controlled version of Junos OS by viewing the package name. A software package for a controlled version of Junos OS is named using the following format:

```
package-name-m.nZx.y-controlled-signed.tgz
```

A software package for a standard version of Junos OS is named using the following format:

```
package-name-m.nZx.y-.tgz
```

To check which version of Junos OS is running on your switch, enter the `show version` command. If the `JUNOS Crypto Software Suite` description appears in the output, you are running the controlled version of Junos OS. If you are running a controlled version of Junos OS, enter the `show system software` command to display the version. The output also shows the version of all loaded software packages.

The process for installing the controlled or standard version of Junos OS software onto your switch is identical to installing any other version of Junos OS software. You must enter the `request system software add` statement to download the Junos OS image, and the `request system reboot` statement to reboot the switch to complete the upgrade procedure.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
23.4R1	Junos OS Release 23.4R1 supports MACsec on 1GbE interfaces on MX304 routers using a QSFP-SFP (QSA) adapter or breakout active optical cable (AOC). Support for MACsec on 1GbE interfaces requires flow control, which is enabled by default.
20.4R1-EVO	Junos OS Evolved Release 20.4R1 introduced support for dynamic power management. MACsec blocks are dynamically powered on and off based on MACsec configuration. When MACsec is configured on an interface, the MACsec block is powered on for that port group. If none of the interfaces in a port group are configured for MACsec, power will bypass the MACsec block. There may be minimal traffic loss during the power block transition.
18.3R1	Starting in Junos OS Release 18.4R2, the MIC-MACSEC-20GE MIC provides 256-bit cipher-suite GCM-AES-256 and GCM-AES-XPN-256.
18.2R1	Starting in Junos OS Release 18.2R1, AES-256 is supported on the EX9200-40XS line card.

15.1	Starting with Junos OS Release 15.1, MACsec is supported on member links of an aggregated Ethernet (ae-) interface bundle, and also regular interfaces that are not part of an interface bundle.

Media Access Control Security (MACsec) over WAN

IN THIS SECTION

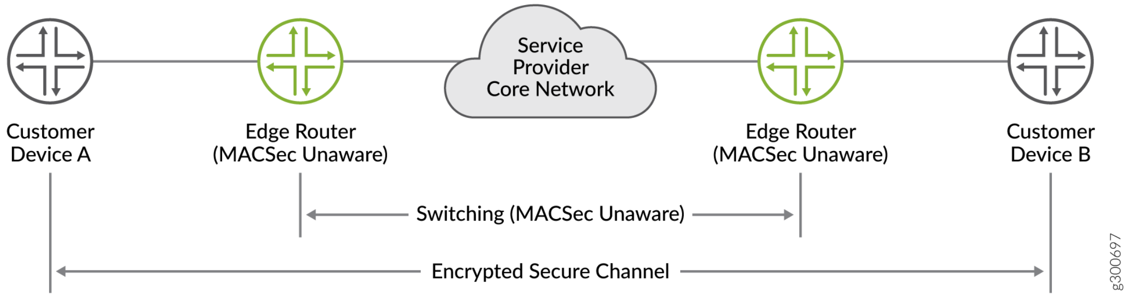
Media Access Control Security (MACsec) is a link layer solution for point-to-point encryption. MACsec can be used to encrypt Layer 2 connections over a service provider WAN to ensure data transmission integrity and confidentiality.

Carrying MACsec over Multiple Hops

To establish a MACsec session, MACsec Key Agreement (MKA) is used to exchange the required keys between the peer nodes. MKA PDUs are transmitted using Extensible Authentication Protocol over LAN (EAPoL) as a transport protocol. EAPoL is a Layer 2 protocol and would normally be locally processed by the switch or router and not propagated further.

In the case where nodes are connected through a service provider network, this presents a challenge. Figure 12 on page 256 shows MACsec carried over a service provider network. MKA must exchange keys between customer devices A and B. The edge routers, or intermediate devices, should not process the EAPoL packets. Instead, they should transparently forward them to the next hop.

Figure 12: MACsec Carried over a Service Provider Network



Service Provider Core Network
Customer Device A
Edge Router (MACSec Unaware)
Edge Router (MACSec Unaware)
Customer Device B
Switching (MACSec Unaware)
Encrypted Secure Channel
g300697

The default destination MAC address for an EAPoL packet is a multicast address. In a service provider network, there might be devices that consume these packets, assuming the packets are meant for them. EAPoL is used by 802.1X and other authentication methods, which might cause the devices to drop the packets, depending on their configuration. This would cause the MKA session to fail. To ensure that the EAPoL packet reaches the correct destination, you can change the destination MAC address so that the service provider network tunnels the packet instead of consuming it.

Configuring VLAN-level MACsec on Logical Interfaces

VLAN-level MACsec allows multiple MKA sessions on a single physical port. This enables service multiplexing with MACsec encryption of point-to-multipoint connections over service provider WANs.

To support VLAN-level MACsec, the MKA protocol packets are sent out with the VLAN tags configured on the logical interface. VLAN tags are transmitted in clear text, which allows intermediate switches that are MACsec-unaware to switch the packets based on the VLAN tags.

When you configure MACsec, you must bind the connectivity association to an interface. To enable VLAN-level MACsec, bind the connectivity association to a logical interface using the following command:

```
[edit security macsec]
user@switch# set interfaces interface-names unit unit-number connectivity-association
connectivity-association-name
```

For complete configuration details, see "Configuring MACsec in Static CAK Mode" on page 260.

Configuring the EAPoL Destination MAC Address for MACsec

MACsec transmits MKA PDUs using EAPoL packets to establish a secure session. By default, EAPoL uses a destination multicast MAC address of 01:80:C2:00:00:03. To prevent these packets from being consumed in a service provider network, you can change the destination MAC address.

To configure the EAPoL destination MAC address, enter one of the following commands.

> **NOTE**: The configuration must match on both peer nodes is order to establish the MACsec session.

- To configure the port access entity multicast address:

```
set security macsec connectivity-association ca-name mka eapol-address pae
```

- To configure a provider bridge multicast address:

```
set security macsec connectivity-association ca-name mka eapol-address provider-bridge
```

- To configure the LLDP multicast address:

```
set security macsec connectivity-association ca-name mka eapol-address lldp-multicast
```

- To configure a unicast destination address:

```
set security macsec connectivity-association ca-name mka eapol-address destination unicast-
mac-address
```

The options are mapped to MAC addresses as follows:

EAPoL Address	MAC Address
pae	01:80:C2:00:00:03

(Continued)

EAPoL Address	MAC Address
`provider-bridge`	01:80:C2:00:00:00
`lldp-multicast`	01:80:C2:00:00:0E
`destination`	*configurable unicast address*

CHAPTER 12

MACsec Examples

IN THIS CHAPTER

Configuring MACsec

IN THIS SECTION

Configuration Overview

Media Access Control Security (MACsec) is an industry-standard security technology that provides secure communication for almost all types of traffic on Ethernet links. MACsec provides point-to-point security on Ethernet links between directly-connected nodes and is capable of identifying and preventing most security threats, including denial of service, intrusion, man-in-the-middle, masquerading, passive wiretapping, and playback attacks. MACsec is standardized in IEEE 802.1AE.

You can configure MACsec to secure point-to-point Ethernet links connecting switches, or on Ethernet links connecting a switch to a host device such as a PC, phone, or server. Each point-to-point Ethernet link that you want to secure using MACsec must be configured independently. You can enable MACsec on switch-to-switch links using dynamic or static connectivity association key (CAK) security mode. Both processes are provided in this document.

For information on configuring MACsec on control and fabric ports of supported SRX Series Firewalls in chassis cluster setup, see Media Access Control Security (MACsec) on Chassis Cluster.

NOTE: On SRX Series Firewalls, you can configure MACsec in routed mode; MACsec is not supported in transparent mode.

Before You Begin

Before enabling MACsec, you must ensure the difference between your interface media maximum transmission unit (MTU) and protocol MTU is large enough to accommodate the additional 32 bytes of MACsec overhead.

For how to configure the interface MTU and protocol MTU, see .

Configuring MACsec in Static CAK Mode

You can enable MACsec using static connectivity association key (CAK) security mode on a point-to-point Ethernet link connecting switches or routers. This can be a switch-to-switch, switch-to-router, or router-to-router link.

BEST PRACTICE: We recommend enabling MACsec using static CAK security mode on links connecting switches or routers. Static CAK security mode ensures security by frequently refreshing to a new random secure association key (SAK) and by only sharing the SAK between the two devices on the MACsec-secured point-to-point link.

When you enable MACsec using static CAK security mode, a preshared key is exchanged between the devices on each end of the point-to-point Ethernet link. The preshared key includes a connectivity association name (CKN) and a connectivity association key (CAK). The CKN and CAK must be manually configured in the connectivity association and must match on both ends of the link to initially enable MACsec.

After the preshared keys are exchanged and verified, the MACsec Key Agreement (MKA) protocol enables MACsec on the link. The MKA is responsible for selecting one of the two devices on the point-to-point link as the key server. The key server then creates a randomized security key that it shares only with the peer device over the MACsec-secured link. The randomized security key enables and maintains MACsec on the point-to-point link. The key server will continue to periodically create and share a randomly-created security key over the point-to-point link for the duration of the MACsec session.

NOTE: If the MACsec session terminates due to a link failure, the MKA key server elects a key server when the link is restored and generates a new SAK.

You enable MACsec using static CAK security mode by configuring a connectivity association on both ends of the link. All configuration is done within the connectivity association but outside of the secure channel. Two secure channels—one for inbound traffic and one for outbound traffic—are automatically created when using static CAK security mode. The automatically-created secure channels do not have any user-configurable parameters. All configuration is done in the connectivity association.

To configure MACsec using static CAK security mode:

1. Create a connectivity association. You can skip this step if you are configuring an existing connectivity association.

   ```
   [edit security macsec]
   user@host# set connectivity-association connectivity-association-name
   ```

   For example, to create a connectivity association named `ca1`, enter:

   ```
   [edit security macsec]
   user@host# set connectivity-association ca1
   ```

2. Configure the MACsec security mode as `static-cak` for the connectivity association:

   ```
   [edit security macsec]
   user@host# set connectivity-association connectivity-association-name security-mode static-cak
   ```

   For example, to configure the MACsec security mode to `static-cak` on connectivity association ca1:

   ```
   [edit security macsec]
   user@host# set connectivity-association ca1 security-mode static-cak
   ```

3. Create the preshared key by configuring the CKN and CAK:

   ```
   [edit security macsec]
   user@host# set connectivity-association connectivity-association-name pre-shared-key ckn
   hexadecimal-number
   ```

```
user@host# set connectivity-association connectivity-association-name pre-shared-key cak
hexadecimal-number
```

The directly-connected peers exchange a preshared key to establish a MACsec-secure link. The pre-shared-key includes the CKN and the CAK, which are hexadecimal numbers. The CKN and the CAK must match on both ends of a link to create a MACsec-secured link.

NOTE: To maximize security, we recommend configuring all digits of a CKN and all digits of a CAK.

If you do not configure all digits of a CKN, or all digits of a CAK, all remaining digits will default to 0. However, you will receive a warning message when you commit the configuration.

After the preshared keys are exchanged and verified by both peers on the link, the MACsec Key Agreement (MKA) protocol enables MACsec. The MKA protocol then elects one of the two directly-connected switches as the key server. The key server then shares a random security with the other device over the MACsec-secure point-to-point link. The key server will continue to periodically create and share a random security key with the other device over the MACsec-secured point-to-point link as long as MACsec is enabled.

To configure a CKN of `37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311` and CAK of `228ef255aa23ff6729ee664acb66e91f` on connectivity association ca1:

```
[edit security macsec]
user@host# set connectivity-association ca1 pre-shared-key ckn
37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311
user@host# set connectivity-association ca1 pre-shared-key cak
228ef255aa23ff6729ee664acb66e91f
```

NOTE: MACsec is not enabled until you attach a connectivity association to an interface. See the final step of this procedure to attach a connectivity association to an interface.

NOTE: In FIPS mode, instead of using `set connectivity-association ca1 pre-shared-key cak` command, you must use the following command:

user@host# **prompt connectivity-association ca1 pre-shared-key cak**

4. (Required on non-EX4300 switches when connecting to EX4300 switches only) Enable SCI tagging:

   ```
   [edit security macsec connectivity-association connectivity-association-name]
   user@host# set include-sci
   ```

   You must enable SCI tagging on a switch that is enabling MACsec on an Ethernet link connecting to an EX4300 or EX4600 switch.

   SCI tags are automatically appended to packets leaving a MACsec-enabled interface on an EX4300 or EX4600 switch, so this option is not available on these switches.

   You should only use this option when connecting a switch to an EX4300 or EX4600 switch, or to a host device that requires SCI tagging. SCI tags are eight octets long, so appending an SCI tag to all traffic on the link adds a significant amount of unneeded overhead.

5. (Optional) Set the MKA key server priority:

   ```
   [edit security macsec connectivity-association connectivity-association-name]
   user@host# set mka key-server-priority priority-number
   ```

   Specifies the key server priority used by the MKA protocol to select the key server. The switch with the lower *priority-number* is selected as the key server.

   The default *priority-number* is 16.

   If the `key-server-priority` is identical on both sides of the link, the MKA protocol selects the interface with the lower MAC address as the key server. Therefore, if this statement is not configured at each end of a MACsec-secured link, the interface with the lower MAC address becomes the key server.

   To change the key server priority to 0 to increase the likelihood that the current device is selected as the key server when MACsec is enabled on the interface using connectivity association `ca1`:

   ```
   [edit security macsec connectivity-association ca1]
   user@host# set mka key-server-priority 0
   ```

   To change the key server priority to 255 to decrease the likelihood that the current device is selected as the key server in connectivity association ca1:

   ```
   [edit security macsec connectivity-association ca1]
   user@host# set mka key-server-priority 255
   ```

6. (Optional) Set the MKA transmit interval:

```
[edit security macsec connectivity-association connectivity-association-name]
user@host# set mka transmit-interval interval
```

The MKA transmit interval setting is the frequency for how often the MACsec Key Agreement protocol data unit (PDU) is sent to the connected device to maintain connectivity on the link. A lower *interval* increases bandwidth overhead on the link; a higher *interval* optimizes MKA protocol communication.

The default *interval* is 2000ms. We recommend increasing the interval to 6000 ms in high-traffic load environments. The transmit interval settings must be identical on both ends of the link when MACsec using static CAK security mode is enabled.

For example, if you wanted to increase the MKA transmit interval to 6000 ms when connectivity association ca1 is attached to an interface:

```
[edit security macsec connectivity-association ca1]
user@host# set mka transmit-interval 6000
```

7. (Optional) Exclude a protocol from MACsec:

```
[edit security macsec connectivity-association connectivity-association-name]
user@host# set exclude-protocol protocol-name
```

When this option is enabled, MACsec is disabled for all packets of the specified protocol that are sent or received on the link. For example, if you did not want Link Level Discovery Protocol (LLDP) to be secured using MACsec:

```
[edit security macsec connectivity-association connectivity-association-name]
user@host# set exclude-protocol lldp
```

When this option is enabled, MACsec is disabled for all packets of the specified protocol—in this case, LLDP—that are sent or received on the link. You can use this option to allow control traffic for some protocols to pass through the MACsec-secured connection without MACsec tags. This provides interoperability with devices, such as IP phones, that do not support MACsec.

8. Assign the connectivity association to an interface:

```
[edit security macsec]
user@host# set interfaces interface-name connectivity-association connectivity-association-
name
```

For example, to assign connectivity association ca1 to interface xe-0/0/1:

```
[edit security macsec]
user@host# set interfaces xe-0/0/1 connectivity-association ca1
```

To assign a connectivity association to a logical interface, use the following command:

```
[edit security macsec]
user@host# set interfaces interface-name unit unit-number connectivity-association
connectivity-association-name
```

NOTE: When assigning a CA to a logical interface, the following limitations apply:

- Configuring a CA on a physical interface and a logical interface is mutually exclusive.
- Logical interfaces with a native VLAN configuration do not support MACsec.
- Logical aggregated interfaces do not support MACsec.

NOTE: On an EX4300 uplink module, the first transceiver plugged into the uplink module determines the PIC mode, as the PIC recognizes the SFP type and programs all of the ports to be either ge- or xe-. Make sure the MACsec configuration on the interface matches the link speed for the uplink module ports.

Assigning the connectivity association to an interface is the final configuration step to enabling MACsec on an interface.

MACsec using static CAK security mode is enabled when a connectivity association on the opposite end of the link is also configured. The connectivity association must contain preshared keys that match on both ends of the link.

SEE ALSO

Configuring MACsec | 259

Configuring MACsec in Dynamic CAK Mode

IN THIS SECTION

In dynamic CAK mode, the peer nodes on the MACsec link generate the security keys dynamically as part of the 802.1X authentication process. You can use dynamic CAK mode to secure a point-to-point link connecting switches or routers. This can be a switch-to-switch, switch-to-router, or router-to-router connection. The devices must act as both authenticator and supplicant for 802.1X authentication so they can authenticate each other.

Dynamic CAK mode provides easier administration than static CAK mode, because the keys do not need to be configured manually. Also, the keys can be centrally-managed from the RADIUS server. However, static CAK mode provides more functionality.

NOTE: Dynamic CAK mode is not supported on logical interfaces.

The following procedure is for configuring dynamic CAK mode on links between switches or routers. To configure dynamic CAK mode on switch-to-host links, see "Configuring MACsec to Secure a Switch-to-Host Link" on page 268.

Before you begin to enable MACsec in dynamic CAK mode, you must configure a RADIUS server. The RADIUS server:

- Must be configured with a server-side certificate.
- Must be using the Extensible Authentication Protocol-Transport Layer Security (EAP-TLS) authentication framework.

For information on configuring the RADIUS server, see *RADIUS Server Configuration for Authentication*.

Configure the Connectivity Association

1. Create a connectivity association. You can skip this step if you are configuring an existing connectivity association.

   ```
   [edit security macsec]
   user@host# set connectivity-association connectivity-association-name
   ```

   For example, to create a connectivity association named `ca1`, enter:

   ```
   [edit security macsec]
   user@host# set connectivity-association ca1
   ```

2. Configure the MACsec security mode as `dynamic` for the connectivity association:

   ```
   [edit security macsec]
   user@host# set connectivity-association connectivity-association-name security-mode dynamic
   ```

   For example, to configure the MACsec security mode to `dynamic` on connectivity association ca1:

   ```
   [edit security macsec]
   user@host# set connectivity-association ca1 security-mode dynamic
   ```

3. Assign the connectivity association to an interface:

   ```
   [edit security macsec]
   user@host# set interfaces interface-name connectivity-association connectivity-association-
   name
   ```

   For example, to assign connectivity association ca1 to interface xe-0/0/1:

   ```
   [edit security macsec]
   user@host# set interfaces xe-0/1/0 connectivity-association ca1
   ```

Configure 802.1X Authentication

Configure 802.1X authentication with EAP-TLS on the interfaces at each end of the point-to-point link. The interfaces must act as both authenticators and supplicants so that the devices can authenticate each other.

1. Configure the interface as an authenticator with the no-reauthentication option:

```
[edit]
user@host# set protocols dot1x authenticator interface interface-name no-reauthentication
```

2. Configure the interface as a supplicant.

```
[edit]
user@host# set protocols dot1x supplicant interface interface-name
```

3. Configure the authentication method as EAP-TLS:

```
[edit]
user@host# set protocols dot1x supplicant interface interface-nameauthentication-method eap-
tls
```

4. Assign a local certificate to the interface:

```
[edit]
user@host# set protocols dot1x supplicant interface interface-namelocal-certificate
certificate-id
```

Configuring MACsec to Secure a Switch-to-Host Link

When configuring MACsec on a switch-to-host link, the MACsec Key Agreement (MKA) keys, which are included as part of 802.1X authentication, are retrieved from a RADIUS server as part of the AAA handshake. A primary key is passed from the RADIUS server to the switch and from the RADIUS server to the host in independent authentication transactions. The primary key is then passed between the switch and the host to create a MACsec-secured connection.

The following requirements must be met in order to enable MACsec on a link connecting a host device to a switch.

The host device:

- must support MACsec and must be running software that allows it to enable a MACsec-secured connection with the switch.

The switch:

- Must support MACsec.
- Must be configured into dynamic connectivity association key (CAK) security mode.

- Must be using 802.1X authentication to communicate with the RADIUS server.

Before you begin to enable MACsec on a switch-to-host link:

- Configure a RADIUS server. The RADIUS server:

  - Must be configured as the user database for 802.1X authentication.

  - Must be using the Extensible Authentication Protocol-Transport Layer Security (EAP-TLS) authentication framework.

  - Must have connectivity to the switch and to the host. The RADIUS server can be multiple hops from the switch or the host.

  See Example: Connecting a RADIUS Server for 802.1X to an EX Series Switch.

- Enable MACsec on the host device.

  The procedures for enabling MACsec on the host device varies by host device, and is beyond the scope of this document.

To configure MACsec using dynamic CAK security mode to secure a switch-to-host Ethernet link:

1. Create a connectivity association. You can skip this step if you are configuring an existing connectivity association.

   ```
   [edit security macsec]
   user@switch# set connectivity-association connectivity-association-name
   ```

   For instance, to create a connectivity association named ca-dynamic1, enter:

   ```
   [edit security macsec]
   user@switch# set connectivity-association ca-dynamic1
   ```

2. Configure the MACsec security mode as dynamic for the connectivity association:

   ```
   [edit security macsec]
   user@switch# set connectivity-association connectivity-association-name security-mode dynamic
   ```

For instance, to configure the MACsec security mode to dynamic on connectivity association ca-dynamic1:

```
[edit security macsec]
user@switch# set connectivity-association ca-dynamic1 security-mode dynamic
```

3. (Optional) Configure the **must-secure** option:

```
[edit security macsec]
user@switch# set connectivity-association connectivity-association-name mka must-secure
```

When the **must-secure** option is enabled, all traffic that is not MACsec-secured that is received on the interface is dropped.

When the **must-secure** option is disabled, all traffic from devices that support MACsec is MACsec-secured while traffic received from devices that do no support MACsec is forwarded through the network.

The **must-secure** option is particularly useful in scenarios where multiple devices, such as a phone and a PC, are accessing the network through the same Ethernet interface. If one of the devices supports MACsec while the other device does not support MACsec, the device that doesn't support MACsec can continue to send and receive traffic over the network—provided the **must-secure** option is disabled—while traffic to and from the device that supports MACsec is MACsec-secured. In this scenario, traffic to the device that is not MACsec-secured must be VLAN-tagged.

4. (Required only if the host device requires SCI tagging) Enable SCI tagging:

```
[edit security macsec connectivity-association connectivity-association-name]
user@switch# set include-sci
```

You should only use this option when connecting a switch to a host that requires SCI tags. SCI tags are eight octets long, so appending an SCI tag to all traffic on the link adds a significant amount of unneeded overhead.

5. (Optional) Set the MKA key server priority:

```
[edit security macsec connectivity-association connectivity-association-name]
user@switch# set mka key-server-priority priority-number
```

Specifies the key server priority used by the MKA protocol to select the key server. The switch with the lower *priority-number* is selected as the key server.

The default *priority-number* is 16. If the **key-server-priority** is identical on both sides of the point-to-point link, the MKA protocol selects the interface with the lower MAC address as the key server. Therefore, if this statement is not configured in the connectivity associations at each end of a MACsec-secured point-to-point link, the interface with the lower MAC address becomes the key server.

To change the key server priority to 0 to increase the likelihood that the current device is selected as the key server when MACsec is enabled on the interface using connectivity association ca1:

```
[edit security macsec connectivity-association ca-dynamic1]
user@switch# set mka key-server-priority 0
```

To change the key server priority to 255 to decrease the likelihood that the current device is selected as the key server in connectivity association ca-dynamic1:

```
[edit security macsec connectivity-association ca-dynamic1]
user@switch# set mka key-server-priority 255
```

6. (Optional) Set the MKA transmit interval:

```
[edit security macsec connectivity-association connectivity-association-name]
user@switch# set mka transmit-interval interval
```

The MKA transmit interval setting sets the frequency for how often the MKA protocol data unit (PDU) is sent to the directly connected device to maintain MACsec connectivity on the link. A lower interval increases bandwidth overhead on the link; a higher interval optimizes MKA protocol communication.

The default interval is 2000ms. We recommend increasing the interval to 6000 ms in high-traffic load environments. The transmit interval settings must be identical on both ends of the link.

For instance, if you wanted to increase the MKA transmit interval to 6000 milliseconds when connectivity association ca-dynamic1 is attached to an interface:

```
[edit security macsec connectivity-association ca-dynamic1]
user@switch# set mka transmit-interval 6000
```

7. (Optional) Exclude a protocol from MACsec:

```
[edit security macsec connectivity-association connectivity-association-name]
user@switch# set exclude-protocol protocol-name
```

   For instance, if you did not want Link Level Discovery Protocol (LLDP) to be secured using MACsec:

```
[edit security macsec connectivity-association ca-dynamic1]
user@switch# set exclude-protocol lldp
```

   When this option is enabled, MACsec is disabled for all packets of the specified protocol—in this case, LLDP—that are sent or received on the link. You can use this option to allow control traffic for some protocols to pass through the MACsec-secured connection without MACsec tags. This provides interoperability with devices, such as IP phones, that do not support MACsec.

8. Assign the connectivity association to an interface:

```
[edit security macsec]
user@switch# set interfaces interface-names connectivity-association connectivity-association-
name
```

   Assigning the connectivity association to an interface is the final configuration step to enabling MACsec on an interface. For instance, to assign connectivity association ca-dynamic1 to interface xe-0/0/1:

```
[edit security macsec]
user@switch# set interfaces xe-0/1/0 connectivity-association ca-dynamic1
```

   NOTE: On an EX4300 uplink module, the first transceiver plugged into the uplink module determines the PIC mode, as the PIC recognizes the SFP type and programs all of the ports to be either ge- or xe-. Make sure the MACsec configuration on the interface matches the link speed for the uplink module ports.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
16.1R2	Starting in Junos OS Release 16.1R2, when Media Access Control Security (MACsec) is enabled on an interface, the interface flow control capability is enabled by default, regardless of the configuration that you set using the (`flow-control` | `no-flow-control`) statement at the `[edit interfaces` *`interface- name`* `gigether-options]` hierarchy level. When MACsec is enabled, additional header bytes are added to the packet by the MACsec PHY. With line rate traffic, when MACsec is enabled and flow control is disabled, the pause frames sent by the MACsec PHY are terminated by the MIC's MAC (enhanced 20-port Gigabit Ethernet MICs on MX Series routers) and not transferred to the Packet Forwarding Engine, causing framing errors. Therefore, when MACsec is enabled on an interface, flow control is also automatically enabled on such an interface.
15.1	Starting with Junos OS Release 15.1, you can configure MACsec to secure point-to-point Ethernet links connecting MX Series routers with MACsec-capable MICs, or on Ethernet links connecting a switch to a host device such as a PC, phone, or server.

Configuring Advanced MACsec Features

IN THIS SECTION

Media Access Control Security (MACsec) is an industry-standard security technology that provides secure communication for almost all types of traffic on Ethernet links. MACsec provides point-to-point security on Ethernet links between directly-connected nodes and is capable of identifying and preventing most security threats, including denial of service, intrusion, man-in-the-middle, masquerading, passive wiretapping, and playback attacks. MACsec is standardized in IEEE 802.1AE.

Configure Encryption Options

IN THIS SECTION

Assign an Encryption Algorithm

You can encrypt all traffic entering or leaving the interface using any of the following MACsec encryption algorithms:

- gcm-aes-128—GCM-AES-128 cipher suite without extended packet numbering (XPN) mode
- gcm-aes-256—GCM-AES-256 cipher suite without XPN
- gcm-aes-xpn-128—GCM-AES-XPN_128 cipher suite with XPN mode
- gcm-aes-xpn-256—GCM-AES-XPN_256 cipher suite with XPN mode

If MACsec encryption is enabled and if no encryption algorithm is specified, the default (gcm-aes-128) encryption algorithm is used without XPN mode.

NOTE: We strongly recommend using XPN when using MACsec on 40G and 100G links.

NOTE:

- The encryption algorithms with XPN mode are not supported on MX-series MPC7E-10G routers.
- Only GCM-AES-128 is supported on MIC-3D-20GE-SFP-E and MIC-3D-20GE-SFP-EH.

```
[edit security macsec connectivity-association <varname>connectivity-association-name</varname>]
user@host# set cipher-suite (gcm-aes-128 | gcm-aes-256 | gcm-aes-xpn-128 | gcm-aes-xpn-256)
```

For example, if you wanted to encrypt using the GCM-AES-XPN-128 algorithm in the connectivity association named `ca1`:

```
[edit security macsec connectivity-association ca1] user@host# set cipher-suite gcm-aes-xpn-128
```

Disable Encryption

The default behavior for MACsec is to encrypt traffic traversing the link. You can disable encryption if you want to use MACsec only to authenticate an endpoint and guarantee integrity of the link. This is called integrity-only mode. Integrity-only mode is useful if you need the unencrypted payload to be visible when carrying MACsec over multiple hops.

When you disable encryption, traffic is forwarded across the Ethernet link in clear text. You can view unencrypted data in the Ethernet frame traversing the link when you are monitoring it. The MACsec header is still applied to the frame, however, and all MACsec data integrity checks are run on both ends of the link to ensure the traffic sent or received on the link has not been tampered with and does not represent a security threat.

To disable encryption, use the following command:

```
[edit security macsec connectivity-association connectivity-association-name]
user@host# set no-encryption
```

Configure an Offset

Offset provides an option between full encyption and no encryption. Configue an offset to expose a set number of bytes of the payload and encrypting the rest. This could be used for intermediate load balancing or for load distribution at the host, in the case of switch-to-host links.

The default offset is 0. All traffic in the connectivity association is encrypted when encryption is enabled and an `offset` is not set.

When the offset is set to 30, the IPv4 header and the TCP/UDP header are unencrypted while encrypting the rest of the traffic. When the offset is set to 50, the IPv6 header and the TCP/UDP header are unencrypted while encrypting the rest of the traffic.

You would typically forward traffic with the first 30 or 50 octets unencrypted if a feature needed to see the data in the octets to perform a function, but you otherwise prefer to encrypt the remaining data in the frames traversing the link. Load balancing features, in particular, typically need to see the IP and TCP/UDP headers in the first 30 or 50 octets to properly load balance traffic.

To configure an offset, use the following command:

```
[edit security macsec connectivity-association connectivity-association-name]
user@host# set offset (0 | 30 | 50)
```

For example, if you wanted to set the offset to 30 in the connectivity association named `ca1`:

```
[edit security macsec connectivity-association ca1]
user@host# set offset 30
```

Configuring Preshared Key Hitless Rollover Keychain (Recommended for Enabling MACsec on Router-to-Router Links)

In the MACsec implementation using static connectivity association key (CAK) prior to release 17.4R1, the user is allowed to configure one static CAK for every connectivity association. Whenever CAK configuration changes, the MACsec session is dropped, resetting peer sessions or interrupting the routing protocol.

For increased security and to prevent session drops when the CAK configuration changes, the hitless rollover keychain feature is implemented. In this implementation, a key chain that has the multiple security keys, key names and start times is used. Each key in the keychain has a unique start time. At the next key's start time, a rollover occurs from the current key to the next key, and the next key becomes the current key. With the implementation of the hitless rollover keychain feature, the MACsec Key Agreement (MKA) protocol establishes MACsec sessions successfully without any session drop when the CAK configuration changes.

For a successful MACsec configuration using preshared key (PSK) hitless rollover keychain:

- The keychain names, keys and start time of each key must be the same in both the participating nodes.
- The order of the keychain names, keys and start time must be same in both the participating nodes.
- The time must be synchronized in the participating nodes.

The existing `authentication-key-chains` and `macsec connectivity-association` commands are used for implementing hitless rollover keychain with the addition of two new attributes:

- `key-name`—Authentication key name, and this `key-name` is used as the CKN for MACsec.
- `pre-shared-key-chain`—The preshared connectivity association keychain name.

To secure a router-to-router Ethernet link by using MACsec with PSK hitless rollover keychain configuration:

NOTE: Ensure that you execute the following steps in both the participating nodes in the same order.

1. Synchronize the time in the participating nodes to the same NTP server.

   ```
   user@host# set date ntp servers
   ```

   For instance, to set the date and time as per the NTP server 192.168.40.1, enter:

   ```
   user@host# set date ntp 192.168.40.1
   ```

2. Configure a set of PSKs in a keychain. A keychain consists of a security key, key name, and start time.

   To configure a keychain:

   a. Create the secret password to use. It is a string of hexadecimal digits up to 64 characters long. The password can include spaces if the character string is enclosed in quotation marks. The keychain's secret-data is used as a CAK.

   ```
   [edit]
   user@host# set security authentication-key-chains key-chain key-chain-name key key secret
   secret-data
   ```

   For instance, to create the secret password 0111223344556677889 9aabbccddeeff for the keychain macsec_key_chain and key 1, enter:

   ```
   [edit]
   user@host# set security authentication-key-chains key-chain macsec_key_chain key 1 secret
   01112233445566778899aabbccddeeff
   ```

   b. Configure the authentication key name. It is a string of hexadecimal digits up to 32 characters long.

   ```
   [edit]
   user@host# set security authentication-key-chains key-chain macsec_key_chain key key key-
   name authentication_key_name
   ```

For instance, to create the key name 01112233445566778899aabbccddeefe, enter:

```
[edit]
user@host# set security authentication-key-chains key-chain macsec_key_chain key 1 key-name 01112233445566778899aabbccddeefe
```

c. Configure the time when the preshared rollover keychain starts.

```
[edit]
user@host# set security authentication-key-chains key-chain macsec_key_chain key key start-time “PSK keychain rollover start time"
```

For instance, if you want the key name with 01112233445566778899aabbccddeefe to start rollover at 2017-12-18.20:55:00 +0000, enter:

```
[edit]
user@host# set security authentication-key-chains key-chain macsec_key_chain key 1 start-time "2017-12-18.20:55:00 +0000"
```

3. Associate the newly created keychain with a MACsec connectivity association.

a. Configure the MACsec security mode for the connectivity association.

```
[edit]
user@host# set security macsec connectivity-association connectivity-association-name security-mode security-mode
```

For instance, to configure the connectivity association ca1 with security mode static-cak, enter:

```
[edit]
user@host# set security macsec connectivity-association ca1 security-mode static-cak
```

b. Associate the preshared keychain name with the connectivity association.

```
[edit]
user@host# set security macsec connectivity-association connectivity-association-name pre-shared-key-chain macsec-key-chain-name
```

For instance, if you want to associate the keychain name macsec_key_chain with the connectivity association ca1, enter:

```
[edit security macsec]
user@host# set security macsec connectivity-association ca1 pre-shared-key-chain
macsec_key_chain
```

4. Assign the configured connectivity association with a specified MACsec interface.

```
[edit]
user@host# set security macsec interfaces interface-name  connectivity-association
connectivity-association-name
```

For instance, to assign the connectivity association ca1 to the interface ge-0/0/1:

```
[edit]
user@host# set security macsec interfaces ge-0/0/1 connectivity-association ca1
```

Configuring MACsec Key Agreement Protocol in Fail Open Mode

You can configure fail open mode for MACsec to prevent traffic from being dropped when the MKA session is inactive. This is recommended for service providers that prioritize network availability over information security.

MACsec maintains data integrity by appending a MACsec header to Ethernet frames transmitted on a MACsec-secured link. When the MKA session is active, traffic is allowed on the link only for frames with a MACsec header. When the MKA session is inactive, frames do not receive a MACsec header. All traffic, both ingress and egress, is dropped. The only exception is EAPoL traffic.

You can configure fail open mode using the `should-secure` CLI statement. This allows traffic on the MACsec-secured link even when the MKA session is inactive. Traffic is transmitted as cleartext, without MACsec headers.

To configure the MKA Protocol in Fail Open Mode:

```
[edit security macsec connectivity-association connectivity-association-name]
 user@host# set mka should-secure;
```

Configuring Replay Protection

MACsec assigns an ID number to each packet on a MACsec-secured link. When replay protection is enabled, the receiving interface checks the ID number of all packets that traversed the MACsec-secured link. If a packet arrives out of sequence and the difference between the packet numbers exceeds the replay protection window size, the receiving interface drops the packet.

For example, if the replay protection window size is set to five and a packet assigned the ID of 1006 arrives on the receiving link immediately after the packet assigned the ID of 1000, the packet with ID 1006 is dropped because it falls outside of the replay protection window.

Replay protection is useful for fighting man-in-the-middle attacks. A packet that is replayed by a man-in-the-middle attacker on the Ethernet link will arrive on the receiving link out of sequence, so replay protection helps ensure the replayed packet is dropped instead of forwarded through the network.

> **NOTE**: You can require that all packets arrive in order by setting the replay window size to 0. Replay protection should not be enabled in cases where packets are expected to arrive out of order.

To enable replay protection use the following command:

```
[edit security macsec connectivity-association connectivity-association-name]
user@host# set replay-protect replay-window-size number-of-packets
```

For example, to enable replay protection with a window size of five on connectivity association `ca1`:

```
[edit security macsec connectivity-association ca1]
user@host# set replay-protect replay-window-size 5
```

Configuring Bounded Delay Protection

You can configure bounded delay protection to ensure that a Media Access Control Security (MACsec) frame will not be delivered after a delay of two seconds or more. This ensures that a delay of MACsec frames resulting from a man-in-the-middle attack will not go undetected.

When you configure bounded delay protection, you must also configure replay protection. This is the window during which duplicate and replay packets are allowed. Bounded delay takes precedence over replay protection. You can increase the effectiveness of bounded delay protection by configuring a lower value for the window size.

Before you configure bounded delay protection, you must configure replay protection. See "Configuring Replay Protection" on page 280.

To configure bounded delay protection, use the following command:

```
[edit security macsec connectivity-association connectivity-association-name mka]
user@host# set bounded-delay
```

NOTE: Bounded delay impacts CPU utilization which can degrade performance. We recommend only configuring bounded delay on interfaces on which it is absolutely required.

Configuring MACsec with Fallback PSK

When you enable MACsec using static CAK security mode, a preshared key (PSK) is exchanged between the devices on each end of the point-to-point Ethernet link. The PSK is includes a connectivity association name (CKN) and a connectivity association key (CAK). The PSK must match across devices for a MACsec session to be established. If there is a mismatch, the session will not be established and all packets will be dropped.

You can configure a fallback PSK to prevent traffic loss in case the primary PSK fails to establish a connection. The fallback PSK is used when primary keys do not match for the initial MACsec negotiation.

If a MACsec session has already been established, and the primary PSK is changed on one device but not the other, the resulting mismatch is resolved by using the older primary PSK. The older primary PSK is a temporary key known as the preceding PSK.

With fallback PSK configured, a MACsec session can be secured with one of the following keys:

- Primary PSK (configurable)—The preferred key.
- Fallback PSK (configurable)—Used when the primary PSK fails to establish a MACsec session.
- Preceding PSK (non-configurable)—When a new primary PSK is configured, the old primary PSK becomes the preceding PSK.

The status of the CAK for each key can be either live, active or in-progress. See Table 8 on page 282 for a description of each status.

Table 8: CAK status descriptions

CAK Status	Description
Live	• CAK has been validated by MKA. • MACsec session is live. • SAK is successfully generated using this key. • CAK is used for encryption and decryption of the MACsec session. • MKA hello packets are sent and received for this key at a configured interval.
Active	• CAK has been validated by MKA. • MACsec session is live. • SAK is not generated using this key. • CAK is not used for encryption and decryption of the MACsec session. • MKA hello packets are sent and received for this key at a configured interval.
In-progress	• No valid live or potential peer is found. • The MACsec session is in-progress to find a peer. • MKA hello packets are sent for this key at a configured interval.

A mismatch of keys occurs when a new PSK is configured on one side of the MACsec link and the other side is either misconfigured or not configured with the new key. The fallback behavior depends on which components of the PSK are changed (CAK, CKN, or both). Each mismatch scenario is described below:

- If the CAK is changed, and the CKN remains the same, the existing MACsec session will be disconnected. A new session will be initiated with the old CKN and new CAK value.

- If the CKN is changed, and the CAK remains the same, the old CKN paired with the existing CAK becomes the preceding PSK, and the session will be live with preceding PSK. A new session is initiated with the newly-created CKN and the CAK, which will be in-progress until the peer node is also configured with the same CKN.

- If both the CAK and the CKN are changed, the old CAK+CKN pair becomes the preceding PSK, and the session will be live with the preceding PSK. A new session is initiated with the new CAK+CKN pair, which will be in-progress until the peer node is also configured with the same CAK+CKN.

> **NOTE**: The preceding PSK takes priority over the fallback PSK, so if the session is live with the preceding PSK, the fallback PSK will not take effect. If you want the session to be live with the fallback PSK, you must configure the `disable-preceding-key` statement.

Fallback PSK is supported for preshared keychains. You can configure a fallback PSK along with a preshared key, or with a preshared keychain. The preshared key and preshared keychain are mutually exclusive.

If only a fallback PSK is configured, and there is no primary PSK, both devices attempt to establish a session with the fallback PSK. If the session comes up, the SAK derived from the fallback PSK is used for data traffic encryption. If the established session is broken, the devices continue attempting to reestablish the session and traffic will be dropped until the session is reestablished.

The fallback PSK is configured as part of the connectivity association (CA). The CA can be configured globally for all interfaces or on a per-interface basis, allowing different fallback keys for different interfaces.

To configure the fallback PSK, configure the CAK and the CKN as part of the CA:

```
[edit security macsec connectivity-association ca-name]
user@switch# set fallback-key cak key
user@switch# set fallback-key ckn key-name
```

The following restrictions apply to fallback PSK configuration:

- Fallback CAK and CKN should not match the preshared key CKN and CAK or any key configured in the keychain under the same CA.
- Security mode configuration must be present to configure the fallback key.
- Key length restrictions for the configured cipher suite apply to the fallback CAK and CKN.

Configuring MACsec with GRES

The graceful switchover (GRES) feature enables a switch or router with redundant routing engines to continue forwarding packets, even if one routing engine (RE) fails. You can configure MACsec to provide uninterrupted service during RE switchover.

The MACsec Key Agreement (MKA) protocol maintains the MACsec session between two nodes on a point-to-point MACsec link. The MKA protocol works at the control plane level between the two nodes. One node acts as the key server and generates a secure association key (SAK) to secure the link.

When the local nodes initiates an RE switchover, it sends a request to the remote peer node to suspend the MACsec session at the control plane. At the data plane, traffic continues to traverse the point-to-point link during suspension. The SAK that was programmed prior to suspension remains in use until the switchover is complete. After the switchover, the key server generates a new SAK to secure the link. The key server will continue to periodically create and share a SAK over the link for as long as MACsec is enabled.

To enable GRES for MACsec, you must configure the `suspend-for` statement on the local node so that it sends a suspension request in the event of an RE switchover. You must also configure the node acting as key server to accept suspension requests using the `suspend-on-request` statement. Otherwise, the key server rejects any suspension requests, resulting in termination of the MACsec session.

When you configure the `suspend-for` and `suspend-on-request` statements, you must also configure GRES and nonstop routing.

NOTE: During GRES, the following MACsec features are disabled:

- Primary, fallback, or preceding key switch.
- Keychain key switch.
- SAK rekey timer.

To enable GRES for MACsec, use the following configuration on the local node:

1. Configure graceful switchover.

```
user@host# set chassis redundancy graceful-switchover
```

2. Configure nonstop routing.

```
user@host# set routing-options nonstop-routing
```

3. Configure the node to send suspension requests when initiating RE switchover.

```
user@host# set security macsec connectivity-association ca-name mka suspend-for
```

4. (Key server only.) Configure the node to accept suspension requests.

```
user@host# set security macsec connectivity-association ca-name mka suspend-on-request
```

Example: Configuring MACsec over an MPLS CCC on EX Series Switches

IN THIS SECTION

This example shows how to enable MACsec to secure sensitive traffic traveling from a user at one site to a user at another site over a basic MPLS CCC.

Requirements

This example uses the following hardware and software components:

- Three EX4550 switches used as the PE and provider switches in the MPLS network
- One EX4550 switch used as the CE switch connecting site A to the MPLS network
- One EX4200 switch that has installed an SFP+ MACsec uplink module used as the CE switch connecting site B to the MPLS network
- Junos OS Release 12.2R1 or later running on all EX4550 switches in the MPLS network (PE1, PE2, or the provider switch)
- Junos OS Release 13.2X50-D15 (controlled version) or later running on the CE switch at site A and the CE switch at site B

NOTE: The controlled version of Juniper Networks Junos operating system (Junos OS) software must be downloaded to enable MACsec. MACsec software support is not available in the domestic version of Junos OS software, which is installed on the switch by default. The controlled version of Junos OS software includes all features and functionality available in the domestic version of Junos OS, while also supporting MACsec. See "Understanding Media Access Control Security (MACsec)" on page 259 for additional information about MACsec software requirements.

- A MACsec feature license installed on the CE switch at site A and the CE switch at site B

NOTE: To purchase a software license for MACsec, contact your Juniper Networks sales representative (https://www.juniper.net/us/en/contact-us/sales-offices). The Juniper Networks sales representative will provide you with a feature license file and a license key. You will be asked to supply the chassis serial number of your switch; you can obtain the serial number by running the `show virtual-chassis` or `show chassis hardware` command.

Overview and Topology

In this example, financially-sensitive company data is often sent between a user at site A and a user at site B. The company wants to ensure that all network traffic traveling from the user at site A to the user at site B is highly secure and cannot be viewed or corrupted by an attacker. The company is using the industry-standard Layer 2 security provided by MACsec, which provides encryption to ensure data cannot be viewed by attackers and integrity checks to ensure transmitted data is not corrupted, to secure all traffic traveling on the CCC through the MPLS cloud connecting the sites. VLANs are configured at both sites to ensure traffic traveling between the two users traverses the sites over the MACsec-secured CCC.

The MPLS network in this example includes two provider edge (PE) switches—PE1 and PE2—and one provider (transit) switch. PE1 connects the customer edge (CE) switch at site A to the MPLS network and PE2 connects the CE switch at site B to the MPLS network. MACsec is enabled on the CCC connecting the CE switches at site A and site B to secure traffic traveling between the sites over the CCC. A VLAN that includes the interfaces that connect the users to the CE switches, interface ge-0/0/0 on the CE switch at site A and interface ge-0/0/2 on the CE switch at site B, and the interfaces that connect the CE switches to the MPLS cloud (ge-0/0/0 on the site A CE switch and xe-0/1/0 on the site B CE switch), is used to direct all traffic between the users onto the MACsec-secured CCC.

Figure 13 on page 287 shows the topology used in this example. The MACsec-secured CCC traffic is labeled `MACsec CCC` in the figure.

Figure 13: MPLS Diagram Between Site A and Site B

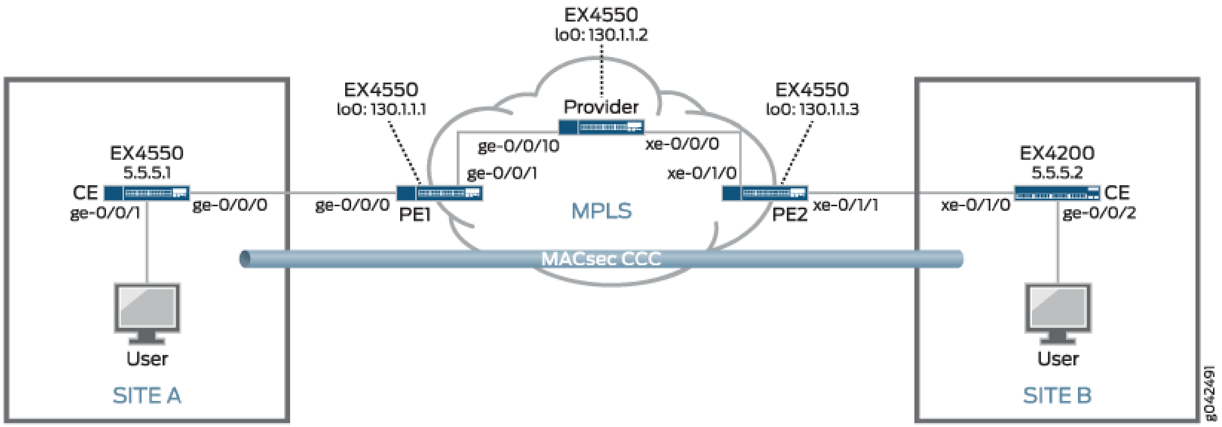


Table 9 on page 288 provides a summary of the MPLS network components in this topology.

Table 10 on page 289 provides a summary of the MACsec connectivity association used in this topology. MACsec is enabled by creating a connectivity association on the interfaces at each end of a link. MACsec is enabled when the interfaces at each end of the link exchange pre-shared keys—the pre-shared keys are defined in the connectivity association—to secure the link for MACsec.

Table 11 on page 290 provides a summary of the VLAN used in this topology. The VLAN is used in this topology to direct all communication from the user at site A to the user at site B onto the MACsec-secured CCC.

Table 9: Components of the MPLS Topology

Component	Description
PE1	PE switch. lo0: • IP address: 130.1.1.1/32 • Participates in OSPF and RSVP. ge-0/0/0: • Customer edge interface connecting site A to the MPLS network. • CCC connecting to xe-0/1/1 on PE2. ge-0/0/1: • Core interface connecting PE1 to the provider switch. • IP address: 10.1.5.2/24 • Participates in OSPF, RSVP, and MPLS.
Provider	Provider switch. lo0: • IP address: 130.1.1.2/32 • Participates in OSPF and RSVP. ge-0/0/10: • Core interface connecting the provider switch to PE1. • IP address: 10.1.5.1/24 • Participates in OSPF, RSVP, and MPLS. xe-0/0/0: • Core interface connecting the provider switch to PE2. • IP address: 10.1.9.1/24 • Participates in OSPF, RSVP, and MPLS.

Table 9: Components of the MPLS Topology *(Continued)*

Component	Description
PE2	PE switch. lo0: • IP address: 130.1.1.3/32 • Participates in OSPF and RSVP. xe-0/1/0 • Core interface connecting PE2 to the provider switch. • IP address: 10.1.9.2/24 • Participates in OSPF, RSVP, and MPLS. xe-0/1/1 • Customer edge interface connecting site B to the MPLS network. • CCC connecting to ge-0/0/0 on PE1.
lsp_to_pe2_xe1 label-switched path	Label-switched path from PE1 to PE2.
lsp_to_pe1_ge0 label-switched path	Label-switched path from PE2 to PE1.

Table 10: MACsec Connectivity Association Summary

Connectivity Association	Description
ccc-macsec	Connectivity association enabling MACsec on CCC connecting site A to site B. The connectivity association is enabled on the following interfaces: • Site A CE switch: ge-0/0/0 • Site B CE switch: xe-0/1/0

Table 11: VLANs Summary

VLAN	Description
macsec	VLAN directing traffic between the user at site A and the user at site B onto the MACsec-secured CCC. The VLAN includes the following interfaces: • Site A CE switch: ge-0/0/0 • Site A CE switch: ge-0/0/1 • Site B CE switch: xe-0/1/0 • Site B CE switch: ge-0/0/2

Configuring MPLS

IN THIS SECTION

This section explains how to configure MPLS on each switch in the MPLS network.

It includes the following sections:

Configuring MPLS on Switch PE1

CLI Quick Configuration

To quickly configure the MPLS configuration on the PE1 switch, use the following commands:

```
[edit]
set protocols ospf traffic-engineering
```

```
                  set protocols ospf area 0.0.0.0 interface lo0.0


                  set protocols ospf area 0.0.0.0 interface ge-0/0/1.0


                  set protocols mpls label-switched-path lsp_to_pe2_xe1 to 130.1.1.3



                  set protocols mpls interface ge-0/0/1.0


                  set protocols rsvp interface lo0.0


                  set protocols rsvp interface ge-0/0/1.0


                  set interfaces lo0 unit 0 family inet address 130.1.1.1/32


                  set interfaces ge-0/0/1 unit 0 family inet address 10.1.5.2/24


                  set interfaces ge-0/0/1 unit 0 family mpls


                  set interfaces ge-0/0/0 unit 0 family ccc


                  set protocols connections remote-interface-switch ge-1-to-pe2 interface
ge-0/0/0.0



                  set protocols connections remote-interface-switch ge-1-to-pe2 transmit-lsp
lsp_to_pe2_xe1


                  set protocols connections remote-interface-switch ge-1-to-pe2 receive-lsp
lsp_to_pe1_ge0
```

Step-by-Step Procedure

To configure MPLS on Switch PE1:

1. Configure OSPF with traffic engineering enabled:

```
[edit protocols]
user@switch-PE1# set ospf traffic-engineering
```

2. Configure OSPF on the loopback address and the core interfaces:

```
[edit protocols]
user@switch-PE1# set ospf area 0.0.0.0 interface lo0.0
user@switch-PE1# set ospf area 0.0.0.0 interface  ge-0/0/1.0
```

3. Configure MPLS on this switch, PE1, with an LSP to the PE2 switch:

```
[edit protocols]
user@switch-PE1# set mpls label-switched-path lsp_to_pe2_xe1 to 130.1.1.3
```

4. Configure MPLS on the core interfaces:

```
[edit protocols]
user@switch-PE1# set mpls interface ge-0/0/1.0
```

5. Configure RSVP on the loopback interface and the core interfaces:

```
[edit protocols]
user@switch-PE1# set rsvp interface lo0.0
user@switch-PE1# set rsvp interface ge-0/0/1.0
```

6. Configure IP addresses for the loopback interface and the core interfaces:

```
[edit]
user@switch-PE1# set interfaces lo0 unit 0 family inet address 130.1.1.1/32
user@switch-PE1# set interfaces ge-0/0/1 unit 0 family inet address 10.1.5.2/24
```

7. Configure `family mpls` on the logical unit of the core interface addresses:

```
[edit]
user@switch-PE1# set interfaces ge-0/0/1 unit 0 family mpls
```

8. Configure the logical unit of the customer edge interface as a CCC:

```
[edit interfaces ge-0/0/0 unit 0]
user@PE-1# set family ccc
```

9. Configure the interface-based CCC from PE1 to PE2:

```
[edit protocols]
user@PE-1# set connections remote-interface-switch ge-1-to-pe2 interface ge-0/0/0.0
user@PE-1# set connections remote-interface-switch ge-1-to-pe2 transmit-lsp lsp_to_pe2_xe1
user@PE-1# set connections remote-interface-switch ge-1-to-pe2 receive-lsp lsp_to_pe1_ge0
```

Results

Display the results of the configuration:

```
user@PE-1> show configuration
```

```
interfaces {
    ge-0/0/0 {
        unit 0 {
            family ccc;
        }
    }
    ge-0/0/1{
        unit 0 {
            family inet {
                address 130.1.5.2/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 130.1.1.1/32;
            }
        }
    }
}
```

```
protocols {
    rsvp {
        interface lo0.0;
        interface ge-0/0/1.0;
    }
    mpls {
        label-switched-path lsp_to_pe2_xe1 {
            to 130.1.1.3;
        }
        interface ge-0/0/1.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0;
            interface ge-0/0/1.0;
        }
    }
    connections {
        remote-interface-switch ge-1-to-pe2 {
            interface ge-0/0/0.0;
            transmit-lsp lsp_to_pe2_xe1;
            receive-lsp lsp_to_pe1_ge0;
        }
    }
}
```

Configuring MPLS on the Provider Switch

CLI Quick Configuration

To quickly configure the MPLS configuration on the provider switch, use the following commands:

```
[edit]
set protocols ospf traffic-engineering


                  set protocols ospf area 0.0.0.0 interface lo0.0


                  set protocols ospf area 0.0.0.0 interface ge-0/0/10.0
                  set protocols ospf area 0.0.0.0 interface xe-0/0/0.0
                  set protocols mpls interface ge-0/0/10.0
                  set protocols mpls interface xe-0/0/0.0
                  set protocols mpls label-switched-path lsp_to_pe2_xe1 to 130.1.1.3

                  set protocols rsvp interface lo0.0
                  set protocols rsvp interface ge-0/0/10.0
                  set protocols rsvp interface xe-0/0/0.0
                  set interfaces lo0 unit 0 family inet address 130.1.1.2/32
                  set interfaces ge-0/0/10 unit 0 family inet address 10.1.5.1/24
                  set interfaces ge-0/0/10 unit 0 family mpls
                  set interfaces xe-0/0/0 unit 0 family inet address 10.1.9.1/24
                  set interfaces xe-0/0/0 unit 0 family mpls
```

Step-by-Step Procedure

To configure the provider switch:

1. Configure OSPF with traffic engineering enabled:

```
[edit protocols]
user@switch-P# set ospf traffic-engineering
```

2. Configure OSPF on the loopback interface and the core interfaces:

```
[edit protocols]
user@switch-P# set ospf area 0.0.0.0 interface lo0.0
user@switch-P# set ospf area 0.0.0.0 interface ge-0/0/10.0
user@switch-P# set ospf area 0.0.0.0 interface xe-0/0/0.0
```

3. Configure MPLS on the core interfaces on the switch:

```
[edit protocols]
user@switch-P# set mpls interface ge-0/0/10.0
user@switch-P# set mpls interface xe-0/0/0.0
```

4. Configure RSVP on the loopback interface and the core interfaces:

```
[edit protocols]
user@switch-P# set rsvp interface lo0.0
user@switch-P# set rsvp interface ge-0/0/10.0
user@switch-P# set rsvp interface xe-0/0/0.0
```

5. Configure IP addresses for the loopback interface and the core interfaces:

```
[edit]
user@switch-P# set interfaces lo0 unit 0 family inet address 130.1.1.2/32
user@switch-P# set interfaces ge-0/0/10 unit 0 family inet address 10.1.5.1/24
user@switch-P# set interfaces xe-0/0/0 unit 0 family inet address 10.1.9.1/24
```

6. Configure `family mpls` on the logical unit of the core interface addresses:

```
[edit]
user@switch-P# set interfaces ge-0/0/10 unit 0 family mpls
user@switch-P# set interfaces xe-0/0/0 unit 0 family mpls
```

7. Configure the LSP to the PE2 switch:

```
[edit]
user@switch-P# set protocols mpls label-switched-path lsp_to_pe2_xe1 to 130.1.1.3
```

Results

Display the results of the configuration:

```
user@switch-P> show configuration
```

```
interfaces {
    ge-0/0/10 {
        unit 0 {
            family inet {
                address 10.1.5.1/24;
            }
            family mpls;
        }
    }
    xe-0/0/0 {
        unit 0 {
            family inet {
                address 10.1.9.1/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 130.1.1.2/32;
            }
        }
    }
}
protocols {
    rsvp {
        interface lo0.0;
        interface ge-0/0/10.0;
        interface xe-0/0/0.0;
    }
    mpls {
        label-switched-path lsp_to_pe2_xe1 {
            to 130.1.1.3;
        }
        interface ge-0/0/10.0;
        interface xe-0/0/0.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0;
            interface ge-0/0/10.0;
            interface xe-0/0/0.0;
        }
    }
}
```

Configuring MPLS on Switch PE2

CLI Quick Configuration

To quickly cconfigure the MPLS configuration on Switch PE2, use the following commands:

```
[edit]
set protocols ospf traffic-engineering


                  set protocols ospf area 0.0.0.0 interface lo0.0


                  set protocols ospf area 0.0.0.0 interface xe-0/1/0.0


                  set protocols mpls label-switched-path lsp_to_pe1_ge0 to 130.1.1.1



                  set protocols mpls interface xe-0/1/0.0


                  set protocols rsvp interface lo0.0
                  set protocols rsvp interface xe-0/1/0.0


                  set interfaces lo0 unit 0 family inet address 130.1.1.3/32


                  set interfaces xe-0/1/0 unit 0 family inet address 10.1.9.2/24


                  set interfaces xe-0/1/0 unit 0 family mpls


                  set interfaces xe-0/1/1 unit 0 family ccc


                  set protocols connections remote-interface-switch xe-1-to-pe1 interface
xe-0/1/1.0



                  set protocols connections remote-interface-switch xe-1-to-pe1 transmit-lsp
lsp_to_pe1_ge0


                  set protocols connections remote-interface-switch xe-1-to-pe1 receive-lsp
lsp_to_pe2_xe1
```

Step-by-Step Procedure

To configure Switch PE2:

1. Configure OSPF with traffic engineering enabled:

   ```
   [edit protocols]
   user@switch-PE2# set ospf traffic-engineering
   ```

2. Configure OSPF on the loopback interface and the core interface:

```
[edit protocols]
user@switch-PE2# set ospf area 0.0.0.0 interface lo0.0
user@switch-PE2# set ospf area 0.0.0.0 interface xe-0/1/0.0
```

3. Configure MPLS on this switch (PE2) with a label-switched path (LSP) to the other PE switch (PE1):

```
[edit protocols]
user@switch-PE2# set mpls label-switched-path lsp_to_pe1_ge0 to 130.1.1.1
```

4. Configure MPLS on the core interface:

```
[edit protocols]
user@switch-PE2# set mpls interface xe-0/1/0.0
```

5. Configure RSVP on the loopback interface and the core interface:

```
[edit protocols]
user@switch-PE2# set rsvp interface lo0.0
user@switch-PE2# set rsvp interface xe-0/1/0.0
```

6. Configure IP addresses for the loopback interface and the core interface:

```
[edit]
user@switch-PE2# set interfaces lo0 unit 0 family inet address 130.1.1.3/32
```

```
user@switch-PE2# set interfaces xe-0/1/0 unit 0 family inet address 10.1.9.2/24
```

7. Configure `family mpls` on the logical unit of the core interface:

```
[edit]
user@switch-PE2# set interfaces xe-0/1/0 unit 0 family mpls
```

8. Configure the logical unit of the customer edge interface as a CCC:

```
[edit interfaces xe-0/1/1 unit 0]
user@switch-PE2# set family ccc
```

9. Configure the interface-based CCC between the primary edge switches:

```
[edit protocols]
user@switch-PE2# set connections remote-interface-switch xe-1-to-pe1 interface xe-0/1/1.0
user@switch-PE2# set connections remote-interface-switch xe-1-to-pe1 transmit-lsp
lsp_to_pe1_ge0
user@switch-PE2# set connections remote-interface-switch xe-1-to-pe1 receive-lsp
lsp_to_pe2_xe1
```

Results

Display the results of the configuration:

```
user@switch-PE2> show configuration
```

```
interfaces {
    xe-0/1/0 {
        unit 0 {
            family inet {
                address 10.1.9.2/24;
            }
            family mpls;
        }
    }
    xe-0/1/1 {
        unit 0 {
            family ccc;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 130.1.1.3/32;
            }
        }
    }
}
protocols {
    rsvp {
        interface lo0.0;
        interface xe-0/1/0.0;
    }
    mpls {
        label-switched-path lsp_to_pe1_ge0 {
            to 130.1.1.1;
        }
        interface xe-0/1/0.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0;
            interface xe-0/1/0.0;
        }
    }
    connections {
        remote-interface-switch xe-1-to-pe1 {
            interface xe-0/1/1.0;
            transmit-lsp lsp_to_pe1_ge0;
            receive-lsp lsp_to_pe2_xe1;
        }
    }
}
```

Configuring MACsec

IN THIS SECTION

This section explains how to configure MACsec on each switch in the topology.

It includes the following sections:

Configuring MACsec on the Site A CE Switch to Secure Traffic to Site B

CLI Quick Configuration

```
[edit]
set security macsec connectivity-association ccc-macsec security-mode static-cak


                  set security macsec connectivity-association ccc-macsec pre-shared-key ckn
37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311
```

```
                 set security macsec connectivity-association ccc-macsec pre-shared-key cak
228ef255aa23ff6729ee664acb66e91f


                 set security macsec interfaces ge-0/0/0 connectivity-association ccc-macsec
```

Step-by-Step Procedure

In this example, the traffic between the users that often exchange financially-sensitive data is sent between the sites on a CCC through the MPLS cloud. MACsec is enabled on the CCC by configuring a MACsec connectivity association on the interfaces on the site A and site B CE switches that connect to the MPLS PE switches. The connectivity associations must have matching connectivity-association names (in this example, **ccc-macsec**), matching CKNs (in this example, **37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311**), and CAKs (in this example, **228ef255aa23ff6729ee664acb66e91f**) in order to establish a MACsec-secure connection.

To enable MACsec on the CCC connecting site A to site B, perform the following procedure on the site A CE switch:

1. Create the connectivity association named **ccc-macsec**, and configure the MACsec security mode as `static-cak`:

   ```
   [edit security macsec]
   user@switch-CE-A# set connectivity-association ccc-macsec security-mode static-cak
   ```

2. Create the pre-shared key by configuring the CKN and CAK:

   ```
   [edit security macsec]
   user@switch-CE-A# set connectivity-association ccc-macsec pre-shared-key ckn
   37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311
   user@switch-CE-A# set connectivity-association ccc-macsec pre-shared-key cak
   228ef255aa23ff6729ee664acb66e91f
   ```

3. Assign the connectivity association to the interface connecting to the PE1 switch:

```
[edit security macsec]
user@switch-CE-A# set interfaces ge-0/0/0 connectivity-association ccc-macsec
```

This completes the steps for configuring the connectivity association on one end of the CCC. MACsec is not enabled until a connectivity association with matching pre-shared keys is enabled on the opposite end of a link, which in this case is the interface on the site B CE switch, of the CCC. The process for configuring the connectivity association on the site B CE switch is described in the following section.

Results

Display the results of the configuration:

```
user@switch-CE-A> show configuration
```

```
security {
    macsec {
        connectivity-association {
            ccc-macsec {
                pre-shared-key {
                    cak "$9$rJ-lWLxNdw24Xxik.PQzreK"; ## SECRET-DATA
                    ckn 37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311;
                }
                security-mode {
                    static-cak;
                }
            }
        }
        interfaces {
            ge-0/0/0 {
                connectivity-association {
                    ccc-macsec;
                }
            }
        }
    }
}
```

Configuring MACsec on the Site B CE Switch to Secure Traffic to Site A

CLI Quick Configuration

```
[edit]
set security macsec connectivity-association ccc-macsec security-mode static-cak


                  set security macsec connectivity-association ccc-macsec pre-shared-key ckn
37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311


                  set security macsec connectivity-association ccc-macsec pre-shared-key cak
228ef255aa23ff6729ee664acb66e91f


                  set security macsec interfaces xe-0/1/0 connectivity-association ccc-macsec
```

Step-by-Step Procedure

Traffic travels from site B to site A over the MPLS network using a CCC. MACsec is enabled on the CCC by configuring a MACsec connectivity association on the interfaces on the site A and site B CE switches that connect to the MPLS PE switches. The connectivity associations must have matching connectivity-association names (in this example, **ccc-macsec**), matching CKNs (**37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311**), and matching CAKs (**228ef255aa23ff6729ee664acb66e91f**) in order to establish a MACsec-secure connection.

To enable MACsec on the CCC connecting site B to site A, perform the following procedure on the site B CE switch:

1. Create the connectivity association named **ccc-macsec**, and configure the MACsec security mode as `static-cak`:

   ```
   [edit security macsec]
   user@switch-CE-B# set connectivity-association ccc-macsec security-mode static-cak
   ```

2. Create the pre-shared key by configuring the CKN and CAK:

   ```
   [edit security macsec]
   user@switch-CE-B# set connectivity-association ccc-macsec pre-shared-key ckn
   37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311
   user@switch-CE-B# set connectivity-association ccc-macsec pre-shared-key cak
   228ef255aa23ff6729ee664acb66e91f
   ```

3. Assign the connectivity association to the interface connecting to Switch PE2:

   ```
   [edit security macsec]
   user@switch-CE-B# set interfaces xe-0/1/0 connectivity-association ccc-macsec
   ```

   MACsec is enabled for the CCC after the pre-shared keys are exchanged, which is shortly after this procedure is completed.

Results

Display the results of the configuration:

```
user@switch-CE-B> show configuration
```

```
security {
    macsec {
        connectivity-association {
            ccc-macsec {
                security-mode {
                    static-cak;
                }
                pre-shared-key {
                    cak "$9$rJ-lWLxNdw24Xxik.PQzreK"; ## SECRET-DATA
                    ckn 37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311;
                }
            }
        }
        interfaces {
            xe-0/1/0 {
                connectivity-association {
                    ccc-macsec;
                }
            }
        }
    }
}
```

Configuring VLANs to Direct Traffic onto the MACsec-Secured CCC

IN THIS SECTION

This section explains how to configure VLANs on the site A and site B CE switches. The purpose of the VLANs is to direct traffic that you want to be MACsec-secured onto the MACsec-secured CCC.

Configuring the VLAN to Direct Traffic to the MACsec CCC on the Site A CE Switch

CLI Quick Configuration

```
[edit]
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members macsec


                  set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members macsec


                  set interfaces vlan unit 50 family inet address 5.5.5.1/24


                  set vlans macsec vlan-id 50


                  set vlans macsec l3-interface vlan.50
```

Step-by-Step Procedure

To create a VLAN (VLAN ID 50) that directs traffic from the user at site A onto the MACsec-secured CCC:

1. Configure the ge-0/0/0 interface into the macsec VLAN:

   ```
   [edit interfaces ge-0/0/0 unit 0]
   user@switch-CE-A# set family ethernet-switching vlan members macsec
   ```

2. Configure the ge-0/0/2 interface into the macsec VLAN:

```
[edit interfaces ge-0/0/2 unit 0]
user@switch-CE-A# set family ethernet-switching vlan members macsec
```

3. Create the IP address for the macsec VLAN broadcast domain:

```
[edit interfaces]
user@switch-CE-A# set vlan unit 50 family inet address 5.5.5.1/24
```

4. Configure the VLAN tag ID to 50 for the macsec VLAN:

```
[edit vlans]
user@switch-CE-A# set macsec vlan-id 50
```

5. Associate a Layer 3 interface with the macsec VLAN:

```
[edit vlans]
user@switch-CE-A# set macsec l3-interface vlan.50
```

Results

Display the results of the configuration:

```
user@switch-CE-A> show configuration
interfaces {
    ge-0/0/0 {
        unit 0 {
            family ethernet-switching {
                vlan members macsec;
            }
        }
    }
    ge-0/0/2 {
        unit 0 {
            family ethernet-switching {
                vlan members macsec;
            }
        }
    }
    vlan {
        unit 50 {
            family inet address 5.5.5.1/24;
        }
    }
}
vlans {
    macsec {
        l3-interface vlan.50;
        vlan-id 50;
    }
}
```

Configuring the VLAN to Direct Traffic to the MACsec CCC on the Site B CE Switch

CLI Quick Configuration

```
[edit]
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members macsec


                  set interfaces xe-0/1/0 unit 0 family ethernet-switching vlan members macsec


                  set interfaces vlan unit 50 family inet address 5.5.5.2/24
                  set vlans macsec vlan-id 50


                  set vlans macsec l3-interface vlan.50
```

Step-by-Step Procedure

To create a VLAN (VLAN ID 50) to direct traffic for the user at site B onto the MACsec-secured CCC:

1. Configure the ge-0/0/2 interface into the macsec VLAN:

   ```
   [edit interfaces ge-0/0/2 unit 0]
   user@switch-CE-B# set family ethernet-switching vlan members macsec
   ```

2. Configure the xe-0/1/0 interface into the macsec VLAN:

   ```
   [edit interfaces xe-0/1/0 unit 0]
   user@switch-CE-B# set family ethernet-switching vlan members macsec
   ```

3. Create the IP address for the macsec VLAN broadcast domain:

   ```
   [edit interfaces]
   user@switch-CE-B# set vlan unit 50 family inet address 5.5.5.2/24
   ```

4. Configure the VLAN tag ID to 50 for the macsec VLAN:

   ```
   [edit vlans]
   user@switch-CE-B# set macsec vlan-id 50
   ```

5. Associate a Layer 3 interface with the macsec VLAN:

```
[edit vlans]
user@switch-CE-B# set macsec l3-interface vlan.50
```

Results

Display the results of the configuration:

```
user@switch-CE-B> show configuration
interfaces {
    ge-0/0/2 {
        unit 0 {
            family ethernet-switching {
                vlan members macsec;
            }
        }
    }
    xe-0/1/0 {
        unit 0 {
            family ethernet-switching {
                vlan members macsec;
            }
        }
    }
    vlan {
        unit 50 {
            family inet address 5.5.5.2/24;
        }
    }
}
vlans {
    macsec {
        l3-interface vlan.50;
        vlan-id 50;
    }
}
```

Verification

IN THIS SECTION

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

Verifying the MACsec Connection

Purpose

Verify that MACsec is operational on the CCC.

Action

Enter the `show security macsec connections` command on one or both of the customer edge (CE) switches.

```
user@switch-CE-A> show security macsec connections
Interface name: ge-0/0/0
        CA name: ccc-macsec
        Cipher suite: GCM-AES-128   Encryption: on
        Key server offset: 0        Include SCI: no
        Replay protect: off         Replay window: 0
```

```
          Outbound secure channels
            SC Id: 00:19:E2:53:CD:F3/1
            Outgoing packet number: 9785
            Secure associations
            AN: 0 Status: inuse Create time: 2d 20:47:54
          Inbound secure channels
            SC Id: 00:23:9C:0A:53:33/1
            Secure associations
            AN: 0 Status: inuse Create time: 2d 20:47:54
```

Meaning

The `Interface name:` and `CA name:` outputs shows that the ccc-macsec connectivity association is operational on interface ge-0/0/0. The output does not appear when the connectivity association is not operational on the interface.

For additional verification that MACsec is operational on the CCC, you can also enter the `show security macsec connections` command on the other CE switch.

Verifying That MACsec-Secured Traffic Is Traversing the CCCs

Purpose

Verify that traffic traversing the CCC is MACsec-secured.

Action

Enter the `show security macsec statistics` command on one or both of the CE switches.

```
user@switch-CE-A> show security macsec statistics
Interface name: ge-0/0/0
    Secure Channel transmitted
        Encrypted packets: 9784
        Encrypted bytes:   2821527
        Protected packets: 0
        Protected bytes:   0
    Secure Association transmitted
        Encrypted packets: 9784
        Protected packets: 0
    Secure Channel received
        Accepted packets:  9791
        Validated bytes:   0
        Decrypted bytes:   2823555
    Secure Association received
        Accepted packets:  9791
        Validated bytes:   0
        Decrypted bytes:   2823555
```

Meaning

The `Encrypted packets` line under the `Secure Channel transmitted` output is incremented each time a packet is sent from the interface that is secured and encrypted by MACsec. The `Encrypted packets` output shows that 9784 encrypted and secured packets have been transmitted from interface ge-0/0/0. MACsec-secured traffic is, therefore, being sent on interface ge-0/0/0.

The `Accepted packets` line under the `Secure Association received` output is incremented each time a packet that has passed the MACsec integrity check is received on the interface. The `Decrypted bytes` line under the `Secure Association received` output is incremented each time an encrypted packet is received and decrypted. The output shows that 9791 MACsec-secured packets have been received on interface ge-0/0/0, and that 2823555 bytes from those packets have been successfully decrypted. MACsec-secured traffic is, therefore, being received on interface ge-0/0/0.

For additional verification, you can also enter the `show security macsec statistics` command on the other CE switch.

Verifying That the MPLS and CCC Protocols Are Enabled on the Provider Edge and Provider Switch Interfaces

Purpose

Verify that MPLS is enabled on the correct interfaces for the PE and provider switches.

Action

Enter the `show interfaces terse` command on both of the PE switches and the provider switch:

```
user@switch-PE1> show interfaces terse

Interface               Admin Link Proto    Local                 Remote
ge-0/0/0                up    up
ge-0/0/0.0              up    up   ccc
ge-0/0/1                up    up
```

```
ge-0/0/1.0              up    up   inet     10.1.5.2/24
                                   mpls
<some output removed for brevity>
```

```
user@switch-P> show interfaces terse

Interface               Admin Link Proto    Local                 Remote
xe-0/0/0                up    up
xe-0/0/0.0              up    up   inet     10.1.9.1/24
                                   mpls
ge-0/0/10               up    up
ge-0/0/10.0             up    up   inet     10.1.5.1/24
                                   mpls
<some output removed for brevity>
```

```
user@switch-PE2> show interfaces terse

Interface               Admin Link Proto    Local                 Remote
xe-0/1/0                up    up
xe-0/1/0.0              up    up   inet     10.1.9.2/24
                                   mpls
xe-0/1/1                up    up
xe-0/1/1.0              up    up   ccc
<some output removed for brevity>
```

Meaning

The output confirms that the MPLS protocol is up for the provider switch interfaces passing MPLS traffic —xe-0/0/0 and ge-0/0/10—and on the PE switch interfaces passing MPLS traffic, which is interface ge-0/0/1 on the PE1 switch and interface xe-0/1/0 on the PE2 switch.

The output also confirms that CCC is enabled on the PE switch interfaces facing the CE switches, which are interface ge-0/0/0 on the PE1 switch and interface xe-0/1/1 on the PE2 switch.

Verifying MPLS Label Operations

Purpose

Verify which interface is being used as the beginning of the CCC and which interface is being used to push the MPLS packet to the next hop.

Action

Enter the `show route forwarding-table family mpls` on one or both of the PE switches.

```
user@switch-PE1> show route forwarding-table family mpls

Routing table: default.mpls
MPLS:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    dscd    50     1
0                  user     0                    recv    49     4
1                  user     0                    recv    49     4
2                  user     0                    recv    49     4
13                 user     0                    recv    49     4
299856             user     0                   Pop    1327     2 ge-0/0/0.0
ge-0/0/0.0  (CCC)  user     0 10.1.5.1          Push 299952  1328     2 ge-0/0/1.0
```

Meaning

This output confirms that the CCC is configured on interface ge-0/0/0.0. The switch receives ingress traffic on ge-0/0/1.0 and pushes label 299952 onto the packet, which exits the switch through interface ge-0/0/1.0. The output also shows that when the switch receives an MPLS packet with label 299856, it pops the label and sends the packet out through interface ge-0/0/0.0

For further verification of MPLS label operations, enter the `show route forwarding-table family mpls` on the other PE switch.

Verifying the Status of the MPLS CCCs

Purpose

Verify that the MPLS CCCs are operating.

Action

Enter the `show connections` command on the PE switches.

```
user@switch-PE1> show connections
CCC and TCC connections [Link Monitoring On]
Legend for status (St):             Legend for connection types:
 UN -- uninitialized                 if-sw:  interface switching
 NP -- not present                   rmt-if: remote interface switching
 WE -- wrong encapsulation           lsp-sw: LSP switching
 DS -- disabled                      tx-p2mp-sw: transmit P2MP switching
 Dn -- down                          rx-p2mp-sw: receive P2MP switching
 -> -- only outbound conn is up     Legend for circuit types:
 <- -- only inbound  conn is up      intf -- interface
 Up -- operational                   oif  -- outgoing interface
 RmtDn -- remote CCC down            tlsp -- transmit LSP
 Restart -- restarting               rlsp -- receive LSP


Connection/Circuit                Type        St      Time last up     # Up trans
ge-1-to-pe2                       rmt-if      Up      May 30 19:01:45         1
  ge-0/0/0.0                        intf      Up
  lsp_to_pe2_xe1                    tlsp      Up
  lsp_to_pe1_ge0                    rlsp      Up
```

```
user@switch-PE2> show connections

CCC and TCC connections [Link Monitoring On]
Legend for status (St):             Legend for connection types:
 UN -- uninitialized                 if-sw:  interface switching
 NP -- not present                   rmt-if: remote interface switching
 WE -- wrong encapsulation           lsp-sw: LSP switching
 DS -- disabled                      tx-p2mp-sw: transmit P2MP switching
 Dn -- down                          rx-p2mp-sw: receive P2MP switching
 -> -- only outbound conn is up     Legend for circuit types:
 <- -- only inbound  conn is up      intf -- interface
 Up -- operational                   oif  -- outgoing interface
 RmtDn -- remote CCC down            tlsp -- transmit LSP
 Restart -- restarting               rlsp -- receive LSP
```

```
Connection/Circuit                Type        St      Time last up     # Up trans
xe-1-to-pe1                       rmt-if      Up      May 30 09:39:15         1
  xe-0/1/1.0                        intf      Up
  lsp_to_pe1_ge0                    tlsp      Up
  lsp_to_pe2_xe1                    rlsp      Up
```

The `show connections` command displays the status of the CCC connections. This output verifies that the CCC interfaces and their associated transmit and receive LSPs are `Up` on both PE switches.

Verifying OSPF Operation

Purpose

Verify that OSPF is running.

Action

Enter the `show ospf neighbor` command the provider or the PE switches, and check the `State` output.

```
user@switch-P> show ospf neighbor

Address          Interface              State     ID               Pri  Dead
10.1.5.2         ge-0/0/10.0            Full      130.1.1.1        128    33
10.1.9.2         xe-0/0/0.0             Full      130.1.1.3        128    38
```

Meaning

The `State` output is `Full` on all interfaces using OSPF, so OSPF is operating.

For further verification on OSPF, enter the `show ospf neighbor` command on the PE switches in addition to the provider switch.

Verifying the Status of the RSVP Sessions

Purpose

Verify the status of the RSVP sessions.

Action

Enter the `show rsvp session` command, and verify that the state is up for each RSVP session.

```
user@switch-P> show rsvp session

Ingress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 2 sessions
To              From            State   Rt Style Labelin Labelout LSPname
130.1.1.1       130.1.1.3       Up       0  1 FF  299936   299856 lsp_to_pe1_ge0
130.1.1.3       130.1.1.1       Up       0  1 FF  299952   299840 lsp_to_pe2_xe1
Total 2 displayed, Up 2, Down 0
```

Meaning

The `State` is `Up` for all connections, so RSVP is operating normally.

For further verification, enter the `show rsvp session` on the PE switches in addition to the provider switch.

RELATED DOCUMENTATION

Configuring MACsec | 259

Example: Configuring MACsec over an MPLS CCC on MX Series Routers

IN THIS SECTION

This example shows how to enable MACsec to secure sensitive traffic traveling from a user at one site to a user at another site over a basic MPLS CCC.

Requirements

This example uses the following hardware and software components:

- Three MX Series routers used as the PE and provider routers in the MPLS network
- One MX Series router used as the CE router connecting site A to the MPLS network
- One MX240, MX480, or MX960 router with the enhanced 20-port Gigabit Ethernet MIC (model number MIC-3D-20GE-SFP-E) used as the CE router connecting site B to the MPLS network
- Junos OS Release 15.1R1 or later running on all MX Series routers in the MPLS network (PE1, PE2, or the provider router)
- Junos OS Release 15.1R1 or later running on the CE router at site A and the CE router at site B

Overview and Topology

In this example, financially-sensitive company data is often sent between a user at site A and a user at site B. The company wants to ensure that all network traffic traveling from the user at site A to the user at site B is highly secure and cannot be viewed or corrupted by an attacker. The company is using the industry-standard Layer 2 security provided by MACsec, which provides encryption to ensure data cannot be viewed by attackers and integrity checks to ensure transmitted data is not corrupted, to secure all traffic traveling on the CCC through the MPLS cloud connecting the sites. VLANs are configured at both sites to ensure traffic traveling between the two users traverses the sites over the MACsec-secured CCC.

The MPLS network in this example includes two provider edge (PE) routers—PE1 and PE2—and one provider (transit) router. PE1 connects the customer edge (CE) router at site A to the MPLS network and PE2 connects the CE router at site B to the MPLS network. MACsec is enabled on the CCC connecting the CE routers at site A and site B to secure traffic traveling between the sites over the CCC. A VLAN that includes the interfaces that connect the users to the CE routers, interface ge-0/0/0 on the CE router at site A and interface ge-0/0/2 on the CE router at site B, and the interfaces that connect the CE routers to the MPLS cloud (ge-0/0/0 on the site A CE router and xe-0/1/0 on the site B CE router), is used to direct all traffic between the users onto the MACsec-secured CCC.

Table 12 on page 325 provides a summary of the MPLS network components in this topology.

Table 13 on page 327 provides a summary of the MACsec connectivity association used in this topology. MACsec is enabled by creating a connectivity association on the interfaces at each end of a link. MACsec is enabled when the interfaces at each end of the link exchange pre-shared keys—the pre-shared keys are defined in the connectivity association—to secure the link for MACsec.

Table 14 on page 327 provides a summary of the bridge domain and VLAN IDs used in this topology. The VLAN is used in this topology to direct all communication from the user at site A to the user at site B onto the MACsec-secured CCC.

Table 12: Components of the MPLS Topology

Component	Description
PE1	PE router. lo0: • IP address: 130.1.1.1/32 • Participates in OSPF and RSVP. ge-0/0/0: • Customer edge interface connecting site A to the MPLS network. • CCC connecting to xe-0/1/1 on PE2. ge-0/0/1: • Core interface connecting PE1 to the provider router. • IP address: 10.1.5.2/24 • Participates in OSPF, RSVP, and MPLS.

Table 12: Components of the MPLS Topology *(Continued)*

Component	Description
Provider	Provider router. lo0: • IP address: 130.1.1.2/32 • Participates in OSPF and RSVP. ge-0/0/10: • Core interface connecting the provider router to PE1. • IP address: 10.1.5.1/24 • Participates in OSPF, RSVP, and MPLS. xe-0/0/0: • Core interface connecting the provider router to PE2. • IP address: 10.1.9.1/24 • Participates in OSPF, RSVP, and MPLS.
PE2	PE router. lo0: • IP address: 130.1.1.3/32 • Participates in OSPF and RSVP. xe-0/1/0 • Core interface connecting PE2 to the provider router. • IP address: 10.1.9.2/24 • Participates in OSPF, RSVP, and MPLS. xe-0/1/1 • Customer edge interface connecting site B to the MPLS network. • CCC connecting to ge-0/0/0 on PE1.

Table 12: Components of the MPLS Topology *(Continued)*

Component	Description
lsp_to_pe2_xe1 label-switched path	Label-switched path from PE1 to PE2.
lsp_to_pe1_ge0 label-switched path	Label-switched path from PE2 to PE1.

Table 13: MACsec Connectivity Association Summary

Connectivity Association	Description
ccc-macsec	Connectivity association enabling MACsec on CCC connecting site A to site B. The connectivity association is enabled on the following interfaces: • Site A CE router: ge-0/0/0 • Site B CE router: xe-0/1/0

Table 14: Bridge Domains Summary

Bridge Domain	Description
macsec	VLAN directing traffic between the user at site A and the user at site B onto the MACsec-secured CCC. The bridge domain includes the following interfaces: • Site A CE router: ge-0/0/0 • Site A CE router: ge-0/0/1 • Site B CE router: xe-0/1/0 • Site B CE router: ge-0/0/2

Configuring MPLS

IN THIS SECTION

This section explains how to configure MPLS on each router in the MPLS network.

It includes the following sections:

Configuring MPLS on PE1

CLI Quick Configuration

To quickly configure the MPLS configuration on the PE1 router, use the following commands:

```
[edit]
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols mpls label-switched-path lsp_to_pe2_xe1 to 130.1.1.3

set protocols mpls interface ge-0/0/1.0
set protocols rsvp interface lo0.0
set protocols rsvp interface ge-0/0/1.0
set interfaces lo0 unit 0 family inet address 130.1.1.1/32
set interfaces ge-0/0/1 unit 0 family inet address 10.1.5.2/24
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/0 unit 0 family ccc
set protocols connections remote-interface-switch ge-1-to-pe2 interface ge-0/0/0.0

set protocols connections remote-interface-switch ge-1-to-pe2 transmit-lsp lsp_to_pe2_xe1
set protocols connections remote-interface-switch ge-1-to-pe2 receive-lsp lsp_to_pe1_ge0
```

Step-by-Step Procedure

To configure MPLS on router PE1:

1. Configure OSPF with traffic engineering enabled:

```
[edit protocols]
user@router-PE1# set ospf traffic-engineering
```

2. Configure OSPF on the loopback address and the core interfaces:

```
[edit protocols]
user@router-PE1# set ospf area 0.0.0.0 interface lo0.0
user@router-PE1# set ospf area 0.0.0.0 interface  ge-0/0/1.0
```

3. Configure MPLS on this router, PE1, with an LSP to the PE2 router:

```
[edit protocols]
user@router-PE1# set mpls label-switched-path lsp_to_pe2_xe1 to 130.1.1.3
```

4. Configure MPLS on the core interfaces:

```
[edit protocols]
user@router-PE1# set mpls interface ge-0/0/1.0
```

5. Configure RSVP on the loopback interface and the core interfaces:

```
[edit protocols]
user@router-PE1# set rsvp interface lo0.0
user@router-PE1# set rsvp interface ge-0/0/1.0
```

6. Configure IP addresses for the loopback interface and the core interfaces:

```
[edit]
user@router-PE1# set interfaces lo0 unit 0 family inet address 130.1.1.1/32
user@router-PE1# set interfaces ge-0/0/1 unit 0 family inet address 10.1.5.2/24
```

7. Configure `family mpls` on the logical unit of the core interface addresses:

```
[edit]
user@router-PE1# set interfaces ge-0/0/1 unit 0 family mpls
```

8. Configure the logical unit of the customer edge interface as a CCC:

```
[edit interfaces ge-0/0/0 unit 0]
user@PE-1# set family ccc
```

9. Configure the interface-based CCC from PE1 to PE2:

```
[edit protocols]
user@PE-1# set connections remote-interface-switch ge-1-to-pe2 interface ge-0/0/0.0
user@PE-1# set connections remote-interface-switch ge-1-to-pe2 transmit-lsp lsp_to_pe2_xe1
user@PE-1# set connections remote-interface-switch ge-1-to-pe2 receive-lsp lsp_to_pe1_ge0
```

Results

Display the results of the configuration:

```
user@PE-1> show configuration
```

```
interfaces {
    ge-0/0/0 {
        unit 0 {
            family ccc;
        }
    }
    ge-0/0/1{
        unit 0 {
            family inet {
                address 130.1.5.2/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 130.1.1.1/32;
            }
        }
    }
}
protocols {
    rsvp {
        interface lo0.0;
        interface ge-0/0/1.0;
    }
    mpls {
        label-switched-path lsp_to_pe2_xe1 {
            to 130.1.1.3;
        }
        interface ge-0/0/1.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0;
            interface ge-0/0/1.0;
        }
    }
    connections {
        remote-interface-switch ge-1-to-pe2 {
            interface ge-0/0/0.0;
            transmit-lsp lsp_to_pe2_xe1;
            receive-lsp lsp_to_pe1_ge0;
        }
    }
}
```

Configuring MPLS on the Provider Router

CLI Quick Configuration

To quickly configure the MPLS configuration on the provider router, use the following commands:

```
[edit]
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/10.0
set protocols ospf area 0.0.0.0 interface xe-0/0/0.0
set protocols mpls interface ge-0/0/10.0
set protocols mpls interface xe-0/0/0.0
set protocols mpls label-switched-path lsp_to_pe2_xe1 to 130.1.1.3

set protocols rsvp interface lo0.0
set protocols rsvp interface ge-0/0/10.0
set protocols rsvp interface xe-0/0/0.0
set interfaces lo0 unit 0 family inet address 130.1.1.2/32
set interfaces ge-0/0/10 unit 0 family inet address 10.1.5.1/24
set interfaces ge-0/0/10 unit 0 family mpls
set interfaces xe-0/0/0 unit 0 family inet address 10.1.9.1/24
set interfaces xe-0/0/0 unit 0 family mpls
```

Step-by-Step Procedure

To configure the provider router:

1. Configure OSPF with traffic engineering enabled:

   ```
   [edit protocols]
   user@router-P# set ospf traffic-engineering
   ```

2. Configure OSPF on the loopback interface and the core interfaces:

   ```
   [edit protocols]
   user@router-P# set ospf area 0.0.0.0 interface lo0.0
   user@router-P# set ospf area 0.0.0.0 interface ge-0/0/10.0
   user@router-P# set ospf area 0.0.0.0 interface xe-0/0/0.0
   ```

3. Configure MPLS on the core interfaces on the router:

```
[edit protocols]
user@router-P# set mpls interface ge-0/0/10.0
user@router-P# set mpls interface xe-0/0/0.0
```

4. Configure RSVP on the loopback interface and the core interfaces:

```
[edit protocols]
user@router-P# set rsvp interface lo0.0
user@router-P# set rsvp interface ge-0/0/10.0
user@router-P# set rsvp interface xe-0/0/0.0
```

5. Configure IP addresses for the loopback interface and the core interfaces:

```
[edit]
user@router-P# set interfaces lo0 unit 0 family inet address 130.1.1.2/32
user@router-P# set interfaces ge-0/0/10 unit 0 family inet address 10.1.5.1/24
user@router-P# set interfaces xe-0/0/0 unit 0 family inet address 10.1.9.1/24
```

6. Configure `family mpls` on the logical unit of the core interface addresses:

```
[edit]
user@router-P# set interfaces ge-0/0/10 unit 0 family mpls
user@router-P# set interfaces xe-0/0/0 unit 0 family mpls
```

7. Configure the LSP to the PE2 router:

```
[edit]
user@router-P# set protocols mpls label-switched-path lsp_to_pe2_xe1 to 130.1.1.3
```

Results

Display the results of the configuration:

```
user@router-P> show configuration
```

```
interfaces {
    ge-0/0/10 {
        unit 0 {
            family inet {
                address 10.1.5.1/24;
            }
            family mpls;
        }
    }
    xe-0/0/0 {
        unit 0 {
            family inet {
                address 10.1.9.1/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 130.1.1.2/32;
            }
        }
    }
}
protocols {
    rsvp {
        interface lo0.0;
        interface ge-0/0/10.0;
        interface xe-0/0/0.0;
    }
    mpls {
        label-switched-path lsp_to_pe2_xe1 {
            to 130.1.1.3;
        }
        interface ge-0/0/10.0;
        interface xe-0/0/0.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0;
            interface ge-0/0/10.0;
            interface xe-0/0/0.0;
        }
    }
}
```

Configuring MPLS on PE2

CLI Quick Configuration

To quickly configure the MPLS configuration on router PE2, use the following commands:

```
[edit]
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface xe-0/1/0.0
set protocols mpls label-switched-path lsp_to_pe1_ge0 to 130.1.1.1

set protocols mpls interface xe-0/1/0.0
set protocols rsvp interface lo0.0
set protocols rsvp interface xe-0/1/0.0
set interfaces lo0 unit 0 family inet address 130.1.1.3/32
set interfaces xe-0/1/0 unit 0 family inet address 10.1.9.2/24
set interfaces xe-0/1/0 unit 0 family mpls
set interfaces xe-0/1/1 unit 0 family ccc
set protocols connections remote-interface-switch xe-1-to-pe1 interface xe-0/1/1.0

set protocols connections remote-interface-switch xe-1-to-pe1 transmit-lsp lsp_to_pe1_ge0
set protocols connections remote-interface-switch xe-1-to-pe1 receive-lsp lsp_to_pe2_xe1
```

Step-by-Step Procedure

To configure router PE2:

1. Configure OSPF with traffic engineering enabled:

```
[edit protocols]
user@router-PE2# set ospf traffic-engineering
```

2. Configure OSPF on the loopback interface and the core interface:

```
[edit protocols]
user@router-PE2# set ospf area 0.0.0.0 interface lo0.0
user@router-PE2# set ospf area 0.0.0.0 interface xe-0/1/0.0
```

3. Configure MPLS on this router (PE2) with a label-switched path (LSP) to the other PE router (PE1):

```
[edit protocols]
user@router-PE2# set mpls label-switched-path lsp_to_pe1_ge0 to 130.1.1.1
```

4. Configure MPLS on the core interface:

```
[edit protocols]
user@router-PE2# set mpls interface xe-0/1/0.0
```

5. Configure RSVP on the loopback interface and the core interface:

```
[edit protocols]
user@router-PE2# set rsvp interface lo0.0
user@router-PE2# set rsvp interface xe-0/1/0.0
```

6. Configure IP addresses for the loopback interface and the core interface:

```
[edit]
user@router-PE2# set interfaces lo0 unit 0 family inet address 130.1.1.3/32
user@router-PE2# set interfaces xe-0/1/0 unit 0 family inet address 10.1.9.2/24
```

7. Configure `family mpls` on the logical unit of the core interface:

```
[edit]
user@router-PE2# set interfaces xe-0/1/0 unit 0 family mpls
```

8. Configure the logical unit of the customer edge interface as a CCC:

```
[edit interfaces xe-0/1/1 unit 0]
user@router-PE2# set family ccc
```

9. Configure the interface-based CCC between the primary edge routers:

```
[edit protocols]
user@router-PE2# set connections remote-interface-switch xe-1-to-pe1 interface xe-0/1/1.0
user@router-PE2# set connections remote-interface-switch xe-1-to-pe1 transmit-lsp
lsp_to_pe1_ge0
user@router-PE2# set connections remote-interface-switch xe-1-to-pe1 receive-lsp
lsp_to_pe2_xe1
```

Results

Display the results of the configuration:

```
user@router-PE2> show configuration
```

```
interfaces {
    xe-0/1/0 {
        unit 0 {
            family inet {
                address 10.1.9.2/24;
            }
            family mpls;
        }
    }
    xe-0/1/1 {
        unit 0 {
            family ccc;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 130.1.1.3/32;
            }
        }
    }
}
protocols {
    rsvp {
        interface lo0.0;
        interface xe-0/1/0.0;
    }
    mpls {
        label-switched-path lsp_to_pe1_ge0 {
            to 130.1.1.1;
        }
        interface xe-0/1/0.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface lo0.0;
            interface xe-0/1/0.0;
        }
    }
    connections {
        remote-interface-switch xe-1-to-pe1 {
            interface xe-0/1/1.0;
            transmit-lsp lsp_to_pe1_ge0;
            receive-lsp lsp_to_pe2_xe1;
        }
    }
}
```

Configuring MACsec

IN THIS SECTION

This section explains how to configure MACsec on each router in the topology.

It includes the following sections:

Configuring MACsec on the Site A CE Router to Secure Traffic to Site B

CLI Quick Configuration

```
[edit]
set security macsec connectivity-association ccc-macsec security-mode static-cak
set security macsec connectivity-association ccc-macsec pre-shared-key ckn
37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311
set security macsec connectivity-association ccc-macsec pre-shared-key cak
228ef255aa23ff6729ee664acb66e91f
set security macsec interfaces ge-0/0/0 connectivity-association ccc-macsec
```

Step-by-Step Procedure

In this example, the traffic between the users that often exchange financially-sensitive data is sent between the sites on a CCC through the MPLS cloud. MACsec is enabled on the CCC by configuring a MACsec connectivity association on the interfaces on the site A and site B CE routers that connect to the MPLS PE routers. The connectivity associations must have matching connectivity-association names (in this example, **ccc-macsec**), matching CKNs (in this example, **37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311**), and CAKs (in this example, **228ef255aa23ff6729ee664acb66e91f**) in order to establish a MACsec-secure connection.

To enable MACsec on the CCC connecting site A to site B, perform the following procedure on the site A CE router:

1. Create the connectivity association named **ccc-macsec**, and configure the MACsec security mode as `static-cak`:

   ```
   [edit security macsec]
   user@router-CE-A# set connectivity-association ccc-macsec security-mode static-cak
   ```

2. Create the pre-shared key by configuring the CKN and CAK:

   ```
   [edit security macsec]
   user@router-CE-A# set connectivity-association ccc-macsec pre-shared-key ckn
   37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311
   user@router-CE-A# set connectivity-association ccc-macsec pre-shared-key cak
   228ef255aa23ff6729ee664acb66e91f
   ```

3. Assign the connectivity association to the interface connecting to the PE1 router:

   ```
   [edit security macsec]
   user@router-CE-A# set interfaces ge-0/0/0 connectivity-association ccc-macsec
   ```

   This completes the steps for configuring the connectivity association on one end of the CCC. MACsec is not enabled until a connectivity association with matching pre-shared keys is enabled on the opposite end of a link, which in this case is the interface on the site B CE router, of the CCC. The process for configuring the connectivity association on the site B CE router is described in the following section.

Results

Display the results of the configuration:

```
user@router-CE-A> show configuration
```

```
security {
    macsec {
        connectivity-association {
            ccc-macsec {
                pre-shared-key {
                    cak "$9$rJ-lWLxNdw24Xxik.PQzreK"; ## SECRET-DATA
                    ckn 37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311;
```

```
                }
                security-mode {
                    static-cak;
                }
            }
        }
        interfaces {
            ge-0/0/0 {
                connectivity-association {
                    ccc-macsec;
                }
            }
        }
    }
}
```

Configuring MACsec on the Site B CE Router to Secure Traffic to Site A

CLI Quick Configuration

```
[edit]
set security macsec connectivity-association ccc-macsec security-mode static-cak
set security macsec connectivity-association ccc-macsec pre-shared-key ckn
37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311
set security macsec connectivity-association ccc-macsec pre-shared-key cak
228ef255aa23ff6729ee664acb66e91f
set security macsec interfaces xe-0/1/0 connectivity-association ccc-macsec
```

Step-by-Step Procedure

Traffic travels from site B to site A over the MPLS network using a CCC. MACsec is enabled on the CCC by configuring a MACsec connectivity association on the interfaces on the site A and site B CE routers that connect to the MPLS PE routers. The connectivity associations must have matching connectivity-association names (in this example, **ccc-macsec**), matching CKNs (**37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311**), and matching CAKs (**228ef255aa23ff6729ee664acb66e91f**) in order to establish a MACsec-secure connection.

To enable MACsec on the CCC connecting site B to site A, perform the following procedure on the site B CE router:

1. Create the connectivity association named **ccc-macsec**, and configure the MACsec security mode as `static-cak`:

   ```
   [edit security macsec]
   user@router-CE-B# set connectivity-association ccc-macsec security-mode static-cak
   ```

2. Create the pre-shared key by configuring the CKN and CAK:

   ```
   [edit security macsec]
   user@router-CE-B# set connectivity-association ccc-macsec pre-shared-key ckn
   37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311
   user@router-CE-B# set connectivity-association ccc-macsec pre-shared-key cak
   228ef255aa23ff6729ee664acb66e91f
   ```

3. Assign the connectivity association to the interface connecting to router PE2:

   ```
   [edit security macsec]
   user@router-CE-B# set interfaces xe-0/1/0 connectivity-association ccc-macsec
   ```

   MACsec is enabled for the CCC after the pre-shared keys are exchanged, which is shortly after this procedure is completed.

Results

Display the results of the configuration:

```
user@router-CE-B> show configuration
```

```
security {
    macsec {
        connectivity-association {
            ccc-macsec {
                security-mode {
                    static-cak;
                }
                pre-shared-key {
                    cak "$9$rJ-lWLxNdw24Xxik.PQzreK"; ## SECRET-DATA
                    ckn 37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311;
```

```
                }
            }
        }
        interfaces {
            xe-0/1/0 {
                connectivity-association {
                    ccc-macsec;
                }
            }
        }
    }
}
```

Configuring VLANs to Direct Traffic onto the MACsec-Secured CCC

IN THIS SECTION

This section explains how to configure VLANs on the site A and site B CE routers. The purpose of the VLANs is to direct traffic that you want to be MACsec-secured onto the MACsec-secured CCC.

Configuring the Bridge Domain to Direct Traffic to the MACsec CCC on the Site A CE Router

CLI Quick Configuration

```
[edit]
set interfaces ge-0/0/0 unit 0 encapsulation vlan-bridge
set interfaces ge-0/0/0 unit 0 family bridge
set interfaces ge-0/0/2 unit 0 encapsulation vlan-bridge
set interfaces ge-0/0/2 unit 0 family bridge
set bridge-domains macsec vlan-id 50
set bridge-domains macsec domain-type bridge
set bridge-domains macsec vlan-id all
set bridge-domains macsec interface ge-0/0/0
set bridge-domains macsec interface ge-0/0/2
set interfaces irb vlan-id 50 family inet address 5.5.5.1/24
```

Step-by-Step Procedure

To create a bridge domain (VLAN ID 50) that directs traffic from the user at site A onto the MACsec-secured CCC:

1. Configure the ge-0/0/0 interface with VLAN encapsulation and the bridge family.

   ```
   user@router-CE-A# set interfaces ge-0/0/0 unit 0 encapsulation vlan-bridge
   user@router-CE-A# set interfaces ge-0/0/0 unit 0 family bridge vlan-id 50
   ```

2. Configure the ge-0/0/2 interface with VLAN encapsulation and the bridge family.

   ```
   [edit]
   user@router-CE-A#set interfaces ge-0/0/2 unit 0 encapsulation vlan-bridge
   user@router-CE-A#set interfaces ge-0/0/2 unit 0 family bridge vlan-id 50
   ```

3. Define the macsec bridge domain and associate the interfaces, ge-0/0/0 and ge-0/0/2, with the bridge domain.

   ```
   [edit]
   user@router-CE-A# set bridge-domains macsec vlan-id 50
   user@router-CE-A# set bridge-domains macsec domain-type bridge
   user@router-CE-A# set bridge-domains macsec interface ge-0/0/0
   user@router-CE-A# set bridge-domains macsec interface ge-0/0/2
   ```

4. Create the IP address for the macsec bridge domain:

   ```
   [edit]
   user@router-CE-A# set interfaces irb vlan-id 50 family inet address 5.5.5.1/24
   ```

Results

Display the results of the configuration:

```
user@router-CE-A> show configuration
interfaces {
    ge-0/0/0 {
        unit 0 {
            encapsulation vlan-bridge;
                family bridge {
                vlan-id 50;
            }
        }
    }
    ge-0/0/2 {
        unit 0 {
            encapsulation vlan-bridge;
                family bridge {
                vlan-id 50;
            }
        }
    }
    irb {
        vlan-id 50 {
            family inet address 5.5.5.1/24;
        }
    }
}
bridge-domains {
    macsec {
        domain-type bridge;
        vlan-id 50;
        interface ge-0/0/0;
        interface ge-0/0/2;
    }
}
```

Configuring the Bridge Domain to Direct Traffic to the MACsec CCC on the Site B CE Router

CLI Quick Configuration

```
[edit]
set interfaces xe-0/1/0 unit 0 encapsulation vlan-bridge
set interfaces xe-0/1/0 unit 0 family bridge
set interfaces ge-0/0/2 unit 0 encapsulation vlan-bridge
set interfaces ge-0/0/2 unit 0 family bridge
set bridge-domains macsec vlan-id 50
set bridge-domains macsec domain-type bridge
set bridge-domains macsec vlan-id all
set bridge-domains macsec interface ge-0/0/2
set bridge-domains macsec interface xe-0/1/0
set interfaces irb vlan-id 50 family inet address 5.5.5.2/24
```

Step-by-Step Procedure

To create a bridge domain (VLAN ID 50) to direct traffic for the user at site B onto the MACsec-secured CCC:

1. Configure the xe-0/1/0 interface with VLAN encapsulation and the bridge family.

   ```
   user@router-CE-A# set interfaces xe-0/1/0 unit 0 encapsulation vlan-bridge
   user@router-CE-A# set interfaces xe-0/1/0 unit 0 family bridge vlan-id 50
   ```

2. Configure the ge-0/0/2 interface with VLAN encapsulation and the bridge family.

   ```
   [edit]
   user@router-CE-A#set interfaces ge-0/0/2 unit 0 encapsulation vlan-bridge
   user@router-CE-A#set interfaces ge-0/0/2 unit 0 family bridge vlan-id 50
   ```

3. Define the macsec bridge domain and associate the interfaces, xe-0/1/0 and ge-0/0/2, with the bridge domain.

   ```
   [edit]
   user@router-CE-A# set bridge-domains macsec vlan-id 50
   user@router-CE-A# set bridge-domains macsec domain-type bridge
   ```

```
user@router-CE-A# set bridge-domains macsec interface xe-0/1/0
user@router-CE-A# set bridge-domains macsec interface ge-0/0/2
```

4. Create the IP address for the macsec bridge domain:

```
[edit]
user@router-CE-A# set interfaces irb vlan-id 50 family inet address 5.5.5.2/24
```

Results

Display the results of the configuration:

```
user@router-CE-B> show configuration
interfaces {
    ge-0/0/2 {
        unit 0 {
            encapsulation vlan-bridge;
                family bridge {
                vlan-id 50;
            }
        }
    }
    xe-0/1/0 {
        unit 0 {
            encapsulation vlan-bridge;
                family bridge {
                vlan-id 50;
            }
        }
    }
    irb {
        vlan-id 50 {
            family inet address 5.5.5.2/24;
        }
    }
}
bridge-domains {
    macsec {
        domain-type bridge;
        vlan-id 50;
        interface xe-0/1/0;
        interface ge-0/0/2;
    }
}
```

Verification

IN THIS SECTION

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

Verifying the MACsec Connection

Purpose

Verify that MACsec is operational on the CCC.

Action

Enter the `show security macsec connections` command on one or both of the customer edge (CE) switches.

```
user@router-CE-A> show security macsec connections
Interface name: ge-0/0/0
        CA name: ccc-macsec
        Cipher suite: GCM-AES-128   Encryption: on
        Key server offset: 0        Include SCI: no
        Replay protect: off         Replay window: 0
```

```
          Outbound secure channels
            SC Id: 00:19:E2:53:CD:F3/1
            Outgoing packet number: 9785
            Secure associations
            AN: 0 Status: inuse Create time: 2d 20:47:54
          Inbound secure channels
            SC Id: 00:23:9C:0A:53:33/1
            Secure associations
            AN: 0 Status: inuse Create time: 2d 20:47:54
```

Meaning

The `Interface name:` and `CA name:` outputs shows that the ccc-macsec connectivity association is operational on interface ge-0/0/0. The output does not appear when the connectivity association is not operational on the interface.

For additional verification that MACsec is operational on the CCC, you can also enter the `show security macsec connections` command on the other CE switch.

Verifying That MACsec-Secured Traffic Is Traversing the CCCs

Purpose

Verify that traffic traversing the CCC is MACsec-secured.

Action

Enter the `show security macsec statistics` command on one or both of the CE switches.

```
user@router-CE-A> show security macsec statistics
Interface name: ge-0/0/0
    Secure Channel transmitted
        Encrypted packets: 9784
        Encrypted bytes:   2821527
        Protected packets: 0
        Protected bytes:   0
    Secure Association transmitted
        Encrypted packets: 9784
        Protected packets: 0
    Secure Channel received
        Accepted packets:  9791
```

```
          Validated bytes:   0
          Decrypted bytes:   2823555
      Secure Association received
          Accepted packets:  9791
          Validated bytes:   0
          Decrypted bytes:   2823555
```

Meaning

The `Encrypted packets` line under the `Secure Channel transmitted` output is incremented each time a packet is sent from the interface that is secured and encrypted by MACsec. The `Encrypted packets` output shows that 9784 encrypted and secured packets have been transmitted from interface ge-0/0/0. MACsec-secured traffic is, therefore, being sent on interface ge-0/0/0.

The `Accepted packets` line under the `Secure Association received` output is incremented each time a packet that has passed the MACsec integrity check is received on the interface. The `Decrypted bytes` line under the `Secure Association received` output is incremented each time an encrypted packet is received and decrypted. The output shows that 9791 MACsec-secured packets have been received on interface ge-0/0/0, and that 2823555 bytes from those packets have been successfully decrypted. MACsec-secured traffic is, therefore, being received on interface ge-0/0/0.

For additional verification, you can also enter the `show security macsec statistics` command on the other CE switch.

Verifying That the MPLS and CCC Protocols Are Enabled on the Provider Edge and Provider Switch Interfaces

Purpose

Verify that MPLS is enabled on the correct interfaces for the PE and provider switches.

Action

Enter the `show interfaces terse` command on both of the PE routers and the provider switch:

```
user@router-PE1> show interfaces terse

Interface               Admin Link Proto    Local                 Remote
ge-0/0/0                up    up
ge-0/0/0.0              up    up   ccc
ge-0/0/1                up    up
```

```
ge-0/0/1.0              up    up   inet     10.1.5.2/24
                                   mpls
<some output removed for brevity>
```

```
user@router-P> show interfaces terse

Interface               Admin Link Proto    Local                 Remote
xe-0/0/0                up    up
xe-0/0/0.0              up    up   inet     10.1.9.1/24
                                   mpls
ge-0/0/10               up    up
ge-0/0/10.0             up    up   inet     10.1.5.1/24
                                   mpls
<some output removed for brevity>
```

```
user@router-PE2> show interfaces terse

Interface               Admin Link Proto    Local                 Remote
xe-0/1/0                up    up
xe-0/1/0.0              up    up   inet     10.1.9.2/24
                                   mpls
xe-0/1/1                up    up
xe-0/1/1.0              up    up   ccc
<some output removed for brevity>
```

Meaning

The output confirms that the MPLS protocol is up for the provider switch interfaces passing MPLS traffic—xe-0/0/0 and ge-0/0/10—and on the PE router interfaces passing MPLS traffic, which is interface ge-0/0/1 on the PE1 switch and interface xe-0/1/0 on the PE2 router.

The output also confirms that CCC is enabled on the PE router interfaces facing the CE switches, which are interface ge-0/0/0 on the PE1 switch and interface xe-0/1/1 on the PE2 router.

Verifying MPLS Label Operations

Purpose

Verify which interface is being used as the beginning of the CCC and which interface is being used to push the MPLS packet to the next hop.

Action

Enter the `show route forwarding-table family mpls` on one or both of the PE routers.

```
user@router-PE1> show route forwarding-table family mpls

Routing table: default.mpls
MPLS:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    dscd    50     1
0                  user     0                    recv    49     4
1                  user     0                    recv    49     4
2                  user     0                    recv    49     4
13                 user     0                    recv    49     4
299856             user     0                   Pop    1327     2 ge-0/0/0.0
ge-0/0/0.0  (CCC)  user     0 10.1.5.1          Push 299952  1328     2 ge-0/0/1.0
```

Meaning

This output confirms that the CCC is configured on interface ge-0/0/0.0. The switch receives ingress traffic on ge-0/0/1.0 and pushes label 299952 onto the packet, which exits the switch through interface ge-0/0/1.0. The output also shows that when the switch receives an MPLS packet with label 299856, it pops the label and sends the packet out through interface ge-0/0/0.0

For further verification of MPLS label operations, enter the `show route forwarding-table family mpls` on the other PE router.

Verifying the Status of the MPLS CCCs

Purpose

Verify that the MPLS CCCs are operating.

Action

Enter the `show connections` command on the PE routers.

```
user@router-PE1> show connections
CCC and TCC connections [Link Monitoring On]
Legend for status (St):             Legend for connection types:
 UN -- uninitialized                 if-sw:  interface switching
 NP -- not present                   rmt-if: remote interface switching
 WE -- wrong encapsulation           lsp-sw: LSP switching
 DS -- disabled                      tx-p2mp-sw: transmit P2MP switching
 Dn -- down                          rx-p2mp-sw: receive P2MP switching
 -> -- only outbound conn is up     Legend for circuit types:
 <- -- only inbound  conn is up      intf -- interface
 Up -- operational                   oif  -- outgoing interface
 RmtDn -- remote CCC down            tlsp -- transmit LSP
 Restart -- restarting               rlsp -- receive LSP


Connection/Circuit                Type        St      Time last up     # Up trans
ge-1-to-pe2                       rmt-if      Up      May 30 19:01:45         1
  ge-0/0/0.0                        intf      Up
  lsp_to_pe2_xe1                    tlsp      Up
  lsp_to_pe1_ge0                    rlsp      Up
```

```
user@router-PE2> show connections

CCC and TCC connections [Link Monitoring On]
Legend for status (St):             Legend for connection types:
 UN -- uninitialized                 if-sw:  interface switching
 NP -- not present                   rmt-if: remote interface switching
 WE -- wrong encapsulation           lsp-sw: LSP switching
 DS -- disabled                      tx-p2mp-sw: transmit P2MP switching
 Dn -- down                          rx-p2mp-sw: receive P2MP switching
 -> -- only outbound conn is up     Legend for circuit types:
 <- -- only inbound  conn is up      intf -- interface
 Up -- operational                   oif  -- outgoing interface
 RmtDn -- remote CCC down            tlsp -- transmit LSP
 Restart -- restarting               rlsp -- receive LSP
```

```
Connection/Circuit                Type        St      Time last up     # Up trans
xe-1-to-pe1                       rmt-if      Up      May 30 09:39:15         1
  xe-0/1/1.0                        intf      Up
  lsp_to_pe1_ge0                    tlsp      Up
  lsp_to_pe2_xe1                    rlsp      Up
```

The `show connections` command displays the status of the CCC connections. This output verifies that the CCC interfaces and their associated transmit and receive LSPs are `Up` on both PE routers.

Verifying OSPF Operation

Purpose

Verify that OSPF is running.

Action

Enter the `show ospf neighbor` command the provider or the PE routers, and check the `State` output.

```
user@router-P> show ospf neighbor

Address          Interface              State     ID               Pri  Dead
10.1.5.2         ge-0/0/10.0            Full      130.1.1.1        128    33
10.1.9.2         xe-0/0/0.0             Full      130.1.1.3        128    38
```

Meaning

The `State` output is `Full` on all interfaces using OSPF, so OSPF is operating.

For further verification on OSPF, enter the `show ospf neighbor` command on the PE routers in addition to the provider switch.

Verifying the Status of the RSVP Sessions

Purpose

Verify the status of the RSVP sessions.

Action

Enter the `show rsvp session` command, and verify that the state is up for each RSVP session.

```
user@router-P> show rsvp session

Ingress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 2 sessions
To              From            State   Rt Style Labelin Labelout LSPname
130.1.1.1       130.1.1.3       Up       0  1 FF  299936   299856 lsp_to_pe1_ge0
130.1.1.3       130.1.1.1       Up       0  1 FF  299952   299840 lsp_to_pe2_xe1
Total 2 displayed, Up 2, Down 0
```

Meaning

The `State` is `Up` for all connections, so RSVP is operating normally.

For further verification, enter the `show rsvp session` on the PE routers in addition to the provider router.

6

PART

MAC Limiting and Move Limiting

CHAPTER 13

MAC Limiting and Move Limiting Configurations and Examples

IN THIS CHAPTER

Understanding MAC Limiting and MAC Move Limiting

IN THIS SECTION

MAC limiting protects against flooding of the Ethernet switching table, and is enabled on Layer 2 interfaces (ports). MAC move limiting detects MAC movement and MAC spoofing on access interfaces. It is enabled on VLANs.

- *MAC limiting* enhances port security by limiting the number of MAC addresses that can be learned within a VLAN. Limiting the number of MAC addresses protects the switch from flooding of the Ethernet switching table (also known as the MAC forwarding table or Layer 2 forwarding table). Flooding occurs when the number of new MAC addresses that are learned causes the Ethernet switching table to overflow, and previously learned MAC addresses are flushed from the table. The switch then reverts to flooding the previously-learned MAC addresses, which can impact performance and introduce security vulnerabilities.

- *MAC move limiting* provides additional security by controlling the number of MAC address moves that are allowed in a VLAN within one second. A MAC address move occurs when the switch receives a packet with a source MAC address that has already been learned by the switch, but on a different interface. The Ethernet switching table is then updated to reflect the association of the MAC address with the new interface. Because the Ethernet switching table must be updated for each MAC address move, frequent move events can lead to exhaustion of the switch's processing resources. This might occur as the result of a MAC spoofing attack or a loop in the network.

MAC Limiting

With MAC limiting, you limit the MAC addresses that can be learned on Layer 2 access interfaces by either limiting the number of MAC addresses or by specifying allowed MAC addresses:

- Limiting the number of MAC addresses—You configure the maximum number of MAC addresses that can be dynamically learned (added to the Ethernet switching table) per interface. You can specify that incoming packets with new MAC addresses be ignored, dropped, or logged when the limit is exceeded. You can also specify that the interface be shut down or temporarily disabled.

  NOTE: Static MAC addresses do not count toward the limit you specify for dynamic MAC addresses.

- Specifying allowed MAC addresses—You configure the allowed MAC addresses for an interface. Any MAC address that is not in the list of configured addresses is not learned, and the switch logs an appropriate message. An allowed MAC address is bound to a VLAN so that the address is not registered outside the VLAN. If an allowed MAC setting conflicts with a dynamic MAC setting, the allowed MAC setting takes precedence.

MAC limiting is configured on Layer 2 interfaces. You can specify the maximum number of dynamic MAC addresses that can be learned on a single interface, all interfaces, or a specific interface on the basis of its membership within a VLAN (VLAN membership MAC limit).

When you are configuring the maximum MAC limit for an interface, you can choose the action that occurs on incoming packets when the MAC limit is exceeded. You can specify that incoming packets be ignored, dropped, or logged when the limit is exceeded. You can also specify that the interface be shut down or temporarily disabled.

MAC limiting is not enabled by default. The configured value remains active until it is replaced or cleared, and persists through device reboot. For additional information about configuring MAC limit for an interface on a device that supports ELS, see Configuring MAC Limiting (ELS). For additional information about configuring MAC limit for an interface on a device that does not support Enhanced Layer 2 Software (ELS), see "Configuring MAC Limiting (non-ELS)" on page 374.

See Using the Enhanced Layer 2 Software CLI for additional information on ELS.

MAC Move Limiting

With MAC move limiting, you limit the number of times a MAC address can move to a new interface within one second. When MAC move limiting is configured, MAC address movements are tracked by the switch. The first time a MAC address moves is always considered a good move and will not count toward the configured MAC move limit. Monitoring of MAC address moves comes into effect after the first move, even if the MAC move limit is configured as 1.

You configure MAC move limiting on a per-VLAN basis. Although you enable this feature on VLANs, the MAC move limit applies to the number of movements for each individual MAC address rather than the total number of MAC address moves in the VLAN. For example, if the MAC move limit is set to 1, the switch allows an unlimited number of MAC address movements within the VLAN as long as the same MAC address does not move more than once within a second.

You can configure an action to be taken if the MAC address move limit is exceeded. You can specify that incoming packets be ignored, dropped, or logged when the limit is exceeded. You can also specify that the interface be shut down or temporarily disabled.

MAC move limiting is not enabled by default. For additional information about configuring MAC move limiting on a device that does not support ELS, see Configuring MAC Move Limiting (non-ELS). For additional information about configuring MAC move limiting on a device that supports ELS, see "Configuring MAC Move Limiting (ELS)" on page 408.

Actions for MAC Limiting and MAC Move Limiting

You can choose to have one of the following actions performed when the MAC limit or the MAC move limit is exceeded:

- `drop`—Drop the packet, but do not generate an alarm.

- `drop-and-log`—Drop the packet and generate an alarm, an SNMP trap, or system log entry.

- `log`—Do not drop the packet but generate an alarm, an SNMP trap, or a system log entry.

- `none`—Forward packets with new source MAC addresses, and learn the new source MAC address.

- `shutdown`—Disable the interface in the VLAN and generate an alarm, an SNMP trap, or a system log entry.

- `vlan-member-shutdown`—(EX9200 only) Starting in Junos OS Release 15.1 for MAC Limiting and MAC Move Limiting on EX9200 Switches, the `vlan-member-shutdown` statement is supported to block an interface on the basis of its membership in a specific VLAN and generate an alarm, an SNMP trap, or a system log entry.

In the event of shutdown, you can configure the switch to automatically restore the disabled interfaces to service after a specified period of time. To configure autorecovery on a device that supports ELS, see "Configuring Autorecovery for Port Security Events" on page 758. To configure autorecovery on a device that does not support ELS, see "Configuring Autorecovery for Port Security Events" on page 758.

NOTE: To view system log entries for mac limit features, you will need to configure system logging with severity as log notice. See Overview of System Logging.

NOTE: If you do not configure the switch for autorecovery from the disabled condition, you can bring up the disabled interfaces by running one of the following commands:

- (For devices that support ELS)— `clear ethernet-switching recovery-timeout`

- (For devices that do not support ELS)— `clear ethernet-switching port-error`

NOTE: With existing dot1x sessions:

- When we set the MAC limit for the first time, existing dot1x sessions are cleared and port moves to Connecting state.

- When we increase the MAC limit, sessions are not cleared and port remains in Authenticated state.

- When we decrease the MAC limit or delete the switch-options configs, existing dot1x sessions are cleared and port moves to Connecting state.

In summary, when interface MAC limit configured is lower than the number of MACs learnt, MAC flush happens. When interface MAC limit configured is greater than the number of MACs learnt, there is no impact

NOTE: Commit checks have been introduced to prevent misconfiguration. Only interfaces configured for L2 will be allowed to be configured under any of these hierarchies.

- set routing-instances <routing-instance-name> vlans <vlans-name> switch-options interface <interface-name>
- set routing-instances <routing-instance-name> bridge-domains <bridge-domain-name> bridge-options interface <interface-name>
- set vlans <vlans-name> switch-options interface <interface-name>
- set bridge-domains <bridge-domain-name> bridge-options interface <interface-name>
- set vlans <vlans-name> switch-options mac-move-limit interface <interface-name>

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
15.1	Starting in Junos OS Release 15.1 for MAC Limiting and MAC Move Limiting on EX9200 Switches, the `vlan-member-shutdown` statement is supported to block an interface on the basis of its membership in a specific VLAN and generate an alarm, an SNMP trap, or a system log entry.

RELATED DOCUMENTATION

Port Security Features | 2

Configuring MAC Limiting (ELS)

Configuring Autorecovery for Port Security Events | 758

Adding a Static MAC Address Entry to the Ethernet Switching Table on a Switch with ELS Support

Understanding MAC Limiting on Layer 3 Routing Interfaces

IN THIS SECTION

Overview

The MAC limiting feature provides a mechanism for limiting MAC addresses on devices that are connected to a Layer 3 routed Gigabit Ethernet (GE), Fast Ethernet (FE), or 10 Gigabit Ethernet (XE) interface. With MAC filters, you can allow traffic with specific source MAC. Software-based MAC limiting is supported. MAC limiting is applicable only on interfaces with plain Ethernet or VLAN tagged encapsulation.

Both the physical interface level `source-address-filter` and *logical interface* level `accept-source-mac` configurations are supported on SRX100, SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, and SRX650 devices. (Platform support depends on the Junos OS release in your installation.) The following considerations apply when you configure the `source-address-filter` and `accept-source-mac` statements:

- If only the logical level `accept-source-mac` statement is configured, traffic from only those configured MAC addresses will be allowed on the logical interface.
- If only the physical interface level `source-address-filter` statement is configured, the physical interface's *allowed* MAC addresses are also considered the *allowed* addresses for all the logical interfaces belonging to the physical interface. Incoming packets from any other source MAC addresses are dropped.
- If the physical interface level `source-address-filter` is configured under `gigether-options` (or `fastether-options`) and `accept-source-mac` is configured for one or more of its logical interfaces or VLANs, the allowed list of addresses is a combination of MAC addresses specified in both the statements. For logical interfaces and VLANs where the `accept-source-mac` statement is not configured, the physical interface's *allowed* list of addresses is considered.

You can configure an interface to receive packets from specific MAC addresses. To do this, specify the MAC addresses in the `source-address-filter` or `accept-source-mac` statements:

- `Logical level MAC filter configuration on an untagged interface`

```
ge-0/0/10 {
    unit 0 {
        accept-source-mac {
            mac-address 00:22:33:44:55:66;
            mac-address 00:26:88:e9:a3:01;
        }
        family inet {
            address 60.60.60.1/24;
        }
    }
}
```

- Physical level MAC filter configuration on an untagged interface

```
ge-0/0/10 {
    gigether-options {
        source-address-filter {
            00:55:55:55:55:66;
            00:26:88:e9:a3:01;
        }
    }
    unit 0 {
        family inet {
            address 60.60.60.1/24;
        }
    }
}
```

- Physical and logical level MAC filter configurations on a tagged interface

```
ge-0/0/10 {
    vlan-tagging;
    gigether-options {
        source-address-filter {
            00:26:88:e9:a3:01;
        }
    }
    unit 0 {
        vlan-id 40;
        accept-source-mac {
            mac-address 00:22:33:44:55:66;
        }
        family inet {
            address 40.40.40.1/24;
        }
    }
    unit 1 {
        vlan-id 60;
        accept-source-mac {
            mac-address 00:55:55:55:55:66;
        }
        family inet {
            address 60.60.60.1/24;
        }
    }
}
```

NOTE: On untagged Gigabit Ethernet interfaces, you must not configure the `source-address-filter` and the `accept-source-mac` statements simultaneously. If these statements are configured for the same interfaces at the same time, an error message appears. However, in the case of tagged VLANs, both these statements can be configured simultaneously, if no identical MAC addresses are specified.

Limitations

The following limitations apply to MAC limiting support on Layer 3 routed GE, AE, FE, or XE interfaces:

- You can configure only 32 MAC addresses per device (except on aggregated Ethernet interfaces, where the limit is 64 addresses per logical interface).

- Only software-based MAC filtering is supported. Software-based MAC filtering impacts performance. The performance impact is proportional to the number of MAC addresses configured.

- MAC-based policer or rate limiting is not supported.

- You cannot configure broadcast or multicast address in the `source-address-filter` statement.

- MAC filtering is not supported on aggregated Ethernet (AE) interfaces (it *is* supported on some platforms; see Feature Explorer for platform specifics); or on Fabric Ethernet, Point-to-Point Protocol over Ethernet (PPPoE), *Routed VLAN interface* (*RVI*), or VLAN interfaces.

  MAC filtering is not supported on chassis clusters.

- If you configure MAC filtering on the AE interface, you must configure the interface with `accept-source-mac` (that is, not with `source-address-filter`) and with `family ethernet-switching`.

RELATED DOCUMENTATION

Understanding Interface Logical Properties

Understanding and Using Persistent MAC Learning

IN THIS SECTION

Understanding Persistent MAC Learning (Sticky MAC)

Persistent MAC learning, also known as sticky MAC, is a port security feature that enables an interface to retain dynamically learned MAC addresses when the switch is restarted or if the interface goes down and is brought back online.

Persistent MAC address learning is disabled by default. You can enable persistent MAC address learning in conjunction with MAC limiting to restrict the number of persistent MAC addresses. You enable this feature on interfaces.

Configure persistent MAC learning on an interface to:

- Prevent traffic losses for trusted workstations and servers because the interface does not have to relearn the addresses from ingress traffic after a restart.

- Protect the switch against security attacks. Use persistent MAC learning in combination with MAC limiting to protect against attacks, such as Layer 2 denial-of-service (DoS) attacks, overflow attacks on the Ethernet switching table, and DHCP starvation attacks, by limiting the MAC addresses allowed while still allowing the interface to dynamically learn a specified number of MAC addresses. The interface is secured because after the limit has been reached, additional devices cannot connect to the port.

By configuring persistent MAC learning along with MAC limiting, you enable interfaces to learn MAC addresses of trusted workstations and servers from the time when you connect the interface to your network until the limit for MAC addresses is reached, and ensure that after this limit is reached, new devices will not be allowed to connect to the interface even if the switch restarts. As an alternative to using persistent MAC learning with MAC limiting, you can statically configure each MAC address on each port or allow the port to continuously learn new MAC addresses after restarts or interface-down events. Allowing the port to continuously learn MAC addresses represents a security risk.

NOTE:

- While a switch is restarting or an interface is coming back up, there might be a short delay before the interface can learn more MAC addresses. This delay occurs while the system re-enters previously learned persistent MAC addresses into the forwarding database for the interface.
- From Junos OS Release 22.4R1 onwards, you can enable persistent MAC learning on both trunk (VLAN-tagged) and access interfaces.

Consider the following configuration guidelines when configuring persistent MAC learning:

- You cannot enable persistent MAC learning on an interface on which 802.1x authentication is configured.
- You cannot enable persistent MAC learning on an interface that is part of a redundant trunk group.
- You cannot enable persistent MAC learning on an interface on which **no-mac-learning** is enabled.

TIP: If you move a device within your network that has a persistent MAC address entry on the switch, use the `clear ethernet-switching table persistent-learning` *<interface | mac-address>* command to clear the persistent MAC address entry from the interface. If you move the device and do not clear the persistent MAC address from the original port it was learned on, then the new port will not learn the MAC address of the device and the device will not be able to connect. If the original port is down when you move the device, then the new port will learn the MAC address and the device can connect. However, if you do not clear the persistent MAC address on the original port, then when the port restarts, the system reinstalls the persistent MAC address in the forwarding table for that port. If this occurs, the persistent MAC address is removed from the new port and the device loses connectivity.

Configuring Persistent MAC Learning (ELS)

NOTE: This section describes using Junos OS with support for the Enhanced Layer 2 Software (ELS). For more information on ELS, see Using the Enhanced Layer 2 Software CLI

To configure persistent MAC learning on an interface and limit the number of allowed MAC addresses:

1. Enable persistent MAC learning on an interface:

   ```
   [edit switch-options]
   user@switch# set interface interface-name  persistent-learning
   ```

2. Configure the MAC limit on an interface, and specify the action that the switch takes after the specified limit is exceeded:

   ```
   [edit switch-options]
   user@switch# set interface interface-name interface-mac-limit limit  packet-action action
   ```

   After you set a new MAC limit for the interface, the system clears existing entries in the MAC address forwarding table associated with the interface.

Values for *action* are:

`drop` Drop packets with new source MAC addresses, and do not learn the new source MAC addresses.

`drop-and-log` (EX Series switches only) Drop packets with new source MAC addresses, and generate an alarm, an SNMP trap, or a system log entry.

`log` (EX Series switches only) Hold packets with new source MAC addresses, and generate an alarm, an SNMP trap, or a system log entry.

`none` (EX Series switches only) Forward packets with new source MAC addresses, and learn the new source MAC address.

`shutdown` (EX Series switches only) Disable the specified interface, and generate an alarm, an SNMP trap, or a system log entry.

TIP: If you move a device within your network that has a persistent MAC address entry on the switch, use the `clear ethernet-switching table persistent-learning` command to clear the persistent MAC address entry from the interface. If you move the device and do not clear the persistent MAC address from the original port it was learned on, then the new port will not learn the MAC address of the device and the device will not be able to connect.

If the original port is down when you move the device, then the new port will learn the MAC address and the device can connect. However, if you do not clear the persistent MAC address on the original port, then when the port restarts, the system reinstalls the persistent MAC address in the forwarding table for that port. If this occurs, the persistent MAC address is removed from the new port and the device loses connectivity.

Configuring Persistent MAC Learning (non-ELS)

Persistent MAC address learning, also known as sticky MAC, is disabled by default. You can enable it to allow dynamically learned MAC addresses to be retained on an interface across restarts of the switch.

NOTE: This section describes using Junos OS without support for the Enhanced Layer 2 Software (ELS). For more information on ELS, see Using the Enhanced Layer 2 Software CLI

Use persistent MAC address learning to:

- Help prevent traffic losses for trusted workstations and servers because the interface does not have to relearn the addresses from ingress traffic after a restart.

- Protect the switch against security attacks—use persistent MAC learning in combination with MAC limiting to protect against attacks while still avoiding the need to statically configure MAC addresses. When the initial learning of MAC addresses up to the number specified by the MAC limit is done, new addresses will not be allowed even after a reboot. The port is secured because after the limit has been reached, additional devices cannot connect to the interface.

The first devices that send traffic after you connect are learned during the initial connection period. You can monitor the MAC addresses and provide the same level of security as if you statically configured each MAC address on each interface, except with less manual effort. Persistent MAC learning also helps prevent traffic loss for trusted workstations and servers because the interface does not have to relearn the addresses from ingress traffic.

To configure persistent MAC learning on an interface and limit the number of allowed MAC addresses:

1. Enable persistent MAC learning on an interface:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface interface-name  persistent-learning
```

2. Configure the MAC limit on an interface, and specify the action that the switch takes after the specified limit is exceeded:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface interface-name mac-limit limit  packet-action action
```

After you set a new MAC limit for the interface, the system clears existing entries in the MAC address forwarding table associated with the interface.

Verifying That Persistent MAC Learning Is Working Correctly

IN THIS SECTION

Purpose

Verify that persistent MAC learning, also known as sticky MAC, is working on the interface. Persistent MAC learning allows retention of dynamically learned MAC addresses on an interface across restarts of the switch (or if the interface goes down).

Action

Display the MAC addresses that have been learned. The following sample output shows the results when persistent MAC learning is enabled on interface ge-0/0/42:

show ethernet-switching table persistent-mac

```
user@switch> show ethernet-switching table
Ethernet-switching table: 8 entries, 2 learned, 5 persistent entries
  VLAN              MAC address       Type         Age Interfaces
  default           *                 Flood          - All-members
  default           00:10:94:00:00:02 Persistent        0 ge-0/0/42.0
  default           00:10:94:00:00:03 Persistent        0 ge-0/0/42.0
  default           00:10:94:00:00:04 Persistent        0 ge-0/0/42.0
  default           00:10:94:00:00:05 Persistent        0 ge-0/0/42.0
  default           00:10:94:00:00:06 Persistent        0 ge-0/0/42.0
  default           00:21:59:c8:0c:50 Learn             0 ae0.0
  default           02:21:59:c8:0c:44 Learn             0 ae0.0
```

Meaning

The sample output shows that learned MAC addresses are stored in the Ethernet switching table as persistent entries. If the switch is rebooted or the interface goes down and comes back up, these addresses will be restored to the table.

SEE ALSO

Configuring Port Security (non-ELS) | 11
Example: Configuring Port Security (non-ELS) | 15

Configuring MAC Limiting

IN THIS SECTION

Configuring MAC Limiting (ELS)

IN THIS SECTION

This topic describes the different ways of configuring a limitation on MAC addresses in packets that are received and forwarded by the device.

NOTE: The tasks presented in this section uses Junos OS for EX Series switches, QFX3500 and QFX3600 switches, and PTX Series routers that support the Enhanced Layer 2 Software (ELS) configuration style. See Using the Enhanced Layer 2 Software CLI for more information about ELS configurations.

- For information on configuring an interface to automatically recover from a shutdown caused by MAC limiting, see "Configuring Autorecovery for Port Security Events" on page 758. If you do not configure the device for autorecovery from the disabled condition, you can bring up the disabled interfaces by running the `clear ethernet-switching recovery-timeout` command.

The different ways of setting a MAC limit are described in the following sections:

Platform-Specific Configuring Global MAC Limit Behavior

Table 15: Platform-Specific Behavior

Platform	Difference
Junos Evolved ACX7000 series	ACX7000 currently does not support packet-action Default MAC numbers are not applicable for ACX7000

Limiting the Number of MAC Addresses Learned by an Interface

NOTE: On PTX Series routers, you can limit the number of MAC addresses learned by an interface only.

To secure a port, you can set the maximum number of MAC addresses that can be learned by an interface.

- Set the MAC limit on an interface, and specify an action that the device takes after the specified limit is exceeded.

If you want to set the MAC limit on an interface that is part of the default routing instance:

```
[edit switch-options]
user@switch# set interface interface-name interface-mac-limit limit packet-action action
```

If you want to set the MAC limit on an interface that is part of a routing instance:

```
[edit routing-instances]
user@switch# set routing-instance-name switch-options interface interface-name interface-mac-
limit limit
```

If you want to set the MAC limit on all interfaces that are part of the default routing instance:

```
[edit switch-options]
user@switch# set interface-mac-limit limit
```

If you want to set the MAC limit on all interfaces that are part of a routing instance:

```
[edit routing-instances]
user@switch# set routing-instance-name switch-options interface-mac-limit limit
```

After you set a new MAC limit for the interface, the system clears existing entries in the MAC address forwarding table associated with the interface.

Limiting the Number of MAC Addresses Learned by a VLAN

IN THIS SECTION

To limit the number of MAC addresses learned by a VLAN, perform the following steps:

Platform-Specific Configuring MAC Limit Behavior

Table 16: Platform-Specific Behavior

Platform	Difference
Junos Evolved ACX7000 series	ACX7000 currently does not support packet-action Default MAC numbers are not applicable for ACX7000

Set the maximum number of MAC addresses that can be learned by a VLAN, and specify an action that the device takes after the specified limit is exceeded:

```
[edit vlans]
user@switch# set vlan-name switch-options mac-table-size limit packet-action action
```

Limiting the Number of MAC Addresses Learned by an Interface in a VLAN

To limit the number of MAC addresses learned by an interface in a VLAN, perform the following steps:

1. Set the maximum number of MAC addresses that can be learned by an interface in a VLAN, and specify an action that the device takes after the specified limit is exceeded:

   ```
   [edit vlans]
   user@switch# set vlan-name switch-options interface-mac-limit limit packet-action action
   ```

2. Set the maximum number of MAC addresses that can be learned by one *or* all interfaces in the VLAN, and specify an action that the device takes after the specified limit is exceeded:

   NOTE: If you specify a MAC limit and packet action for all interfaces in the VLAN *and* a specific interface in the VLAN, the MAC limit and packet action specified at the specific interface level takes precedence. Also, at the VLAN interface level, only the `drop` and `drop-and-log` options are supported.

```
[edit vlans]
user@switch# set vlan-name switch-options interface interface-name interface-mac-limit limit
packet-action action
```

```
[edit vlans]
user@switch# set vlan-name switch-options interface-mac-limit limit packet-action action
```

After you set new MAC limits for a VLAN by using the `mac-table-size` statement or for interfaces associated with a VLAN by using the `interface-mac-limit` statement, the system clears the corresponding existing entries in the MAC address forwarding table.

NOTE: On a QFX Series Virtual Chassis, if you include the `shutdown` option at the `[edit vlans` *`vlan-name`* `switch-options interface` *`interface-name`* `interface-mac-limit packet-action]` hierarchy level and issue the `commit` operation, the system generates a commit error. The system does not generate an error if you include the `shutdown` option at the `[edit switch-options interface` *`interface-name`* `interface-mac-limit packet-action]` hierarchy level.

Configuring MAC Limiting (non-ELS)

IN THIS SECTION

This task uses Junos OS for EX Series switches and QFX3500 and QFX3600 switches that does not support the Enhanced Layer 2 Software (ELS) configuration style.

This topic describes various ways of configuring a limitation on MAC addresses in packets that are received and forwarded by the switch.

Before you can change a MAC limit that was previously set for an interface or a VLAN, you must first clear existing entries in the MAC address forwarding table that correspond to the change you want to make. Thus, to change the limit on an interface, first clear the MAC address forwarding table entries for that interface. To change the limit on all interfaces and VLANs, clear all MAC address forwarding table entries. To change the limit on a VLAN, clear the MAC address forwarding table entries for that VLAN.

To clear MAC addresses from the forwarding table:

- Clear MAC address entries from a specific interface (here, the interface is **ge-0/0/1**) in the forwarding table:

  ```
  user@switch> clear ethernet-switching-table interface ge-0/0/1
  ```

- Clear all MAC address entries in the forwarding table:

  ```
  user@switch>clear ethernet-switching-table
  ```

- Clear MAC address entries from a specific VLAN (here, the VLAN is vlan-abc):

  ```
  user@switch> clear ethernet-switching-table vlan vlan-abc
  ```

The different ways of setting a MAC limit are described in the following sections:

Limiting the Number of MAC Addresses That Can be Learned on Interfaces

To configure MAC limiting for port security by setting a maximum number of MAC addresses that can be learned on interfaces.

- Apply the MAC limit on a single interface (here, the interface is ge-0/0/1):

  ```
  [edit ethernet-switching-options secure-access-port]
  user@switch# set interface ge-0/0/1 mac-limit 10
  ```

  When no action is specified for configuring the MAC limit on an interface, the device performs the default action **drop** if the limit is exceeded.

- Apply the MAC limit on a single access interface, on the basis of its membership within a specific VLAN (here, the interface is ge-0/0/1 and the VLAN is v1.

  ```
  [edit ethernet-switching-options secure-access-port]
  user@switch# set interface ge-0/0/1 vlan v1 mac-limit 5
  ```

With this type of configuration, the device drops any additional packets if the limit is exceeded, and also logs a message.

- Apply the limit to all access interfaces:

  ```
  [edit ethernet-switching-options secure-access-port]
  user@switch# set interface all mac-limit 10
  ```

  When no action is specified for configuring the MAC limit on all interfaces, the device performs the default action **drop** if the limit is exceeded:

Specifying MAC Addresses That Are Allowed

You must clear existing entries in the MAC address forwarding table prior to changing the MAC address limit.

To configure MAC limiting for port security by specifying allowed MAC addresses:

- On a single interface (here, the interface is ge-0/0/2):

  ```
  [edit ethernet-switching-options secure-access-port]
  user@switch# set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:80
  user@switch# set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:81
  user@switch# set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:83
  ```

- On all interfaces:

  ```
  [edit ethernet-switching-options secure-access-port]
  user@switch#set interface all allowed-mac 00:05:85:3A:82:80
  user@switch#set interface all allowed-mac 00:05:85:3A:82:81
  user@switch#set interface all allowed-mac 00:05:85:3A:82:83
  ```

Configuring MAC Limiting for VLANs

You must clear existing entries in the MAC address forwarding table before you can change the MAC address limit.

MAC limiting for a VLAN restricts the MAC addresses that can be learned for that VLAN, but does *not* drop the packet. Therefore, setting the MAC limit on a VLAN is not considered a port-security feature.

NOTE: The configuration of specific allowed MAC addresses does not apply to VLANs.

To configure MAC limiting for a VLAN using the CLI:

- Limit the number of dynamic MAC addresses on a VLAN:

  If the MAC limit on a specific VLAN is exceeded, the device logs the MAC addresses of packets that cause the limit to be exceeded. No other action is possible.

  ```
  [edit vlans]
  user@switch# set vlan-abc mac-limit 20
  ```

  NOTE: When you are applying a MAC limit on a VLAN, do not set `mac-limit` to 1 for a VLAN composed of Routed VLAN Interfaces (RVIs) or a VLAN composed of aggregated Ethernet bundles using LACP. In these cases, setting the `mac-limit` to 1 prevents the device from learning MAC addresses other than the automatic addresses:

  - For RVIs, the first MAC address inserted into the forwarding database is the MAC address of the RVI.

  - For aggregated Ethernet bundles using LACP, the first MAC address inserted into the forwarding database in the forwarding table is the source address of the protocol packet.

  If the VLAN is composed of regular access or trunk interfaces, you can set the `mac-limit` to 1 if you choose to do so.

RELATED DOCUMENTATION

Example: Protecting against Ethernet Switching Table Overflow Attacks | 392

Verifying That MAC Limiting Is Working Correctly | 398

Override a MAC Limit Applied to All Interfaces | 407

Configuring Autorecovery for Port Security Events | 758

Understanding MAC Limiting and MAC Move Limiting for Port Security

Understanding Bridging and VLANs on Switches

Configuring MAC Limiting on MX Series Routers

IN THIS SECTION

This topic describes the different ways of configuring a limitation on MAC addresses in packets that are received and forwarded by MX Series routers.

Limiting the Number of MAC Addresses Learned by an Interface

To secure a port, you can set the maximum number of MAC addresses that can be learned by an interface.

MX Series routers support only the **drop** action. If the action is not specified, the router performs the default action **drop** if the limit is exceeded.

- Set the MAC limit on an interface, and specify the action that the router takes after the specified limit is exceeded.

  If you want to set the MAC limit on an interface that is part of the default routing instance:

  ```
  [edit switch-options]
  user@switch# set interface interface-name interface-mac-limit limit packet-action action
  ```

  If you want to set the MAC limit on an interface that is part of a routing instance:

  ```
  [edit routing-instances]
  user@switch# set routing-instance-name switch-options interface interface-name interface-mac-
  limit limit
  ```

  If you want to set the MAC limit on all interfaces that are part of the default routing instance:

  ```
  [edit switch-options]
  user@switch# set interface-mac-limit limit
  ```

If you want to set the MAC limit on all interfaces that are part of a routing instance:

```
[edit routing-instances]
user@switch# set routing-instance-name switch-options interface-mac-limit limit
```

After you set a new MAC limit for the interface, the system clears existing entries in the MAC address forwarding table associated with the interface.

Limiting the Number of MAC Addresses Learned by a Bridge Domain

To limit the number of MAC addresses learned by a bridge domain, perform the following steps:

Set the maximum number of MAC addresses that can be learned by a bridge domain, and specify an action that the device takes after the specified limit is exceeded:

```
[edit bridge-domains]
user@switch# set bridge-domain-name bridge-options mac-table-size limit packet-action action
```

Limiting the Number of MAC Addresses Learned by an Interface in a Bridge Domain

To limit the number of MAC addresses learned by an interface in a bridge domain, perform the following steps:

1. Set the maximum number of MAC addresses that can be learned by an interface in a bridge domain, and specify an action that the device takes after the specified limit is exceeded:

   ```
   [edit bridge-domains]
   user@switch# set bridge-domain-name bridge-options interface-mac-limit limit packet-action action
   ```

2. Set the maximum number of MAC addresses that can be learned by one *or* all interfaces in the bridge domain, and specify an action that the device takes after the specified limit is exceeded:

   > **NOTE**: If you specify a MAC limit and packet action for all interfaces in the bridge domain *and* a specific interface in the bridge domain, the MAC limit and packet action

specified at the specific interface level takes precedence. Also, at the bridge domain interface level, only the drop option is supported.

```
[edit bridge-domains]
user@switch# set bridge-domain-name bridge-options interface interface-name interface-mac-limit
limit packet-action action
```

```
[edit bridge-domains]
user@switch# set bridge-domain-name bridge-options interface-mac-limit limit packet-action action
```

Configuring MAC Limiting (J-Web Procedure)

MAC limiting protects against flooding of the Ethernet switching table on an EX Series switch. MAC limiting sets a limit on the number of MAC addresses that can be learned on a single Layer 2 access interface (port).

Junos OS provides two MAC limiting methods:

- Maximum number of dynamic MAC addresses allowed per interface—If the limit is exceeded, incoming packets with new MAC addresses are dropped.

- Specific “allowed” MAC addresses for the access interface—Any MAC address that is not in the list of configured addresses is not learned.

You configure MAC limiting for each interface, not for each VLAN. You can specify the maximum number of dynamic MAC addresses that can be learned on a single Layer 2 access interface or on all Layer 2 access interfaces. The default action that the switch will take if that maximum number is exceeded is **drop**—drop the packet and generate an alarm, an SNMP trap, or a system log entry.

To enable MAC limiting on one or more interfaces using the J-Web interface:

1. Select **Configure>Security>Port Security**.
2. Select one or more interfaces from the **Interface List**.
3. Click the **Edit** button. If a message appears asking whether you want to enable port security, click **Yes**.
4. To set a dynamic MAC limit:

   a. Type a limit value in the **MAC Limit** box.

   b. Select an action from the **MAC Limit Action** box (optional). The switch takes this action when the MAC limit is exceeded. If you do not select an action, the switch applies the default action, **drop**.

      - Log—Generate a system log entry.

- Drop—Drop the packets and generate a system log entry. (Default)
- Shutdown—Shut down the VLAN and generate a system log entry. You can mitigate the effect of this option by configuring the switch for autorecovery from the disabled state and specifying a **disable timeout** value.
- None— No action to be taken.

5. To add allowed MAC addresses:

   a. Click **Add**.

   b. Type the allowed MAC address and click **OK**.

   Repeat this step to add more allowed MAC addresses.

6. Click **OK** when you have finished setting MAC limits.
7. Click **OK** after the configuration has been successfully delivered.

> **NOTE**: You can enable or disable port security on the switch at any time by clicking the **Activate** or **Deactivate** button on the Port Security Configuration page. If security status is shown as **Disabled** when you try to edit settings for any VLANs or interfaces (ports), a message asking whether you want to enable port security appears.

SEE ALSO

Example: Protecting against DHCP Starvation Attacks | 382

Verifying That MAC Limiting Is Working Correctly | 398

Understanding MAC Limiting and MAC Move Limiting | 357

Example: Configuring MAC Limiting

IN THIS SECTION

Example: Protecting against DHCP Starvation Attacks

IN THIS SECTION

In a DHCP starvation attack, an attacker floods an Ethernet LAN with DHCP requests from spoofed (counterfeit) MAC addresses, causing the switch's overworked DHCP server to stop assigning IP addresses and lease times to legitimate DHCP clients on the switch (hence the name starvation). Requests from those clients are either dropped or directed to a rogue DHCP server set up by the attacker.

This example describes how to configure MAC limiting, a port security feature, to protect the switch against DHCP starvation attacks:

Requirements

This example uses the following hardware and software components:

- One EX Series or QFX3500 switch
- Junos OS Release 9.0 or later for EX Series switches, or Junos OS Release 12.1 or later for the QFX Series switch
- A DHCP server to provide IP addresses to network devices on the switch

Before you configure MAC limiting, a port security feature, to mitigate DHCP starvation attacks, be sure you have:

- Connected the DHCP server to the switch.
- Configured the VLAN **employee-vlan** on the switch.

Overview and Topology

IN THIS SECTION

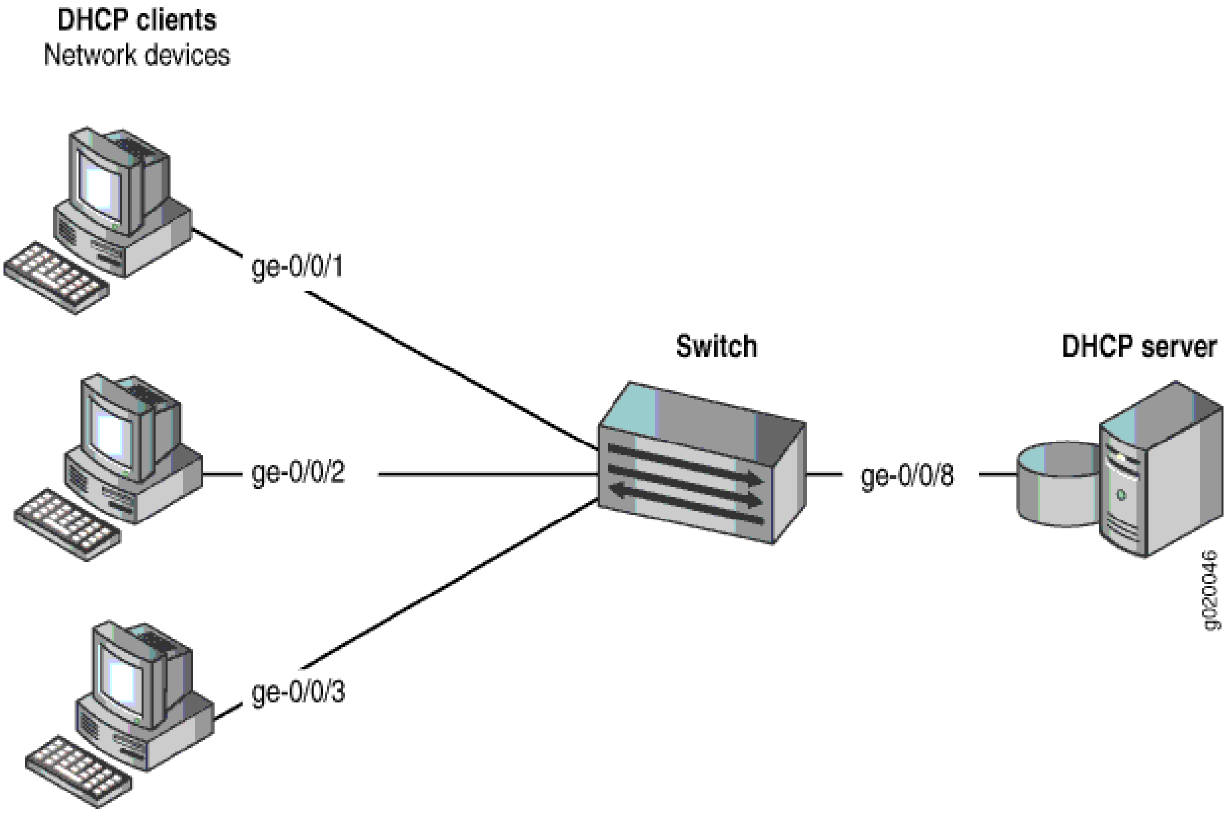
Ethernet LANs are vulnerable to address spoofing and DoS attacks on network devices. This example describes how to protect the switch against one common type of attack, a DHCP starvation attack.

This example shows how to configure port security features on a switch connected to a DHCP server. The setup for this example includes the VLAN **employee-vlan** on the switch. The procedure for creating that VLAN on an EX Series switch is described in the topic, Example: Setting Up Bridging with Multiple VLANs for EX Series Switches. The procedure is not repeated here.

Figure 14 on page 384 illustrates the topology for this example.

Topology

Figure 14: Network Topology for Basic Port Security



The components of the topology for this example are shown in Table 17 on page 384.

Table 17: Components of the Port Security Topology

Properties	Settings
Switch hardware	QFX3500 switch
VLAN name and ID	**employee-vlan**
Interfaces in **employee-vlan**	**ge-0/0/1, ge-0/0/2, ge-0/0/3, ge-0/0/8**
Interface for DHCP server	**ge-0/0/8**

In this example, the switch has already been configured as follows:

- Secure port access is activated on the switch.
- No MAC limit is set on any of the interfaces.
- DHCP snooping is disabled on the VLAN **employee-vlan**.
- All access interfaces are untrusted, which is the default setting.

Configuration

IN THIS SECTION

To configure the MAC limiting port security feature to protect the switch against DHCP starvation attacks:

Procedure

CLI Quick Configuration

To quickly configure MAC limiting, copy the following commands and paste them into the switch terminal window:

```
[edit ethernet-switching-options secure-access-port]
 set interface ge-0/0/1 mac-limit 3 action drop
set interface ge-0/0/2 mac-limit 3 action drop
```

Step-by-Step Procedure

Configure MAC limiting:

1. Configure a MAC limit of **3** on **ge-0/0/1** and specify that packets with new addresses be dropped if the limit has been exceeded on the interface:

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set interface ge-0/0/1mac-limit 3 action drop
   ```

2. Configure a MAC limit of **3** on **ge-0/0/2** and specify that packets with new addresses be dropped if the limit has been exceeded on the interface:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface ge-0/0/2 mac-limit 3 action drop
```

Results

Check the results of the configuration:

```
[edit ethernet-switching-options secure-access-port]
user@switch# show
interface ge-0/0/1.0 {
    mac-limit 3 action drop;
}
interface ge-0/0/2.0 {
    mac-limit 3 action drop;
}
```

Verification

IN THIS SECTION

To confirm that the configuration is working properly:

Verifying That MAC Limiting Is Working Correctly on the Switch

Purpose

Verify that MAC limiting is working on the switch.

Action

Send some DHCP requests from network devices (here they are DHCP clients) connected to the switch.

Display the MAC addresses learned when DHCP requests are sent from hosts on **ge-0/0/1** and from hosts on **ge-0/0/2**, with both interfaces set to a MAC limit of **3** with the action **drop**:

```
user@switch> show ethernet-switching table
Ethernet-switching table:  7 entries, 6 learned

  VLAN           MAC address         Type         Age    Interfaces

  default        *                   Flood          -    ge-0/0/2.0
  default        00:05:85:3A:82:77   Learn          0    ge-0/0/1.0
  default        00:05:85:3A:82:79   Learn          0    ge-0/0/1.0
  default        00:05:85:3A:82:80   Learn          0    ge-0/0/1.0
  default        00:05:85:3A:82:81   Learn          0    ge-0/0/2.0
  default        00:05:85:3A:82:83   Learn          0    ge-0/0/2.0
  default        00:05:85:3A:82:85   Learn          0    ge-0/0/2.0
```

Meaning

The sample output shows that with a MAC limit of **3** for each interface, the DHCP request for a fourth MAC address on **ge-0/0/2** was dropped because it exceeded the MAC limit.

Because only 3 MAC addresses can be learned on each of the two interfaces, attempted DHCP starvation attacks will fail.

SEE ALSO

Example: Configuring Port Security (non-ELS) | 15

Configuring MAC Limiting (non-ELS) | 374

Understanding MAC Limiting and MAC Move Limiting for Port Security

Example: Protecting against Rogue DHCP Server Attacks

IN THIS SECTION

In a rogue DHCP server attack, an attacker has introduced a rogue server into the network, allowing it to give IP address leases to the network's DHCP clients and to assign itself as the gateway device.

This example describes how to configure a DHCP server interface as untrusted to protect the switch from a rogue DHCP server:

Requirements

This example uses the following hardware and software components:

- One EX Series switch or one QFX3500 switch
- Junos OS Release 9.0 or later for EX Series switches or Junos OS Release 12.1 or later for the QFX Series
- A DHCP server to provide IP addresses to network devices on the switch

Before you configure an untrusted DHCP server interface to mitigate rogue DHCP server attacks, be sure you have:

- Connected the DHCP server to the switch.
- Enabled DHCP snooping on the VLAN.
- Configured a VLAN on the switch. See the task for your platform:
  - Example: Setting Up Bridging with Multiple VLANs.

Overview and Topology

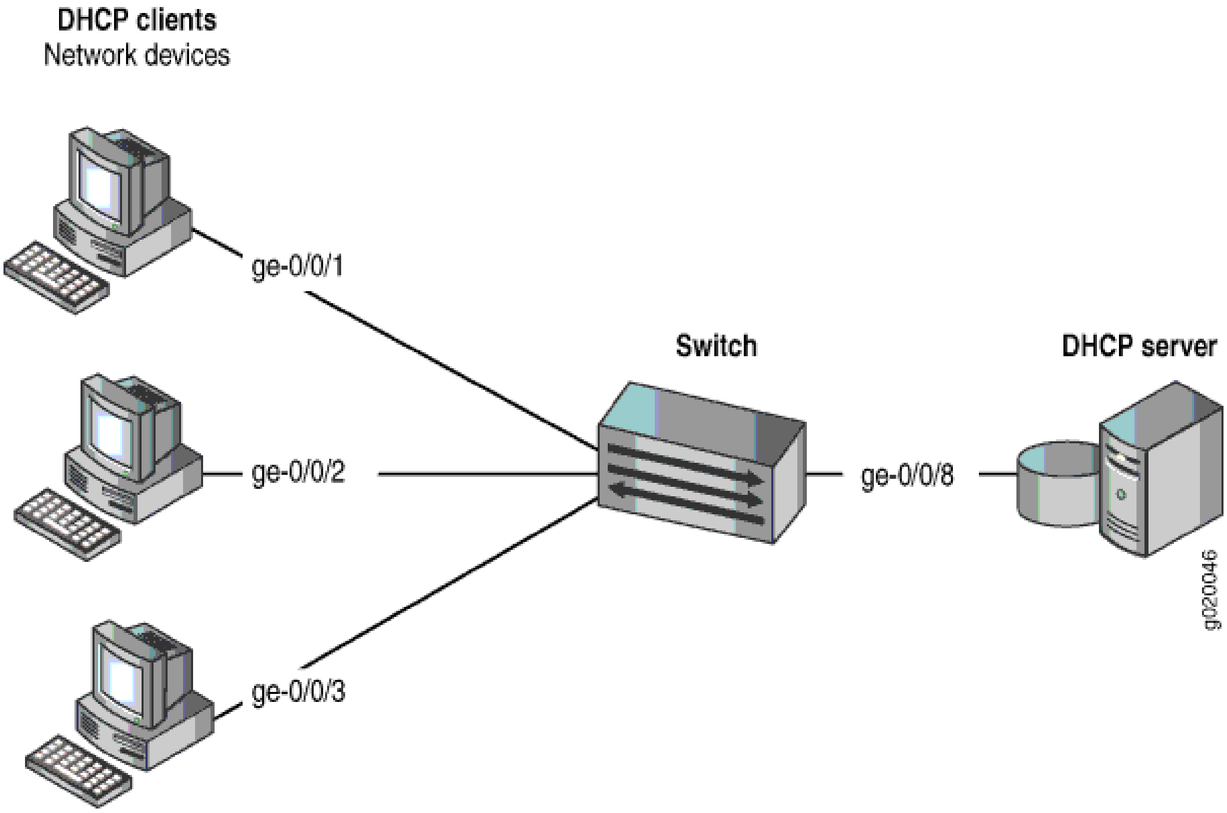
IN THIS SECTION

Ethernet LANs are vulnerable to address spoofing and DoS attacks on network devices. This example describes how to protect the switch from rogue DHCP server attacks.

This example shows how to explicitly configure an untrusted interface on an EX3200-24P switch and a QFX3500 switch. Figure 15 on page 389 illustrates the topology for this example.

Topology

Figure 15: Network Topology for Basic Port Security



The components of the topology for this example are shown in Table 18 on page 389.

Table 18: Components of the Port Security Topology

Properties	Settings
Switch hardware	One EX3200-24P, 24 ports (8 PoE ports) or one QFX3500 switch
VLAN name and ID	**employee-vlan**, tag **20**

Table 18: Components of the Port Security Topology *(Continued)*

Properties	Settings
VLAN subnets	192.0.2.16/28 192.0.2.17 through 192.0.2.30 192.0.2.31 is the subnet's broadcast address
Interfaces in **employee-vlan**	ge-0/0/1, ge-0/0/2, ge-0/0/3, ge-0/0/8
Interface for DHCP server	ge-0/0/8

In this example, the switch has already been configured as follows:

- Secure port access is activated on the switch.
- DHCP snooping is enabled on the VLAN **employee-vlan**.
- The interface (port) where the rogue DHCP server has connected to the switch is currently trusted.

Configuration

IN THIS SECTION

To configure the DHCP server interface as untrusted because the interface is being used by a rogue DHCP server:

Procedure

CLI Quick Configuration

To quickly set the rogue DHCP server interface as untrusted, copy the following command and paste it into the switch terminal window:

```
[edit ethernet-switching-options secure-access-port]
set interface ge-0/0/8 no-dhcp-trusted
```

Step-by-Step Procedure

To set the DHCP server interface as untrusted:

- Specify the interface (port) from which DHCP responses are not allowed:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface ge-0/0/8 no-dhcp-
trusted
```

Results

Check the results of the configuration:

```
[edit ethernet-switching-options secure-access-port]
user@switch# show
interface ge-0/0/8.0 {
    no-dhcp-trusted;
}
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying That the DHCP Server Interface Is Untrusted

Purpose

Verify that the DHCP server is untrusted.

Action

1. Send some DHCP requests from network devices (here they are DHCP clients) connected to the switch.

2. Display the DHCP snooping information when the port on which the DHCP server connects to the switch is not trusted.

Meaning

There is no output from the command because no entries are added to the DHCP snooping database.

SEE ALSO

Understanding and Using Trusted DHCP Servers | 415
Example: Configuring Port Security (non-ELS) | 15
show dhcp snooping binding

Example: Protecting against Ethernet Switching Table Overflow Attacks

IN THIS SECTION

In an Ethernet switching table overflow attack, an intruder sends so many requests from new MAC addresses that the Ethernet switching table fills up and then overflows, forcing the switch to broadcast all messages.

This example describes how to configure MAC limiting and allowed MAC addresses, two port security features, to protect the switch from Ethernet switching table attacks:

Requirements

This example uses the following hardware and software components:

- One EX Series switch or QFX3500 switch
- Junos OS Release 9.0 or later for EX Series switches or Junos OS 12.1 or later for the QFX Series.
- A DHCP server to provide IP addresses to network devices on the switch

Before you configure specific port security features to mitigate common access-interface attacks, be sure you have:

- Connected the DHCP server to the switch.
- Configured a VLAN on the switch. See the task for your platform:

Overview and Topology

IN THIS SECTION

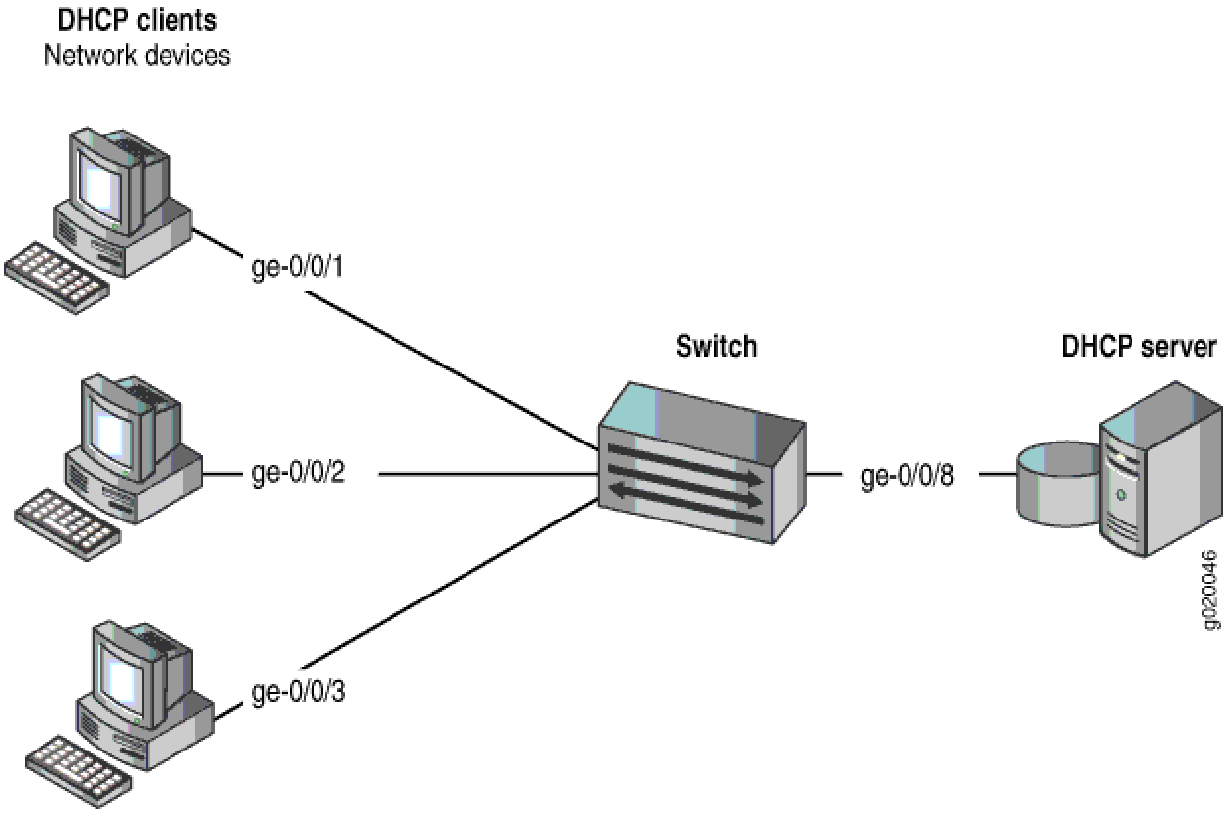
Ethernet LANs are vulnerable to address spoofing and DoS attacks on network devices. This example describes how to protect the switch from an attack on the Ethernet switching table that causes the table to overflow and thus forces the switch to broadcast all messages.

This example shows how to configure port security features on a switch connected to a DHCP server.

The setup for this example includes the VLAN **employee-vlan** on the switch. The procedure for creating that VLAN is described in the topic Example: Setting Up Bridging with Multiple VLANs for EX Series Switches and Example: Setting Up Bridging with Multiple VLANs for the QFX Series. That procedure is not repeated here. Figure 16 on page 394 illustrates the topology for this example.

Topology

Figure 16: Network Topology for Basic Port Security



The components of the topology for this example are shown in Table 19 on page 394.

Table 19: Components of the Port Security Topology

Properties	Settings
Switch hardware	One EX Series switch or one QFX3500 switch
VLAN name and ID	**employee-vlan**, tag **20**
VLAN subnets	**192.0.2.16/28** **192.0.2.17** through **192.0.2.30** **192.0.2.31** is subnet's broadcast address

Table 19: Components of the Port Security Topology *(Continued)*

Properties	Settings
Interfaces in **employee-vlan**	ge-0/0/1, ge-0/0/2, ge-0/0/3, ge-0/0/8
Interface for DHCP server	ge-0/0/8

In this example, use the MAC limit feature to control the total number of MAC addresses that can be added to the Ethernet switching table for the specified interface. Use the allowed MAC addresses feature to ensure that the addresses of network devices whose network access is critical are guaranteed to be included in the Ethernet switching table.

In this example, the switch has already been configured as follows:

- Secure port access is activated on the switch.
- No MAC limit is set on any of the interfaces.
- All access interfaces are untrusted, which is the default setting.

Configuration

IN THIS SECTION

To configure MAC limiting and some allowed MAC addresses to protect the switch against Ethernet switching table overflow attacks:

Procedure

CLI Quick Configuration

To quickly configure MAC limiting, clear the MAC forwarding table, and configure some allowed MAC addresses, copy the following commands and paste them into the switch terminal window:

```
[edit ethernet-switching-options secure-access-port]
 set interface ge-0/0/1 mac-limit 4 action drop
set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:80
set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:81
set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:83
set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:85
exit
exit
clear ethernet-switching-table interface ge-0/0/1
```

Step-by-Step Procedure

Configure MAC limiting and some allowed MAC addresses:

1. Configure a MAC limit of **4** on **ge-0/0/1** and specify that incoming packets with different addresses be dropped once the limit is exceeded on the interface:

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set interface ge-0/0/1 mac-limit (Access Port Security) 4 action drop
   ```

2. Clear the current entries for interface ge-0/0/1 from the MAC address forwarding table :

   ```
   user@switch# clear ethernet-switching-table interface ge-0/0/1
   ```

3. Configure the allowed MAC addresses on **ge-0/0/2**:

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:80
   user@switch# set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:81
   user@switch# set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:83
   user@switch# set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:85
   ```

Results

Check the results of the configuration:

```
[edit ethernet-switching-options secure-access-port]
user@switch# show
interface ge-0/0/1.0 {
    mac-limit 4 action drop;
```

```
}
interface ge-0/0/2.0 {
    allowed-mac [ 00:05:85:3a:82:80 00:05:85:3a:82:81 00:05:85:3a:82:83 00:05:85
    :3a:82:85 ];
}
```

Verification

IN THIS SECTION

To confirm that the configuration is working properly:

Verifying That MAC Limiting Is Working Correctly on the Switch

Purpose

Verify that MAC limiting is working on the switch.

Action

Display the MAC cache information after DHCP requests have been sent from hosts on **ge-0/0/1**, with the interface set to a MAC limit of **4** with the action **drop**, and after four allowed MAC addresses have been configured on interface **ge/0/0/2**:

```
user@switch>  show ethernet-switching table
Ethernet-switching table:  5 entries, 4 learned

  VLAN              MAC address         Type         Age    Interfaces

  employee-vlan     00:05:85:3A:82:71   Learn          0    ge-0/0/1.0
  employee-vlan     00:05:85:3A:82:74   Learn          0    ge-0/0/1.0
  employee-vlan     00:05:85:3A:82:77   Learn          0    ge-0/0/1.0
  employee-vlan     00:05:85:3A:82:79   Learn          0    ge-0/0/1.0
  employee-vlan     *                   Flood          0    ge-0/0/1.0
  employee-vlan     00:05:85:3A:82:80   Learn          0    ge-0/0/2.0
```

```
    employee-vlan     00:05:85:3A:82:81   Learn          0    ge-0/0/2.0
    employee-vlan     00:05:85:3A:82:83   Learn          0    ge-0/0/2.0
    employee-vlan     00:05:85:3A:82:85   Learn          0    ge-0/0/2.0
    employee-vlan     *                   Flood          -    ge-0/0/2.0
```

Meaning

The sample output shows that with a MAC limit of **4** for the interface, the DHCP request for a fifth MAC address on **ge-0/0/1** was dropped because it exceeded the MAC limit and that only the specified allowed MAC addresses have been learned on the **ge-0/0/2** interface.

SEE ALSO

Example: Configuring Port Security (non-ELS) | **15**

Configuring MAC Limiting (non-ELS) | **374**

Configuring MAC Move Limiting (non-ELS)

Verifying That MAC Limiting Is Working Correctly

IN THIS SECTION

MAC limiting protects against flooding of the Ethernet switching table by setting a limit on the number of MAC addresses that can be learned on a single Layer 2 access interface (port)..

Junos OS provides two methods for MAC limiting for port security:

- Maximum number of MAC addresses—You configure the maximum number of dynamic MAC addresses allowed per interface. When the limit is exceeded, incoming packets with new MAC addresses can be ignored, dropped, or logged. You can also specify that the interface be shut down or temporarily disabled.

- Allowed MAC addresses—You configure specific “allowed” MAC addresses for the access interface. Any MAC address that is not in the list of configured addresses is not learned, and the switch logs an appropriate message. The allowed MAC method binds MAC addresses to a VLAN so that the address is not registered outside the VLAN. If an allowed MAC setting conflicts with a dynamic MAC setting, the allowed MAC setting takes precedence.

Junos OS also allows you to set a MAC limit on VLANs. However, setting a MAC limit on VLANs is not considered a port security feature, because the switch does not prevent incoming packets that cause the MAC limit to be exceeded from being forwarded; it only logs the MAC addresses of these packets.

NOTE: The information in this topic is for non-ELS platforms. For ELS platforms, refer Configuring MAC Limiting (ELS) to read on MAC limiting.

Verifying That MAC Limiting for Dynamic MAC Addresses Is Working Correctly

IN THIS SECTION

Purpose

Verify that MAC limiting for dynamic MAC addresses is working on the switch.

Action

Display the MAC addresses that have been learned. The following sample output shows the results when two packets were sent from hosts on ge-0/0/1 and five packets requests were sent from hosts on ge-0/0/2, with both interfaces set to a MAC limit of **4** with the default action **drop**:

```
user@switch> show ethernet-switching table
Ethernet-switching table:  7 entries, 6 learned
```

```
VLAN                MAC address         Type          Age    Interfaces

employee-vlan       *                   Flood           -    ge-0/0/2.0
employee-vlan       00:05:85:3A:82:77   Learn           0    ge-0/0/1.0
employee-vlan       00:05:85:3A:82:79   Learn           0    ge-0/0/1.0
employee-vlan       00:05:85:3A:82:80   Learn           0    ge-0/0/2.0
employee-vlan       00:05:85:3A:82:81   Learn           0    ge-0/0/2.0
employee-vlan       00:05:85:3A:82:83   Learn           0    ge-0/0/2.0
employee-vlan       00:05:85:3A:82:85   Learn           0    ge-0/0/2.0
```

Meaning

The sample output shows that with a MAC limit of **4** for each interface, the packet for a fifth MAC address on ge-0/0/2 was dropped because it exceeded the MAC limit. The address was not learned, and thus an asterisk (*) rather than an address appears in the **MAC address** column in the first line of the sample output.

Verifying That MAC Limiting for a Specific Interface Within a Specific VLAN Is Working Correctly

IN THIS SECTION

Purpose

Verify that MAC limiting for a specific interface based on its membership within a specific VLAN is working on the switch.

Action

Display the detailed statistics for MAC addresses that have been learned:

```
user@switch> show ethernet-switching statistics mac-learning     interface ge-0/0/28 detail



    Interface: ge-0/0/28.0
    Learning message from local packets:   0
    Learning message from transit packets: 5
    Learning message with error:           0
      Invalid VLAN:               0       Invalid MAC:                  0
      Security violation:         0       Interface down:               0
      Incorrect membership:       0       Interface limit:              0
      MAC move limit:             0       VLAN limit:                   0
                                          VLAN membership limit:        20
      Invalid VLAN index:         0       Interface not learning:       0
      No nexthop:                 0       MAC learning disabled:        0
      Others:                     0
```

Meaning

The `VLAN membership limit` shows the number of packets that were dropped because of the VLAN membership MAC limit for interface ge-0/0/28.0 was exceeded. In this case, 20 packets were dropped.

Verifying That Allowed MAC Addresses Are Working Correctly

IN THIS SECTION

Purpose

Verify that allowed MAC addresses are working on the switch.

Action

Display the MAC address cache information after allowed MAC addresses have been configured on an interface. The following sample shows the MAC address cache after 5 allowed MAC addresses were on interface ge-0/0/2. In this instance, the interface was also set to a dynamic MAC limit of 4 with the default action **drop**.

```
user@switch> show ethernet-switching table
Ethernet-switching table:  5 entries, 4 learned

  VLAN                MAC address         Type         Age    Interfaces

  employee-vlan       00:05:85:3A:82:80   Learn          0    ge-0/0/2.0
  employee-vlan       00:05:85:3A:82:81   Learn          0    ge-0/0/2.0
  employee-vlan       00:05:85:3A:82:83   Learn          0    ge-0/0/2.0
  employee-vlan       00:05:85:3A:82:85   Learn          0    ge-0/0/2.0
  employee-vlan       *                   Flood          -    ge-0/0/2.0
```

Meaning

Because the MAC limit value for this interface was set to **4**, only four of the five configured allowed addresses were learned and thus added to the MAC address cache. Because the fifth address was not learned, an asterisk (*) rather than an address appears in the **MAC address** column in the last line of the sample output.

Verifying Results of Various Action Settings When the MAC Limit Is Exceeded

IN THIS SECTION

Purpose

Verify the results provided by the various action settings for MAC limits—**drop**, **log**, **shutdown** and **none** —when the limits are exceeded.

Action

Display the results of the various action settings.

NOTE: You can view log messages by using the `show log messages` command. You can also have the log messages displayed by configuring the monitor start messages with the `monitor start messages` command.

- **drop** action—For MAC limiting configured with a **drop** action and with the MAC limit set to **5**:

```
user@switch>    show ethernet-switching
table
Ethernet-switching table: 6 entries, 5 learned

  VLAN                MAC address         Type         Age    Interfaces

  employee-vlan       *                   Flood          -    ge-0/0/2.0
  employee-vlan       00:05:85:3A:82:80   Learn          0    ge-0/0/2.0
  employee-vlan       00:05:85:3A:82:81   Learn          0    ge-0/0/2.0
  employee-vlan       00:05:85:3A:82:83   Learn          0    ge-0/0/2.0
  employee-vlan       00:05:85:3A:82:85   Learn          0    ge-0/0/2.0
  employee-vlan       00:05:85:3A:82:88   Learn          0    ge-0/0/2.0
```

- **log** action—For MAC limiting configured with a **log** action and with MAC limit set to **5**:

```
user@switch> show ethernet-switching table
Ethernet-switching table: 74 entries, 73 learned

  VLAN                MAC address         Type         Age    Interfaces

  employee-vlan       *                   Flood          -    ge-0/0/2.0
  employee-vlan       00:05:85:3A:82:80   Learn          0    ge-0/0/2.0
  employee-vlan       00:05:85:3A:82:81   Learn          0    ge-0/0/2.0
  employee-vlan       00:05:85:3A:82:82   Learn          0    ge-0/0/2.0
  employee-vlan       00:05:85:3A:82:83   Learn          0    ge-0/0/2.0
  employee-vlan       00:05:85:3A:82:84   Learn          0    ge-0/0/2.0
  employee-vlan       00:05:85:3A:82:85   Learn          0    ge-0/0/2.0
  employee-vlan       00:05:85:3A:82:87   Learn          0    ge-0/0/2.0
  employee-vlan       00:05:85:3A:82:88   Learn          0    ge-0/0/2.0
  . . .
```

- **shutdown** action—For MAC limiting configured with a **shutdown** action and with MAC limit set to **3**:

  ```
  user@switch> show ethernet-switching table
  Ethernet-switching table: 4 entries, 3 learned

    VLAN                 MAC address         Type         Age    Interfaces

    employee–vlan        *                   Flood          -    ge-0/0/2.0
    employee–vlan        00:05:85:3A:82:82   Learn          0    ge-0/0/2.0
    employee–vlan        00:05:85:3A:82:84   Learn          0    ge-0/0/2.0
    employee–vlan        00:05:85:3A:82:87   Learn          0    ge-0/0/2.0
  ```

- **none** action—If you set a MAC limit to apply to all interfaces on the switch, you can override that setting for a particular interface by specifying this action for that interface. See "Override a MAC Limit Applied to All Interfaces" on page 407.

Meaning

For the **drop** action results—The sixth MAC address exceeded the MAC limit. The request packet for that address was dropped. Only five MAC addresses have been learned on ge-0/0/2.

For the **log** action results—The sixth MAC address exceeded the MAC limit. No MAC addresses were blocked.

For the **shutdown** action results—The fourth MAC address exceeded the MAC limit. Only three MAC addresses have been learned on ge-0/0/2. The interface ge-0/0/1 is shut down.

For more information about interfaces that have been shut down, use the `show ethernet-switching interfaces` command.

```
user@switch> show ethernet-switching interfaces

Interface    State  VLAN members        Tag   Tagging  Blocking



bme0.32770   down   mgmt                      untagged unblocked

ge-1/0/0.0   down   v1                        untagged MAC limit exceeded

ge-1/0/1.0   up     v1                        untagged unblocked
```

```
ge-1/0/2.0   up     v1                        untagged unblocked

me0.0        up     mgmt                      untagged unblocked
```

NOTE: You can configure the switch to recover automatically from this type of error condition by specifying the `port-error-disable` statement with a **disable timeout** value. The switch automatically restores the disabled interface to service when the disable timeout expires. The **port-error-disable** configuration does not apply to already existing error conditions. It impacts only error conditions that are detected after **port-error-disable** has been enabled and committed. To clear an already existing error condition and restore the interface to service, use the `clear ethernet-switching port-error` command.

Verifying That Interfaces Are Shut Down

IN THIS SECTION

Purpose

Verify that an interface is shut down when the MAC limit is exceeded.

Action

For more information about interfaces that have been shut down because the MAC limit was exceeded, use the `show ethernet-switching interfaces` command.

```
user@switch> show ethernet-switching interfaces

Interface    State  VLAN members        Tag   Tagging  Blocking



bme0.32770           down   mgmt     untagged unblocked

xe-0/0/0.0              down   v1       untagged MAC limit exceeded
```

```
xe- 0/0/1.0              up     v1       untagged unblocked

xe-0/0/2.0               up     v1       untagged unblocked

me0.0                up     mgmt     untagged unblocked
```

NOTE: You can configure interfaces to recover automatically when the MAC limit has been exceeded by specifying the `port-error-disable` statement with a **disable timeout** value. The switch automatically restores the disabled interface to service when the disable timeout expires. The **port-error-disable** configuration does not apply to preexisting error conditions—it affects only error conditions that are detected after the `port-error-disable` statement has been enabled and the configuration has been committed. To clear a preexisting error condition and restore the interface to service, use the `clear ethernet-switching port-error` command.

Customizing the Ethernet Switching Table Display to View Information for a Specific Interface

IN THIS SECTION

Purpose

You can use the `show ethernet-switching table` command to view information about the MAC addresses learned on a specific interface.

Action

For example, to display the MAC addresses learned on ge-0/0/2 interface, type:

```
user@switch> show ethernet-switching table interface ge-0/0/2.0
Ethernet-switching table: 1 unicast entries
```

```
VLAN              MAC address       Type         Age Interfaces

v1                *                 Flood          - All-members

v1                00:00:06:00:00:00 Learn          0 ge-2/0/0.0
```

Meaning

The MAC limit value for ge-0/0/2 was set to **1**, and the output shows that only one MAC address was learned and thus added to the MAC address cache. An asterisk (*) rather than an address appears in the **MAC address** column in the first line of the sample output.

RELATED DOCUMENTATION

Configuring MAC Limiting (non-ELS) | **374**

Configuring Autorecovery for Port Security Events | **758**

Example: Protecting Against DHCP Snooping Database Attacks | **476**

Example: Protecting against Ethernet Switching Table Overflow Attacks | **392**

Example: Protecting against DHCP Starvation Attacks | **382**

Monitoring Port Security

Override a MAC Limit Applied to All Interfaces

If you set a MAC limit in your port security settings to apply to all interfaces on the EX Series switch, you can override that setting for a particular interface by specifying action the **none**.

NOTE: A non-ELS style configuration is used in this topic. Refer Configuring MAC Limiting (ELS) to read about using ELS and non-ELS style configuration to configure MAC limiting.

To use the **none** action to override a MAC limit setting:

1. Set the MAC limit for all interfaces to have a limit of, for example, **5** using the action **drop**:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface allmac-limit 5 action drop
```

2. Then change the action for one interface (here, **ge-0/0/2**) with this command. You don't need to specify a limit value.

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface ge-0/0/2 mac-limit action none
```

NOTE: In MX and SRX Series Firewalls, the 1 and 10-Gigabit SFP or SFP+ optical interfaces are always named as xe even if a 1-Gigabit SFP is inserted. However, in EX and QFX series devices, the interface name is shown as ge or xe based on the speed of the optical device inserted.

RELATED DOCUMENTATION

Configuring MAC Limiting (non-ELS) | **374**

Example: Configuring Port Security (non-ELS) | **15**

Verifying That MAC Limiting Is Working Correctly | **398**

Example: Protecting against Ethernet Switching Table Overflow Attacks | **392**

Example: Protecting against DHCP Starvation Attacks | **382**

Configuring MAC Move Limiting (ELS)

NOTE: This topic uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. For ELS details, see Using the Enhanced Layer 2 Software CLI.

When MAC move limiting is configured, the switch tracks MAC address movements on access and trunk interfaces. A MAC address move occurs when the switch receives a packet with a source MAC address that has already been learned by the switch, but on a different interface. If a MAC address changes more than the configured number of times within one second, the changes to MAC addresses are dropped, logged or ignored, or the interface is shut down, as specified in the configuration.

MAC move limiting is not configured by default.

You can choose to have one of the following actions performed when the MAC move limit is exceeded:

- `drop`—(EX2300, EX3400 and EX4300) Drop the packet, but do not generate an alarm.
- `drop-and-log`—(EX2300, EX3400 and EX4300 only) Drop the packet and generate an alarm, an SNMP trap, or system log entry.
- `log`—(EX4300 and EX9200) Do not drop the packet but generate an alarm, an SNMP trap, or a system log entry.
- `none`—(EX4300 and EX9200) Forward packets with new source MAC addresses, and learn the new source MAC address.
- `shutdown`—Disable the interface in the VLAN and generate an alarm, an SNMP trap, or a system log entry. If you configure an interface with the `recovery-timeout` statement, the disabled interfaces recover automatically upon expiration of the specified disable timeout. If you do not configure the switch for autorecovery from the disabled condition, you can bring up the disabled interfaces by running the `clear ethernet-switching recovery-timeout` command.
- `vlan-member-shutdown`—(EX9200 only) Block an interface on the basis of its membership in a specific VLAN and generate an alarm, an SNMP trap, or a system log entry. If you configure an interface with the `recovery-timeout` statement, the disabled interfaces recover automatically upon expiration of the specified disable timeout. If you do not configure `recovery-timeout`, then the interface remains blocked for 180 seconds, after which it is automatically restored. You can recover all of the blocked interfaces by running the `clear ethernet-switching recovery-timeout` command, or recover a specific interface by using the `set ethernet-switching recovery-timeout interface` *`interface-name`* `vlan` *`vlan-name`* command.

To configure a MAC move limit for MAC addresses within a specific VLAN:

- To limit the number of MAC address movements that can be made by an individual MAC address within the specified VLAN:

  ```
  [edit edit vlans vlan-name  switch-options]
  user@switch# set mac-move-limit limit
  ```

- To limit the number of MAC address movements that can be made by an individual MAC address and to specify the action to be taken when the limit is reached:

  ```
  [edit edit vlans vlan-name  switch-options]
  user@switch# set mac-move-limit limit packet-action action
  ```

The switch performs the specified action if it tracks that an individual MAC address within the specified VLAN has moved more than the specified number of times within one second.

- Starting in Junos OS Release 15.1 for EX9200 Switches with configured actions for MAC Move Limiting, you can determine the priority for an interface involved in the MAC move to be selected for the action. To determine the priority for an interface involved in the MAC move:

```
[edit edit vlans vlan-name  switch-options]
user@switch# set mac-move-limit interface interface-name action-priority value
```

The interface with the lowest value configured for `action-priority` has the highest priority.

> **NOTE**: You can use the action priority to decrease the likelihood of blocking a trusted interface. The trusted interface should have the lowest priority if the configured action is `shutdown` or `vlan-member-shutdown`. To assign a low priority, configure a high value for `action-priority`.

> **NOTE**: `mac-move-limit` configuration does not work with dot1x authenticated MAC addresses.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
15.1	Starting in Junos OS Release 15.1 for EX9200 Switches with configured actions for MAC Move Limiting, you can determine the priority for an interface involved in the MAC move to be selected for the action.

RELATED DOCUMENTATION

Understanding MAC Limiting and MAC Move Limiting | 357
Configuring MAC Limiting (ELS)
Configuring Persistent MAC Learning (ELS) | 366

Verifying That MAC Move Limiting Is Working Correctly

IN THIS SECTION

Purpose

Verify that MAC move limiting is working on the switch.

Action

Display the MAC addresses in the Ethernet switching table when MAC move limiting has been configured for a VLAN. The following sample shows the results after two of the hosts on **ge-0/0/2** sent packets after the MAC addresses for those hosts had moved to other interfaces more than five times in 1 second. The VLAN, **employee-vlan**, was set to a MAC move limit of **5** with the action **drop**:

```
user@switch> show ethernet-switching table
```

```
Ethernet-switching table:  7 entries, 4 learned

  VLAN                MAC address        Type         Age    Interfaces

  employee-vlan       00:05:85:3A:82:77  Learn          0    ge-0/0/1.0
  employee-vlan       00:05:85:3A:82:79  Learn          0    ge-0/0/1.0
  employee-vlan       00:05:85:3A:82:80  Learn          0    ge-0/0/2.0
  employee-vlan       00:05:85:3A:82:81  Learn          0    ge-0/0/2.0
  employee-vlan       *                  Flood          -    ge-0/0/2.0
  employee-vlan       *                  Flood          -    ge-0/0/2.0
```

Meaning

The last two lines of the sample output show that MAC addresses for two hosts on **ge-0/0/2** were not learned, because the hosts had been moved back and forth from the original interfaces more than five times in 1 second.

RELATED DOCUMENTATION

Configuring MAC Move Limiting (non-ELS)

Configuring MAC Move Limiting (J-Web Procedure)

Configuring Autorecovery for Port Security Events | **758**

Example: Configuring Port Security (non-ELS) | **15**

Monitoring Port Security

Verifying That the Port Error Disable Setting Is Working Correctly

IN THIS SECTION

Purpose

Verify that the port error disable setting is working as expected for MAC limited, MAC move limited, and rate-limited interfaces on an EX Series switch, or that MAC limited and storm control interfaces are working as expected for QFX Series switches or NFX Series devices.

Action

Display information about interfaces:

```
user@switch> show ethernet-switching interfaces
Interface    State    VLAN members           Blocking
ge-0/0/0.0   up       T1122                  unblocked
ge-0/0/1.0   down     default                MAC limit exceeded
ge-0/0/2.0   down     default                MAC move limit exceeded
ge-0/0/3.0   down     default                Storm control in effect
ge-0/0/4.0   down     default                unblocked
```

Meaning

For interfaces disabled by port security features, the sample output from the `show ethernet-switching interfaces` command shows the reason that the down interface is disabled:

- **MAC limit exceeded**—The interface is temporarily disabled because of a MAC limit error. The disabled interface is automatically restored to service when the `disable-timeout` expires.
- **MAC move limit exceeded**—The interface is temporarily disabled because of a MAC move limit error. The disabled interface is automatically restored to service when the `disable-timeout` expires.
- **Storm control in effect** —The interface is temporarily disabled because of a storm control error. The disabled interface is automatically restored to service when the `disable-timeout` expires.

RELATED DOCUMENTATION

Configuring Autorecovery for Port Security Events | **758**

Understanding MAC Limiting and MAC Move Limiting for Port Security

mac-limit

mac-move-limit

disable-timeout

7

PART

DHCP Protection

CHAPTER 14

DHCPv4 and DHCPv6

IN THIS CHAPTER

Understanding and Using Trusted DHCP Servers

IN THIS SECTION

Understanding Trusted and Untrusted Ports and DHCP Servers

DHCP servers provide IP addresses and other configuration information to the network’s DHCP clients. Using trusted ports for the DHCP server protects against rogue DHCP servers sending leases.

Untrusted ports drop traffic from DHCP servers to prevent unauthorized servers from providing any configuration information to clients.

By default, all trunk ports are trusted for DHCP and all access ports are untrusted.

You can configure an override of the default behavior to set a trunk port as untrusted, which blocks all ingress DHCP server messages from that interface. This is useful for preventing a rogue DHCP server attack, in which an attacker has introduced an unauthorized server into the network. The information provided to DHCP clients by this server has the potential to disrupt their network access. The unauthorized server might also assign itself as the default gateway device for the network. An attacker can then sniff the network traffic and perpetrate a man-in-the-middle attack—that is, it misdirects traffic intended for a legitimate network device to a device of its choice.

You can also configure an access port as trusted. If you attach a DHCP server to an access port, you must configure the port as trusted. Before you do so, ensure that the server is physically secure—that is, that access to the server is monitored and controlled.

Enabling a Trusted DHCP Server (ELS)

NOTE: This example uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style.

You can configure any interface on a switch that connects to a DHCP server as a trusted interface (port). Configuring a DHCP server on a trusted interface protects against rogue DHCP servers sending leases.

By default, all access interfaces are untrusted, and all trunk interfaces are trusted. However, you can override the default setting for access interfaces by configuring a group of access interfaces within a VLAN, specifying an interface to belong to that group, and then configuring the group as trusted.

Before you can configure a trusted DHCP server, you must configure a VLAN. See Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure).

To configure an untrusted access interface as a trusted interface for a DHCP server by using the CLI :

1. Configure a group within a VLAN with a specific access interface:

```
[edit vlans vlan-name forwarding-options dhcp-security ]
user@switch# set group group-name interface interface-name
```

2. Configure that group as `trusted` to make the specified interface contained within the group a trusted interface:

```
[edit vlans vlan-name forwarding-options dhcp-security group group-name]
user@switch# set overrides trusted
```

Enabling a Trusted DHCP Server (non-ELS)

You can protect against rogue DHCP servers sending rogue leases on your network by using trusted DHCP servers and ports. By default, for DHCP, all trunk ports are trusted, and all access ports are untrusted. And you can only set up DHCP server on an interface; that is, using a VLAN is not supported.

Trusted ports allow DHCP servers to provide IP addresses and other information to requesting devices. Untrusted ports drop traffic from DHCP servers to prevent unauthorized servers from providing any configuration information to clients.

To configure a port to host a DHCP server, enter the following command from the Junos CLI:

```
[edit ethernet-switching-options secure-access port]
user@switch# set interface ge-0/0/8 dhcp-trusted
```

where, the interface, **ge-0/0/8** is any trusted and physically secure interface that is valid for your network.

SEE ALSO

Example: Configuring Port Security (non-ELS) | **15**

Monitoring Port Security

Enabling a Trusted DHCP Server (MX Series Routers)

You can configure any interface on a switching device that connects to a DHCP server as a trusted interface (port). Configuring a DHCP server on a trusted interface protects against rogue DHCP servers sending leases.

By default, all access interfaces are untrusted, and all trunk interfaces are trusted. However, you can override the default setting for access interfaces by configuring a group of access interfaces within a bridge domain, specifying an interface to belong to that group, and then configuring the group as trusted.

Before you can configure a trusted DHCP server, you must configure a bridge domain.

To configure an untrusted access interface as a trusted interface for a DHCP server by using the CLI :

1. Configure a group within a bridge domain with a specific access interface:

   ```
   [edit bridge-domains bridge-domain-name forwarding-options dhcp-security]
   user@device# set group group-name interface interface-name
   ```

2. Configure that group as `trusted` to make the specified interface contained within the group a trusted interface:

```
[edit bridge-domains bridge-domain-name forwarding-options dhcp-security group group-name]
user@device# set overrides trusted
```

Verifying That a Trusted DHCP Server Is Working Correctly

IN THIS SECTION

Purpose

Verify that a DHCP trusted server is working on the switch. See what happens when the DHCP server is trusted and then untrusted.

Action

Send some DHCP requests from network devices (here they are DHCP clients) connected to the switch.

Display the DHCP snooping information when the interface on which the DHCP server connects to the switch is trusted. The following output results when requests are sent from the MAC addresses and the server has provided the IP addresses and leases:

```
user@switch>                      show dhcp snooping binding
DHCP Snooping Information:
MAC Address         IP Address    Lease    Type     VLAN          Interface

-----------------   ----------    -----    ----     ----          ---------


00:05:85:3A:82:77    192.0.2.17    600    dynamic  employee—vlan   ge-0/0/1.0


00:05:85:3A:82:79    192.0.2.18    653    dynamic  employee—vlan   ge-0/0/1.0


00:05:85:3A:82:80    192.0.2.19    720    dynamic  employee—vlan   ge-0/0/2.0
```

```
00:05:85:3A:82:81    192.0.2.20    932    dynamic   employee—vlan   ge-0/0/2.0

00:05:85:3A:82:83    192.0.2.21   1230    dynamic   employee—vlan   ge-0/0/2.0

00:05:85:27:32:88    192.0.2.22   3200    dynamic   employee—vlan   ge-0/0/2.0
```

Meaning

When the interface on which the DHCP server connects to the switch has been set to trusted, the output (see preceding sample) shows, for each MAC address, the assigned IP address and lease time—that is, the time, in seconds, remaining before the lease expires.

If the DHCP server had been configured as untrusted, no entries would be added to the DHCP snooping database and nothing would be shown in the output of the `show dhcp snooping binding` command.

SEE ALSO

- Example: Configuring Port Security (non-ELS) | **15**
- Example: Protecting against Rogue DHCP Server Attacks | **387**
- Monitoring Port Security
- Troubleshooting Port Security

Configuring a Trunk Interface as Untrusted for DHCP Security (CLI Procedure)

Before you can configure a group of interfaces, you must configure a VLAN. See Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure).

Untrusted trunk interfaces support the following DHCP security features when they are enabled on the VLAN:

- DHCP and DHCPv6 snooping
- Dynamic ARP inspection
- IPv6 neighbor discovery inspection

To configure a trunk interface as untrusted, you must configure a group of interfaces within a VLAN, add the trunk interface to the group, and then configure the group as untrusted. A group must have at least one interface.

To configure a trunk interface as untrusted for DHCP security:

1. Configure a group within a VLAN with the trunk interface as a member:

   ```
   [edit vlans vlan-name forwarding-options dhcp-security]
   user@switch# set group group-name interface interface-name
   ```

2. Configure the group as `untrusted` to make the specified interface contained within the group an untrusted interface:

   ```
   [edit vlans vlan-name forwarding-options dhcp-security group group-name]
   user@switch# set overrides untrusted
   ```

Example: Protecting against Rogue DHCP Server Attacks

IN THIS SECTION

In a rogue DHCP server attack, an attacker has introduced a rogue server into the network, allowing it to give IP address leases to the network's DHCP clients and to assign itself as the gateway device.

This example describes how to configure a DHCP server interface as untrusted to protect the switch from a rogue DHCP server:

Requirements

This example uses the following hardware and software components:

- One EX Series switch or one QFX3500 switch
- Junos OS Release 9.0 or later for EX Series switches or Junos OS Release 12.1 or later for the QFX Series

- A DHCP server to provide IP addresses to network devices on the switch

Before you configure an untrusted DHCP server interface to mitigate rogue DHCP server attacks, be sure you have:

- Connected the DHCP server to the switch.
- Enabled DHCP snooping on the VLAN.
- Configured a VLAN on the switch. See the task for your platform:
  - Example: Setting Up Bridging with Multiple VLANs.

Overview and Topology

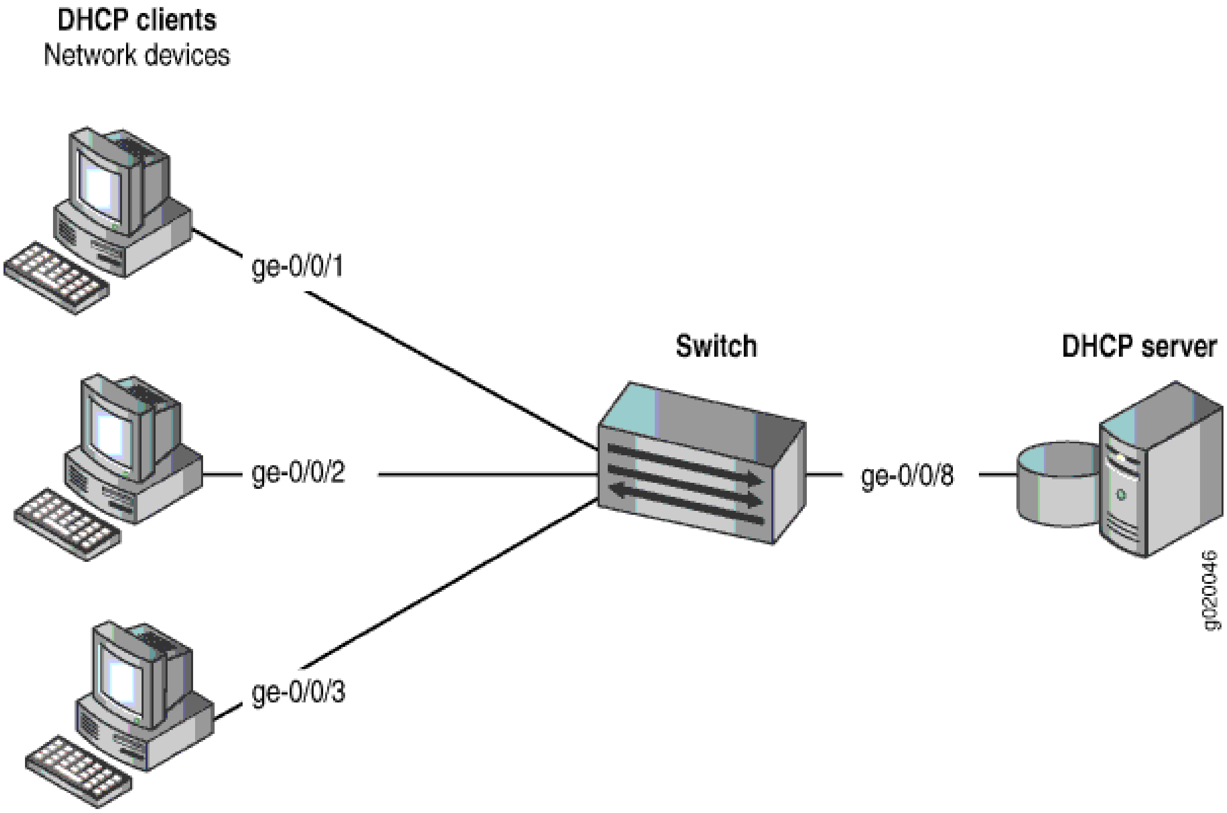
IN THIS SECTION

Ethernet LANs are vulnerable to address spoofing and DoS attacks on network devices. This example describes how to protect the switch from rogue DHCP server attacks.

This example shows how to explicitly configure an untrusted interface on an EX3200-24P switch and a QFX3500 switch. Figure 17 on page 422 illustrates the topology for this example.

Topology

Figure 17: Network Topology for Basic Port Security



The components of the topology for this example are shown in Table 20 on page 422.

Table 20: Components of the Port Security Topology

Properties	Settings
Switch hardware	One EX3200-24P, 24 ports (8 PoE ports) or one QFX3500 switch
VLAN name and ID	**employee-vlan**, tag **20**
VLAN subnets	**192.0.2.16/28** **192.0.2.17** through **192.0.2.30** **192.0.2.31** is the subnet's broadcast address

Table 20: Components of the Port Security Topology *(Continued)*

Properties	Settings
Interfaces in **employee-vlan**	ge-0/0/1, ge-0/0/2, ge-0/0/3, ge-0/0/8
Interface for DHCP server	ge-0/0/8

In this example, the switch has already been configured as follows:

- Secure port access is activated on the switch.
- DHCP snooping is enabled on the VLAN **employee-vlan**.
- The interface (port) where the rogue DHCP server has connected to the switch is currently trusted.

Configuration

IN THIS SECTION

To configure the DHCP server interface as untrusted because the interface is being used by a rogue DHCP server:

Procedure

CLI Quick Configuration

To quickly set the rogue DHCP server interface as untrusted, copy the following command and paste it into the switch terminal window:

```
[edit ethernet-switching-options secure-access-port]
set interface ge-0/0/8 no-dhcp-trusted
```

Step-by-Step Procedure

To set the DHCP server interface as untrusted:

- Specify the interface (port) from which DHCP responses are not allowed:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface ge-0/0/8 no-dhcp-trusted
```

Results

Check the results of the configuration:

```
[edit ethernet-switching-options secure-access-port]
user@switch# show
interface ge-0/0/8.0 {
    no-dhcp-trusted;
}
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying That the DHCP Server Interface Is Untrusted

Purpose

Verify that the DHCP server is untrusted.

Action

1. Send some DHCP requests from network devices (here they are DHCP clients) connected to the switch.

2. Display the DHCP snooping information when the port on which the DHCP server connects to the switch is not trusted.

Meaning

There is no output from the command because no entries are added to the DHCP snooping database.

RELATED DOCUMENTATION

Understanding and Using Trusted DHCP Servers | 415

Example: Configuring Port Security (non-ELS) | 15

show dhcp snooping binding

DHCPv6 Rapid Commit

IN THIS SECTION

Configuring DHCPv6 Rapid Commit (MX Series, EX Series)

You can configure the DHCPv6 local server to support the DHCPv6 Rapid Commit option (DHCPv6 option 14). When rapid commit is enabled, the server recognizes the Rapid Commit option in Solicit messages sent from the DHCPv6 client. (DHCPv6 clients are configured separately to include the DHCPv6 Rapid Commit option in the Solicit messages.) The server and client then use a two-message exchange (Solicit and Reply) to configure clients, rather than the default four-message exchange (Solicit, Advertise, Request, and Reply). The two-message exchange provides faster client configuration, and is beneficial in environments in which networks are under a heavy load.

You can configure the DHCPv6 local server to support the Rapid Commit option globally, for a specific group, or for a specific interface. By default, rapid commit support is disabled on the DHCPv6 local server.

To configure the DHCPv6 local server to support the DHCPv6 Rapid Commit option:

1. Specify that you want to configure the `overrides` options:

```
[edit system services dhcp-local-server dhcpv6]
user@host# edit overrides
```

2. Enable rapid commit support:

```
[edit system services dhcp-local-server dhcpv6 overrides]
user@host# set  rapid-commit
```

SEE ALSO

> *Overriding the Default DHCP Local Server Configuration Settings*

Configuring the DHCPv6 Client Rapid Commit Option

The DHCPv6 client can obtain configuration parameters from a DHCPv6 server through a rapid two-message exchange (solicit and reply). When the rapid commit option is enabled by both the DHCPv6 client and the DHCPv6 server, the two-message exchange is used, rather than the default four-method exchange (solicit, advertise, request, and reply). The two-message exchange provides faster client configuration and is beneficial in environments in which networks are under a heavy load.

To configure the DHCPv6 client to support the DHCPv6 rapid commit option on SRX300, SRX320, SRX340, SRX550M, and SRX1500 devices:

1. Specify the DHCPv6 client interface.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet6 dhcpv6-client
```

2. Configure the two-message exchange option for address assignment.

```
[edit interfaces ge-0/0/0 unit 0 family inet6 dhcpv6-client]
user@host# set rapid-commit
```

Using Lightweight DHCPv6 Relay Agent (LDRA)

In Layer 2 networks that have many nodes on a single link, a DHCP server would normally be unaware of how a DHCP client is attached to the network. In a DHCPv6 deployment, you can use a Lightweight DHCPv6 Relay Agent (LDRA) to add relay agent information to a DHCPv6 message to identify the client-facing interface of the access node that received the message. The server can use this information to assign IP addresses, prefixes, and other configuration parameters for the client.

DHCPv6 relay agents are typically used to forward DHCPv6 messages between clients and servers or other relay agents when they are not on the same IPv6 link node. The relay agent can add information to the messages before relaying them. When the client and server reside on the same IPv6 link, LDRA enables a switching device to perform the function of intercepting DHCPv6 messages and inserting relay agent information that can be used for client identification. The LDRA acts as a relay agent, but without performing the routing function necessary to forward messages to a server or relay agent that resides on a different IPv6 link.

When the LDRA receives a DHCPv6 Solicit message from a client, it encapsulates that message within a DHCPv6 Relay-Forward message, which it then forwards to the server or another relay agent. Before it forwards the Relay-Forward message, the LDRA can also insert relay information by using one or more of the following options:

- `option-16` (Vendor ID)—Option 16 provides the server with information about the vendor that manufactured the hardware on which the DHCPv6 client is running. Option 16 is the DHCPv6 equivalent of the `vendor-id` suboption of DHCP option 82.

- `option-18` (Interface ID)—A unique identifier for the interface on which the client DHCPv6 packet is received. Suboptions can be configured to include a prefix with the interface ID or to change the type of information used to identify the interface. Option 18 is the DHCPv6 equivalent of the `circuit-id` suboption of DHCP option 82.

- `option-37` (Remote ID)—A unique identifier for the remote host. Suboptions can be configured to include a prefix with the remote ID or to change the interface portion of the ID. Option 37 is the DHCPv6 equivalent of the `remote-id` suboption of DHCP option 82.

You must configure LDRA if you configure DHCPv6 options at the `[edit vlan` *`vlan-name`* `forwarding-options dhcp-security dhcpv6-options]` hierarchy level. Option 16, option 37, and option 79 are included in the Relay-Forward message only if they are explicitly configured. Option 18 is mandatory in Relay-Forward messages and is included even if it is not explicitly configured. However, suboptions of option 18 are included only if they are configured using the `option-18` statement at the `[edit vlan` *`vlan-name`* `forwarding-options dhcp-security dhcpv6-options]` hierarchy level.

To configure LDRA to enable DHCPv6 options:

1. Configure the switch as an LDRA.

```
[edit vlans vlan-name forwarding-options dhcp-security]
user@switch# set light-weight-dhcpv6-relay
```

2. Configure the switch to insert DHCPv6 options in the Relay-Forward message to provide additional information about the client to the server or to another relay agent.

   - To insert option 16:

   ```
   [edit vlans vlan-name forwarding-options dhcp-security]
   user@switch# set dhcpv6-options option-16
   ```

   - To insert option 18:

   ```
   [edit vlans vlan-name forwarding-options dhcp-security]
   user@switch# set dhcpv6-options option-18
   ```

   - To insert option 37:

   ```
   [edit vlans vlan-name forwarding-options dhcp-security]
   user@switch# set dhcpv6-options option-37
   ```

   - To insert option 79:

   ```
   [edit vlans vlan-name forwarding-options dhcp-security]
   user@switch# set dhcpv6-options option-79
   ```

3. (Optional) Configure a prefix to include additional information with DHCPv6 option 18 or DHCPv6 option 37. For example, to configure a prefix for option 37 to include the switch’s hostname:

```
[edit vlans vlan-name forwarding-options dhcp-security dhcpv6-options]
user@switch# set option-37 prefix host-name
```

4. (Optional) Change the type of information used to identify the interface. For example, to specify that option 18 contain the interface description for the logical unit rather than the interface name (the default):

```
[edit vlans vlan-name forwarding-options dhcp-security dhcpv6-options]
user@switch# set option-18 use-interface-description logical
```

> **NOTE**: To use the interface description rather than the interface name for identifying the interface, the interface description must be specified under interface unit (`set interfaces ge-0/0/0 unit 0 description` *`description`*). If you do not do this, then the interface name is used by default.

RELATED DOCUMENTATION

Understanding DHCP Option 82 | **501**

Configuring Persistent Bindings in the DHCP or DHCPv6 (ELS)

> **NOTE**: This task uses Junos OS with support for the Enhanced Layer 2 Software (ELS) configuration style.If your switch runs software that does not support ELS, see "Configuring Persistent Bindings in the DHCP or DHCPv6 (non-ELS)" on page 431. For ELS details, see Using the Enhanced Layer 2 Software CLI.

By default, IP-MAC address bindings in the DHCP snooping database do not persist through device reboots. You can improve network performance by configuring the IP-MAC address bindings in the DHCP snooping database to persist through reboots so that the table does not need to be rebuilt after rebooting. Do this by configuring a storage location for the DHCP snooping database file, where you must specify how frequently the device writes the database entries into the DHCP snooping database file.

> **NOTE**: You can also configure persistent bindings for IPv6 addresses and MAC addresses on devices that support DHCPv6 snooping.
>
> DHCPv6 is not supported on the MX Series routers.

The DHCP snooping database of IP-MAC bindings is created when you enable any of the port security features for a specific VLAN or bridge domain in either of the following hierarchy levels:

- `[edit vlans vlan-name forwarding-options dhcp-security]`
- `[edit bridge-domains bridge-domain-name forwarding-options dhcp-security]`

On devices that support DHCPv6, enabling any port security features will automatically enable DHCPv6 snooping. DHCP snooping and DHCPv6 snooping are not enabled by default.

> **TIP**: By default, the IP-MAC bindings are lost when the switch is rebooted, and the DHCP clients (the network devices, or hosts) must reacquire bindings. However, you can configure the bindings to persist by setting the `dhcp-snooping-file` statement to store the database file either locally or remotely. When you configure and enable 802.1x dynamic VLAN, the DHCP snooping entries also get deleted. Due to this, it is recommended to configure for a DHCP server to store lease information for clients and provide them with a predictable IP address even after you reboot the client (DHCP persistence).

To configure a *local* storage location for the DHCP snooping database file:

- For DHCP snooping:

  ```
  [edit system processes]
  user@device# set dhcp-service dhcp-snooping-file local-pathname write-interval seconds
  ```

  For example:

  ```
  [edit system processes]
  user@device# set dhcp-service dhcp-snooping-file /var/tmp/test.log write-interval 60
  ```

- For DHCPv6 snooping:

  ```
  [edit system processes]
  user@device# set dhcp-service dhcpv6-snooping-file local-pathname write-interval seconds
  ```

  For example:

  ```
  [edit system processes]
  user@device# set dhcp-service dhcpv6-snooping-file /var/tmp/test.log write-interval 60
  ```

To configure a *remote* storage location for IP-MAC bindings, use `tftp://`*`ip-address`* or `ftp://`*`hostname/path`* as the remote URL, or the local pathname for the storage location of the DHCP or DHCPv6 snooping database file:

- For DHCP snooping:

  ```
  [edit system processes]
  user@device# set dhcp-service dhcp-snooping-file remote_url write-interval seconds
  ```

  For example:

  ```
  [edit system processes]
  user@device# set dhcp-service dhcp-snooping-file tftp://@14.1.2.1 write-interval 60
  ```

- For DHCPv6 snooping:

  ```
  [edit system processes]
  user@device# set dhcp-service dhcpv6-snooping-file remote_url write-interval seconds
  ```

  For example:

  ```
  [edit system processes]
  user@device# set dhcp-service dhcpv6-snooping-file tftp://@14.1.2.1 write-interval 60
  ```

.

RELATED DOCUMENTATION

Understanding DHCP Snooping (non-ELS) | **445**

Configuring Persistent Bindings in the DHCP or DHCPv6 (non-ELS)

NOTE: This task uses Junos OS without support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see"Configuring

Persistent Bindings in the DHCP or DHCPv6 (ELS)" on page 429 instead. For ELS details, see Using the Enhanced Layer 2 Software CLI.

By default, IP-MAC bindings in the DHCP snooping database do not persist through switch reboots. You can configure the IP-MAC bindings in the DHCP snooping database to persist through switch reboots by configuring a storage location for the DHCP snooping database file. When specifying the location for the DHCP snooping database, you must also specify how frequently the switch writes the database entries into the DHCP snooping database file.

The DHCP snooping database of IP-MAC bindings is created when you enable DHCP snooping. DHCP snooping is not enabled by default. You can configure DHCP snooping on a specific VLAN or on all VLANs. See "Enabling DHCP Snooping (non-ELS)" on page 456.

To configure a local storage location for the DHCP snooping database file:

- For DHCPv4 snooping:

  ```
  [edit ethernet-switching-options]
  user@switch# set secure-access-port dhcp-snooping-file location local-pathname write-interval seconds
  ```

  For example:

  ```
  [edit ethernet-switching-options]
  user@switch# set secure-access-port dhcp-snooping-file location /var/tmp/test.log write-interval 60
  ```

- For DHCPv6 snooping:

  ```
  [edit ethernet-switching-options]
  user@switch# set secure-access-port dhcpv6-snooping-file location local-pathname write-interval seconds
  ```

  For example:

  ```
  [edit ethernet-switching-options]
  user@switch# set secure-access-port dhcpv6-snooping-file location /var/tmp/test.log write-interval 60
  ```

To configure a remote storage location for IP-MAC bindings, use tftp://ip-address or ftp://hostname/path as the remote URL or the local pathname for the storage location of the DHCP or DHCPv6 snooping database file:

- For DHCPv4 snooping:

  ```
  [edit ethernet-switching-options]
  user@switch# set secure-access-port dhcp-snooping-file location remote_url write-interval
  seconds
  ```

  For example:

  ```
  [edit ethernet-switching-options]
  user@switch# set secure-access-port dhcp-snooping-file location ftp://test:Test123@14.1.2.1
  write-interval 60
  ```

- For DHCPv6 snooping:

  ```
  [edit ethernet-switching-options]
  user@switch# set secure-access-port dhcpv6-snooping-file location remote_url write-interval
  seconds
  ```

  For example:

  ```
  [edit ethernet-switching-options]
  user@switch# set secure-access-port dhcpv6-snooping-file location ftp://test:Test123@14.1.2.1
  write-interval 60
  ```

NOTE: If you save the DHCP or DHCPv6 snooping file to a remote server using TFTP, then the CLI returns a message that the save process is initiated. The CLI remains accessible during the save process; however, if you attempt to save a file while the previous save is still pending, the CLI returns an error message.

NOTE: Specify any requisite user credentials for the FTP server before you specify the IP address or hostname. In this example, **test** is the username and **Test123** is the password for FTP server 14.1.2.1.

When you are storing the DHCP snooping database at a remote location, you might also want to specify a timeout value for remote read and write operations. See `timeout`. This configuration is optional.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1X53-D40	If you save the DHCP or DHCPv6 snooping file to a remote server using TFTP, then the CLI returns a message that the save process is initiated.

RELATED DOCUMENTATION

Understanding DHCP Snooping (non-ELS) | 445

CHAPTER 15

DHCP Snooping

IN THIS CHAPTER

Understanding DHCP Snooping (ELS)

IN THIS SECTION

NOTE: This topic includes information about enabling Dynamic Host Configuration Protocol (DHCP) snooping when using Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs Junos OS software that does not support ELS, see "Understanding DHCP Snooping (non-ELS)" on page 445. For ELS details, see Using the Enhanced Layer 2 Software CLI.

DHCP snooping enables the switching device, which can be either a switch or a router, to monitor DHCP messages received from untrusted devices connected to the switching device. When DHCP snooping is enabled on a VLAN, the system examines DHCP messages sent from untrusted hosts associated with the VLAN and extracts their IP addresses and lease information. This information is used to build and maintain the DHCP snooping database. Only hosts that can be verified using this database are allowed access to the network.

DHCP Snooping Basics

DHCP allocates IP addresses dynamically, *leasing* addresses to devices so that the addresses can be reused when they are no longer needed by the devices to which they were assigned. Hosts and end devices that require IP addresses obtained through DHCP must communicate with a DHCP server across the LAN.

DHCP snooping acts as a guardian of network security by keeping track of valid IP addresses assigned to downstream network devices by a trusted DHCP server (the server is connected to a trusted network port).

By default, all trunk ports on the switch are trusted and all access ports are untrusted for DHCP snooping.

Starting with Junos OS Release 18.4R1, DHCP snooping occurs on trusted ports for the following Juniper Series switches, EX2300, EX4600, and QFX5K. Prior to Junos OS Release 18.4R1, for these devices, this was true only for DHCPv6 snooping. In addition, DHCP snooping occurs on trusted ports for EX9200 Series switches, and Fusion Enterprises, that are running Junos OS Release 19.1R1 and later.

You can configure an access port as trusted, or a trunk port as untrusted, using the *overrides* configuration statement with either the *trusted* or *untrusted* option.

When DHCP snooping is enabled, the lease information from the server is used to create the DHCP snooping table, also known as the DHCP binding table. The table shows current IP-MAC address bindings, as well as lease time, type of binding, names of associated VLANs and interfaces.

Entries in the DHCP snooping table are updated in the following events:

- When a network device releases an IP address (sends a DHCPRELEASE message). In this event, the associated mapping entry is deleted from the database.

- When you move a network device from one VLAN to another. In this event, typically the device needs to acquire a new IP address. Therefore, its entry in the database, including the VLAN name, is updated.
- When the lease time (timeout value) assigned by the DHCP server expires. In this event, the associated entry is deleted from the database.
- When the network device renews its lease by sending a unicast DHCPREQUEST message and receiving a positive response from the DHCP server. In this event, the lease time is updated in the database.
- If the network device cannot reach the DHCP server that originally granted the lease, it sends a broadcast DHCPREQUEST message and rebinds to the DHCP server that responds. In this event, the client receives a new IP address and the binding is updated in the DHCP snooping table.
- Starting in Junos OS Release 14.1X53-D35, if a network device with a fixed IP allocation from the DHCP server is replaced by a new device with a different MAC address, the new IP-MAC address binding is stored until the server sends a DHCPACK message; then, the entry in the DHCP snooping table is updated with the new address binding.

TIP: By default, the IP-MAC bindings are lost when the switch is rebooted, and the DHCP clients (the network devices, or hosts) must reacquire bindings. However, you can configure the bindings to persist by setting the `dhcp-snooping-file` statement to store the database file either locally or remotely. When you configure and enable 802.1x dynamic VLAN, the DHCP snooping entries also get deleted. Due to this, it is recommended to configure for a DHCP server to store lease information for clients and provide them with a predictable IP address even after you reboot the client (DHCP persistence).

You can configure the switch to snoop DHCP server responses only from specific VLANs. Doing this prevents spoofing of DHCP server messages.

Enabling DHCP Snooping

When you are using the DHCP snooping feature, it is important that you understand about enabling the DHCP snooping feature.

On Junos OS device, you cannot configure DHCP snooping feature as an independent feature. Whenever you configure DHCP security for a specific VLAN of the device, the DHCP snooping is automatically enabled on that VLAN to perform it’s task.

For example:

- When you enable DHCP security on a specific VLAN, DHCP snooping gets automatically enabled on that VLAN.

Junos OS enables DHCP snooping on a switch:

- When you configure the following option for any port security features:
  - dhcp-security statement at the [edit vlans vlan-name forwarding-options] hierarchy level.

Prior to Junos OS Release 17.1, Junos OS enabled DHCP snooping automatically if you configure any of the following port security features within [edit vlans vlan-name forwarding-options] hierarchy:

- Dynamic ARP inspection (DAI
- IP source guard
- DHCP option 82
- Static IP

Starting in Junos OS Release 17.1R1, you can configure DHCP snooping or DHCPv6 snooping on a VLAN without enabling other port security features. Use the following configuration statement to enable DHCP snooping:

[set vlans *vlan-name* forwarding-options dhcp-security].

For additional information about enabling DHCP snooping, see " Configuring Port Security (ELS)" on page 9

NOTE: To disable DHCP snooping, you must delete the dhcp-security statement from the configuration. DHCP snooping is not disabled automatically when you disable other port security features.

DHCP Snooping Process

The DHCP snooping process consists of the following steps:

NOTE: When DHCP snooping is enabled for a VLAN, all DHCP packets sent from network devices in that VLAN are subjected to DHCP snooping. The final IP-MAC binding occurs when the DHCP server sends a DHCPACK packet to the DHCP client.

1. The network device sends a DHCPDISCOVER packet to request an IP address.
2. The switch forwards the packet to the DHCP server.
3. The server sends a DHCPOFFER packet to offer an address. If the DHCPOFFER packet is from a trusted interface, the switch forwards the packet to the network device.

4. The network device sends a DHCPREQUEST packet to accept the IP address. The switch adds an IP-MAC placeholder binding to the DHCP snooping table. The entry is considered a placeholder until a DHCPACK packet is received from the server. Until then, the IP address could still be assigned to some other host.

5. The server sends a DHCPACK packet to assign the IP address or a DHCPNAK packet to deny the address request.

6. The switch updates the DHCP database according to the type of packet received:

   - If the switch receives a DHCPACK packet, it updates lease information for the IP-MAC address bindings in its database.

   - If the switch receives a DHCPNACK packet, it deletes the placeholder.

NOTE: The DHCP database is updated only after the DHCPREQUEST packet is sent.

For general information about the messages that the DHCP client and DHCP server exchange during the assignment of an IP address for the client, see the *Junos OS System Basics Configuration Guide* .

DHCPv6 Snooping

Starting in Junos OS Release 14.1X53-D10, DHCP snooping is supported for IPv6 packets on EX 9200 Switches.DHCP snooping is also supported for IPv6 packets. The process for DHCPv6 snooping is similar to that for DHCP snooping, but uses different names for the messages exchanged between the client and server to assign IPv6 addresses. Table 21 on page 439 shows DHCPv6 messages and their DHCPv4 equivalents.

Table 21: DHCPv6 Messages and DHCPv4 Equivalent Messages

Sent by	DHCPv6 Messages	DHCPv4 Equivalent Messages
Client	SOLICIT	DHCPDISCOVER
Server	ADVERTISE	DHCPOFFER
Client	REQUEST, RENEW, REBIND	DHCPREQUEST
Server	REPLY	DHCPACK/DHCPNAK

Table 21: DHCPv6 Messages and DHCPv4 Equivalent Messages *(Continued)*

Sent by	DHCPv6 Messages	DHCPv4 Equivalent Messages
Client	RELEASE	DHCPRELEASE
Client	INFORMATION-REQUEST	DHCPINFORM
Client	DECLINE	DHCPDECLINE
Client	CONFIRM	none
Server	RECONFIGURE	DHCPFORCERENEW
Client	RELAY-FORW, RELAY-REPLY	none

Rapid Commit for DHCPv6

The DHCPv6 Rapid Commit option can shorten the exchange of messages between the client and server. When supported by the server and set by the client, this option shortens the exchange from a four-way relay to a two-message handshake. For more information about enabling the Rapid Commit option, see *Configuring DHCPv6 Rapid Commit (MX Series, EX Series)*.

When the Rapid Commit option is enabled, the exchange of messages is as follows:

1. The DHCPv6 client sends out a SOLICIT message that contains a request that rapid assignment of address, prefix, and other configuration parameters be preferred.

2. If the DHCPv6 server supports rapid assignment, it responds with a REPLY message, which contains the assigned IPv6 address and prefix and other configuration parameters.

DHCP Server Access

A switch’s access to the DHCP server can be configured in three ways:

Switch, DHCP Clients, and the DHCP Server Are All on the Same VLAN

When the switch, DHCP clients, and DHCP server are *all members of the same VLAN,* the DHCP server can be connected to the switch in one of two ways:

NOTE: To enable DHCP snooping on the VLAN, configure the *dhcp-security* statement at the `[edit vlans` *`vlan-name`* `forwarding-options]` hierarchy.

- (See Figure 18 on page 441.) The server is directly connected to the same switch as the one connected to the DHCP clients (the hosts, or network devices, that are requesting IP addresses from the server). The VLAN is enabled for DHCP snooping to protect the untrusted access ports. The trunk port is configured by default as a trusted port.
- (See Figure 19 on page 442.) The server is connected to an intermediary switch (Switch 2) that is connected through a trunk port to the switch (Switch 1) that the DHCP clients are connected to. Switch 2 is being used as a transit switch. The VLAN is enabled for DHCP snooping to protect the untrusted access ports of Switch 1. The trunk port is configured by default as a trusted port. In Figure 19 on page 442, ge-0/0/11 is a trusted trunk port.

Figure 18: DHCP Server Connected Directly to a Switch

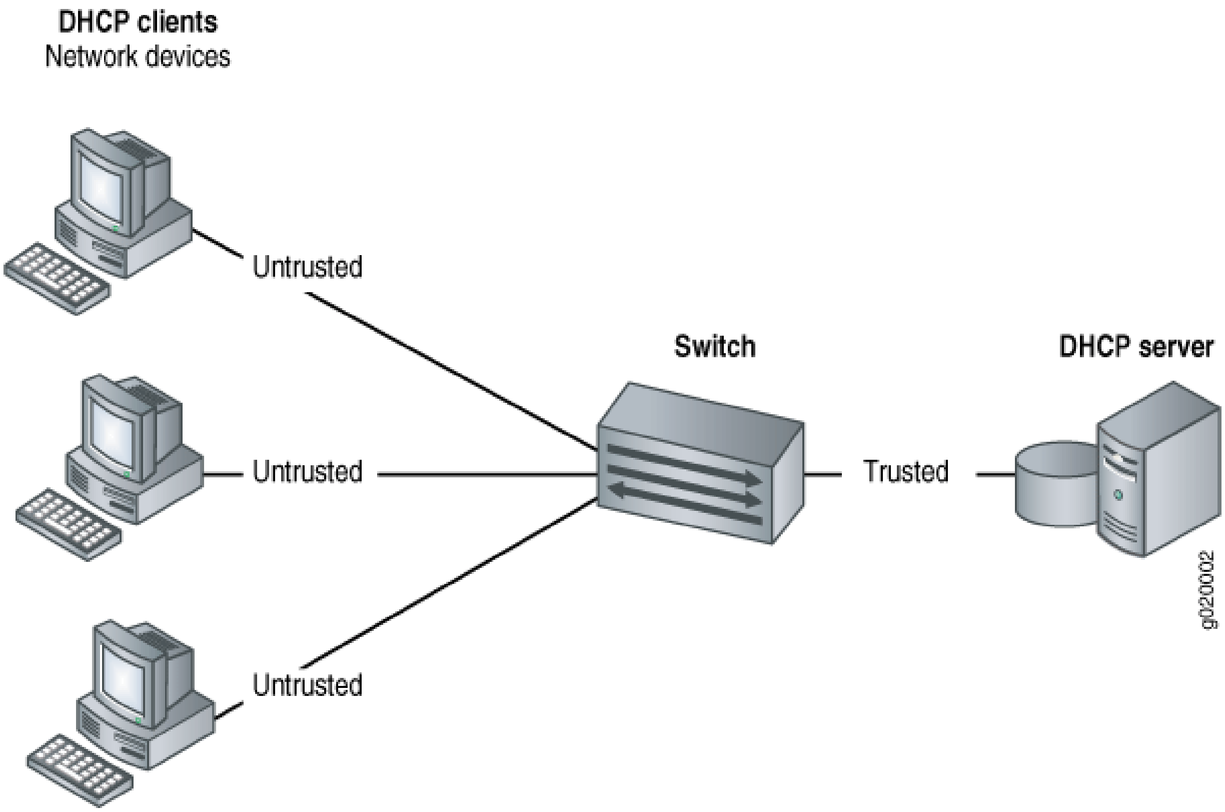
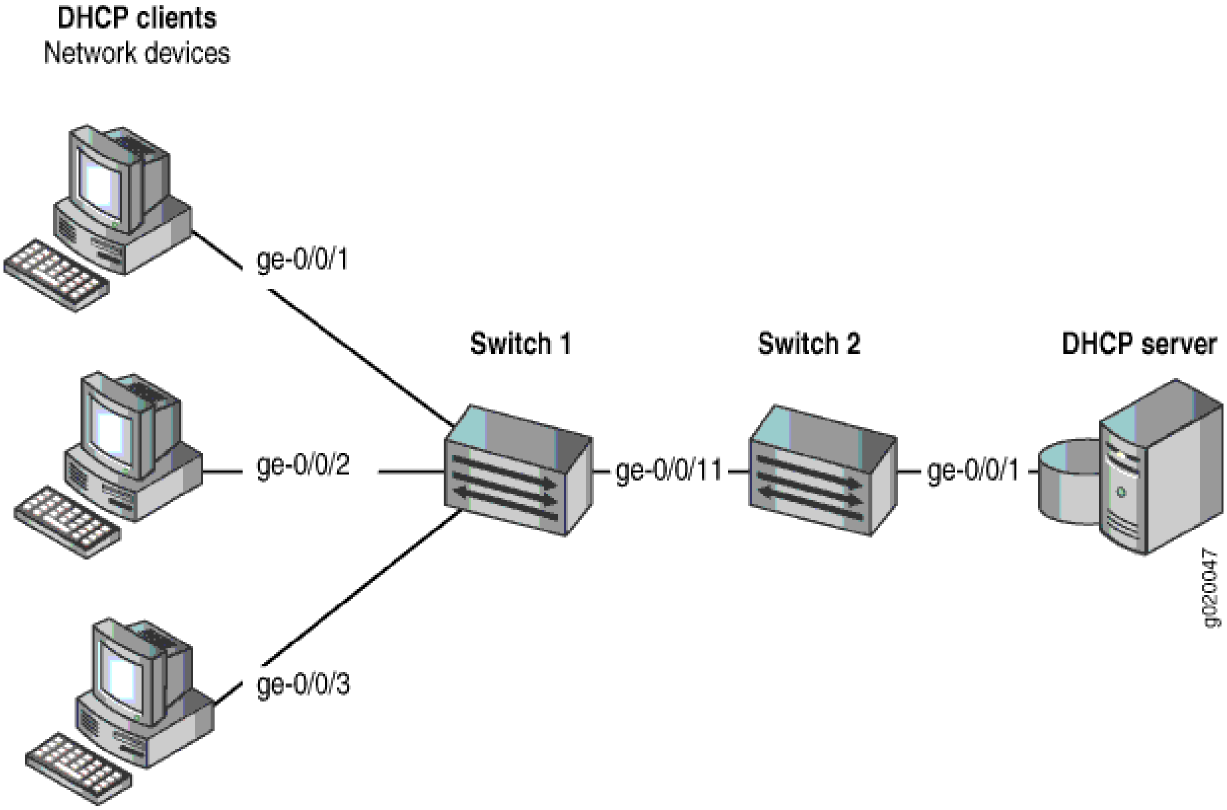


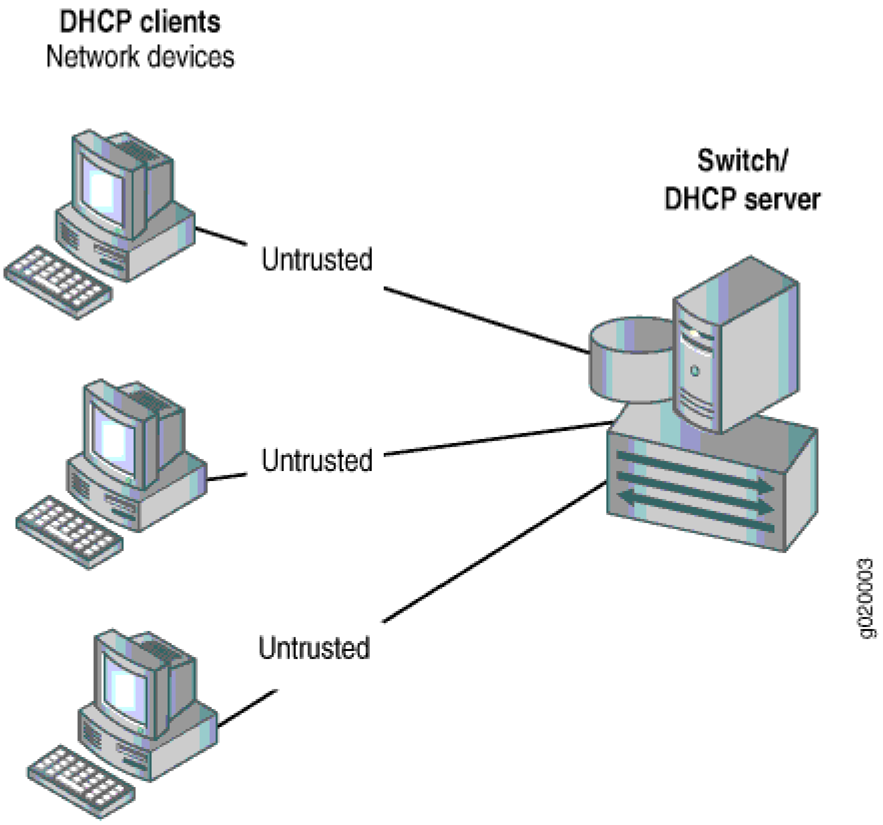
Figure 19: DHCP Server Connected Directly to Switch 2, with Switch 2 Connected to Switch 1 Through a Trusted Trunk Port



Switch Acts as the DHCP Server

You can configure DHCP local server options on the switch, which enables the switch to function as an extended DHCP local server. In Figure 20 on page 443, the DHCP clients are connected to the extended DHCP local server through untrusted access ports.

Figure 20: Switch Is the DHCP Server



Switch Acts as a Relay Agent

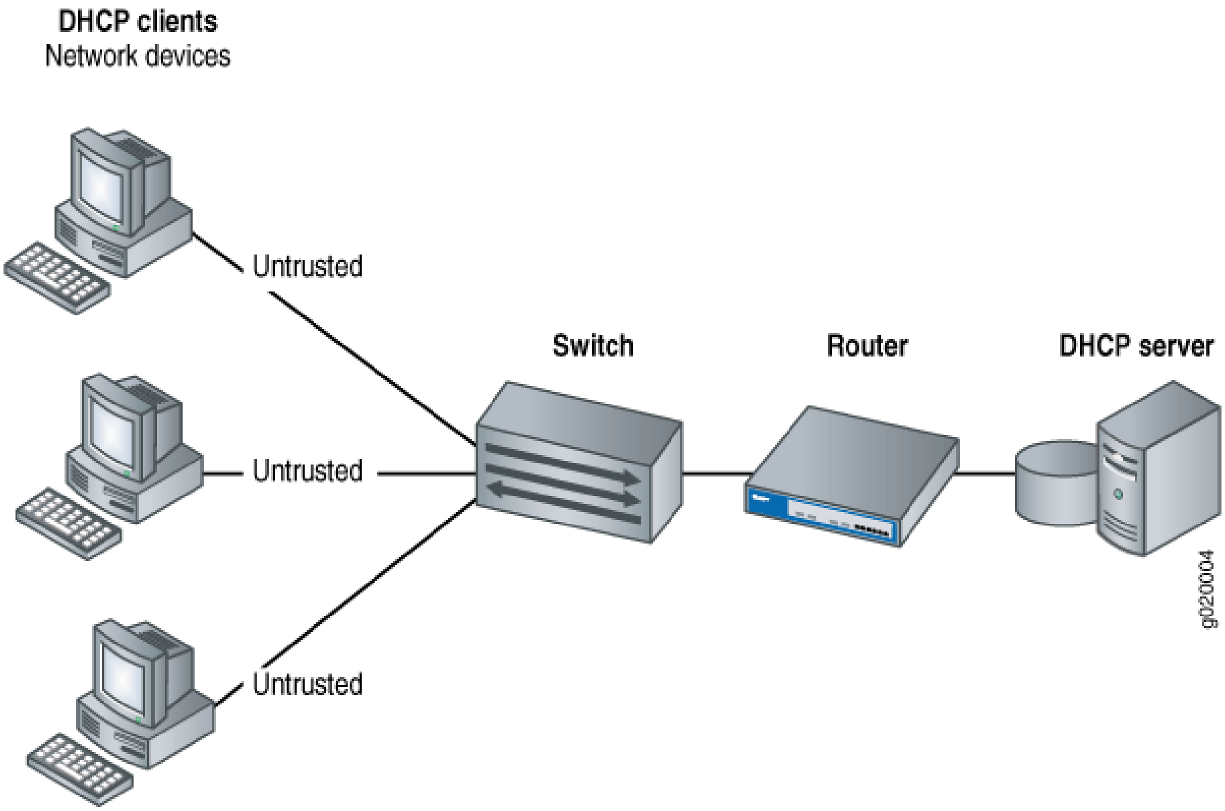
The switch functions as a relay agent when the DHCP clients or the DHCP server is connected to the switch through a Layer 3 interface (on a switch or a router). The Layer 3 interfaces on the switch are configured as routed VLAN interfaces (RVIs)—also called integrated routing and bridging (IRB) interfaces. The trunk interfaces are trusted by default.

The switch can act as a relay agent in these two scenarios:

- The DHCP server and clients are in different VLANs.

- The switch is connected to a router that is, in turn, connected to the DHCP server. See Figure 21 on page 444.

Figure 21: Switch Acting as a Relay Agent Through a Router to the DHCP Server



Static IP Address Additions to the DHCP Snooping Database

You can add static (fixed) IP addresses and bind them to fixed MAC addresses in the DHCP snooping database. These bindings are labeled static in the database, while those bindings that have been added through the process of DHCP snooping are labeled dynamic. Static IPv6 address assignment is also available for DHCPv6. For configuration details, see "Configuring Static DHCP IP Addresses for DHCP snooping" on page 460.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1X53-D35	Starting in Junos OS Release 14.1X53-D35, a network device with a fixed IP allocation from the DHCP server is replaced by a new device with a different MAC address.
14.1X53-D10	Starting in Junos OS Release 14.1X53-D10, DHCP snooping is supported for IPv6 packets on EX 9200 Switches.

13.2X51-D20	Starting in Junos OS Release 17.1R1, you can configure DHCP snooping or DHCPv6 snooping on a VLAN without enabling other port security features by configuring the `dhcp-security` CLI statement at the `[edit vlans` *`vlan-name`* `forwarding-options dhcp-security]`.

RELATED DOCUMENTATION

Port Security Features | 2
Configuring Port Security (ELS) | 9
Understanding and Using Trusted DHCP Servers | 415
DHCP/BOOTP Relay for Switches Overview
Configuring Persistent Bindings in the DHCP or DHCPv6 (ELS) | 429
Understanding DHCP Option 82 | 501

Understanding DHCP Snooping (non-ELS)

IN THIS SECTION

NOTE: This topic includes information about enabling Dynamic Host Configuration Protocol (DHCP) snooping for Junos EX Series switches that do not support the Enhanced Layer 2 Software (ELS). If your switch runs a version of Junos that supports

ELS, see "Understanding DHCP Snooping (ELS)" on page 435. For ELS details, see Using the Enhanced Layer 2 Software CLI.

DHCP snooping enables the switching device, which can be either a switch or a router, to monitor DHCP messages received from untrusted devices connected to the switching device. When DHCP snooping is enabled on a VLAN, the system examines DHCP messages sent from untrusted hosts associated with the VLAN and extracts their IP addresses and lease information. This information is used to build and maintain the DHCP snooping database. Only hosts that can be verified using this database are allowed access to the network.

DHCP Snooping Basics

The Dynamic Host Configuration Protocol (DHCP) allocates IP addresses dynamically, *leasing* addresses to devices so that the addresses can be reused when no longer needed. Hosts and end devices that require IP addresses obtained through DHCP must communicate with a DHCP server across the LAN.

DHCP snooping acts as a guardian of network security by keeping track of valid IP addresses assigned to downstream network devices by a trusted DHCP server (the server is connected to a trusted network port).

By default, all trunk ports on the switch are trusted and all access ports are untrusted for DHCP snooping.

When DHCP snooping is enabled, the lease information from the switching device is used to create the DHCP snooping table, also known as the binding table. The table shows the IP-MAC binding, as well as the lease time for the IP address, type of binding, VLAN name, and interface for each host.

NOTE: DHCP snooping is disabled in the default configuration of the switching device. You must explicitly enable DHCP snooping by setting `examine-dhcp` at the `[edit ethernet-switching-options secure-access-port]` hierarchy level.

Entries in the DHCP snooping database are updated in these events:

- When a DHCP client releases an IP address (sends a DHCPRELEASE message). In this event, the associated mapping entry is deleted from the database.
- If you move a network device from one VLAN to another. In this event, typically the device needs to acquire a new IP address. Therefore, its entry in the database, including its VLAN ID, is updated.
- When the lease time (timeout value) assigned by the DHCP server expires. In this event, the associated entry is deleted from the database.

TIP: By default, the IP-MAC bindings are lost when the switching device is rebooted and DHCP clients (the network devices, or hosts) must reacquire bindings. However, you can configure the bindings to persist by setting the `dhcp-snooping-file` statement to store the database file either locally or remotely.

You can configure the switching device to snoop DHCP server responses from particular VLANs only. This prevents spoofing of DHCP server messages.

You configure DHCP snooping per VLAN, not per interface (port). DHCP snooping is disabled by default on switching devices.

DHCP Snooping Process

The basic process of DHCP snooping consists of the following steps:

NOTE: When DHCP snooping is enabled for a VLAN, all DHCP packets sent from the network devices in that VLAN are subjected to DHCP snooping. The final IP-MAC binding occurs when the DHCP server sends DHCPACK to the DHCP client.

1. The network device sends a DHCPDISCOVER packet to request an IP address.

2. The switching device forwards the packet to the DHCP server.

3. The server sends a DHCPOFFER packet to offer an address. If the DHCPOFFER packet is from a trusted interface, the switching device forwards the packet to the DHCP client.

4. The network device sends a DHCPREQUEST packet to accept the IP address. The switching device adds an IP-MAC placeholder binding to the database. The entry is considered a placeholder until a DHCPACK packet is received from the server. Until then, the IP address could still be assigned to some other host.

5. The server sends a DHCPACK packet to assign the IP address or a DHCPNAK packet to deny the address request.

6. The switching device updates the DHCP snooping database according to the type of packet received:

   - If the switching device receives a DHCPACK packet, it updates lease information for the IP-MAC bindings in its database.

   - If the switching device receives a DHCPNACK packet, it deletes the placeholder.

NOTE: The DHCP snooping database is updated only after the DHCPREQUEST packet has been sent.

For general information about the messages that the DHCP client and DHCP server exchange during the assignment of an IP address for the client, see the Junos OS Administration Library.

DHCPv6 Snooping

DHCPv6 snooping is the equivalent of DHCP snooping for IPv6. The process for DHCPv6 snooping is similar to that for DHCP snooping, but uses different names for the messages exchanged between the client and server to assign IPv6 addresses. Table 22 on page 448 shows DHCPv6 messages and their DHCP equivalents.

Table 22: DHCPv6 Messages and Equivalent DHCPv4 Messages

Sent by	DHCPv6 Messages	Equivalent DHCP Messages
Client	SOLICIT	DHCPDISCOVER
Server	ADVERTISE	DHCPOFFER
Client	REQUEST, RENEW, REBIND	DHCPREQUEST
Server	REPLY	DHCPACK/DHCPNAK
Client	RELEASE	DHCPRELEASE
Client	INFORMATION-REQUEST	DHCPINFORM
Client	DECLINE	DHCPDECLINE
Client	CONFIRM	none
Server	RECONFIGURE	DHCPFORCERENEW

Table 22: DHCPv6 Messages and Equivalent DHCPv4 Messages *(Continued)*

Sent by	DHCPv6 Messages	Equivalent DHCP Messages
Client	RELAY-FORW, RELAY-REPLY	none

Rapid Commit for DHCPv6

DHCPv6 provides for a Rapid Commit option (DHCPv6 option 14), which, when supported by the server and set by the client, shortens the exchange from a four-way relay to a two-message handshake. For more information about enabling the Rapid Commit option, see *Configuring DHCPv6 Rapid Commit (MX Series, EX Series)*.

In the rapid commit process:

1. The DHCPv6 client sends out a SOLICIT message that contains a request that rapid assignment of address, prefix, and other configuration parameters be preferred.

2. If the DHCPv6 server supports rapid assignment, it responds with a REPLY message, which contains the assigned IPv6 address and prefix and other configuration parameters.

DHCP Server Access

You can configure a switching device's access to the DHCP server in three ways:

Switching Device, DHCP Clients, and DHCP Server Are All on the Same VLAN

When the switching device, DHCP clients, and DHCP server are *all members of the same VLAN*, the DHCP server can be connected to the switching device in one of two ways:

- The server is directly connected to the same switching device as the one connected to the DHCP clients (the hosts, or network devices, that are requesting IP addresses from the server). The VLAN is enabled for DHCP snooping to protect the untrusted access ports. The trunk port is configured by default as a trusted port. See Figure 22 on page 450.

- The server is connected to an intermediary switching device (Switching Device 2). The DHCP clients are connected to Switching Device 1, which is connected through a trunk port to Switching Device 2. Switching Device 2 is being used as a transit device. The VLAN is enabled for DHCP snooping to protect the untrusted access ports. The trunk port is configured by default as a trusted port. As shown in Figure 23 on page 451, ge-0/0/11 is a trusted trunk port.

Figure 22: DHCP Server Connected Directly to a Switching Device

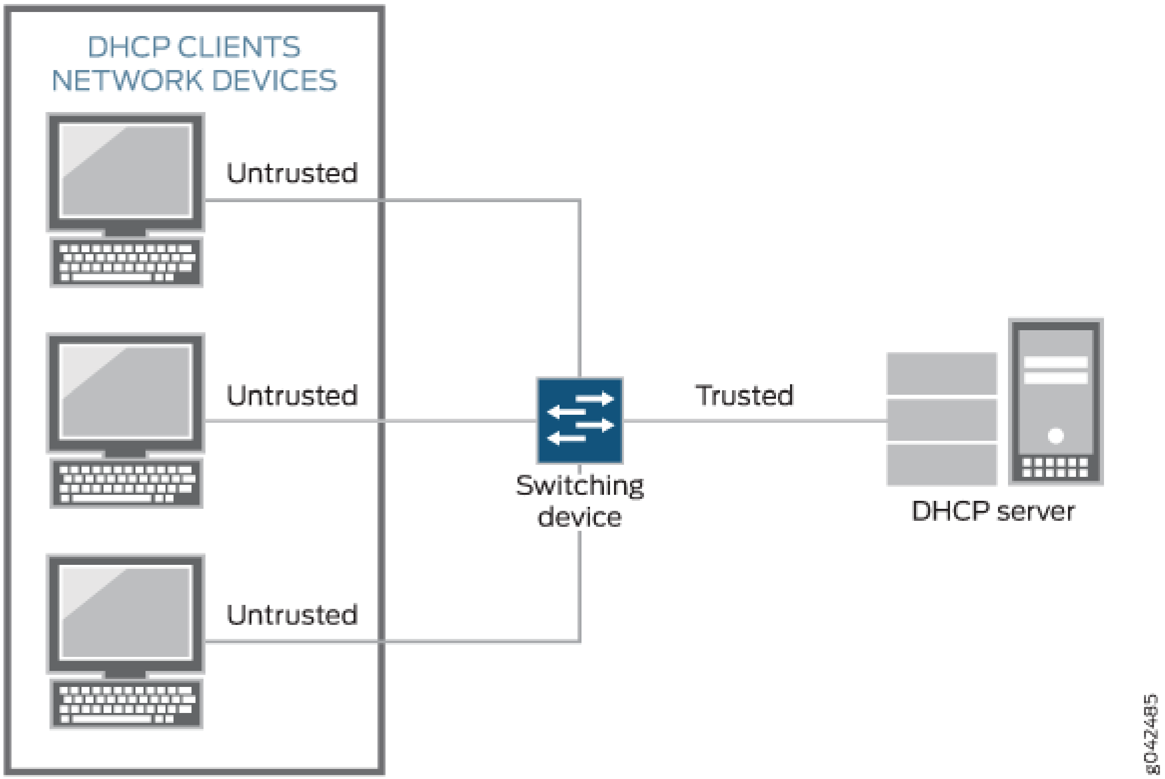
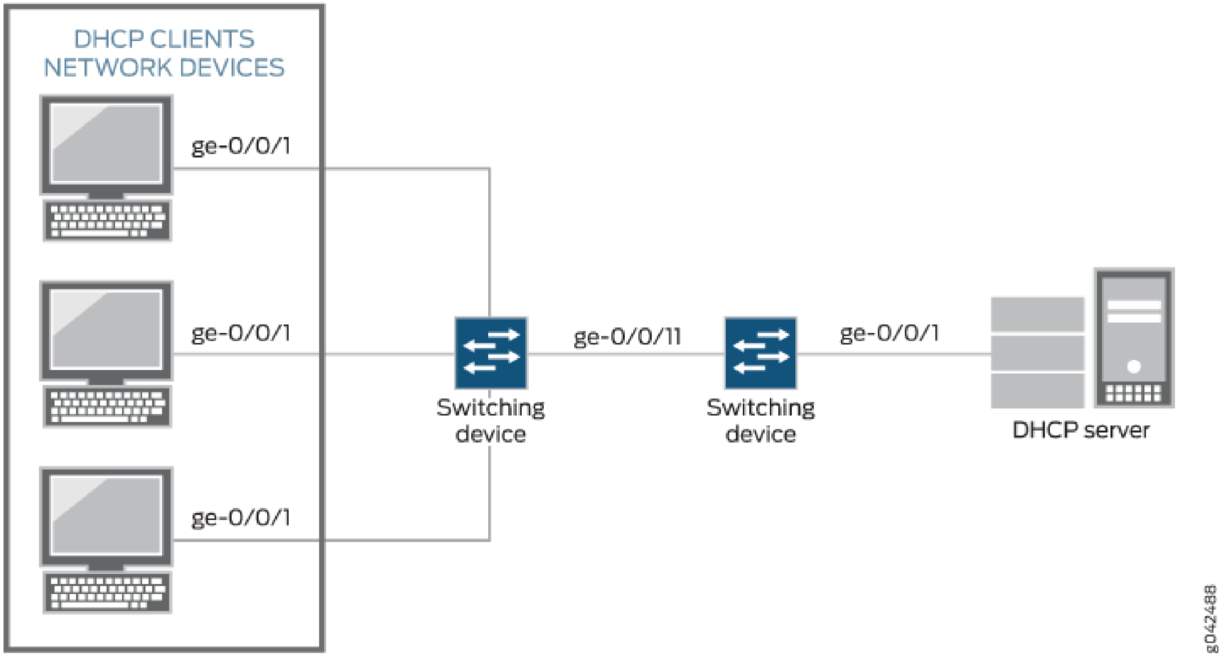


Figure 23: DHCP Server Connected Directly to Switching Device 2, with Switching Device 2 Connected to Switching Device 1 Through a Trusted Trunk Port

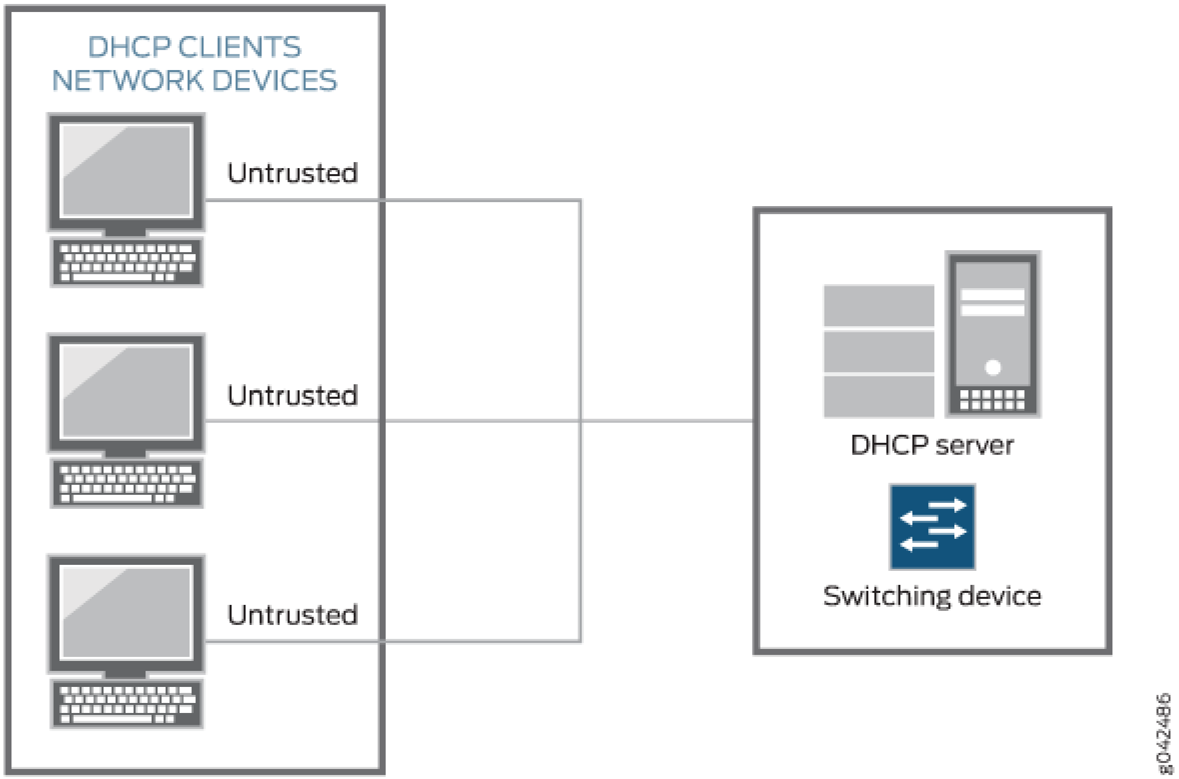


Switching Device Acts as DHCP Server

NOTE: The switching device acting as a DHCP server is not supported on the QFX Series.

The switching device itself is configured as a DHCP server; this is known as a *local configuration*. See Figure 24 on page 452.

Figure 24: Switching Device Is the DHCP Server



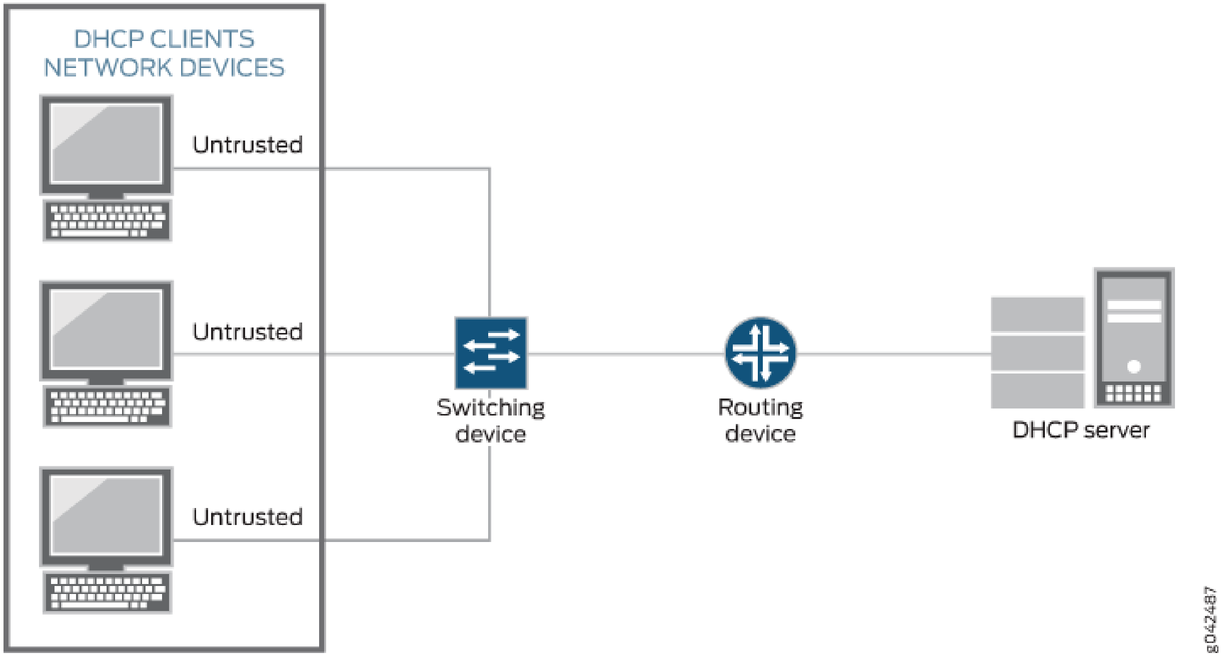
Switching Device Acts as Relay Agent

The switching device functions as a relay agent when the DHCP clients or the DHCP server is connected to the device through a Layer 3 interface. The Layer 3 interfaces on the switching device are configured as routed VLAN interfaces (RVIs), which are also known as integrated routing and bridging (IRB) interfaces. The trunk interfaces are trusted by default.

These two scenarios illustrate the switching device acting as a relay agent:

- The DHCP server and clients are in different VLANs.
- The switching device is connected to a router that is in turn connected to the DHCP server. See Figure 25 on page 453.

Figure 25: Switching Device Acting as Relay Agent Through Router to DHCP Server



Static IP Address Additions to the DHCP Snooping Database

You can add specific static IP addresses to the database as well as have the addresses dynamically assigned through DHCP snooping. To add static IP addresses, you supply the IP address, the MAC address of the device, the interface on which the device is connected, and the VLAN with which the interface is associated. No lease time is assigned to the entry. The statically configured entry never expires.

Snooping DHCP Packets That Have Invalid IP Addresses

If you enable DHCP snooping on a VLAN and then devices on that VLAN send DHCP packets that request invalid IP addresses, these invalid IP addresses are stored in the DHCP snooping database until they are deleted when their default timeout is reached. To eliminate this unnecessary consumption of space in the DHCP snooping database, the switching device drops the DCHP packets that request invalid IP addresses, preventing the snooping of these packets. The invalid IP addresses are:

- 0.0.0.0
- 128.0.*x.x*
- 191.255.*x.x*

- 192.0.0.*x*
- 223.255.255.*x*
- 224.*x*.*x*.*x*
- 240.*x*.*x*.*x* to 255.255.255.255

Prioritizing Snooped Packets

NOTE: Prioritizing snooped packets is not supported on the QFX Series and the EX4600 switch.

You can use class-of-service (CoS) forwarding classes and queues to prioritize DHCP snooped packets for a specified VLAN. This type of configuration places the DHCP snooped packets for that VLAN in a specified egress queue, so that the security procedure does not interfere with the transmission of high-priority traffic.

RELATED DOCUMENTATION

Port Security Features | 2

Understanding and Using Trusted DHCP Servers | 415

Configuring Static DHCP IP Addresses for DHCP snooping (MX routers) | 463

Understanding DHCP Snooping Trust-All Configuration

IN THIS SECTION

The DHCP Snooping Trust-All Configuration feature streamlines network security management by enabling you to mark all access ports within a VLAN as trusted with a single command. This functionality significantly reduces the administrative burden of configuring each port individually. Override configurations maintain precedence, ensuring specific security settings for individual ports are preserved. This feature integrates seamlessly with existing DHCP snooping mechanisms and supports comprehensive verification through new CLI commands, allowing administrators to efficiently manage and audit DHCP security configurations across VLANs, interfaces, routing instances, and logical systems.

Benefits of DHCP Snooping Trust-All Configuration

- Simplifies configuration by allowing network administrators to mark all access ports within a VLAN as trusted using a single command.
- Ensures consistency in VLAN security settings by uniformly applying trust status to all access ports, while still permitting specific ports to be individually configured as untrusted if necessary.
- Enhances manageability by integrating with existing DHCP snooping mechanisms and providing comprehensive verification tools through new CLI commands.
- Maintains existing security postures with the precedence of override configurations, ensuring that critical security settings for individual ports are not unintentionally altered.
- Supports high availability mechanisms like Graceful Routing Engine Switchover (GRES) without additional impact, ensuring seamless and reliable network performance.

Overview

The DHCP Snooping Trust-All Configuration feature introduces a significant enhancement to the DHCP snooping mechanism by allowing you to mark all access ports within a VLAN as trusted with a single command. This streamlined approach is facilitated through the `set vlans <vlan> forwarding-options dhcp-security trust-all` command. By applying this command, you eliminate the need for repetitive configurations on each individual port, thus simplifying the management of network security settings and significantly reducing the administrative workload. This configuration option is particularly useful in environments with numerous access ports, where consistency and efficiency are paramount.

The trust-all configuration interacts seamlessly with existing override configurations, ensuring that any specific trusted or untrusted settings previously applied to individual ports take precedence. This hierarchy of configurations guarantees that critical security postures are preserved even when the trust-all command is executed. If a port is explicitly marked as untrusted, it will remain untrusted despite the overarching trust-all setting applied to the VLAN, thus maintaining the integrity of your network's security policies.

To aid in the verification and auditing of these security configurations, new CLI commands have been introduced. Commands such as `show dhcp-security vlans` and its detailed variants enable you to view the DHCP security settings at various levels, including VLAN, interface, routing instance, and logical system. These commands provide comprehensive insights into the current state of your network's security configurations, ensuring that the trust-all settings are correctly applied and that any overrides are accurately reflected. This capability enhances your ability to manage and troubleshoot DHCP security settings effectively.

Implementation and Verification

To implement the DHCP Snooping Trust-All Configuration, access the VLAN configuration mode using the command hierarchy `[edit vlans vlan-name forwarding-options dhcp-security]`. Once in this mode, apply the trust-all setting with the command `set vlans <vlan> forwarding-options dhcp-security trust-all`. This command marks all access ports within the specified VLAN as trusted. To ensure the configurations are correctly applied and to validate your settings, use the `show` commands provided.

For example, the `show dhcp-security vlans` command displays a summary of all VLANs' DHCP security configurations, while `show dhcp-security vlans <vlan-name> detail` provides detailed information for a specific VLAN. These commands help verify that the trust-all configuration is active and functioning as intended. Additionally, commands like `show dhcp-security vlans interface <intf-name>` allow you to drill down into the settings for individual interfaces, ensuring that any override configurations are correctly implemented and that the network's security posture is maintained.

Enabling DHCP Snooping (non-ELS)

IN THIS SECTION

DHCP snooping enables the switch to monitor and control DHCP messages received from untrusted devices connected to the switch. The switch builds and maintains a database of valid bindings between IP address and MAC addresses (IP-MAC bindings) called the DHCP snooping database.

NOTE: If you configure DHCP snooping for all VLANs and you enable a different port security feature on a specific VLAN, you must also explicitly enable DHCP snooping on that VLAN. Otherwise, the default value of no DHCP snooping applies to that VLAN.

Enabling DHCP Snooping

You configure DHCP snooping per VLAN, not per interface (port). By default, DHCP snooping is disabled for all VLANs. You can enable DHCP snooping on all VLANs or on specific VLANs.

To enable DHCP snooping:

- On a specific VLAN:

```
[edit ethernet-switching-options secure-access port]
user@switch# set vlan vlan-name examine-dhcp
```

- On all VLANs:

```
[edit ethernet-switching-options secure-access port]
user@switch# set vlan all examine-dhcp
```

To enable DHCPv6 snooping:

- On a specific VLAN:

```
[edit ethernet-switching-options secure-access port]
user@switch# set vlan vlan-name examine-dhcpv6
```

- On all VLANs:

```
[edit ethernet-switching-options secure-access port]
user@switch# set vlan all examine-dhcpv6
```

TIP: By default, the IP-MAC bindings are lost when the switch is rebooted and DHCP clients (the network devices, or hosts) must reacquire bindings. However, you can configure the bindings to persist by setting the switch to store the database file either locally or remotely. See "Configuring Persistent Bindings in the DHCP or DHCPv6 (non-ELS)" on page 431.

TIP: For private VLANs (PVLANs), enable DHCP snooping on the primary VLAN. If you enable DHCP snooping only on a community VLAN, DHCP messages coming from PVLAN trunk ports are not snooped.

Applying CoS Forwarding Classes to Prioritize Snooped Packets

On EX Series switches you might need to use class of service (CoS) to protect packets from critical applications from being dropped during periods of network congestion and delay, and might also need to configure the port security features of DHCP snooping on the ports through which those packets enter or leave.

NOTE: Prioritizing snooped packets by using CoS forwarding classes is not supported on the QFX Series switch.

To apply CoS forwarding classes and queues to snooped packets:

1. Create a user-defined forwarding class to be used for prioritizing snooped packets:

   ```
   [edit class-of-service]
   user@switch# set forwarding-classes class class-name queue-num queue-number
   ```

2. Enable DHCP snooping on a specific VLAN or on all VLANs and apply the required forwarding class on the snooped packets:

   - On a specific VLAN:

     ```
     [edit ethernet-switching-options secure-access port]
     user@switch# set vlan vlan-name examine-dhcp forwarding-class class-name
     ```

   - On all VLANs:

     ```
     [edit ethernet-switching-options secure-access port]
     user@switch# set vlan all examine-dhcp forwarding-class class-name
     ```

   NOTE: Replace `examine-dhcp` with `examine-dhcpv6` to enable DHCPv6 snooping.

Verifying That DHCP Snooping Is Working Correctly

IN THIS SECTION

Purpose

Verify that DHCP snooping is working on the switch and that the DHCP snooping database is correctly populated with both dynamic and static bindings.

Action

Send some DHCP requests from network devices (here they are DHCP clients) connected to the switch.

Display the DHCP snooping information when the interface on which the DHCP server connects to the switch is trusted. The following output results when requests are sent from the MAC addresses and the server has provided the IP addresses and leases:

```
user@switch> show dhcp snooping binding
DHCP Snooping Information:
MAC address         IP address   Lease (seconds) Type     VLAN      Interface


00:05:85:3A:82:77  192.0.2.17   600             dynamic  employee  ge-0/0/1.0

00:05:85:3A:82:79  192.0.2.18   653             dynamic  employee  ge-0/0/1.0

00:05:85:3A:82:80  192.0.2.19   720             dynamic  employee  ge-0/0/2.0

00:05:85:3A:82:81  192.0.2.20   932             dynamic  employee  ge-0/0/2.0

00:05:85:3A:82:83  192.0.2.21   1230            dynamic  employee  ge-0/0/2.0

00:05:85:27:32:88  192.0.2.22   -               static   data      ge-0/0/4.0
```

Meaning

When the interface on which the DHCP server connects to the switch has been set to trusted, the output (see preceding sample) shows, for each MAC address, the assigned IP address and lease time—that is, the time, in seconds, remaining before the lease expires. Static IP addresses have no assigned lease time. The statically configured entry never expires.

If the DHCP server had been configured as untrusted, no entries would be added to the DHCP snooping database and nothing would be shown in the output of the `show dhcp snooping binding` command.

RELATED DOCUMENTATION

Example: Protecting Against Address Spoofing and Layer 2 DoS Attacks | 464
Example: Protecting Against ARP Spoofing Attacks | 482
Example: Prioritizing Snooped and Inspected Packet | 489
Monitoring Port Security
Understanding DHCP Snooping (non-ELS) | 445

Configuring Static DHCP IP Addresses

IN THIS SECTION

Configuring Static DHCP IP Addresses for DHCP snooping (ELS)

NOTE: This task uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Configuring Static DHCP IP Addresses for DHCP snooping (non-ELS)" on page 462. For ELS details, see Using the Enhanced Layer 2 Software CLI.

You can add static (fixed) IP addresses and bind them to fixed MAC addresses in the DHCP snooping database. These bindings are labeled *static* in the database, while those bindings that have been added through the process of DHCP snooping are labeled *dynamic*. Static IPv6 address assignment is also available for DHCPv6. This feature is supported on aggregated Ethernet interfaces.

Before you can perform this procedure, you must configure the VLAN. See Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure).

To configure a static IP address to MAC address (IP-MAC) binding in the DHCP snooping database, you must first create a group of access interfaces under the `[edit vlans vlan-name forwarding-options dhcp-security]` hierarchy. Creating this group automatically enables DHCP snooping, which is a prerequisite for creating the DHCP snooping database. On switches that support DHCPv6, creating the group of interfaces will automatically enable both DHCP and DHCPv6 snooping. Then you can configure a specific interface within the group to have one or more static IP-MAC address bindings.

To configure a static IP-MAC address binding in the DHCP snooping database:

-
  ```
  [edit vlans vlan-name forwarding-options dhcp-security]
  user@switch# set group group-name interface interface-name static-ip ip-address mac mac-address
  ```

To configure a static IPv6-MAC address binding in the DHCPv6 snooping database:

-
  ```
  [edit vlans vlan-name forwarding-options dhcp-security]
  user@switch# set group group-name interface interface-name static-ipv6 ip-address mac mac-address
  ```

In the following example, a device with static IP allocation is connected to the ge-0/0/1 interface, which belongs to vlan-A. To configure this device to connect to the external network:

```
[edit]
user@switch# set vlans vlan-A forwarding-options dhcp-security group static-group interface
ge-0/0/1 static-ip 10.1.1.6 mac 00:00:00:44:44:06
```

To verify that the configuration is configured on the device:

```
user@switch> show configuration vlans vlan-A
vlan-id 100;
forwarding-options {
    dhcp-security {
        ip-source-guard;
```

```
        group static-group {
            interface ge-0/0/1 {
                static-ip 10.1.1.6 mac 00:00:00:44:44:06
            }
        }
    }
}
```

To verify that a binding entry is created for the static client:

```
user@switch> show dhcp-security binding
    IP address        MAC address         Vlan     Expires State   Interface
    10.1.1.6          00:00:00:44:44:06   vlan-A   0       STATIC  ge-0/0/1
```

SEE ALSO

show dhcp-security binding

Enabling DHCP Snooping (non-ELS) | **456**

Understanding DHCP Snooping (non-ELS) | **445**

Configuring Static DHCP IP Addresses for DHCP snooping (non-ELS)

You can add static (fixed) IP addresses and bind them to fixed MAC addresses in the DHCP snooping database. These bindings are labeled *static* in the database, while those bindings that have been added through the process of DHCP snooping are labeled *dynamic*.

NOTE: This task uses Junos OS for EX Series switches that do not support Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does support ELS, see "Configuring Static DHCP IP Addresses for DHCP snooping (ELS)" on page 460. For ELS details, see Using the Enhanced Layer 2 Software CLI.

To configure a static IP-MAC address binding in the DHCP snooping database:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface interface-name  static-ip ip-address  vlan data-vlan mac mac-address
```

To configure a static IP-MAC address binding in the DHCPv6 snooping database:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface interface-name static-ipv6 ip-address vlan data-vlan mac mac-address
```

To view results of the configuration steps before committing the configuration, type the `show` command at the user prompt.

To commit these changes to the active configuration, type the `commit` command at the user prompt.

SEE ALSO

Enabling DHCP Snooping (non-ELS) | **456**

Understanding DHCP Snooping (non-ELS) | **445**

secure-access-port

secure-access-port

Configuring Static DHCP IP Addresses for DHCP snooping (MX routers)

You can add static (fixed) IP addresses and bind them to fixed MAC addresses in the DHCP snooping database. These bindings are labeled as *static* in the database, while those bindings that have been added through the process of DHCP snooping are labeled *dynamic*.

To configure a static IP address/MAC address binding in the DHCP snooping database, you must first create a group of access interfaces under `[edit bridge-domains` *`bridge-domain-name`* `forwarding-options dhcp-security]`. Creating this group automatically enables DHCP snooping, which is a prerequisite for creating the DHCP snooping database. The following procedure shows the configuration in two steps, but it can be done in one. You can then configure a specific interface within the group to have one or more static IP-MAC address bindings.

To configure a static IP address and MAC address binding in the DHCP snooping database:

1. Create a group by including an access interface:

   ```
   [edit bridge-domains bd-name forwarding-options dhcp-security]
   user@device# set group group-name interface interface-name
   ```

2. Configure a static IP address:

```
[edit bridge-domains bd-name forwarding-options dhcp-security]
user@device# set group group-name interface interface-name static-ip ip-address mac mac-address
```

SEE ALSO

show dhcp-security binding

Understanding DHCP Snooping (non-ELS) | 445

Example: Protecting Against Address Spoofing and Layer 2 DoS Attacks

IN THIS SECTION

You can configure DHCP snooping, dynamic ARP inspection (DAI), and MAC limiting on the access interfaces of a switch to protect the switch and the Ethernet LAN against address spoofing and Layer 2 denial-of-service (DoS) attacks. To obtain the basic settings for these features, you can use the switch's default configuration for port security, configure the MAC limit, and enable DHCP snooping and DAI on a VLAN. You can configure these features when the DHCP server is connected to a switch that is different from the one to which the DHCP clients (network devices) are connected.

This example describes how to configure port security features on a switch whose hosts obtain IP addresses and lease times from a DHCP server connected to a second switch:

Requirements

This example uses the following hardware and software components:

- One EX Series switch or QFX3500 switch—*Switch 1* in this example.
- An additional EX Series switch or QFX3500 switch—*Switch 2* in this example. You do not configure port security on this second switch.
- Junos OS Release 9.0 or later for EX Series switches or Junos OS Release 12.1 or later for the QFX Series.
- A DHCP server connected to Switch 2. You use the server to provide IP addresses to network devices connected to Switch 1.
- At least two network devices (hosts) that you connect to access interfaces on Switch 1. These devices are DHCP clients.

Before you configure DHCP snooping, DAI, and MAC limiting port security features, be sure you have:

- Connected the DHCP server to Switch 2.
- Configured a VLAN on Switch 1. See the task for your platform:
  - Example: Setting Up Bridging with Multiple VLANs for EX Series Switches
  - Example: Setting Up Bridging with Multiple VLANs for the QFX Series

Overview and Topology

IN THIS SECTION

Ethernet LANs are vulnerable to address spoofing and DoS attacks on network devices. To protect the devices from such attacks, you can configure:

- DHCP snooping to validate DHCP server messages
- DAI to protect against ARP spoofing
- MAC limiting to constrain the number of MAC addresses the switch adds to its MAC address cache

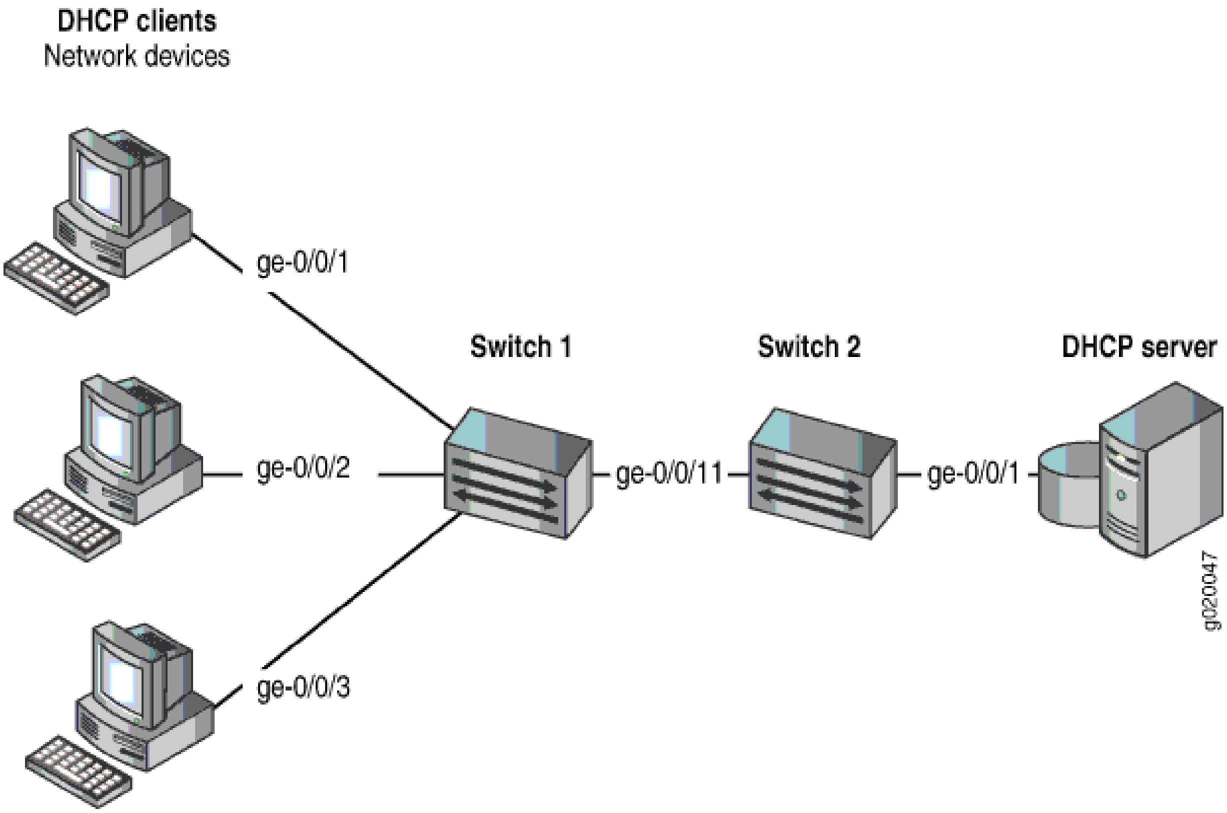
This example shows how to configure these port security features on Switch 1. Switch 1 is connected to another switch (Switch 2), which is not configured with port security features. Switch 2 is connected to a DHCP server (see Figure 26 on page 466.) Network devices (hosts) that are connected to Switch 1 send requests for IP addresses (these network devices are DHCP clients). Those requests are transmitted from Switch 1 to Switch 2 and then to the DHCP server connected to Switch 2. Responses to the requests are transmitted along the reverse path of the one followed by the requests.

The setup for this example includes the VLAN `employee-vlan` on both switches.

Figure 26 on page 466 shows the network topology for the example.

Topology

Figure 26: Network Topology for Port Security Setup with Two Switches on the Same VLAN



The components of the topology for this example are shown in Table 23 on page 466.

Table 23: Components of Port Security Setup on Switch 1 with a DHCP Server Connected to Switch 2

Properties	Settings
Switch hardware	One EX Series switch or one QFX3500 switch (Switch 1), and an additional EX Series switch or QFX3500 switch (Switch 2)

Table 23: Components of Port Security Setup on Switch 1 with a DHCP Server Connected to Switch 2 *(Continued)*

Properties	Settings
VLAN name and ID	`employee-vlan`, tag `20`

VLAN subnets	`192.0.2.16/28`
	`192.0.2.17` through `192.0.2.30`
	`192.0.2.31` is subnet's broadcast address

Trunk interface on both switches	ge-0/0/11
Access interfaces on Switch 1	ge-0/0/1, ge-0/0/2, and ge-0/0/3
Access interface on Switch 2	ge-0/0/1
Interface for DHCP server	ge-0/0/1 on Switch 2

Switch 1 is initially configured with the default port security setup. In the default configuration on the switch:

- Secure port access is activated on the switch.
- The switch does not drop any packets, which is the default setting.
- DHCP snooping and DAI are disabled on all VLANs.
- All access interfaces are untrusted and trunk interfaces are trusted; these are the default settings.

In the configuration tasks for this example, you configure a VLAN on both switches.

In addition to configuring the VLAN, you enable DHCP snooping on Switch 1. In this example, you also enable DAI and a MAC limit of `5` on Switch 1.

Because the interface that connects Switch 2 to Switch 1 is a trunk interface, you do not need to configure this interface to be trusted. As noted above, trunk interfaces are automatically trusted, so DHCP messages coming from the DHCP server to Switch 2 and then on to Switch 1 are trusted.

Configuring a VLAN, Interfaces, and Port Security Features on Switch 1

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure a VLAN, interfaces, and port security features, copy the following commands and paste them into the switch terminal window:

```
[edit]
set vlans employee-vlan vlan-id 20
set interfaces ge-0/0/11 unit 0 family ethernet-switching port-mode trunk
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members 20
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members 20
set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members 20
set interfaces ge-0/0/11 unit 0 family ethernet-switching vlan members 20
set ethernet-switching-options secure-access-port interface ge-0/0/1 mac-limit 5 action drop
set ethernet-switching-options secure-access-port vlan employee-vlan arp-inspection
set ethernet-switching-options secure-access-port vlan employee-vlan examine-dhcp
clear ethernet-switching table interface ge-0/0/1
```

Step-by-Step Procedure

To configure MAC limiting, a VLAN, and interfaces on Switch 1 and enable DAI and DHCP on the VLAN:

1. Configure the VLAN `employee-vlan` with VLAN ID `20`:

   ```
   [edit vlans]
   user@switch1# set employee-vlan vlan-id 20
   ```

2. Configure an interface on Switch 1 as a trunk interface:

```
[edit interfaces]
user@switch1# set ge-0/0/11 unit 0 family ethernet-switching port-mode
trunk
```

3. Associate the VLAN with interfaces ge-0/0/1, ge-0/0/2, ge-0/0/3, and ge-0/0/11:

```
[edit interfaces]
user@switch1# set ge-0/0/1 unit 0 family ethernet-switching vlan members 20
user@switch1# set ge-0/0/2 unit 0 family ethernet-switching vlan members 20
user@switch1# set ge-0/0/3 unit 0 family ethernet-switching vlan members 20
user@switch1# set ge-0/0/11 unit 0 family ethernet-switching vlan members 20
```

4. Enable DHCP snooping on the VLAN:

```
[edit ethernet-switching-options secure-access-port]
user@switch1# set vlan employee-vlan examine-
dhcp
```

5. Enable DAI on the VLAN:

```
[edit ethernet-switching-options secure-access-port]
user@switch1# set vlan employee-vlan arp-inspection
```

6. Configure a MAC limit of `5` on ge-0/0/1 and use the default action, `drop` (packets with new addresses are dropped if the limit is exceeded):

```
[edit ethernet-switching-options secure-access-port]
user@switch1# set interface ge-0/0/1 mac-limit 5 drop
```

7. Clear the existing MAC address table entries from interface ge-0/0/1:

```
user@switch1# clear ethernet-switching table interface ge-0/0/1
```

Results

Display the results of the configuration:

```
[edit]
user@switch1# show
ethernet-switching-options {
    secure-access-port {
        interface ge-0/0/1.0{
            mac-limit 5 action drop;
        }
        vlan employee-vlan {
            arp-inspection;
            examine-dhcp;
        }
    }
}
interfaces {
    ge-0/0/1 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members 20;
                }
            }
        }
    }
    ge-0/0/2 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members 20;
                }
            }
        }
    }
    ge-0/0/3 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    port-mode trunk;
                    members 20;
                }
            }
        }
    }
    ge-0/0/11 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members 20;
                }
            }
        }
    }
}
vlans {
    employee-vlan {
        vlan-id 20;
    }
}
```

Configuring a VLAN and Interfaces on Switch 2

IN THIS SECTION

To configure the VLAN and interfaces on Switch 2:

Procedure

CLI Quick Configuration

To quickly configure the VLAN and interfaces on Switch 2, copy the following commands and paste them into the switch terminal window:

```
[edit]
set vlans employee-vlan vlan-id 20
set interfaces ge-0/0/11 unit 0 family ethernet-switching port-mode trunk
set interfaces ge-0/0/11 unit 0 family ethernet-switching vlan members 20
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members 20
```

Step-by-Step Procedure

To configure the VLAN and interfaces on Switch 2:

1. Configure the VLAN `employee-vlan` with VLAN ID `20`:

```
[edit vlans]
user@switch1# set employee-vlan vlan-id 20
```

2. Configure an interface on Switch 2 as a trunk interface:

```
[edit interfaces]
user@switch2# set ge-0/0/11 unit 0 ethernet-switching port-mode trunk
```

3. Associate the VLAN with interfaces ge-0/0/1 and ge-0/0/11:

```
[edit interfaces]
user@switch2# set ge-0/0/1 unit 0 family ethernet-switching vlan members 20
user@switch2# set ge-0/0/11 unit 0 family ethernet-switching vlan members
20
```

Results

Display the results of the configuration:

```
[edit]
user@switch2# show
interfaces {
    ge-0/0/1 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members 20;
                }
            }
        }
    }
    ge-0/0/11 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members 20;
                }
            }
        }
    }
}
vlans {
    employee-vlan {
        vlan-id 20;
    }
}
```

Verification

IN THIS SECTION

To confirm that the configuration is working properly.

Verifying That DHCP Snooping Is Working Correctly on Switch 1

Purpose

Verify that DHCP snooping is working on Switch 1.

Action

Send some DHCP requests from network devices (here they are DHCP clients) connected to the switch.

issue the operational mode command **show dhcp snooping binding** to display the DHCP snooping information when the interface through which Switch 2 sends the DHCP server replies to clients connected to Switch 1 is trusted. The server has provided the IP addresses and leases:

```
user@switch1> show dhcp snooping binding
DHCP Snooping Information:
MAC Address         IP Address    Lease    Type     VLAN          Interface

-----------------   ----------    -----    ----     ----          ---------

00:05:85:3A:82:77    192.0.2.17    600    dynamic  employee-vlan   ge-0/0/1.0

00:05:85:3A:82:79    192.0.2.18    653    dynamic  employee-vlan   ge-0/0/1.0

00:05:85:3A:82:80    192.0.2.19    720    dynamic  employee-vlan   ge-0/0/1.0

00:05:85:3A:82:81    192.0.2.20    932    dynamic  employee-vlan   ge-0/0/1.0

00:05:85:3A:82:83    192.0.2.21   1230    dynamic  employee-vlan   ge-0/0/1.0

00:05:85:3A:82:90    192.0.2.20    932    dynamic  employee-vlan   ge-0/0/2.0

00:05:85:3A:82:91    192.0.2.21   1230    dynamic  employee-vlan   ge-0/0/3.0
```

Meaning

The output shows, for each MAC address, the assigned IP address and lease time—that is, the time, in seconds, remaining before the lease expires.

Verifying That DAI Is Working Correctly on Switch 1

Purpose

Verify that DAI is working on Switch 1.

Action

Send some ARP requests from network devices connected to the switch.

Issue the operational mode command **show arp inspection statistics** to display the DAI information:

```
user@switch1> show arp inspection statistics
ARP inspection statistics:
Interface    Packets received  ARP inspection pass  ARP inspection failed
----------   ----------------  -------------------  ---------------------
ge-0/0/1.0                  7                    5                      2
ge-0/0/2.0                 10                   10                      0
ge-0/0/3.0                 18                   15                      3
```

Meaning

The output shows the number of ARP packets received and inspected per interface, with a listing of how many packets passed and how many failed the inspection on each interface. The switch compares the ARP requests and replies against the entries in the DHCP snooping database. If a MAC address or IP address in the ARP packet does not match a valid entry in the database, the packet is dropped.

Verifying That MAC Limiting Is Working Correctly on Switch 1

Purpose

Verify that MAC limiting is working on Switch 1.

Action

Issue the operational mode command **show ethernet-switching table** to display the MAC addresses that are learned when DHCP requests are sent from hosts on ge-0/0/1:

```
user@switch1> show ethernet-switching table
```

```
Ethernet-switching table:  6 entries, 5 learned

  VLAN                MAC address        Type         Age    Interfaces

  employee-vlan       00:05:85:3A:82:77  Learn          0    ge-0/0/1.0
```

```
employee-vlan       00:05:85:3A:82:79   Learn          0    ge-0/0/1.0
employee-vlan       00:05:85:3A:82:80   Learn          0    ge-0/0/1.0
employee-vlan       00:05:85:3A:82:81   Learn          0    ge-0/0/1.0
employee-vlan       00:05:85:3A:82:83   Learn          0    ge-0/0/1.0
employee-vlan       *                   Flood          -    ge-0/0/1.0
```

Meaning

The output shows that five MAC addresses have been learned for interface `ge-0/0/1`, which corresponds to the MAC limit of `5` set in the configuration. The last line of the output shows that a sixth MAC address request was dropped, as indicated by the asterisk (*) in the `MAC address` column.

RELATED DOCUMENTATION

Example: Configuring Port Security (non-ELS) | **15**

Configuring Port Security (non-ELS) | **11**

Configuring Port Security (J-Web Procedure)

secure-access-port

secure-access-port

show arp inspection statistics

show dhcp snooping binding

show ethernet-switching table

Example: Protecting Against DHCP Snooping Database Attacks

IN THIS SECTION

In one type of attack on the DHCP snooping database, an intruder introduces a DHCP client on an untrusted access interface with a MAC address identical to that of a client on another untrusted interface. The intruder then acquires the DHCP lease of that other client, thus changing the entries in the DHCP snooping table. Subsequently, what would have been valid ARP requests from the legitimate client are blocked.

This example describes how to configure allowed MAC addresses, a port security feature, to protect the switch from DHCP snooping database alteration attacks:

Requirements

This example uses the following hardware and software components:

- One EX Series switch or one QFX3500 switch
- Junos OS Release 11.4 or later for EX Series switches or Junos OS Release 12.1 or later for the QFX Series
- A DHCP server to provide IP addresses to network devices on the switch

Before you configure specific port security features to mitigate common access-inteface attacks, be sure you have:

- Connected the DHCP server to the switch.
- Configured a VLAN on the switch. See the task for your platform:
  - Example: Setting Up Bridging with Multiple VLANs for EX Series Switches
  - Example: Setting Up Bridging with Multiple VLANs for the QFX Series

Overview and Topology

IN THIS SECTION

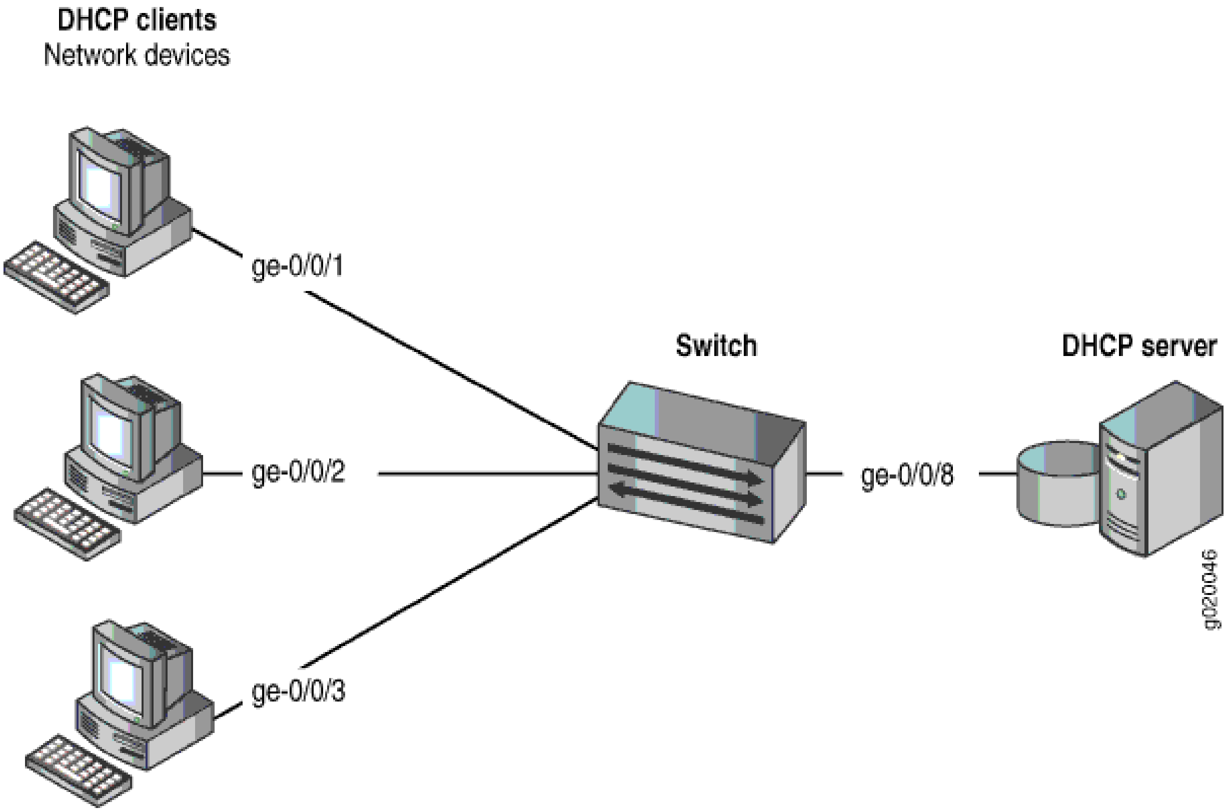
Ethernet LANs are vulnerable to address spoofing and DoS attacks on network devices. This example describes how to protect the switch from an attack on the DHCP snooping database that alters the MAC addresses assigned to some clients.

This example shows how to configure port security features on a switch that is connected to a DHCP server.

The setup for this example includes the VLAN **employee-vlan** on the switch. Figure 27 on page 478 illustrates the topology for this example.

Topology

Figure 27: Network Topology for Basic Port Security



The components of the topology for this example are shown in Table 24 on page 478.

Table 24: Components of the Port Security Topology

Properties	Settings
Switch hardware	One EX3200-24P, 24 ports (8 PoE ports) or one QFX3500 switch
VLAN name and ID	**employee-vlan**, tag **20**

Table 24: Components of the Port Security Topology *(Continued)*

Properties	Settings
VLAN subnets	**192.0.2.16/28** **192.0.2.17** through **192.0.2.30** **192.0.2.31** is the subnet's broadcast address
Interfaces in **employee-vlan**	**ge-0/0/1**, **ge-0/0/2**, **ge-0/0/3**, **ge-0/0/8**
Interface for DHCP server	**ge-0/0/8**

In this example, the switch has already been configured as follows:

- Secure port access is activated on the switch.
- DHCP snooping is enabled on the VLAN **employee-vlan**.
- All access ports are untrusted, which is the default setting.

Configuration

IN THIS SECTION

To configure allowed MAC addresses to protect the switch against DHCP snooping database alteration attacks:

Procedure

CLI Quick Configuration

To quickly configure some allowed MAC addresses on an interface, copy the following commands and paste them into the switch terminal window:

```
[edit ethernet-switching-options secure-access-port]

set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:80
set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:81
set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:83
set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:85
set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:88
```

Step-by-Step Procedure

To configure some allowed MAC addresses on an interface:

Configure the five allowed MAC addresses on an interface:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:80
user@switch# set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:81
user@switch# set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:83
user@switch# set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:85
user@switch# set interface ge-0/0/2 allowed-mac 00:05:85:3A:82:88
```

Results

Check the results of the configuration:

```
[edit ethernet-switching-options secure-access-port]
user@switch# show
interface ge-0/0/2.0 {
    allowed-mac [ 00:05:85:3a:82:80 00:05:85:3a:82:81 00:05:85:3a:82:83 00:05:85
    :3a:82:85 00:05:85:3a:82:88 ];
}
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying That Allowed MAC Addresses Are Working Correctly on the Switch

Purpose

Verify that allowed MAC addresses are working on the switch.

Action

Display the MAC cache information:

```
user@switch> show ethernet-switching table
Ethernet-switching table:  6 entries, 5 learned

  VLAN                 MAC address         Type         Age    Interfaces

  employee-vlan        00:05:85:3A:82:80   Learn          0    ge-0/0/2.0
  employee-vlan        00:05:85:3A:82:81   Learn          0    ge-0/0/2.0
  employee-vlan        00:05:85:3A:82:83   Learn          0    ge-0/0/2.0
  employee-vlan        00:05:85:3A:82:85   Learn          0    ge-0/0/2.0
  employee-vlan        00:05:85:3A:82:88   Learn          0    ge-0/0/2.0
  employee-vlan        *                   Flood          -    ge-0/0/2.0
```

Meaning

The output shows that the five MAC addresses configured as allowed MAC addresses have been learned and are displayed in the MAC cache. The last MAC address in the list, one that had not been configured as allowed, has not been added to the list of learned addresses.

RELATED DOCUMENTATION

Example: Configuring Port Security (non-ELS) | 15

Configuring MAC Limiting (non-ELS) | 374

Example: Protecting Against ARP Spoofing Attacks

IN THIS SECTION

In an ARP spoofing attack, the attacker associates its own MAC address with the IP address of a network device connected to the switch. Traffic intended for that IP address is now sent to the attacker instead of being sent to the intended destination. The attacker can send faked, or “spoofed,” ARP messages on the LAN.

> **NOTE**: When dynamic ARP inspection (DAI) is enabled, the switch logs the number of invalid ARP packets that it receives on each interface, along with the sender’s IP and MAC addresses. You can use these log messages to discover ARP spoofing on the network. ARP probe packets are not subjected to dynamic ARP inspection. The switch always forwards such packets.

This example describes how to configure DHCP snooping and dynamic ARP inspection (DAI), two port security features, to protect the switch against ARP spoofing attacks:

Requirements

This example uses the following hardware and software components:

- One EX Series switch or one QFX3500 switch
- Junos OS Release 11.4 or later for EX Series switches or Junos OS Release 12.1 or later for the QFX Series
- A DHCP server to provide IP addresses to network devices on the switch

Before you configure DHCP snooping and DAI (two port security features) to mitigate ARP spoofing attacks, be sure you have:

- Connected the DHCP server to the switch.
- Configured a VLAN on the switch. See the task for your platform:

- Example: Setting Up Bridging with Multiple VLANs for EX Series Switches
- Example: Setting Up Bridging with Multiple VLANs on Switches for QFX Series Switches

Overview and Topology

IN THIS SECTION

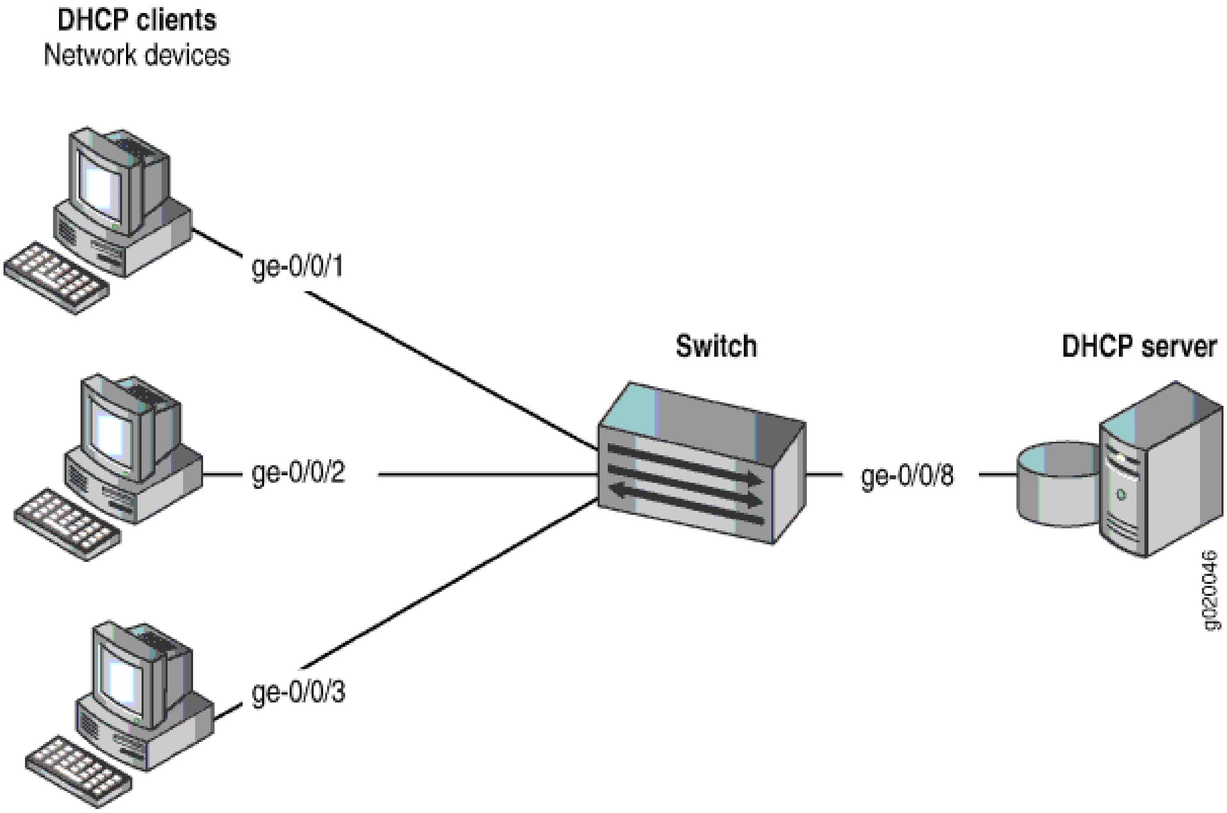
Ethernet LANs are vulnerable to address spoofing and DoS attacks on network devices. This example describes how to protect the switch against one common type of attack, an ARP spoofing attack.

In an ARP spoofing attack, the attacker sends faked ARP messages, thus creating various types of problems on the LAN—for example, the attacker might launch a man-in-the middle attack.

This example shows how to configure port security features on a switch that is connected to a DHCP server. The setup for this example includes the VLAN **employee-vlan** on the switch. The procedure for creating that VLAN is described in the topic Example: Setting Up Bridging with Multiple VLANs for EX Series Switches and Example: Setting Up Bridging with Multiple VLANs on Switches for the QFX Series. That procedure is not repeated here. Figure 28 on page 484 illustrates the topology for this example.

Topology

Figure 28: Network Topology for Basic Port Security



The components of the topology for this example are shown in Table 25 on page 484.

Table 25: Components of the Port Security Topology

Properties	Settings
Switch hardware	One EX3200-24P, 24 ports (8 PoE ports) or one QFX3500 switch
VLAN name and ID	**employee-vlan**, tag **20**
VLAN subnets	**192.0.2.16/28** **192.0.2.17** through **192.0.2.30** **192.0.2.31** is the subnet's broadcast address

Table 25: Components of the Port Security Topology *(Continued)*

Properties	Settings
Interfaces in **employee-vlan**	ge-0/0/1,ge-0/0/2, ge-0/0/3, ge-0/0/8
Interface for DHCP server	ge-0/0/8

In this example, the switch has already been configured as follows:

- Secure port access is activated on the switch.
- DHCP snooping is disabled on the VLAN **employee-vlan**.
- All access ports are untrusted, which is the default setting.

Configuration

IN THIS SECTION

To configure DHCP snooping and dynamic ARP inspection (DAI) to protect the switch against ARP attacks:

Procedure

CLI Quick Configuration

To quickly configure DHCP snooping and dynamic ARP inspection (DAI), copy the following commands and paste them into the switch terminal window:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface ge-0/0/8 dhcp-trusted
user@switch# set vlan employee-vlan examine-dhcp
user@switch# set vlan employee-vlan arp-inspection
```

Step-by-Step Procedure

Configure DHCP snooping and dynamic ARP inspection (DAI) on the VLAN:

1. Set the **ge-0/0/8** interface as trusted:

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set interface ge-0/0/8 dhcp-trusted
   ```

2. Enable DHCP snooping on the VLAN:

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set vlan employee-vlan examine-dhcp
   ```

3. Enable DAI on the VLAN:

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set vlan employee-vlan arp-inspection
   ```

Results

Check the results of the configuration:

```
[edit ethernet-switching-options secure-access-port]
user@switch# show
interface ge-0/0/8.0 {
    dhcp-trusted;
}
vlan employee-vlan {
    arp-inspection;
    examine-dhcp;
}
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying That DHCP Snooping Is Working Correctly on the Switch

Purpose

Verify that DHCP snooping is working on the switch.

Action

Send some DHCP requests from network devices (here they are DHCP clients) connected to the switch.

Display the DHCP snooping information when the port on which the DHCP server connects to the switch is trusted. The following output results when requests are sent from the MAC addresses and the server has provided the IP addresses and leases:

```
user@switch> show dhcp-snooping binding
DHCP Snooping Information:
MAC Address          IP Address     Lease    Type     VLAN            Interface

-----------------    ----------     -----    ----     ----            ---------

00:05:85:3A:82:77     192.0.2.17     600    dynamic  employee-vlan    ge-0/0/1.0

00:05:85:3A:82:79     192.0.2.18     653    dynamic  employee-vlan    ge-0/0/1.0

00:05:85:3A:82:80     192.0.2.19     720    dynamic  employee-vlan    ge-0/0/2.0

00:05:85:3A:82:81     192.0.2.20     932    dynamic  employee-vlan    ge-0/0/2.0

00:05:85:3A:82:83     192.0.2.21    1230    dynamic  employee-vlan    ge-0/0/2.0
```

```
00:05:85:27:32:88    192.0.2.22   3200    dynamic  employee-vlan   ge-0/0/3.0
```

Meaning

When the interface on which the DHCP server connects to the switch has been set to trusted, the output (see preceding sample) shows, for each MAC address, the assigned IP address and lease time—that is, the time, in seconds, remaining before the lease expires.

Verifying That DAI Is Working Correctly on the Switch

Purpose

Verify that DAI is working on the switch.

Action

Send some ARP requests from network devices connected to the switch.

Display the DAI information:

```
user@switch> show arp inspection statistics
ARP inspection statistics:
Interface       Packets received  ARP inspection pass   ARP inspection failed
---------------  ---------------   -------------------- ---------------------
ge-0/0/1.0                     7                    5                     2
ge-0/0/2.0                    10                   10                     0
ge-0/0/3.0                    12                   12                     0
```

Meaning

The sample output shows the number of ARP packets received and inspected per interface, with a listing of how many packets passed and how many failed the inspection on each interface. The switch compares the ARP requests and replies against the entries in the DHCP snooping database. If a MAC address or IP address in the ARP packet does not match a valid entry in the database, the packet is dropped.

RELATED DOCUMENTATION

Example: Configuring Port Security (non-ELS) | 15

Enabling DHCP Snooping (non-ELS) | 456

Enabling DHCP Snooping (J-Web Procedure)

Enabling Dynamic ARP Inspection (non-ELS) | 531

Enabling Dynamic ARP Inspection (J-Web Procedure)

secure-access-port

show arp inspection statistics

show dhcp snooping binding

Example: Prioritizing Snooped and Inspected Packet

IN THIS SECTION

On EX Series switches you might need to use class of service (CoS) to protect packets from critical applications from being dropped during periods of network congestion and delay and you might also need the port security features of DHCP snooping and dynamic ARP inspection (DAI) on the same ports through which those critical packets are entering and leaving. You can combine the advantages of both these features by using CoS forwarding classes and queues to prioritize snooped and inspected packets. This type of configuration places the snooped and inspected packets in the desired egress queue, ensuring that the security procedure does not interfere with the transmittal of this high-priority traffic. This is especially important for traffic that is sensitive to jitter and delay, such as voice traffic.

This example shows how to configure the switch to prioritize snooped and inspected packets in heavy network traffic.

Requirements

This example uses the following hardware and software components:

- One EX Series switch
- Junos OS Release 11.2 or later for EX Series switches
- A DHCP server to provide IP addresses to network devices on the switch

Before you specify CoS forwarding classes for snooped and inspected packets, be sure you have:

- Connected the DHCP server to the switch.
- Configured the VLAN **VLAN200** on the switch. See Configuring VLANs for EX Series Switches.
- Configured two interfaces, **ge-0/0/1** and **ge-0/0/8**, to belong to **VLAN200**.

Overview and Topology

IN THIS SECTION

Ethernet LANs are vulnerable to address spoofing and DoS attacks on network devices. To protect the devices from such attacks, you can configure DHCP snooping to validate DHCP server messages and DAI to protect against MAC spoofing. If you have to deal with periods of heavy network congestion and you want to ensure that sensitive traffic is not disrupted, you can combine the port security features with CoS forwarding classes to prioritize the handling of the snooped and inspected security packets.

In the default switch configuration:

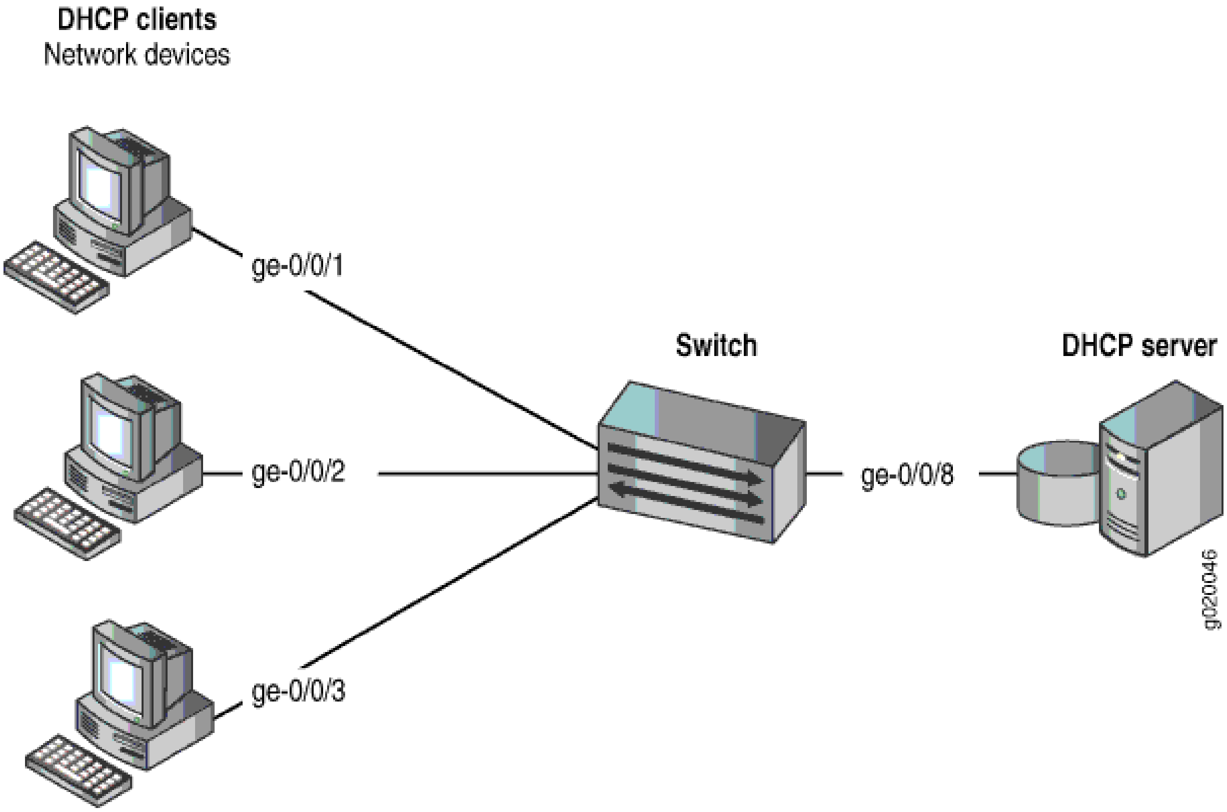
- Secure port access is activated on the switch.
- DHCP snooping and DAI are disabled on all VLANs.
- All access ports are untrusted and all trunk ports are trusted for DHCP snooping.

This example shows how to combine the DHCP snooping and DAI security features with prioritized forwarding of snooped and inspected packets.

The setup for this example includes the VLAN **VLAN200** on the switch. Figure 29 on page 491 illustrates the topology for this example.

Topology

Figure 29: Network Topology for Using CoS Forwarding Classes to Prioritize Snooped and Inspected Packets



The components of the topology for this example are shown in Table 26 on page 491.

Table 26: Components of the Topology for Using CoS Forwarding Classes to Prioritize Snooped and Inspected Packets

Properties	Settings
Switch hardware	EX Series switch
VLAN name	VLAN200
Interfaces in VLAN200	ge-0/0/1,ge-0/0/2,ge-0/0/3,ge-0/0/8

Table 26: Components of the Topology for Using CoS Forwarding Classes to Prioritize Snooped and Inspected Packets *(Continued)*

Properties	Settings
Interface for DHCP server	ge-0/0/8

In the configuration tasks for this example, you create a user-defined forwarding class **c1**, you enable DHCP snooping and DAI on VLAN200, and you assign the snooped and inspected packets to forwarding class **c1** and queue **6**. Queues 6 and 7 are reserved for high priority, control packets. The packets that are subjected to DHCP snooping and DAI are control (not data) packets; therefore, it is appropriate to place these snooped and inspected high-priority control packets in queue 6. (Queue 7 is higher priority than queue 6 and can also be used for this purpose.)

Configuration

IN THIS SECTION

To configure DHCP snooping and DAI on VLAN200, and to prioritize the snooped and inspected packets:

Procedure

CLI Quick Configuration

To quickly configure DHCP snooping and DAI with prioritized forwarding of snooped and inspected packets, copy the following commands and paste them into the switch terminal window:

```
[edit]
                         set class-of-service forwarding-classes class c1 queue 6
                         set ethernet-switching-options security-access-port vlan VLAN200
examine-dhcp forwarding-class c1
set ethernet-switching-options security-access-port vlan VLAN200 arp-inspection forwarding-class
c1
```

Step-by-Step Procedure

Configure DHCP and DAI with prioritized forwarding of snooped and inspected packets:

1. Create a user-defined forwarding class to be used for prioritizing the snooped and inspected packets.

   ```
   [edit class-of-service]
   user@switch# set forwarding-classes class c1 queue 6
   ```

2. Enable DHCP snooping on the VLAN and apply forwarding class **c1** to the snooped packets:

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set vlan VLAN200 examine-dhcp forwarding-class c1
   ```

3. Enable DAI on the VLAN and apply forwarding class **c1** to the inspected packets:

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set vlan VLAN200 arp-inspection forwarding-class c1
   ```

Results

Check the results of the configuration:

```
[edit ethernet-switching-options secure-access-port]
user@switch# show
vlan VLAN200 {
    arp-inspection forwarding-class c1;
    examine-dhcp forwarding-class c1;
}
[edit class-of-service]
user@switch# show
}
    forwarding-classes {
        class c1 queue-num 6;
    }
```

Verification

IN THIS SECTION

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

Verifying That Prioritized Forwarding Is Working Correctly on the Snooped Packets

Purpose

Verify that prioritized forwarding is working on the DHCP snooped packets.

Action

Send some DHCP requests from network devices to the switch. Display the output queue for one of the interfaces in VLAN200 to make sure that the packets are being transmitted in the designated queue:

```
user@switch> show interfaces ge 0/0/1 extensive
Egress queues: 8 supported, 5 in use
  Queue counters:       Queued packets  Transmitted packets   Dropped packets
    0 best-effort                    0                    0                 0
    1 assured-forw                   0                    0                 0
    5 expedited-fo                   0                    0                 0
    6 c1                             0                 3209                 0
    7 network-cont                   0               126371                0
```

Meaning

The command output shows that packets have been transmitted on forwarding class **c1** queue 6.

Continue testing by changing the setting of **examine-dhcp forwarding-class** to use one of the default queues, such as best-effort, and repeat the `show interfaces` command to compare the difference in the output. You can tell that the setting is working correctly by seeing the difference in the number of transmitted packets reported for forwarding class **c1** queue 6.

Verifying That Prioritized Forwarding Is Working Correctly on the DAI Inspected Packets

Purpose

Verify that prioritized forwarding is working on the DAI inspected packets.

Action

Send some ARP requests from network devices to the switch. Display the output queue for one of the interfaces in VLAN200 to make sure that the packets are being transmitted in the designated queue:

```
user@switch> show interfaces ge-0/0/1 extensive
Egress queues: 8 supported, 5 in use
  Queue counters:       Queued packets  Transmitted packets   Dropped packets
    0 best-effort                    0                    0                 0
    1 assured-forw                   0                    0                 0
    5 expedited-fo                   0                    0                 0
    6 c1                             0                 3209                 0
    7 network-cont                   0                126371                0
```

Meaning

The command output shows that packets have been transmitted on forwarding class **c1** queue 6.

Continue testing by changing the setting of **arp-inspection forwarding-class** to use one of the default queues, such as best-effort, and repeat the `show interfaces` command to compare the difference in the output. You can tell that the setting is working correctly by seeing the difference in the number of transmitted packets reported for forwarding class **c1** queue 6.

RELATED DOCUMENTATION

Example: Protecting Against ARP Spoofing Attacks | 482

Configuring DHCP Security with Q-in-Q Tunneling in Service Provider Style

IN THIS SECTION

Junos OS supports two different styles of configuration for switch interfaces: service provider style and enterprise style. The service provider style requires more configuration but provides greater flexibility. The enterprise style is easier to configure but offers less functionality.

With the enterprise style of configuration, logical interfaces are placed into Layer 2 mode by specifying ethernet-switching as the interface family. The ethernet-switching option can only be configured on a single logical unit, unit 0. You cannot bind a VLAN ID to unit 0, because these interfaces operate either in trunk mode, which supports traffic with various VLAN tags, or in access mode, which supports untagged traffic.

Some switching features, such as Q-in-Q tunneling, cannot be configured on logical interface unit 0. Q-in-Q tunneling requires the logical interface to transmit VLAN-tagged frames. To enable a logical interface to receive and forward VLAN-tagged Ethernet frames, you must bind the logical interface to that VLAN. Because the enterprise style does not allow binding of a VLAN ID to unit 0, you must use the service provider style to configure Q-in-Q tunneling.

To support DHCP security along with Q-in-Q tunneling, you can configure the following DHCP security features using the service provider style:

- DHCP snooping (DHCPv4 and DHCPv6)
- Dynamic ARP inspection
- Neighbor discovery inspection
- DHCP option 82
- DHCPv6 option 18 and option 37
- Lightweight DHCPv6 relay agent

You can combine the service provider and enterprise styles of configuration on the same physical interface using flexible Ethernet services encapsulation. With flexible Ethernet services encapsulation, you can configure encapsulations at the logical interface level instead of at the physical interface level. Defining multiple per-unit Ethernet encapsulations makes it easier to customize Ethernet-based services to multiple hosts connected to the same physical interface. For more information, see *Flexible Ethernet Services Encapsulation* .

NOTE: EX4300 switches do not support configuration of service provider style and enterprise style on the same physical interface.

Example: DHCP Security and Q-in-Q Tunneling with Service Provider Style Configuration

When configuring a physical interface to support only the service provider style, configure the `extended-vlan-bridge` encapsulation type to support bridging features. You must also configure native VLAN tagging on the physical interface so that it can operate in trunk mode and transmit Ethernet frames with VLAN tags for multiple VLANs. Configure flexible VLAN tagging on the interface to transmit packets with 802.1Q VLAN single-tagged and dual-tagged frames.

The following example configuration encapsulates physical interface ge-0/0/11 for service provider configuration and defines logical unit 111. VLAN ID v111 is bound to unit 111, and Q-in-Q tunneling is configured on logical interface ge-0/0/11.111. The configuration enables DHCP snooping, dynamic ARP inspection, and DHCP option 82 on VLAN v111.

```
set interfaces ge-0/0/11 flexible-vlan-tagging
set interfaces ge-0/0/11 native-vlan-id 112
set interfaces ge-0/0/11 encapsulation extended-vlan-bridge
set interfaces ge-0/0/11 input-native-vlan-push enable
set interfaces ge-0/0/11 unit 111 vlan-id-list 111-112
set interfaces ge-0/0/11 unit 111 input-vlan-map push
set interfaces ge-0/0/11 unit 111 output-vlan-map pop
set vlans V111 interface ge-0/0/11.111
set vlans V111 forwarding-options dhcp-security group TRUSTED overrides trusted
set vlans V111 forwarding-options dhcp-security group TRUSTED interface ge-0/0/11.111
set vlans V111 forwarding-options dhcp-security arp-inspection
set vlans V111 forwarding-options dhcp-security option-82 remote-id use-interface-description
logical
```

Example: DHCP Security and Q-in-Q Tunneling with Flexible Ethernet Services Encapsulation

The flexible Ethernet services encapsulation type enables a physical interface to support both styles of configuration. To support the service provider style, flexible Ethernet services allows for encapsulations to be configured at the logical interface level instead of the physical interface. To support the enterprise style, flexible Ethernet services allows the `ethernet-switching` family to be configured on any logical interface unit number.

The following example configuration encapsulates physical interface ge-0/0/11 with `flexible-ethernet-services` to support service provider and enterprise style configurations. Two logical units are defined on the physical interface: unit 111 for service provider style, and unit 0 for enterprise style. The `vlan-bridge` encapsulation enables bridging features on unit 111, and the `ethernet-switching` family enables bridging features on unit 0. Q-in-Q tunneling is configured on logical interface ge-0/0/11.111.

VLAN v111 is bound to unit 111 and has the following DHCP security features:

- DHCP snooping with option 82 and trusted override
- Dynamic ARP inspection

VLAN EP_v222 is bound to unit 0 and has the following DHCP security features:

- DHCP snooping with option 82
- Dynamic ARP inspection
- Neighbor discovery inspection

> **NOTE**: Interfaces with service provider style configuration are untrusted by default for DHCP. On interfaces with enterprise style configuration, access interfaces are untrusted and trunk interfaces are trusted.

```
set interfaces ge-0/0/11 flexible-vlan-tagging
set interfaces ge-0/0/11 native-vlan-id 112
set interfaces ge-0/0/11 encapsulation flexible-ethernet-services
set interfaces ge-0/0/11 input-native-vlan-push enable
set interfaces ge-0/0/11 unit 111 encapsulation vlan-bridge
set interfaces ge-0/0/11 unit 111 vlan-id-list 111-112
set interfaces ge-0/0/11 unit 111 input-vlan-map push
set interfaces ge-0/0/11 unit 111 output-vlan-map pop
set interfaces ge-0/0/11 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/0/11 unit 0 family ethernet-switching vlan members EP_V222
set vlans V111 interface ge-0/0/11.111
set vlans V111 forwarding-options dhcp-security group TRUSTED overrides trusted
set vlans V111 forwarding-options dhcp-security group TRUSTED interface ge-0/0/11.111
set vlans V111 forwarding-options dhcp-security arp-inspection
set vlans V111 forwarding-options dhcp-security option-82 remote-id use-interface-description
logical
set vlans EP_V222 vlan-id 222
set vlans EP_V222 forwarding-options dhcp-security arp-inspection
set vlans EP_V222 forwarding-options dhcp-security neighbor-discovery-inspection
set vlans EP_V222 forwarding-options dhcp-security option-82 remote-id use-interface-description
logical
```

Example: DHCP Security and Q-in-Q Tunneling with Support for Swap-Push/Pop-Swap

Q-in-Q tunneling and VLAN translation allow service providers to create an L2 Ethernet connection between two customer sites. Providers can segregate different customers' VLAN traffic on a link.

Q-in-Q tunneling with L2 swap-push/pop-swap support is a specific scenario in which the customer VLAN (C-VLAN) tag is swapped with the `inner-vlan-id` tag, and the service-provider-defined service VLAN (S-VLAN) tag is pushed on it (for traffic flowing from customer to service provider site). This traffic is sent to the service provider network double-tagged (S-VLAN + C-VLAN). For the traffic flowing from the service provider network to the customer network, the S-VLAN tag is removed, and the C-VLAN tag is replaced with the VLAN ID configured on the UNI logical interface.

The following example shows the swap-push/pop-swap dual tag operations.

1. Swap-push—For incoming-single tagged frame from UNI, the C-VLAN (VLAN ID 100) swaps with configured inner-VLAN ID (200) on logical interface and the S-VLAN (VLAN ID 900) pushes on to the frame. The double-tagged frame egresses out of NNI.

2. Pop-swap—For incoming double-tagged frame from NNI, the S-VLAN tag pops (VLAN ID 900) from the frame and the logical interface's VLAN ID 100 replaces the C-VLAN tag. The single-tagged frame egresses out of UNI.

To support DHCP security along with Q-in-Q tunneling, you can configure the following DHCP security features:

- DHCP snooping (DHCPv4 and DHCPv6)
- Dynamic ARP inspection
- DHCPv6 source-guard
- Neighbor discovery inspection
- DHCP option 82

- DHCPv6 option 37

```
set interfaces ge-0/0/1 description UNI
set interfaces ge-0/0/1 flexible-vlan-tagging
set interfaces ge-0/0/1 encapsulation flexible-ethernet-services
set interfaces ge-0/0/1 unit 100 encapsulation vlan-bridge
set interfaces ge-0/0/1 unit 100 vlan-id 100
set interfaces ge-0/0/1 unit 100 input-vlan-map swap-push
set interfaces ge-0/0/1 unit 100 input-vlan-map vlan-id 900
set interfaces ge-0/0/1 unit 100 input-vlan-map inner-vlan-id 200
set interfaces ge-0/0/1 unit 100 output-vlan-map pop-swap

set interfaces ge-0/0/2 description NNI
set interfaces ge-0/0/2 flexible-vlan-tagging
set interfaces ge-0/0/2 encapsulation flexible-ethernet-services
set interfaces ge-0/0/2 unit 900 encapsulation vlan-bridge
set interfaces ge-0/0/2 unit 900 vlan-id 900

set vlans vlan-900 interface ge-0/0/1.100
set vlans vlan-900 interface ge-0/0/2.900
set vlans vlan-900 forwarding-options dhcp-security arp-inspection
set vlans vlan-900 forwarding-options dhcp-security ip-source-guard
set vlans vlan-900 forwarding-options dhcp-security neighbor-discovery-inspection
set vlans vlan-900 forwarding-options dhcp-security ipv6-source-guard
set vlans vlan-900 forwarding-options dhcp-security group trusted overrides trusted
set vlans vlan-900 forwarding-options dhcp-security group trusted overrides no-option82
set vlans vlan-900 forwarding-options dhcp-security group trusted overrides no-dhcpv6-options
set vlans vlan-900 forwarding-options dhcp-security group trusted interface ge-0/0/2.900
```

If you configure the logical interface with a VLAN ID list and the input-vlan-map and output-vlan-map is configured as swap-push/pop-swap, it results in undesired behavior as the traffic regressing out of the UNI has a logical unit number instead of the original customer VLAN ID from VLAN ID list configured.

CHAPTER 16

DHCP Option 82

IN THIS CHAPTER

Understanding DHCP Option 82

IN THIS SECTION

You can use DHCP option 82, also known as the DHCP relay agent information option, to help protect supported Juniper devices against attacks including spoofing (forging) of IP addresses and MAC addresses, and DHCP IP address starvation.

In a common scenario, various hosts are connected to the network via untrusted access interfaces on the switch, and these hosts request and are assigned IP addresses from the DHCP server. Bad actors can spoof DHCP requests using forged network addresses, however, to gain an improper connection to the network.

To protect against this vulnerability, RFC 3046, *DHCP Relay Agent Information Option*, http://tools.ietf.org/html/rfc3046 describes a standard known as Option 82 which defines how for the DHCP server can use the location of a DHCP client when assigning IP addresses or other parameters to the client.

DHCP Option 82 Overview

If DHCP option 82 is enabled on a VLAN or bridge domain, then when a network device—a DHCP client—that is connected to the VLAN or bridge domain on an untrusted interface sends a DHCP request, the switching device inserts information about the client's network location into the packet header of that request. The switching device then sends the request to the DHCP server. The DHCP server reads the option 82 information in the packet header and uses it to implement the IP address or another parameter for the client. See "Suboption Components of Option 82" on page 503 for more information about option 82.

NOTE: On EX4300 switches, DHCP option 82 information is added to DHCP packets received on trusted interfaces as well as untrusted interfaces.

If option 82 is enabled on a VLAN or bridge domain, the following sequence of events occurs when a DHCP client sends a DHCP request:

1. The switching device receives the request and inserts the option 82 information in the packet header.

2. The switching device forwards (or relays) the request to the DHCP server.

3. The server uses the DHCP option 82 information to formulate its reply and sends a response to the switching device. It does not alter the option 82 information.

4. The switching device strips the option 82 information from the response packet.

5. The switching device forwards the response packet to the client.

To use the DHCP option 82 feature, you must ensure that the DHCP server is configured to accept option 82. If the DHCP server is not configured to accept option 82, then when it receives requests containing option 82 information, it does not use the information for setting parameters and it does not echo the information in its response message.

NOTE: If your switching device is an EX Series switch and uses Junos OS with Enhanced Layer 2 Software (ELS) configuration style, you can enable DHCP option 82 only for a specific VLAN. See "Setting Up DHCP Option 82 on the Switch with No Relay (ELS)" on page 517.

If your switching device is an EX Series switch and does *not* use Junos OS with Enhanced Layer 2 Software (ELS) configuration style, you can enable DHCP option 82 either for a specific VLAN or for all VLANs. See "Setting Up DHCP Option 82 on the Switch with No Relay (non-ELS)" on page 519.

Suboption Components of Option 82

Option 82 as implemented on a switching device comprises the suboptions circuit ID, remote ID, and vendor ID. These suboptions are fields in the packet header:

- circuit ID—Identifies the circuit (interface or VLAN) on the switching device on which the request was received. The circuit ID contains the interface name and VLAN name, with the two elements separated by a colon—for example, ge-0/0/10:vlan1, where ge-0/0/10 is the interface name and vlan1 is the VLAN name. If the request packet is received on a Layer 3 interface, the circuit ID is just the interface name—for example, ge-0/0/10.

  Use the prefix option to add an optional prefix to the circuit ID. If you enable the prefix option, the hostname for the switching device is used as the prefix; for example, device1:ge-0/0/10:vlan1, where device1 is the hostname.

  You can also specify that the interface description be used rather than the interface name or that the VLAN ID be used rather than the VLAN name.

- remote ID—Identifies the remote host. See *remote-id* for details.

- vendor ID—Identifies the vendor of the host. If you specify the `vendor-id` option but do not enter a value, the default value Juniper is used. To specify a value, you type a character string.

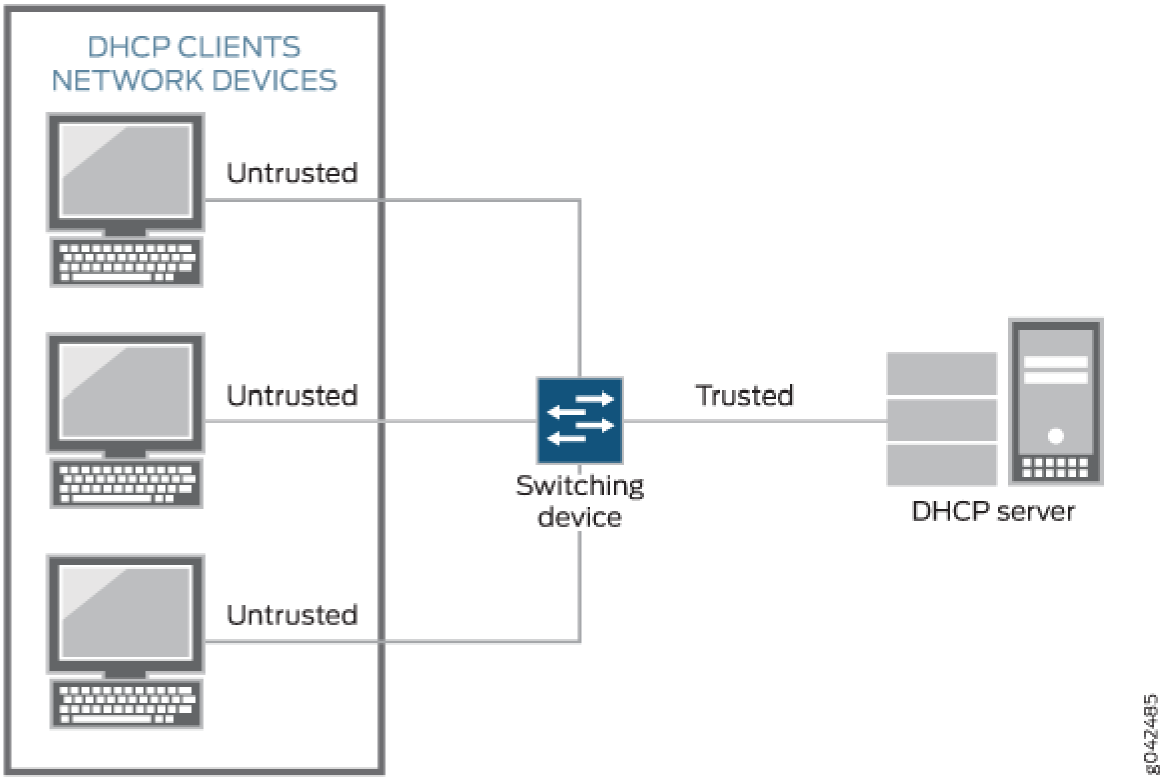
Switching Device Configurations That Support Option 82

Switching device configurations that support option 82 are:

Switching Device, DHCP Clients, and the DHCP Server Are on the Same VLAN or Bridge Domain

If the switching device, the DHCP clients, and the DHCP server are all on the same VLAN or bridge domain, the switching device forwards the requests from the clients on untrusted access interfaces to the server on a trusted interface. See Figure 30 on page 504.

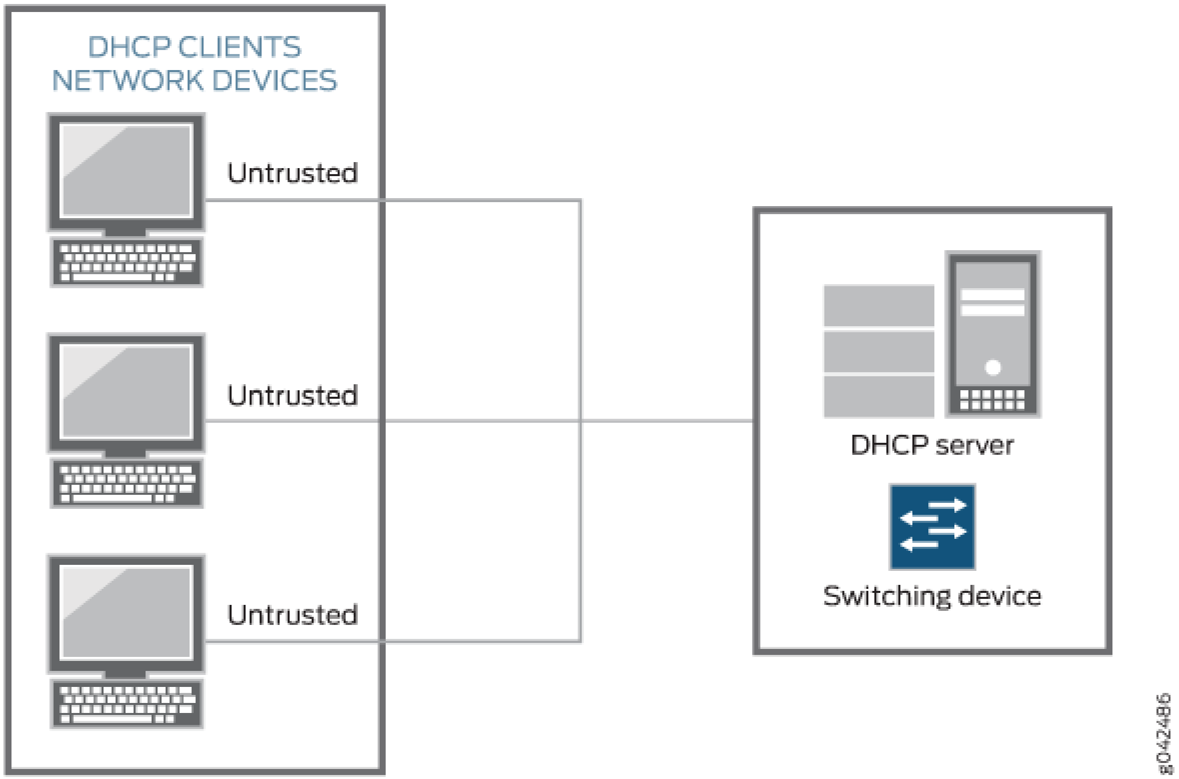
Figure 30: DHCP Clients, Switching Device, and the DHCP Server Are All on the Same VLAN or Bridge Domain



Switching Device Acts as a Relay Agent

The switching device functions as a relay agent (extended relay server) when the DHCP clients or the DHCP server is connected to the switching device through a Layer 3 interface. On the switching device, these interfaces are configured as routed VLAN interfaces (RVIs). Figure 31 on page 505 illustrates a scenario for the switching device acting as an extended relay server; in this instance, the switching device relays requests to the server. This figure shows the relay agent and server on the same network, but they can also be on different networks–that is, the relay agent can be external.

Figure 31: Switching Device Acting as an Extended Relay Server



DHCPv6 Options

DHCPv6 provides several options that can be used to insert information into the DHCPv6 request packets that are relayed to a server from a client. These options are equivalent to the sub-options of DHCP option 82.

- Option 37—Identifies the remote host. Option 37 is equivalent to the `remote-id` sub-option of DHCP option 82.

- Option 18—Identifies the interface on which the DHCP request packet was received from the client. Option 18 is equivalent to the `circuit-id` sub-option of DHCP option 82.

- Option 16—Identifies the vendor of the hardware on which the client is hosted. Option 16 is equivalent to the `vendor-id` sub-option of DHCP option 82.

DHCPv6 options are not enabled automatically when DHCPv6 snooping is enabled on a VLAN. They must be configured using the `dhcpv6-options` statement.

RELATED DOCUMENTATION

Example: Setting Up DHCP Option 82 | 508

Setting Up DHCP Option 82 on the Switch with No Relay (ELS) | 517

DHCP Option-82 Customization with EVPN/SR E-LAN/E-Tree

IN THIS SECTION

The DHCPv4 Option-82 customization feature, in conjunction with EVPN/SR E-LAN and E-Tree support, offers precise control over DHCP relay agent information by allowing you to configure custom hexadecimal or ASCII values for the circuit ID and remote ID. This capability ensures more granular policy enforcement based on detailed client data. Additionally, you can leverage enhanced visibility and management with the inclusion of physical interface details in the "show dhcp relay binding" command output. The system also supports the customization of DHCPv6 options, facilitating custom values for relay-agent-interface-id and relay-agent-remote-id. Furthermore, integrating Option-82 values into RADIUS VSA improves compatibility with non-standard authentication methods, while the unhidden NAK options provide feedback on unknown renew or rebind requests, enhancing client management. These features collectively bolster the flexibility, control, and serviceability of your network environment.

Benefits of DHCPv4 Option-82 Customization with EVPN/SR E-LAN/E-Tree Support

- Provides granular policy enforcement by allowing the configuration of custom hexadecimal or ASCII values for circuit ID and remote ID, ensuring more precise control over client data.
- Enhances network visibility and management by displaying the physical interface associated with each subscriber in the "show dhcp relay binding" command output, aiding in troubleshooting and network administration.
- Improves integration with non-standard authentication methods by including Option-82 values in RADIUS VSA, facilitating compatibility and flexibility in authentication processes.

- Supports customized DHCPv6 relay-agent options, enabling tailored configurations for relay-agent-interface-id and relay-agent-remote-id, which can improve network deployment for DHCPv6 environments.

- Provides better client management with the capability of sending DHCPNAKs for unknown rebind or renew requests, ensuring that unauthorized or misconfigured devices are promptly identified and addressed.

Overview

The DHCPv4 Option-82 customization feature with EVPN/SR E-LAN/E-Tree support allows you to define custom circuit ID and remote ID values in either ASCII or hexadecimal formats. This capability enhances your network management by providing more granular control over the information relayed to the DHCP server. By configuring these custom values, you can enforce specific policies based on detailed client data, which can be crucial in complex network environments where precise client identification is necessary.

To configure custom circuit ID and remote ID values, you can use the following CLI commands:

```
set forwarding-options dhcp-relay relay-option-82 circuit-id user-defined string <ascii-string> set forwarding-
options dhcp-relay relay-option-82 circuit-id user-defined hex-string <hexadecimal-string> set forwarding-options
dhcp-relay relay-option-82 remote-id user-defined string <ascii-string> set forwarding-options dhcp-relay relay-
option-82 remote-id user-defined hex-string <hexadecimal-string>
```

These commands allow you to specify custom strings that replace the default circuit ID and remote ID values, thus tailoring the information passed to the DHCP server for more accurate policy application and client tracking.

Additionally, this feature supports DHCPv6 environments by extending similar customization capabilities to the relay-agent-interface-id (Option 18) and relay-agent-remote-id (Option 37). You can configure these options using the following commands:

```
set forwarding-options dhcp-relay dhcpv6 relay-agent-interface-id user-defined string <ascii-string> set
forwarding-options dhcp-relay dhcpv6 relay-agent-interface-id user-defined hex-string <hexadecimal-string> set
forwarding-options dhcp-relay dhcpv6 relay-agent-remote-id user-defined string <ascii-string> set forwarding-
options dhcp-relay dhcpv6 relay-agent-remote-id user-defined hex-string <hexadecimal-string>
```

By utilizing these commands, you can ensure that custom relay agent information is accurately included in DHCPv6 messages, thus providing consistent policy enforcement and client management across both DHCPv4 and DHCPv6 protocols.

Trace Options for DHCP

To facilitate detailed debugging and logging of DHCP processes, you can enable trace options. This capability helps in diagnosing and resolving network issues effectively by providing comprehensive logs of DHCP activities. Use the following configuration to set up trace options:

```
system { processes { dhcp-service { traceoptions { file jdhcpd size 1g; level all; flag all; } } } }
```

By configuring these trace options, you gain visibility into the DHCP server's operations, allowing you to monitor and troubleshoot DHCP-related events with detailed logs. This improved serviceability ensures that network administrators can quickly identify and address issues, thereby maintaining the overall health and performance of the network.

Example: Setting Up DHCP Option 82

IN THIS SECTION

You can use DHCP option 82, also known as the DHCP relay agent information option, to help protect the switch against attacks such as spoofing (forging) of IP addresses and MAC addresses, and DHCP IP address starvation. Option 82 provides information about the network location of a DHCP client, and the DHCP server uses this information to implement IP addresses or other parameters for the client.

You can configure the DHCP option 82 feature in various topologies:

- The switch functions as a relay agent when the DHCP clients or the DHCP server is connected to the switch through a Layer 3 interface. On the switch, these interfaces are configured as routed VLAN interfaces, or RVIs. The switch relays the clients' requests to the server and then forwards the server's replies to the clients.

    - For EX Series switches, the configuration for this topology is the same for both Enhanced Layer 2 Software (ELS) and non-ELS.

- The switch, DHCP clients, and DHCP server are all on the same VLAN. The switch forwards the clients' requests to the server and forwards the server's replies to the clients.

    - If your switch is an EX Series, see "Setting Up DHCP Option 82 on the Switch with No Relay (ELS)" on page 517 for both ELS and non-ELS instructions.

- The switching device, DHCP clients, and DHCP server are all on the same bridge domain. The switching device forwards the clients' requests to the server and forwards the server's responses to the clients. This topic describes this configuration.

Before you configure DHCP option 82 on the switch, make sure the DHCP server is configured to accept DHCP option 82. If the server is not configured for DHCP option 82, the server does not use the DHCP option 82 information in the requests sent to it when it formulates its reply messages.

Example: Setting Up DHCP Option 82 on a VLAN

IN THIS SECTION

Requirements

This example describes how to configure DHCP option 82 on a switch that acts as a relay agent and is on the same VLAN as the DHCP clients, but is on a different VLAN from the DHCP server. The example includes the following hardware and software components:

- One EX4200-24P switch or one QFX3500 switch
- Junos OS Release 9.3 or later for EX Series switches or Junos OS Release 12.1 or later for the QFX Series
- A DHCP server to provide IP addresses to network devices on the switch

Overview and Topology

In this example, you configure option 82 on the switch. The switch is configured as a BOOTP relay agent (See DHCP/BOOTP Relay for Switches Overview for more information). The switch connects to the DHCP server through the routed VLAN interface (RVI), as described for QFX in Configuring IRB Interfaces on Switches and for EX Series switches in Configuring Routed VLAN Interfaces on Switches (CLI Procedure). The switch and clients are members of the **employee** VLAN (for details, see Configuring VLANs on Switches for the EX and QFX Series). The DHCP server is a member of the **corporate** VLAN.

If DHCP option 82 is enabled on the switch, then when a network device—a DHCP client—that is connected to the switch on an untrusted interface sends a DHCP request, the switch inserts information about the client's network location into the packet header of that request. The switch then sends the request (in this setting, it relays the request) to the DHCP server. The DHCP server reads the option 82 information in the packet header and uses it to implement the IP address or other parameter for the client.

When option 82 is enabled on the switch, then this sequence of events occurs when a DHCP client sends a DHCP request:

1. The switch receives the request and inserts the option 82 information in the packet header.

2. The switch relays the request to the DHCP server.

3. The server uses the DHCP option 82 information to formulate its reply and sends a response back to the switch. It does not alter the option 82 information.

4. The switch strips the option 82 information from the response packet.

5. The switch forwards the response packet to the client.

Configuration

IN THIS SECTION

To configure DHCP option 82:

Procedure

CLI Quick Configuration

To quickly configure DHCP option 82, copy the following commands and paste them into the switch terminal window:

```
set forwarding-options helpers bootp dhcp-option82
set forwarding-options helpers bootp dhcp-option82 circuit-id prefix hostname
 set forwarding-options helpers bootp dhcp-option82 circuit-id use-vlan-id
set forwarding-options helpers bootp dhcp-option82 remote-id
set forwarding-options helpers bootp dhcp-option82 remote-id prefix mac
set forwarding-options helpers bootp dhcp-option82 remote-id use-string employee-switch1
set forwarding-options helpers bootp dhcp-option82 vendor-id
```

Step-by-Step Procedure

To configure DHCP option 82 (replace values in italics with values for your own network):

1. Specify DHCP option 82 for the **employee** VLAN on the BOOTP server.

- On all interfaces that connect to the server:

  ```
  [edit forwarding-options helpers bootp]
  user@switch# set dhcp-option82
  ```

- On a specific interface that connects to the server:

  ```
  [edit forwarding-options helpers bootp]
  user@switch# set interface ge-0/0/10 dhcp-option82
  ```

  The remaining steps are optional. They show configurations for all interfaces; include the specific interface designation to configure any of the following options on a specific interface:

2. Configure a prefix for the circuit ID suboption (the prefix is always the hostname of the switch):

   ```
   [edit forwarding-options helpers bootp]
   user@switch# set dhcp-option82 circuit-id prefix hostname
   ```

3. To specify that the circuit ID suboption value should contain the interface description rather than the interface name (the default):

   NOTE: When you use the interface description rather than the interface name, the interface description has to be specified under interface unit ("set interfaces ge-0/0/0 unit 0 description "client"). If you do not do this, then the interface name is used.

   ```
   [edit forwarding-options helpers bootp]
   user@switch# set dhcp-option82 circuit-id use-interface-description
   ```

4. Specify that the circuit ID suboption value contains the VLAN ID rather than the VLAN name (the default):

   ```
   [edit forwarding-options helpers bootp]
   user@switch# set dhcp-option82 circuit-id use-vlan-id
   ```

5. Specify that the remote ID suboption be included in the DHCP option 82 information:

```
[edit forwarding-options helpers bootp]
user@switch# set dhcp-option82 remote-id
```

6. Configure a prefix for the remote ID suboption (here, the prefix is the MAC address of the switch):

```
[edit forwarding-options helpers bootp]
user@switch# set dhcp-option82 remote-id prefix mac
```

- Or, to specify that the prefix for the remote ID suboption be the hostname of the switch rather than the MAC address of the switch (the default):

  ```
  [edit forwarding-options helpers bootp]
  user@switch# set dhcp-option82 remote-id prefix hostname
  ```

  To specify that the remote ID suboption value should contain the interface description:

  ```
  [edit forwarding-options helpers bootp]
  user@switch# set dhcp-option82 remote-id use-interface-description
  ```

7. Specify that the remote ID suboption value contains a character string (here, the string is **employee-switch1**):

```
[edit forwarding-options helpers bootp]
user@switch# set dhcp-option82 remote-id use-string employee-switch1
```

8. Configure a vendor ID suboption value, and use the default value. To use the default value, (which is **Juniper**), do not type a character string after the **vendor-id** option keyword. Otherwise, specify a value such as show here:

```
[edit forwarding-options helpers bootp]
user@switch# set dhcp-option82 vendor-id mystring
```

Results

To view results of the configuration steps before committing the configuration, type the `show` command at the user prompt.

To commit these changes to the active configuration, type the `commit` command at the user prompt.

Check the results of the configuration:

```
[edit forwarding-options helpers bootp]
user@switch# show
dhcp-option82 {
    circuit-id {
        prefix hostname;
        use-vlan-id;
    }
    remote-id {
        prefix mac;
        use-string employee-switch1;
    }
    vendor-id;
}
```

SEE ALSO

> Example: Setting Up DHCP Option 82 Using the Same VLAN | **522**
>
> Understanding DHCP Option 82 | **501**
>
> http://tools.ietf.org/html/rfc3046.

Configuring DHCP Option 82 on a Router with Bridge Domain

Before you configure DHCP option 82 on the switching device, perform these tasks:

- Connect and configure the DHCP server.

> **NOTE**: Your DHCP server must be configured to accept DHCP option 82. If the server is not configured for DHCP option 82, the server does not use the DHCP option 82 information in the requests sent to it when it formulates its reply messages.

- Configure a bridge domain on the switching device and associate the interfaces on which the clients and the server connect, to the switch with that bridge domain.

To configure DHCP option 82:

1. Specify DHCP option 82 for the bridge domain that you configured:

   ```
   [edit bridge-domains bridge-domain-name forwarding-options dhcp-security]
   user@device# set option-82
   ```

   NOTE: If you want to enable DHCP option 82 on all bridge domains, you must configure it separately for each specific bridge domain.

   The remaining steps are optional.

2. Configure the prefix for the circuit ID suboption to include the hostname or the routing instance name for the bridge domain:

   ```
   [edit bridge-domains bridge-domain-name forwarding-options dhcp-security option-82]
   user@device# set circuit-id prefix (host-name | routing-instance-name)
   ```

3. Specify that the circuit ID suboption value contains the interface description rather than the interface name (the default):

   ```
   [edit bridge-domains bridge-domain-name forwarding-options dhcp-security option-82]
   user@device# set circuit-id use-interface-description
   ```

4. Specify that the circuit ID suboption value contains the VLAN ID rather than the VLAN name (the default):

   ```
   [edit bridge-domains bridge-domain-name forwarding-options dhcp-security option-82]
   user@device# set circuit-id use-vlan-id
   ```

5. Specify that the remote ID suboption is included in the DHCP option 82 information:

   ```
   [edit bridge-domains bridge-domain-name forwarding-options dhcp-security option-82]
   user@device# set remote-id
   ```

NOTE: If you do not specify a keyword after `remote-id`, the default value for the `remote-id` suboption is the interface name.

6. Specify that the remote ID suboption is the hostname of the switch:

```
[edit bridge-domains bridge-domain-name forwarding-options dhcp-security option-82]
user@device# set remote-id host-name
```

7. Specify that the remote ID suboption value contains the interface description:

```
[edit bridge-domains bridge-domain-name forwarding-options dhcp-security option-82]
user@device# set remote-id use-interface-description
```

8. Specify that the remote ID suboption value contains a character string:

```
[edit bridge-domains bridge-domain-name forwarding-options dhcp-security option-82]
user@device# set remote-id use-string mystring
```

9. Configure a vendor ID suboption:

- To use the default value (the default value is `Juniper`), do not type a character string after the `vendor-id` option keyword:

```
[edit bridge-domains bridge-domain-name forwarding-options dhcp-security option-82]
user@device# set vendor-id
```

- To configure it so that the vendor ID suboption value contains a character string value that you specify rather than `Juniper` (the default):

```
[edit bridge-domains bridge-domain-name forwarding-options dhcp-security option-82]
user@device# set vendor-id use-string mystring
```

SEE ALSO

Understanding DHCP Option 82 | 501

Understanding DHCP Snooping (non-ELS) | 445

Example: Setting Up DHCP Option 82 (No Relay)

IN THIS SECTION

You can use DHCP option 82, also known as the DHCP relay agent information option, to help protect the switch against attacks such as spoofing (forging) of IP addresses and MAC addresses, and DHCP IP address starvation. Option 82 provides information about the network location of a DHCP client, and the DHCP server uses this information to implement IP addresses or other parameters for the client.

NOTE: DHCP option 82 is not supported on the QFX10000 switches.

You can configure the DHCP option 82 feature in several topologies:

- The switch, DHCP clients, and DHCP server are all on the same VLAN. The switch forwards the clients' requests to the server and forwards the server's replies to the clients. This topic describes this configuration.

- The switch functions as a relay agent when the DHCP clients or the DHCP server is connected to the switch through a Layer 3 interface. This means that the relay agent and server can be on different networks–that is, the relay agent can be external. On the switch, these interfaces are configured as routed VLAN interfaces (RVIs) or, the interfaces are configured as integrated routing and bridging (IRB) interfaces. In either case, the switch relays the clients' requests to the server and then forwards the server's replies to the clients. These configurations are described in "Example: Setting Up DHCP Option 82" on page 508.

Before you configure DHCP option 82 on the switch, perform these tasks:

- Connect and configure the DHCP server.

  NOTE: Your DHCP server must be configured to accept DHCP option 82. If the server is not configured for DHCP option 82, the server does not use the DHCP option 82 information in the requests sent to it when it formulates its reply messages.

- Configure a VLAN on the switch and associate the interfaces on which the clients and the server connect to the switch with that VLAN. See Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)

Setting Up DHCP Option 82 on the Switch with No Relay (ELS)

NOTE: This task uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Setting Up DHCP Option 82 on the Switch with No Relay (non-ELS)" on page 519. For ELS details, see Using the Enhanced Layer 2 Software CLI.

To configure DHCP option 82:

1. Specify DHCP option 82 for the VLAN that you configured.

```
[edit vlans vlan-name forwarding-options dhcp-security]
user@switch# set option-82
```

NOTE: If you want to enable DHCP option 82 on all VLANs, you must configure it separately for each specific VLAN.

The remaining steps are optional.

2. Configure the prefix for the circuit ID suboption to include the switch’s hostname or the routing instance name for the VLAN:

```
[edit vlans vlan-name forwarding-options dhcp-security option-82]
user@switch# set circuit-id prefix (host-name | routing-instance-name)
```

3. Specify that the circuit ID suboption value contains the interface description rather than the interface name (the default):

```
[edit vlans vlan-name forwarding-options dhcp-security option-82]
user@switch# set circuit-id use-interface-description
```

NOTE: Starting in Junos OS Release 14.1X53-D25, when you use the interface description rather than the interface name, the interface description has to be specified under interface unit.When you use the interface description rather than the interface name, the interface description has to be specified under interface unit ("set interfaces ge-0/0/0 unit 0 description "client"). If you do not do this, then the interface name is used.

4. Specify that the circuit ID suboption value contains the VLAN ID rather than the VLAN name (the default):

```
[edit vlans vlan-name forwarding-options dhcp-security option-82]
user@switch# set circuit-id use-vlan-id
```

5. Specify that the remote ID suboption is included in the DHCP option 82 information:

```
[edit vlans vlan-name forwarding-options dhcp-security option-82]
user@switch# set remote-id
```

NOTE: If you do not specify a keyword after `remote-id`, the default value for the `remote-id` suboption is the interface name.

6. Specify that the remote ID suboption is the hostname of the switch:

```
[edit vlans vlan-name forwarding-options dhcp-security option-82]
user@switch# set remote-id host-name
```

7. Specify that the remote ID suboption value contains the interface description:

```
[edit vlans vlan-name forwarding-options dhcp-security option-82]
user@switch# set remote-id use-interface-description
```

8. Specify that the remote ID suboption value contains a character string:

```
[edit vlans vlan-name forwarding-options dhcp-security option-82]
user@switch# set remote-id use-string mystring
```

9. Configure a vendor ID suboption:

- To use the default value (the default value is `Juniper`), do not type a character string after the `vendor-id` option keyword:

  ```
  [edit vlans vlan-name forwarding-options dhcp-security option-82]
  user@switch# set vendor-id
  ```

- To configure that the vendor ID suboption value contains a character string value that you specify rather than `Juniper` (the default):

  ```
  [edit vlans vlan-name forwarding-options dhcp-security option-82]
  user@switch# set vendor-id use-string mystring
  ```

SEE ALSO

Example: Setting Up DHCP Option 82 | **508**

http://tools.ietf.org/html/rfc3046.

Setting Up DHCP Option 82 on the Switch with No Relay (non-ELS)

> **NOTE**: This task uses Junos OS for EX Series switches that do not include support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does support ELS, see "Setting Up DHCP Option 82 on the Switch with No Relay (ELS)" on page 517. For ELS details, see Using the Enhanced Layer 2 Software CLI.

To configure DHCP option 82:

> **NOTE**: Replace values displayed in italics with values for your configuration.

1. Specify DHCP option 82 for all VLANs associated with the switch or for a specified VLAN. (You can also configure the feature for a VLAN range.)

   - On a specific VLAN:

     ```
     [edit ethernet-switching-options secure-access-port]
     user@switch# set vlan employee dhcp-option82
     ```

- On all VLANs:

  ```
  [edit ethernet-switching-options secure-access-port]
  user@switch# set vlan all dhcp-option82
  ```

  The remaining steps are optional.

2. To configure a prefix for the circuit ID suboption (the prefix is always the hostname of the switch):

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set vlan employee dhcp-option82 circuit-id prefix hostname
   ```

3. To specify that the circuit ID suboption value should contain the interface description rather than the interface name (the default):

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set vlan employee dhcp-option82 circuit-id use-interface-description
   ```

   NOTE: When you use the interface description rather than the interface name, the interface description has to be specified under interface unit ("set interfaces ge-0/0/0 unit 0 description "client"). If you do not do this, then the interface name is used.

4. To specify that the circuit ID suboption value should contain the VLAN ID rather than the VLAN name (the default):

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set vlan employee dhcp-option82 circuit-id use-vlan-id
   ```

5. To specify that the remote ID suboption be included in the DHCP option 82 information:

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set vlan employee dhcp-option82 remote-id
   ```

6. To configure a prefix for the remote ID suboption (here, the prefix is the MAC address of the switch):

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set vlan employee dhcp-option82 remote-id prefix mac
   ```

7. To specify that the prefix for the remote ID suboption be the hostname of the switch rather than the MAC address of the switch (the default):

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set vlan employee dhcp-option82 remote-id prefix hostname
   ```

8. To specify that the remote ID suboption value should contain the interface description:

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set vlan employee dhcp-option82 remote-id use-interface-description
   ```

9. To specify that the remote ID suboption value should contain a character string:

   ```
   [edit ethernet-switching-options secure-access-port]
   user@switch# set vlan employee dhcp-option82 remote-id use-string mystring
   ```

10. To configure a vendor ID suboption and use the default value (the default value is **Juniper**), do not type a character string after the **vendor-id** option keyword:

    ```
    [edit ethernet-switching-options secure-access-port]
    user@switch# set vlan employee dhcp-option82 vendor-id
    ```

11. To specify that the vendor ID suboption value should contain a character string value that you specify rather than **Juniper** (the default):

    ```
    [edit ethernet-switching-options secure-access-port]
    user@switch# set vlan employee dhcp-option82 vendor-id mystring
    ```

To view results of the configuration steps before committing the configuration, type the `show` command at the user prompt.

To commit these changes to the active configuration, type the `commit` command at the user prompt.

SEE ALSO

> http://tools.ietf.org/html/rfc3046.

Example: Setting Up DHCP Option 82 Using the Same VLAN

IN THIS SECTION

This example describes how to configure DHCP option 82 on a switch with DHCP clients, DHCP server, and switch all on the same VLAN:

Requirements

This example uses the following hardware and software components:

- One EX Series or QFX Series switch
- Junos OS Release 9.3 or later for EX Series switches or Junos OS Release 12.1 or later for the QFX Series
- A DHCP server to provide IP addresses to network devices on the switch

Overview and Topology

IN THIS SECTION

If DHCP option 82 is enabled on the switch, then when a network device—a DHCP client—that is connected to the switch on an untrusted interface sends a DHCP request, the switch inserts information about the client's network location into the packet header of that request. The switch then sends the request to the DHCP server. The DHCP server reads the option 82 information in the packet header and uses it to implement the IP address or other parameter for the client.

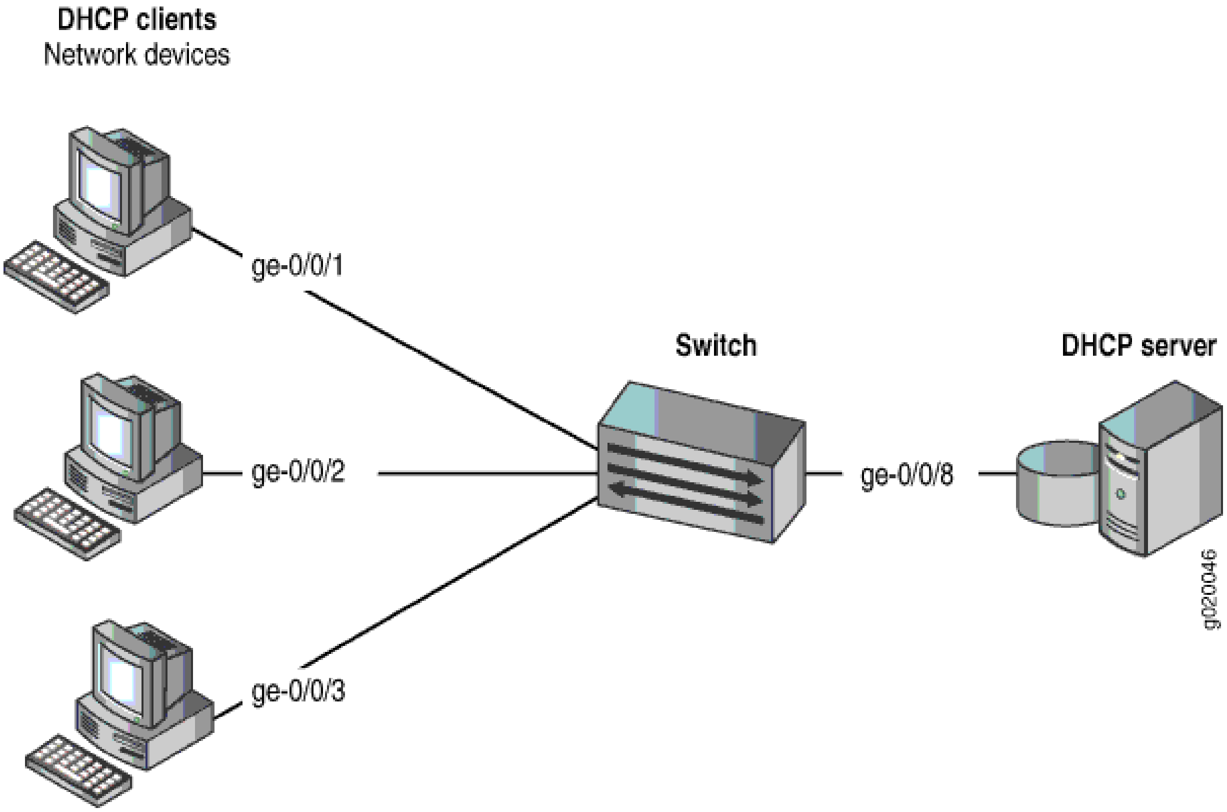
DHCP option 82 is enabled on an individual VLAN or on all VLANs on the switch.

When option 82 is enabled on the switch, then this sequence of events occurs when a DHCP client sends a DHCP request:

1. The switch receives the request and inserts the option 82 information in the packet header.
2. The switch forwards the request to the DHCP server.
3. The server uses the DHCP option 82 information to formulate its reply and sends a response back to the switch. It does not alter the option 82 information.
4. The switch strips the option 82 information from the response packet.
5. The switch forwards the response packet to the client.

Figure 32 on page 523 illustrates the topology for this example.

Figure 32: Network Topology for Configuring DHCP Option 82 on a Switch That Is on the Same VLAN as the DHCP Clients and the DHCP Server



Topology

In this example, you configure DHCP option 82 on the switch. The switch connects to the DHCP server on interface **ge-0/0/8**. The DHCP clients connect to the switch on interfaces **ge-0/0/1**, **ge-0/0/2**, and **ge-0/0/3**.

The switch, server, and clients are all members of the **employee** VLAN – be sure to configure the **employee** VLAN on the switch and associated the interfaces on which the clients and the server connect to the switch with the **employee** VLAN.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure DHCP option 82, copy the following commands and paste them into the switch terminal window:

```
set ethernet-switching-options secure-access-port vlan employee dhcp-option82
set ethernet-switching-options secure-access-port vlan employee dhcp-option82 circuit-id prefix hostname
set ethernet-switching-options secure-access-port vlan employee dhcp-option82 circuit-id use-vlan-id
set ethernet-switching-options secure-access-port vlan employee dhcp-option82 remote-id
set ethernet-switching-options secure-access-port vlan employee dhcp-option82 remote-id prefix mac
set ethernet-switching-options secure-access-port vlan employee dhcp-option82 remote-id use-string employee-switch1
 set ethernet-switching-options secure-access-port vlan employee dhcp-option82 vendor-id
```

Step-by-Step Procedure

To configure DHCP option 82:

1. Specify DHCP option 82 for the **employee** VLAN:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan employee dhcp-
option82
```

2. Configure a prefix for the circuit ID suboption (the prefix is always the hostname of the switch):

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan employee dhcp-option82 circuit-id prefix hostname
```

3. Specify that the circuit ID suboption value contains the VLAN ID rather than the VLAN name (the default):

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan employee dhcp-option82 circuit-id use-vlan-id
```

4. Specify that the remote ID suboption be included in the DHCP option 82 information:

```
[edit ethernet-switching-options secure-accesswitch# set vlan employee dhcp-option82 remote-
id
```

5. Configure a prefix for the remote ID suboption (here, the prefix is the MAC address of the switch):

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan employee dhcp-option82 remote-id prefix mac
```

6. Specify that the remote ID suboption value contain a character string (here, the string is **employee-switch1**):

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan employee dhcp-option82 remote-id use-string employee-switch1
```

7. Configure a vendor ID suboption value, and use the default value. To use the default value, do not type a character string after the **vendor-id** option keyword:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan employee dhcp-option82 vendor-id
```

Results

Check the results of the configuration:

```
[edit ethernet-switching-options secure-access-port]
user@switch# show
vlan employee {
    dhcp-option82 {
        circuit-id {
            prefix hostname;
            use-vlan-id;
        }
        remote-id {
            prefix mac;
            use-string employee-switch1;
        }
        vendor-id;
    }
}
```

SEE ALSO

Example: Setting Up DHCP Option 82 | **508**

http://tools.ietf.org/html/rfc3046.

Configuring VLANs for EX Series Switches

Configuring VLANs on Switches

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1X53-D25	Starting in Junos OS Release 14.1X53-D25, when you use the interface description rather than the interface name, the interface description has to be specified under interface unit.

CHAPTER 17

Dynamic ARP Inspection (DAI)

IN THIS CHAPTER

Understanding and Using Dynamic ARP Inspection (DAI)

IN THIS SECTION

Dynamic ARP inspection (DAI) protects switching devices against Address Resolution Protocol (ARP) packet spoofing (also known as ARP poisoning or ARP cache poisoning).

DAI inspects ARPs on the LAN and uses the information in the DHCP snooping database on the switch to validate ARP packets and to protect against ARP spoofing. ARP requests and replies are compared against entries in the DHCP snooping database, and filtering decisions are made based on the results of those comparisons. When an attacker tries to use a forged ARP packet to spoof an address, the switch compares the address with entries in the database. If the media access control (MAC) address or IP address in the ARP packet does not match a valid entry in the DHCP snooping database, the packet is dropped

Understanding ARP Spoofing and Inspection

IN THIS SECTION

ARP packets are sent to the Routing Engine and are rate-limited to protect the switching device from CPU overload.

Address Resolution Protocol

Sending IP packets on a multi-access network requires mapping an IP address to an Ethernet MAC address.

Ethernet LANs use ARP to map MAC addresses to IP addresses.

The switching device maintains this mapping in a cache that it consults when forwarding packets to network devices. If the ARP cache does not contain an entry for the destination device, the host (the DHCP client) broadcasts an ARP request for that device's address and stores the response in the cache.

ARP Spoofing

ARP spoofing is one way to initiate man-in-the-middle attacks. The attacker sends an ARP packet that spoofs the MAC address of another device on the LAN. Instead of the switching device sending traffic to the proper network device, it sends the traffic to the device with the spoofed address that is impersonating the proper device. If the impersonating device is the attacker's machine, the attacker receives all the traffic from the switch that must have gone to another device. The result is that traffic from the switching device is misdirected and cannot reach its proper destination.

One type of ARP spoofing is gratuitous ARP, which is when a network device sends an ARP request to resolve its own IP address. In normal LAN operation, gratuitous ARP messages indicate that two devices have the same MAC address. They are also broadcast when a network interface card (NIC) in a device is changed and the device is rebooted, so that other devices on the LAN update their ARP caches. In malicious situations, an attacker can poison the ARP cache of a network device by sending an ARP response to the device that directs all packets destined for a certain IP address to go to a different MAC address instead.

To prevent MAC spoofing through gratuitous ARP and through other types of spoofing, the switches examine ARP responses through DAI.

Dynamic ARP Inspection

DAI examines ARP requests and responses on the LAN and validates ARP packets. The switch intercepts ARP packets from an access port and validates them against the DHCP snooping database. If no IP-MAC entry in the database corresponds to the information in the ARP packet, DAI drops the ARP packet and the local ARP cache is not updated with the information in that packet. DAI also drops ARP packets when the IP address in the packet is invalid. ARP probe packets are not subjected to dynamic ARP inspection. The switch always forwards such packets.

Junos OS for EX Series switches and the QFX Series uses DAI for ARP packets received on access ports because these ports are untrusted by default. Trunk ports are trusted by default, and therefore ARP packets bypass DAI on them.

You configure DAI for each VLAN, not for each interface (port). By default, DAI is disabled for all VLANs.

If you set an interface to be a DHCP trusted port, it is also trusted for ARP packets.

NOTE:

- If your switching device is an EX Series switch and uses Junos OS with support for the Enhanced Layer 2 Software (ELS) configuration style, see "Enabling a Trusted DHCP Server (ELS)" on page 416 for information about configuring an access interface to be a DHCP trusted port.

For packets directed to the switching device to which a network device is connected, ARP queries are broadcast on the VLAN. The ARP responses to those queries are subjected to the DAI check.

For DAI, all ARP packets are trapped to the Packet Forwarding Engine. To prevent CPU overloading, ARP packets destined for the Routing Engine are rate-limited.

If the DHCP server goes down and the lease time for an IP-MAC entry for a previously valid ARP packet runs out, that packet is blocked.

Prioritizing Inspected Packets

NOTE: Prioritizing inspected packets is not supported on the QFX Series and the EX4600 switch.

You can use class-of-service (CoS) forwarding classes and queues to prioritize DAI packets for a specified VLAN. This type of configuration places inspected packets for that VLAN in the egress queue, that you specify, ensuring that the security procedure does not interfere with the transmission of high-priority traffic.

Enabling Dynamic ARP Inspection (ELS)

NOTE: This task uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Enabling Dynamic ARP Inspection (non-ELS)" on page 531.

Dynamic ARP inspection (DAI) protects switches against ARP spoofing. DAI inspects ARP packets on the LAN and uses the information in the DHCP snooping database on the switch to validate ARP packets and to protect against ARP cache poisoning.

Before you can enable DAI on a VLAN, you must configure the VLAN. See Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure).

To enable DAI on a VLAN by using the CLI:

```
[edit vlans vlan-name forwarding-options dhcp-security]
user@switch# set  arp-inspection
```

SEE ALSO

Example: Configuring IP Source Guard and Dynamic ARP Inspection to Protect the Switch from IP Spoofing and ARP Spoofing | **573**

Enabling Dynamic ARP Inspection (non-ELS)

IN THIS SECTION

NOTE: This task uses Junos OS for EX Series switches that do not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does support ELS, see "Enabling Dynamic ARP Inspection (ELS)" on page 531. For ELS details, see Using the Enhanced Layer 2 Software CLI.

Dynamic ARP inspection (DAI) protects switches against ARP spoofing. DAI inspects ARP packets on the LAN and uses the information in the DHCP snooping database on the switch to validate ARP packets and to protect against ARP cache poisoning.

Enabling DAI on a VLAN

You configure DAI for each VLAN, not for each interface (port). By default, DAI is disabled for all VLANs.

To enable DAI on a VLAN or all VLANs:

- On a single VLAN:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan vlan-name arp-inspection
```

- On all VLANs:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan all arp-inspection
```

Enabling DAI on a bridge domain

See Configuring a Bridge Domain to set up a bridge domain if necessary.

- To enable DAI on a bridge domain:

```
[edit bridge-domains bridge-domain-name forwarding-options dhcp-security]
user@device# set  arp-inspection
```

RELATED DOCUMENTATION

Example: Configuring Port Security (non-ELS) | 15

Example: Prioritizing Snooped and Inspected Packet | 489

Monitoring Port Security

Applying CoS Forwarding Classes to Prioritize Inspected Packets

You might need to use class of service (CoS) to protect packets from critical applications from being dropped during periods of network congestion and delay and you might also need the port security features of DHCP snooping on the same ports through which those critical packets are entering and leaving.

To apply CoS forwarding classes and queues to DAI packets:

1. Create a user-defined forwarding class to be used for prioritizing DAI packets:

   ```
   [edit class-of-service]
   user@switch# set forwarding-classes class class-name queue queue-number
   ```

2. Enable DAI on a specific VLAN or on all VLANs and apply the desired forwarding class on the DAI packets:

   - On a specific VLAN:

     ```
     [edit ethernet-switching-options secure-access port]
     user@switch# set vlan vlan-name arp-inspection  forwarding-class class-name
     ```

   - On all VLANs:

     ```
     [edit ethernet-switching-options secure-access port]
     user@switch# set vlan all arp-inspection forwarding-class class-name
     ```

Verifying That DAI Is Working Correctly

IN THIS SECTION

Purpose

Verify that dynamic ARP inspection (DAI) is working on the switch.

Action

Send some ARP requests from network devices connected to the switch.

Display the DAI information:

```
user@switch>                      show arp inspection statistics
ARP inspection statistics:
Interface        Packets received  ARP inspection pass   ARP inspection failed
---------------  ---------------  -------------------- ---------------------
ge-0/0/1.0                     7                    5                      2
ge-0/0/2.0                    10                   10                      0
ge-0/0/3.0                    12                   12                      0
```

Meaning

The sample output shows the number of ARP packets received and inspected per interface, with a listing of how many packets passed and how many failed the inspection on each interface. The switch compares the ARP requests and replies against the entries in the DHCP snooping database. If a MAC address or IP address in the ARP packet does not match a valid entry in the database, the packet is dropped.

SEE ALSO

Example: Protecting Against Address Spoofing and Layer 2 DoS Attacks | 464

Example: Protecting Against ARP Spoofing Attacks | 482

Configuring Static ARP Table Entries For Mapping IP Addresses to MAC Addresses

By default, the device responds to an Address Resolution Protocol (ARP) request only if the destination address of the ARP request is on the local network of the incoming interface. For Fast Ethernet or Gigabit Ethernet interfaces, you can configure static ARP entries that associate the IP addresses of nodes on the same Ethernet subnet with their media access control (MAC) addresses. These static ARP entries enable the device to respond to ARP requests even if the destination address of the ARP request is not local to the incoming Ethernet interface.

Also, unlike dynamically learned ARP entries, static ARP entries do not age out. You can also configure static ARP entries in a troubleshooting situation or if your device is unable to learn a MAC address dynamically.

> **NOTE**: By default, an ARP policer is installed that is shared among all the Ethernet interfaces on which you have configured the `family inet` statement. By including the `arp` statement at the `[edit interfaces` *interface-name* `unit` *logical-unit-number* `family inet policer]` hierarchy level, you can apply a specific ARP-packet policer to an interface. This feature is not available on EX Series switches.

To configure static ARP entries:

1. In the configuration mode, at the `[edit]` hierarchy level, configure the router interface on which the ARP table entries for the router is configured.

```
[edit]
user@host# edit interfaces interface-name
```

2. Configure the protocol family, the logical unit of the interface, and the interface address of the router interface at the `[edit interfaces` *interface-name*`]` hierarchy level. While configuring the protocol family, specify `inet` as the protocol family.

> **NOTE**: When you need to conserve IP addresses, you can configure an Ethernet interface to be unnumbered by including the `unnumbered-address` statement at the `[edit interfaces` *interface-name* `unit` *logical-unit-number* `family inet]` hierarchy level.

```
[edit interfaces interface-name]
user@host# edit unit logical-unit-number family inet address interface-address
```

3. Configure a static ARP entry by specifying the IP address and the MAC address that are to be mapped to each other. The IP address specified must be part of the subnet defined in the enclosing `address` statement. The MAC address must be specified as hexadecimal bytes in the following formats: *nnnn*.*nnnn*.*nnnn* or *nn*:*nn*:*nn*:*nn*:*nn*:*nn*format. For instance, you can use either `0011.2233.4455` or `00:11:22:33:44:55`.

```
[edit interfaces interface-name unit logical-unit-number family inet address interface-address
user@host# set arp ip-address mac mac-address
```

4. Configure another static ARP entry by specifying the IP address and the MAC address that are to be mapped to each other. You can also associate a multicast MAC address with a unicast IP address by including the `multicast-mac` option with the `arp` statement. You can optionally configure the router to respond to ARP requests for the specified IP address by using the `publish` option with the `arp` statement.

   NOTE: For unicast MAC addresses only, if you include the `publish` option, the router or switch replies to proxy ARP requests.

   ```
   [edit interfaces interface-name unit logical-unit-number family inet address interface-address
   user@host# set arp ip-address multicast-mac mac-address publish
   ```

NOTE: The Junos OS supports the IPv6 static neighbor discovery cache entries, similar to the static ARP entries in IPv4.

RELATED DOCUMENTATION

arp

Management Ethernet Interface Overview

Applying Policers

Configuring an Unnumbered Interface

Ethernet Interfaces User Guide for Routing Devices

8

PART

IP Source Guard

CHAPTER 18

Understanding IP Source Guard

IN THIS CHAPTER

Understanding IP Source Guard for Port Security on Switches

IN THIS SECTION

Ethernet LAN switches are vulnerable to attacks that involve spoofing (forging) of source IP addresses or source MAC addresses. You can use the IP source guard access port security feature to mitigate the effects of these attacks.

IP Address Spoofing

Hosts on access interfaces can spoof source IP addresses and source MAC addresses by flooding the switch with packets containing invalid addresses. Such attacks combined with other techniques such as TCP SYN flood attacks can cause denial-of-service (DoS) attacks. With source IP address or source MAC address spoofing, the system administrator cannot identify the source of the attack. The attacker can spoof addresses on the same subnet or on a different subnet.

How IP Source Guard Works

IP source guard examines each packet sent from a host attached to an untrusted access interface on the switch. The IP address, MAC address, VLAN and interface associated with the host is checked against entries stored in the DHCP snooping database. If the packet header does not match a valid entry in the DHCP snooping database, the switch does not forward the packet—that is, the packet is discarded.

NOTE:

- If your switch uses Junos OS for EX Series with support for the Enhanced Layer 2 Software (ELS) configuration style, DHCP snooping is enabled automatically when you enable IP source guard on a VLAN. See "Configuring IP Source Guard (ELS)" on page 545.

- If your switch uses Junos OS for EX Series without support the Enhanced Layer 2 Software (ELS) configuration style and you enable IP source guard on a VLAN, you must also explicitly enable DHCP snooping on that VLAN. Otherwise, the default value of no DHCP snooping applies to the VLAN. See "Configuring IP Source Guard (non-ELS)" on page 541.

IP source guard examines packets sent from untrusted access interfaces on those VLANs. By default, access interfaces are untrusted and trunk interfaces are trusted. IP source guard does not examine packets that have been sent to the switch by devices connected to trusted interfaces so that a DHCP server can be connected to that interface to provide dynamic IP addresses.

NOTE: On an EX9200 switch, you can set a trunk interface as `untrusted` so that it supports IP source guard.

IPv6 Source Guard

IPv6 source guard is available on switches that support DHCPv6 snooping. To determine whether your switch supports DHCPv6 snooping, see Feature Explorer.

The DHCP Snooping Table

IP source guard obtains information about IP address to MAC address bindings (IP-MAC binding) from the DHCP snooping table, also known as the DHCP binding table. The DHCP snooping table is populated either through dynamic DHCP snooping or through configuration of specific static IP address to MAC address bindings. For more information about the DHCP snooping table, see "Understanding DHCP Snooping (ELS)" on page 435.

To display the DHCP snooping table, issue the operational mode command that appears in the switch CLI.

For DHCP snooping:

- (For non-ELS switches) `show ip-source-guard`
- (ELS switches only) `show dhcp-security binding`

For DHCPv6 snooping:

- (For non-ELS switches) `show dhcpv6 snooping binding`
- (ELS switches only) `show dhcp-security ipv6 binding`

Typical Uses of Other Junos OS Features with IP Source Guard

You can configure IP source guard with various other port security features including:

- VLAN tagging (used for voice VLANs)
- GRES (*graceful Routing Engine switchover*)
- *Virtual Chassis* configurations
- Link aggregation groups (LAGs)
- 802.1X user authentication in single supplicant, single-secure supplicant, or multiple supplicant mode.

> **NOTE**: While implementing 801.X user authentication in single-secure supplicant or multiple supplicant mode, use the following configuration guidelines:
>
> - If the 802.1X interface is part of an untagged MAC-based VLAN and you want to enable IP source guard and DHCP snooping on that VLAN, you must enable IP source guard and DHCP snooping on all dynamic VLANs in which the interface has untagged membership. This also applies to IPv6 source guard and DHCPv6 snooping.
> - If the 802.1X interface is part of a tagged MAC-based VLAN and you want to enable IP source guard and DHCP snooping on that VLAN, you must enable IP source guard

and DHCP snooping on all dynamic VLANs in which the interface has tagged membership. This also applies to IPv6 source guard and DHCPv6 snooping.

RELATED DOCUMENTATION

Configuring IP Source Guard (non-ELS)

IN THIS SECTION

You can use the IP source guard access port security feature on EX Series switches to mitigate the effects of source IP address spoofing and source MAC address spoofing. If IP source guard determines that a host connected to an access interface has sent a packet with an invalid source IP address or source MAC address in the packet header, it ensures that the switch does not forward the packet—that is, the packet is discarded.

You enable the IP source guard feature on VLANs. You can enable it on a specific VLAN, on all VLANs, or on a VLAN range.

NOTE: IP source guard applies only to access interfaces and only to untrusted interfaces. If you enable IP source guard on a VLAN that includes trunk interfaces or an interface set to *dhcp-trusted*, the CLI shows an error when you try to commit the configuration.

NOTE: You can use IP source guard together with 802.1X user authentication in single supplicant, single-secure supplicant, or multiple supplicant mode.

While implementing 801.X user authentication in single-secure supplicant or multiple supplicant mode, use the following configuration guidelines:

- If the 802.1X interface is part of an untagged MAC-based VLAN and you want to enable IP source guard and DHCP snooping on that VLAN, you must enable IP source guard and DHCP snooping on all dynamic VLANs in which the interface has untagged membership.

- If the 802.1X interface is part of a tagged MAC-based VLAN and you want to enable IP source guard and DHCP snooping on that VLAN, you must enable IP source guard and DHCP snooping on all dynamic VLANs in which the interface has tagged membership.

Configuring IP Source Guard

Before you configure IP source guard, be sure that you have:

Explicitly enabled DHCP snooping on the specific VLAN or specific VLANs on which you will configure IP source guard. See "Enabling DHCP Snooping (non-ELS)" on page 456. If you configure IP source guard on specific VLANs rather than on all VLANs, you must also enable DHCP snooping explicitly on those VLANs. Otherwise, the default value of no DHCP snooping applies to that VLAN.

To configure IP source guard:

- On a specific VLAN:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan vlan-name ip-source-guard
```

- On all VLANs:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan all ip-source-guard
```

- On a VLAN range:

1. Set the VLAN range:

```
[edit vlans]
user@switch# set vlan-name vlan-range vlan-id-low-vlan-id-high
```

2. Associate an interface with the VLAN-range and set the port mode to **access**:

```
[edit interfaces]
user@switch# set interface-name unit 0 family ethernet-switching port-mode access vlan
members vlan-name
```

3. Enable IP source guard on the VLAN:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan vlan-name ip-source-guard
```

To commit these changes to the active configuration, type the `commit` command at the user prompt.

Configuring IPv6 Source Guard

Before you configure IPv6 source guard, be sure that you have:

- Explicitly enabled DHCPv6 snooping on the specific VLAN or specific VLANs on which you will configure IPv6 source guard. See "Enabling DHCP Snooping (non-ELS)" on page 456. If you configure IPv6 source guard on specific VLANs rather than on all VLANs, you must also enable DHCPv6 snooping explcitly on those VLANs. Otherwise, the default value of no DHCPv6 snooping applies to that VLAN.

- Set the maximum number of IPv6 source guard sessions:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set ipv6-source-guard-sessions max-number maximum-
number
```

> **NOTE**: After setting or changing the maximum number of IPv6 source guard sessions and committing the configuration, you must reboot the switch for the configuration to take effect.

To configure IPv6 source guard:

- On a specific VLAN:

  ```
  [edit ethernet-switching-options secure-access-port]
  user@switch# set vlan vlan-name ipv6-source-guard
  ```

- On all VLANs:

  ```
  [edit ethernet-switching-options secure-access-port]
  user@switch# set vlan all ipv6-source-guard
  ```

- On a VLAN range:

  1. Set the VLAN range):

     ```
     [edit vlans]
     user@switch# set vlan-name vlan-range vlan-id-low-vlan-id-high
     ```

  2. Associate an interface with a VLAN-range and set the port mode to **access**:

     ```
     [edit interfaces]
     user@switch# set interface-name unit 0 family ethernet-switching port-mode access vlan
     members vlan-name
     ```

  3. Enable IPv6 source guard on the VLAN:

     ```
     [edit ethernet-switching-options secure-access-port]
     user@switch# set vlan vlan-name ipv6-source-guard
     ```

To commit these changes to the active configuration, type the `commit` command at the user prompt.

Disabling IP Source Guard

You can disable IP source guard for a specific VLAN after you have enabled the feature for all VLANs, or for all VLANs.

- To disable IP source guard on a specific VLAN:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan vlan-name  no-ip-source-guard
```

- To disable IP source guard on all VLANs:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan all no-ipv6-source-guard
```

NOTE: Replace `no-ip-source-guard` with `no-ipv6-source-guard` to disable IPv6 source guard.

RELATED DOCUMENTATION

Understanding IP Source Guard for Port Security on Switches | 538

Example: Configuring IP Source Guard on a Data VLAN That Shares an Interface with a Voice VLAN | 549

Example: Configuring IP Source Guard with Other EX Series Switch Features to Mitigate Address-Spoofing Attacks on Untrusted Access Interfaces | 560

Understanding IP Source Guard for Port Security on Switches | 538

Configuring IP Source Guard (ELS)

NOTE: This task uses Junos OS with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switching device runs software that does not support ELS, see "Configuring IP Source Guard (non-ELS)" on page 541. For ELS details, see Using the Enhanced Layer 2 Software CLI.

NOTE: On EX9200 switches, IP source guard is not supported in an MC-LAG scenario.

You can use the IP source guard access port security feature to mitigate the effects of source IP address spoofing and source MAC address spoofing. If IP source guard determines that a host connected to an access interface has sent a packet with an invalid source IP address or source MAC address in the packet header, then IP source guard ensures that the switch does not forward the packet—that is, the packet is discarded.

You configure the IP source guard feature on a specific VLAN. When you configure IP source guard on a VLAN, the switch automatically enables DHCP snooping on that VLAN.

IPv6 source guard is supported on switches with support for DHCPv6 snooping. On these switches, configuring IP source guard or IPv6 source guard on a VLAN automatically enables DHCP snooping and DHCPv6 snooping on that VLAN.

Before you can configure IP source guard or IPv6 source guard on a VLAN, you must configure the VLAN. See the documentation that describes setting up basic bridging and a VLAN for your switch.

IP source guard and IPv6 source guard can be applied only to untrusted interfaces. Access interfaces are untrusted by default.

IP source guard and IPv6 source guard can be used together with 802.1X user authentication in single supplicant, single-secure supplicant, or multiple supplicant mode.

To configure IP source guard on a specific VLAN by using the CLI:

```
[edit vlans vlan-name forwarding-options dhcp-security]
user@switch# set  ip-source-guard
```

To configure IPv6 source guard on a specific VLAN by using the CLI:

```
[edit vlans vlan-name forwarding-options dhcp-security]
user@switch# set  ipv6-source-guard
```

RELATED DOCUMENTATION

Example: Configuring IP Source Guard and Dynamic ARP Inspection to Protect the Switch from IP Spoofing and ARP Spoofing | **573**

Example: Configuring IPv6 Source Guard and Neighbor Discovery Inspection to Protect a Switch from IPv6 Address Spoofing | **581**

Understanding IP Source Guard for Port Security on Switches | **538**

Verifying That IP Source Guard Is Working Correctly

IN THIS SECTION

Purpose

Verify that IP source guard is enabled and is mitigating the effects of any source IP spoofing attacks on the EX Series switch.

Action

Display the IP source guard database.

```
user@switch> show ip-source-guard
IP source guard information:
Interface    Tag  IP Address   MAC Address        VLAN

ge-0/0/12.0  0    10.10.10.7   00:30:48:92:A5:9D  vlan100

ge-0/0/13.0  0    10.10.10.9   00:30:48:8D:01:3D  vlan100

ge–0/0/13.0  100  *            *                  voice
```

Meaning

The IP source guard database table contains the VLANs enabled for IP source guard, the untrusted access interfaces on those VLANs, the VLAN 802.1Q tag IDs if there are any, and the IP addresses and MAC addresses that are bound to one another. If a switch interface is associated with multiple VLANs and some of those VLANs are enabled for IP source guard and others are not, the VLANs that are not enabled for IP source guard have a star (*) in the **IP Address** and **MAC Address** fields. See the entry for the **voice** VLAN in the preceding sample output.

RELATED DOCUMENTATION

Configuring IP Source Guard (non-ELS) | **541**

CHAPTER 19

IP Source Guard Examples

IN THIS CHAPTER

Example: Configuring IP Source Guard on a Data VLAN That Shares an Interface with a Voice VLAN

IN THIS SECTION

Ethernet LAN switches are vulnerable to attacks that involve spoofing (forging) of source IP addresses or source MAC addresses. These spoofed packets are sent from hosts connected to untrusted access interfaces on the switch. You can enable the IP source guard port security feature on EX Series switches to mitigate the effects of such attacks. If IP source guard determines that a source IP address and a source MAC address in a binding in an incoming packet are not valid, the switch does not forward the packet.

If two VLANs share an interface, you can configure IP source guard on just one of the VLANs; in this example, you configure IP source guard on an untagged data VLAN but not on the tagged voice VLAN. You can use 802.1X user authentication to validate the device connections on the data VLAN.

This example describes how to configure IP source guard with 802.1X user authentication on a data VLAN, with a voice VLAN on the same interface:

Requirements

This example uses the following hardware and software components:

- One EX Series switch
- Junos OS Release 9.2 or later for EX Series switches
- A DHCP server to provide IP addresses to network devices on the switch
- A RADIUS server to provide 802.1X authentication

Before you configure IP source guard for the data VLANs, be sure you have:

- Connected the DHCP server to the switch.
- Connected the RADIUS server to the switch and configured user authentication on the server. See Example: Connecting a RADIUS Server for 802.1X to an EX Series Switch.
- Configured the VLANs. See Example: Setting Up Bridging with Multiple VLANs for EX Series Switches for detailed information about configuring VLANs.

Overview and Topology

IN THIS SECTION

IP source guard checks the IP source address and MAC source address in a packet sent from a host attached to an untrusted access interface on the switch. If IP source guard determines that the packet header contains an invalid source IP address or source MAC address, it ensures that the switch does not forward the packet—that is, the packet is discarded.

When you configure IP source guard, you enable on it on one or more VLANs. IP source guard applies its checking rules to untrusted access interfaces on those VLANs. By default, on EX Series switches, access interfaces are untrusted and trunk interfaces are trusted. IP source guard does not check packets that have been sent to the switch by devices connected to either trunk interfaces or trusted access interfaces—that is, interfaces configured with **dhcp-trusted** so that a DHCP server can be connected to that interface to provide dynamic IP addresses.

IP source guard obtains information about IP-address/MAC-address/VLAN bindings from the DHCP snooping database. It causes the switch to validate incoming IP packets against the entries in that database.

Topology

The topology for this example includes one EX-3200-24P switch, a PC and an IP phone connected on the same interface, a connection to a DHCP server, and a connection to a RADIUS server for user authentication.

NOTE: The 802.1X user authentication applied in this example is for single supplicants. You can also use IP source guard with 802.1X user authentication for single-secure supplicant or multiple supplicant mode. If you are implementing IP source guard with 802.1X authentication in single-secure supplicant or multiple supplicant mode, you must use the following configuration guidelines:

- If the 802.1X interface is part of an untagged MAC-based VLAN and you want to enable IP source guard and DHCP snooping on that VLAN, you must enable IP source guard and DHCP snooping on all dynamic VLANs in which the interface has untagged membership.

- If the 802.1X interface is part of a tagged MAC-based VLAN and you want to enable IP source guard and DHCP snooping on that VLAN, you must enable IP source guard and DHCP snooping on all dynamic VLANs in which the interface has tagged membership.

TIP: You can set the **ip-source-guard** flag in the `traceoptions (Access Port Security)` statement for debugging purposes.

This example shows how to configure a static IP address to be added to the DHCP snooping database.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure IP source guard on a data VLAN, copy the following commands and paste them into the switch terminal window:

```
set ethernet-switching-options voip interface ge-0/0/14.0 vlan voice
set ethernet-switching-options secure-access-port interface ge-0/0/24.0 dhcp-trusted
set ethernet-switching-options secure-access-port interface ge-0/0/14 static-ip 10.1.1.1 mac
00:11:11:11:11:11 vlan data
set ethernet-switching-options secure-access-port vlan data examine-dhcp
set ethernet-switching-options secure-access-port vlan data ip-source-guard
set interfaces ge-0/0/24 unit 0 family ethernet-switching vlan members data
set vlans voice vlan-id 100
set protocols lldp-med interface ge-0/0/14.0
set protocols dot1x authenticator authentication-profile-name profile52
set protocols dot1x authenticator interface ge-0/0/14.0 supplicant single
```

Step-by-Step Procedure

To configure IP source guard on the data VLAN:

1. Configure the VoIP interface:

```
[edit ethernet-switching-options]
user@switch# set voip interface ge-0/0/14.0 vlan voice
```

2. Configure the interface on which the DHCP server is connected to the switch as a trusted interface and add that interface to the data VLAN:

```
[edit ethernet-switching-options]
user@switch# set secure-access-port interface ge-0/0/24.0 dhcp-
trusted
[edit interfaces]
user@switch# set ge-0/0/24 unit 0 family ethernet-switching vlan members data
```

3. Configure a static IP address on an interface on the data VLAN (optional)

```
[edit ethernet-switching-options]
user@switch# set secure-access-port interface ge-0/0/14 static-ip 10.1.1.1 mac
00:11:11:11:11:11 vlan data
```

4. Configure DHCP snooping and IP source guard on the data VLAN:

```
[edit ethernet-switching-options]
user@switch# set secure-access-port vlan data examine-dhcp
user@switch# set secure-access-port vlan data ip-source-guard
```

5. Configure 802.1X user authentication and LLDP-MED on the interface that is shared by the data VLAN and the voice VLAN:

```
[edit protocols]
user@switch# set lldp-med interface ge-0/0/14.0
user@switch# set dot1x authenticator authentication-profile-name profile52
user@switch# set dot1x authenticator interface ge-0/0/14.0 supplicant single
```

6. Set the VLAN ID for the voice VLAN:

```
[edit vlans]
user@switch# set voice vlan-id 100
```

Results

Check the results of the configuration:

```
[edit ethernet-switching-options]
user@switch# show
voip {
    interface ge-0/0/14.0 {
        vlan voice;
    }
}
secure-access-port {
    interface ge-0/0/14.0 {
        static-ip 10.1.1.1 vlan data mac 00:11:11:11:11:11;
    }
    interface ge-0/0/24.0 {
        dhcp-trusted;
    }
    vlan data {
        examine-dhcp;
        ip-source-guard;
    }
}
```

```
[edit interfaces]
ge-0/0/24 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members data;
            }
        }
    }
}
```

```
[edit vlans]
voice {
    vlan-id 100;
}
```

```
[edit protocols]
lldp-med {
    interface ge-0/0/14.0;
}
dot1x {
    authenticator {
        authentication-profile-name profile52;
        interface {
            ge-0/0/14.0 {
                supplicant single;
            }
        }
    }
}
```

TIP: If you wanted to configure IP source guard on the voice VLAN as well as on the data VLAN, you would configure DHCP snooping and IP source guard exactly as you did for the data VLAN. The configuration result for the voice VLAN under **secure-access-port** would look like this:

```
secure-access-port {
    vlan voice {
        examine-dhcp;
        ip-source-guard;
    }
}
```

Verification

IN THIS SECTION

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

Verifying That 802.1X User Authentication Is Working on the Interface

Purpose

Verify the 802.1X configuration on interface **ge-0/0/14**.

Action

Verify the 802.1X configuration with the operational mode command `show dot1x interface`:

```
user@switch> show dot1x interface ge-0/0/14.0 detail
ge-0/0/14.0
  Role: Authenticator
  Administrative state: Auto
  Supplicant mode: Single
  Number of retries: 3
  Quiet period: 60 seconds
  Transmit period: 30 seconds
  Mac Radius: Disabled
  Mac Radius Restrict: Disabled
  Reauthentication: Enabled
  Configured Reauthentication interval: 3600 seconds
  Supplicant timeout: 30 seconds
  Server timeout: 30 seconds
  Maximum EAPOL requests: 2
  Guest VLAN member: <not configured>
  Number of connected supplicants: 1
    Supplicant: user101, 00:04:0f:fd:ac:fe
      Operational state: Authenticated
      Authentication method: Radius
      Authenticated VLAN: vo11
      Dynamic Filter: <not configured>
      Session Reauth interval: 60 seconds
      Reauthentication due in 50 seconds
```

Meaning

The **Supplicant mode** output field displays the configured administrative mode for each interface. Interface **ge-0/0/14.0** displays **Single** supplicant mode.

Verifying the VLAN Association with the Interface

Purpose

Display the interface state and VLAN membership.

Action

```
user@switch>                              show ethernet-switching
interfaces
 Ethernet-switching table: 0 entries, 0 learned

user@switch> show ethernet-switching interfaces
Interface   State    VLAN members        Blocking
ge-0/0/0.0  down     default             unblocked
ge-0/0/1.0  down     employee            unblocked
ge-0/0/2.0  down     employee            unblocked
ge-0/0/12.0 down     default             unblocked
ge-0/0/13.0 down     default             unblocked
ge-0/0/13.0 down     vlan100             unblocked
ge-0/0/14.0 up       voice               unblocked
                     data                unblocked
ge-0/0/17.0 down     employee            unblocked
ge-0/0/23.0 down     default             unblocked
ge-0/0/24.0 down     data                unblocked
                     employee            unblocked
                     vlan100             unblocked
                     voice               unblocked
```

Meaning

The field **VLAN members** shows that the **ge-0/0/14.0** interface supports both the **data** VLAN and the **voice** VLAN. The **State** field shows that the interface is up.

Verifying That DHCP Snooping and IP Source Guard Are Working on the Data VLAN

Purpose

Verify that DHCP snooping and IP source guard are enabled and working on the data VLAN.

Action

Send some DHCP requests from network devices (here they are DHCP clients) connected to the switch.

Display the DHCP snooping information when the interface on which the DHCP server connects to the switch is trusted. The following output results when requests are sent from the MAC addresses and the server has provided the IP addresses and leases:

```
user@switch>                                  show dhcp snooping binding
DHCP Snooping Information:
MAC address        IP address  Lease (seconds) Type     VLAN      Interface



00:05:85:3A:82:77  192.0.2.17  600             dynamic  employee  ge-0/0/1.0


00:05:85:3A:82:79  192.0.2.18  653             dynamic  employee  ge-0/0/1.0


00:05:85:3A:82:80  192.0.2.19  720             dynamic  employee  ge-0/0/2.0


00:05:85:3A:82:81  192.0.2.20  932             dynamic  employee  ge-0/0/2.0



                        00:30:48:92:A5:9D  10.10.10.7  720             dynamic  vlan100
ge-0/0/13.0


00:30:48:8D:01:3D  10.10.10.9  720             dynamic  data      ge-0/0/14.0


00:30:48:8D:01:5D  10.10.10.8  1230            dynamic  voice     ge-0/0/14.0


00:11:11:11:11:11  10.1.1.1    –               static   data      ge-0/0/14.0


00:05:85:27:32:88  192.0.2.22  –               static   employee  ge-0/0/17.0


00:05:85:27:32:89  192.0.2.23  –               static   employee  ge-0/0/17.0


00:05:85:27:32:90  192.0.2.27  –               static   employee  ge-0/0/17.0
```

View the IP source guard information for the data VLAN.

```
user@switch>                              show ip-source-
guard
IP source guard information:
Interface    Tag  IP Address   MAC Address        VLAN

ge-0/0/13.0  0    10.10.10.7   00:30:48:92:A5:9D  vlan100

ge-0/0/14.0  0    10.10.10.9   00:30:48:8D:01:3D  data

ge-0/0/14.0  0    10.1.1.1     00:11:11:11:11:11  data

ge-0/0/13.0  100  *            *                  voice
```

Meaning

When the interface on which the DHCP server connects to the switch has been set to trusted, the output (see the preceding sample output for `show dhcp snooping binding`) shows, for each MAC address, the assigned IP address and lease time—that is, the time, in seconds, remaining before the lease expires. Static IP addresses have no assigned lease time. Statically configured entries never expire.

The IP source guard database table contains the VLANs enabled for IP source guard, the untrusted access interfaces on those VLANs, the VLAN 802.1Q tag IDs if there are any, and the IP addresses and MAC addresses that are bound to one another. If a switch interface is associated with multiple VLANs and some of those VLANs are enabled for IP source guard and others are not, the VLANs that are not enabled for IP source guard have a star (*) in the **IP Address** and **MAC Address** fields. See the entry for the **voice** VLAN in the preceding sample output.

RELATED DOCUMENTATION

Example: Configuring IP Source Guard with Other EX Series Switch Features to Mitigate Address-Spoofing Attacks on Untrusted Access Interfaces | **560**

Example: Configuring Port Security (non-ELS) | **15**

Example: Setting Up VoIP with 802.1X and LLDP-MED on an EX Series Switch

Configuring IP Source Guard (non-ELS) | **541**

Example: Configuring IP Source Guard with Other EX Series Switch Features to Mitigate Address-Spoofing Attacks on Untrusted Access Interfaces

IN THIS SECTION

Ethernet LAN switches are vulnerable to attacks that involve spoofing (forging) of source IP addresses or source MAC addresses. These spoofed packets are sent from hosts connected to untrusted access interfaces on the switch. You can enable the IP source guard port security feature on EX Series switches to mitigate the effects of such attacks. If IP source guard determines that a source IP address and a source MAC address in a binding in an incoming packet are not valid, the switch does not forward the packet.

You can use IP source guard in combination with other EX Series switch features to mitigate address-spoofing attacks on untrusted access interfaces. This example shows two configuration scenarios:

Requirements

This example uses the following hardware and software components:

- An EX Series switch
- Junos OS Release 9.2 or later for EX Series switches
- A DHCP server to provide IP addresses to network devices on the switch
- A RADIUS server to provide 802.1X authentication

Before you configure IP source guard for the scenarios related in this example, be sure you have:

- Connected the DHCP server to the switch.

- Connected the RADIUS server to the switch and configured user authentication on the RADIUS server. See Example: Connecting a RADIUS Server for 802.1X to an EX Series Switch.
- Configured VLANs on the switch. In this example, we have two VLANs, which are named `DATA` and `GUEST`. The `DATA` VLAN is configured with `vlan-id 300`. The `GUEST` VLAN (which functions as the guest VLAN) is configured with `vlan-id 100`. See Example: Setting Up Bridging with Multiple VLANs for EX Series Switches for detailed information about configuring VLANs.

Overview and Topology

IN THIS SECTION

IP source guard checks the IP source address and MAC source address in a packet sent from a host attached to an untrusted access interface on the switch. If IP source guard determines that the packet header contains an invalid source IP address or source MAC address, it ensures that the switch does not forward the packet—that is, the packet is discarded.

When you configure IP source guard, you enable it on one or more VLANs. IP source guard applies its checking rules to untrusted access interfaces on those VLANs. By default, on EX Series switches, access interfaces are untrusted and trunk interfaces are trusted. IP source guard does not check packets that have been sent to the switch by devices connected to either trunk interfaces or trusted access interfaces—that is, interfaces configured with `dhcp-trusted`. A DHCP server can be connected to a `dhcp-trusted` interface to provide dynamic IP addresses.

IP source guard obtains information about IP-addresses, MAC-addresses, or VLAN bindings from the DHCP snooping database, which enables the switch to validate incoming IP packets against the entries in that database.

Topology

The topology for this example includes an EX Series switch, which is connected to both a DHCP server and to a RADIUS server.

NOTE: The 802.1X user authentication applied in this example is for single-supplicant mode.

You can use IP source guard with 802.1X user authentication for single-secure supplicant or multiple supplicant mode. If you are implementing IP source guard with 802.1X authentication in single-secure supplicant or multiple supplicant mode, you must use the following configuration guidelines:

- If the 802.1X interface is part of an untagged MAC-based VLAN and you want to enable IP source guard and DHCP snooping on that VLAN, you must enable IP source guard and DHCP snooping on all dynamic VLANs in which the interface has untagged membership.
- If the 802.1X interface is part of a tagged MAC-based VLAN and you want to enable IP source guard and DHCP snooping on that VLAN, you must enable IP source guard and DHCP snooping on all dynamic VLANs in which the interface has tagged membership.

In the first configuration example, two clients (network devices) are connected to an access switch. You configure IP source guard and 802.1X user authentication, in combination with two access port security features: DHCP snooping and dynamic ARP inspection (DAI). This setup is designed to protect the switch from IP attacks such as *ping of death* attacks, DHCP starvation, and ARP spoofing.

In the second configuration example, the switch is configured for 802.1X user authentication. If the client fails authentication, the switch redirects the client to a guest VLAN that allows this client to access a set of restricted network features. You configure IP source guard on the guest VLAN to mitigate effects of source IP spoofing.

TIP: You can set the `ip-source-guard` flag in the `traceoptions` statement for debugging purposes.

Configuring IP Source Guard with 802.1X Authentication, DHCP Snooping, and Dynamic ARP Inspection

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure IP source guard with 802.1X authentication and with other access port security features, copy the following commands and paste them into the switch terminal window:

```
[edit]
 set ethernet-switching-options secure-access-port interface ge-0/0/24 dhcp-trusted
 set ethernet-switching-options secure-access-port vlan DATA examine-dhcp
 set ethernet-switching-options secure-access-port vlan DATA arp-inspection
 set ethernet-switching-options secure-access-port vlan DATA ip-source-guard
 set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members DATA
 set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members DATA
 set interfaces ge-0/0/24 unit 0 family ethernet-switching vlan members DATA
 set protocols lldp-med interface ge-0/0/0.0
 set protocols dot1x authenticator authentication-profile-name profile52
 set protocols dot1x authenticator interface ge-0/0/0.0 supplicant single
 set protocols lldp-med interface ge-0/0/1.0
 set protocols dot1x authenticator interface ge-0/0/1.0 supplicant single
```

Step-by-Step Procedure

To configure IP source guard with 802.1X authentication and various port security features:

1. Configure the interface on which the DHCP server is connected to the switch as a trusted interface and add that interface to the DATA VLAN:

   ```
   [edit ethernet-switching-options]
   user@switch# set secure-access-port interface ge-0/0/24 dhcp-
   trusted
   user@switch# set set ge-0/0/24 unit 0 family ethernet-switching vlan members DATA
   ```

2. Associate two other access interfaces (untrusted) with the DATA VLAN:

   ```
   [edit interfaces]
   user@switch# set ge-0/0/0 unit 0 family ethernet-switching vlan members DATA
   user@switch# set ge-0/0/1 unit 0 family ethernet-switching vlan members DATA
   ```

3. Configure 802.1X user authentication and LLDP-MED on the two interfaces that you associated with the DATA VLAN:

   ```
   [edit protocols]
   user@switch# set lldp-med interface ge-0/0/0.0
   user@switch# set dot1x authenticator authentication-profile-name profile52
   user@switch# set dot1x authenticator interface ge-0/0/0.0 supplicant single
   user@switch# set lldp-med interface ge-0/0/1.0
   user@switch# set dot1x authenticator interface ge-0/0/1.0 supplicant single
   ```

4. Configure three access port security features—DHCP snooping, dynamic ARP inspection (DAI), and IP source guard—on the `DATA` VLAN:

   ```
   [edit ethernet-switching-options]
   user@switch# set secure-access-port vlan DATA examine-dhcp
   user@switch# set secure-access-port vlan DATA arp-inspection
   user@switch# set secure-access-port vlan DATA ip-source-guard
   ```

Results

Check the results of the configuration:

```
[edit ethernet-switching-options]
secure-access-port {
    interface ge-0/0/24.0 {
        dhcp-trusted;
    }
    vlan DATA {
        arp-inspection;
        examine-dhcp;
        ip-source-guard;
    }
}
```

```
[edit interfaces]
ge-0/0/0 {
    unit 0 {
```

```
        family ethernet-switching {
            vlan {
                members DATA;
            }
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members DATA;
            }
        }
    }
}
ge-0/0/24 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members DATA;
            }
        }
    }
}
```

```
[edit protocols]
lldp-med {
    interface ge-0/0/0.0;
    interface ge-0/0/1.0;
}
dot1x {
    authenticator {
        authentication-profile-name profile52;
    }
    interface {
        ge-0/0/0.0 {
            supplicant single;
        }
        ge-0/0/1.0 {
            supplicant single;
        }
    }
}
```

Configuring IP Source Guard on a Guest VLAN

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure IP source guard on a guest VLAN, copy the following commands and paste them into the switch terminal window:

```
 [edit]
 set ethernet-switching-options secure-access-port interface ge-0/0/24 dhcp-trusted
 set interfaces ge-0/0/24 unit 0 family ethernet-switching vlan members GUEST
 set ethernet-switching-options secure-access-port vlan GUEST examine-dhcp
 set ethernet-switching-options secure-access-port vlan GUEST ip-source-guard
 set ethernet-switching-options secure-access-port interface ge-0/0/0 static-ip 10.1.1.1 mac
00:11:11:11:11:11 vlan GUEST
 set ethernet-switching-options secure-access-port interface ge-0/0/1 static-ip 10.1.1.2 mac
00:22:22:22:22:22 vlan GUEST
 set interfaces ge-0/0/0 unit 0 family ethernet-switching port-mode access
 set interfaces ge-0/0/1 unit 0 family ethernet-switching port-mode access
 set protocols dot1x authenticator authentication-profile-name profile52
 set protocols dot1x authenticator interface ge-0/0/0 supplicant single
 set protocols dot1x authenticator interface ge-0/0/0 guest-vlan GUEST
 set protocols dot1x authenticator interface ge-0/0/0 supplicant-timeout 2
 set protocols dot1x authenticator interface ge-0/0/1 supplicant single
 set protocols dot1x authenticator interface ge-0/0/1 guest-vlan GUEST
 set protocols dot1x authenticator interface ge-0/0/1 supplicant-timeout 2
```

Step-by-Step Procedure

To configure IP source guard on a guest VLAN:

1. Configure the interface on which the DHCP server is connected to the switch as a trusted interface and add that interface to the GUEST VLAN:

```
[edit ethernet-switching-options]
user@switch# set secure-access-port interface ge-0/0/24 dhcp-
trusted
user@switch# set ge-0/0/24 unit 0 family ethernet-switching vlan members GUEST
```

2. Configure two interfaces for the access port mode:

```
[edit interfaces]
user@switch# set ge-0/0/0 unit 0 family ethernet-switching port-mode access
user@switch# set ge-0/0/1 unit 0 family ethernet-switching port-mode access
```

3. Configure DHCP snooping and IP source guard on the GUEST VLAN:

```
[edit ethernet-switching-options]
user@switch# set secure-access-port vlan GUEST examine-dhcp
user@switch# set secure-access-port vlan GUEST ip-source-guard
```

4. Configure a static IP address on each of two (untrusted) interfaces on the GUEST VLAN (optional):

```
[edit ethernet-switching-options]
user@switch# set secure-access-port interface ge-0/0/0 static-ip 10.1.1.1 mac
00:11:11:11:11:11 vlan GUEST
```

```
[edit ethernet-switching-options]
user@switch# set secure-access-port interface ge-0/0/1  static-ip 10.1.1.2 mac
00:22:22:22:22:22 vlan GUEST
```

5. Configure 802.1X user authentication:

```
[edit protocols]
user@switch# set dot1x authenticator authentication-profile-name profile52
user@switch# set dot1x authenticator interface ge-0/0/0 supplicant single
user@switch# set dot1x authenticator interface ge-0/0/1 supplicant single
user@switch# set dot1x authenticator interface ge-0/0/0 supplicant-timeout 2
user@switch# set dot1x authenticator interface ge-0/0/1 supplicant-timeout 2
```

Results

Check the results of the configuration:

```
[edit protocols]
    dot1x {
        authenticator {
            authentication-profile-name profile52;
        }
        interface {
            ge-0/0/0.0 {
                guest-vlan GUEST;
                supplicant single;
                supplicant-timeout 2;
            }
            ge-0/0/1.0 {
                guest-vlan GUEST;
                supplicant single;
                supplicant-timeout 2;
            }
        }
    }
}
```

```
[edit vlans]
GUEST {
    vlan-id 100;
}
```

```
[edit interfaces]
ge-0/0/0 {
    unit 0 {
        family ethernet-switching {
            port-mode access;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family ethernet-switching {
            port-mode access;
        }
    }
}
ge-0/0/24 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members GUEST;
            }
        }
    }
}
```

```
[edit ethernet-switching-options]
secure-access-port {
    interface ge-0/0/0.0 {
        static-ip 10.1.1.1 vlan GUEST mac 00:11:11:11:11:11;
    }
    interface ge-0/0/1.0 {
        static-ip 10.1.1.2 vlan GUEST mac 00:22:22:22:22:22;
    }
    interface ge-0/0/24.0 {
        dhcp-trusted;
    }
    vlan GUEST {
        examine-dhcp;
        ip-source-guard;
    }
}
```

Verification

IN THIS SECTION

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

Verifying That 802.1X User Authentication Is Working on the Interface

Purpose

Verify that the 802.1X configuration is working on the interface.

Action

```
user@switch> show dot1x interface ge/0/0/0.0 detail
ge-0/0/0.0
  Role: Authenticator
  Administrative state: Auto
  Supplicant mode: Single
  Number of retries: 2
  Quiet period: 30 seconds
  Transmit period: 15 seconds
  Mac Radius: Disabled
  Mac Radius Restrict: Disabled
  Reauthentication: Enabled
  Configured Reauthentication interval: 3600 seconds
  Supplicant timeout: 2 seconds
  Server timeout: 30 seconds
  Maximum EAPOL requests: 1
  Guest VLAN member: GUEST
  Number of connected supplicants: 1
    Supplicant: md5user01, 00:30:48:90:53:B7
      Operational state: Authenticated
      Backend Authentication state: Idle
      Authentcation method: Radius
      Authenticated VLAN: DATA
      Session Reauth interval: 3600 seconds
      Reauthentication due in 3581 seconds
```

Meaning

The `Supplicant mode` field displays the configured administrative mode for each interface. The `Guest VLAN member` field displays the VLAN to which a supplicant is connected when the supplicant is authenticated using a guest VLAN. The `Authenticated VLAN` field displays the VLAN to which the supplicant is connected.

Verifying the VLAN Association with the Interface

Purpose

Verify interface states and VLAN memberships.

Action

```
user@switch> show ethernet-switching interfaces

Interface    State  VLAN members        Tag   Tagging  Blocking
ge-0/0/0.0  up     DATA                101   untagged unblocked
ge-0/0/1.0  up     DATA                101   untagged unblocked
ge-0/0/24   up     DATA                101   untagged unblocked
```

Meaning

The `VLAN members` field shows the associations between VLANs and interfaces. The `State` field shows whether the interfaces are up or down.

For the guest VLAN configuration, the interface is associated with the guest VLAN if and when the supplicant fails 802.1X user authentication.

Verifying That DHCP Snooping Is Working on the VLAN

Purpose

Verify that DHCP snooping is enabled and working on the VLAN. Send some DHCP requests from network devices (DHCP clients) connected to the switch.

Action

```
user@switch> show dhcp snooping binding

DHCP Snooping Information:
MAC address        IP address        Lease (seconds)  Type     VLAN    Interface
00:30:48:90:53:B7  192.0.2.1                 86392  dynamic    DATA    ge-0/0/24.0
```

Meaning

When the interface on which the DHCP server connects to the switch has been set to `dhcp-trusted`, the output shows for each MAC address, the assigned IP address and lease time—that is, the time, in seconds, remaining before the lease expires. Static IP addresses have no assigned lease time. Statically configured entries never expire.

Verifying That IP Source Guard Is Working on the VLAN

Purpose

Verify that IP source guard is enabled and working on the VLAN.

Action

```
user@switch> show ip-source-guard

IP source guard information:
Interface    Tag   IP Address       MAC Address         VLAN
ge-0/0/0.0  0     192.0.2.2        00:30:48:90:63:B7   DATA
ge-0/0/1.0  0     192.0.2.3        00:30:48:90:73:B7   DATA
```

Meaning

The IP source guard database table contains the VLANs for which IP source guard is enabled, the untrusted access interfaces on those VLANs, the VLAN 802.1Q tag IDs if there are any, and the IP addresses and MAC addresses that are bound to one another. If a switch interface is associated with multiple VLANs and some of those VLANs have IP source guard enabled (or configured) while others do not have IP source guard enabled, the VLANs that do not have IP source guard enabled have a star (*) in the `IP Address` and `MAC Address` fields.

RELATED DOCUMENTATION

Example: Configuring Port Security (non-ELS) | 15

Example: Setting Up VoIP with 802.1X and LLDP-MED on an EX Series Switch

Example: Configuring IP Source Guard on a Data VLAN That Shares an Interface with a Voice VLAN | 549

Configuring IP Source Guard (non-ELS) | 541

Example: Configuring IP Source Guard and Dynamic ARP Inspection to Protect the Switch from IP Spoofing and ARP Spoofing

IN THIS SECTION

NOTE: This example uses Junos OS with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Example: Protecting Against ARP Spoofing Attacks" on page 482. For ELS details, see Using the Enhanced Layer 2 Software CLI.

NOTE: On EX9200 switches, DHCP snooping, DAI, and IP source guard are not supported in an MC-LAG scenario.

This example describes how to enable IP source guard and Dynamic ARP Inspection (DAI) on a specified VLAN to protect the switch against spoofed IP/MAC addresses and ARP spoofing attacks. When you enable either IP source guard or DAI, the configuration automatically enables DHCP snooping for the same VLAN.

Requirements

This example uses the following hardware and software components:

NOTE: This example also applies to QFX5100, QFX5110, and QFX5200 switches.

- One EX4300 switch or EX9200 switch
- Junos OS Release 13.2X50-D10 or later for EX Series switches
- A DHCP server to provide IP addresses to network devices on the switch

Before you configure IP source guard to prevent IP/MAC spoofing or DAI to mitigate ARP spoofing attacks, be sure you have:

- Connected the DHCP server to the switch.
- Configured the VLAN to which you are adding DHCP security features.

Overview and Topology

IN THIS SECTION

Ethernet LAN switches are vulnerable to attacks on security that involve spoofing (forging) of source MAC addresses or source IP addresses. These spoofed packets are sent from hosts connected to untrusted access interfaces on the switch. These spoofed packets are sent from hosts connected to untrusted access interfaces on the switch. IP source guard checks the IP source address and MAC source address in a packet sent from a host attached to an untrusted access interface on the switch against entries stored in the DHCP snooping database. If IP source guard determines that the packet header contains an invalid source IP address or source MAC address, it ensures that the switch does not forward the packet—that is, the packet is discarded.

Another type of security attack is ARP spoofing (also known as ARP poisoning or ARP cache poisoning). ARP-spoofing is a way to initiate man-in-the-middle attacks. The attacker sends an ARP packet that spoofs the MAC address of another device on the LAN. Instead of the switch sending traffic to the proper network device, it sends it to the device with the spoofed address that is impersonating the proper device. If the impersonating device is the attacker's machine, the attacker receives all the traffic from the switch that should have gone to another device. The result is that traffic from the switch is misdirected and cannot reach its proper destination.

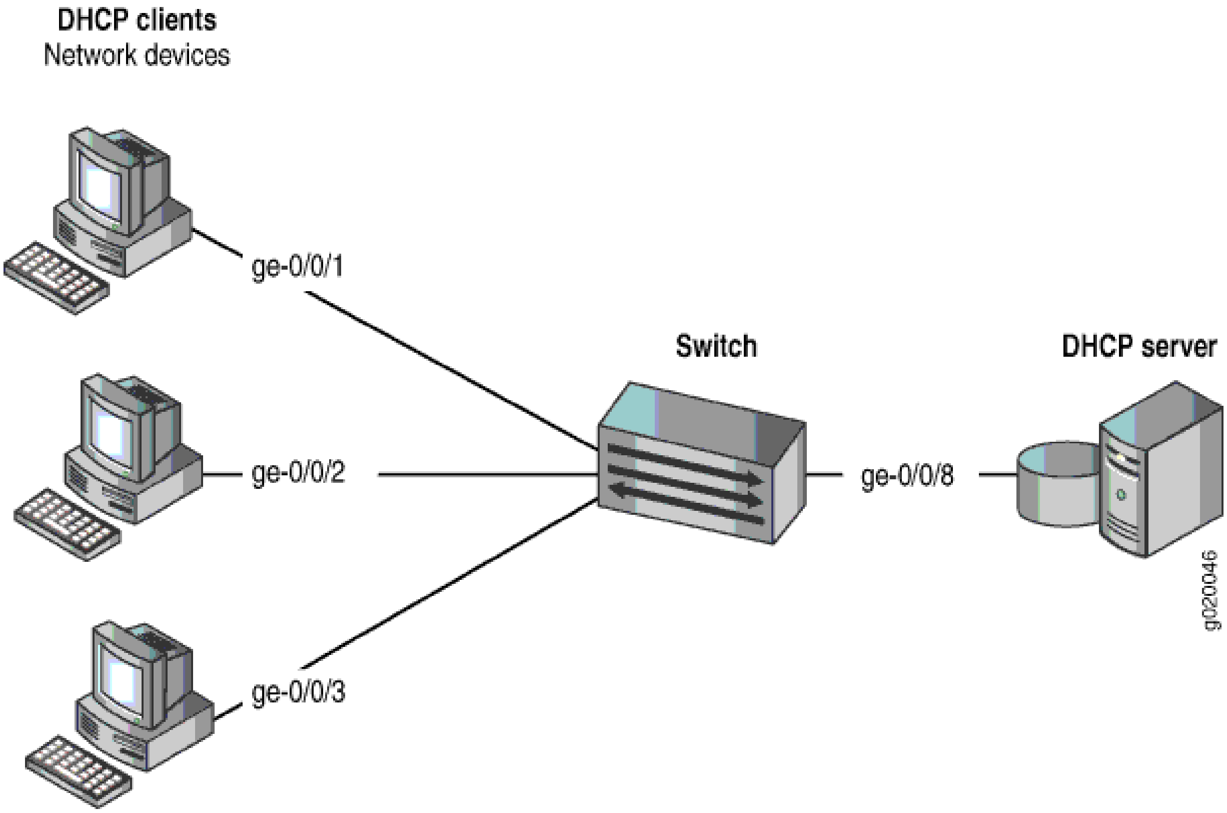
NOTE: When dynamic ARP inspection (DAI) is enabled, the switch logs the number of invalid ARP packets that it receives on each interface, along with the sender's IP and MAC addresses. You can use these log messages to discover ARP spoofing on the network.

This example shows how to configure these important port security features on a switch that is connected to a DHCP server. The setup for this example includes the VLAN `employee-vlan` on the switch. Figure 33 on page 576 illustrates the topology for this example.

NOTE: The trunk interface connecting to the DHCP server interface is a trusted port by default. If you attach a DHCP server to an access port, you must configure the port as trusted. Before you do so, ensure that the server is physically secure—that is, that access to the server is monitored and controlled. For more information on trusted and untrusted ports for DHCP, see "Understanding and Using Trusted DHCP Servers" on page 415.

Topology

Figure 33: Network Topology for Basic Port Security



The components of the topology for this example are shown in Table 27 on page 576.

Table 27: Components of the Port Security Topology

Properties	Settings
Switch hardware	One EX4300 or EX9200 switch
VLAN name and ID	`employee-vlan`, tag `20`

VLAN subnets	`192.0.2.16/28`
	`192.0.2.17` through `192.0.2.30`
	`192.0.2.31` is the subnet's broadcast address

Table 27: Components of the Port Security Topology *(Continued)*

Properties	Settings
Interfaces in employee-vlan	ge-0/0/1, ge-0/0/2, ge-0/0/3, ge-0/0/8
Interface connecting to DHCP server	ge-0/0/8

In this example, the switch has already been configured as follows:

- All access ports are untrusted, which is the default setting.
- The trunk port (ge-0/0/8) is trusted, which is the default setting.
- The VLAN (employee-vlan) has been configured to include the specified interfaces.

Configuration

IN THIS SECTION

To configure IP source guard and DAI (and thereby, also automatically configure DHCP snooping) to protect the switch against IP spoofing and ARP attacks:

Procedure

CLI Quick Configuration

To quickly configure IP source guard and DAI (and thereby, also automatically configure DHCP snooping), copy the following commands and paste them into the switch terminal window:

```
[edit]
set vlans employee-vlan forwarding-options dhcp-security ip-source-guard
set vlans employee-vlan forwarding-options dhcp-security arp-inspection
```

Step-by-Step Procedure

Configure IP source guard and DAI (and thereby, also automatically configure DHCP snooping) on the VLAN:

1. Configure IP source guard on the VLAN:

```
[edit vlans employee-vlan forwarding-options dhcp-security]
user@switch# set ip-source-guard
```

2. Enable DAI on the VLAN:

```
[edit vlans employee-vlan forwarding-options dhcp-security]
user@switch# set arp-inspection
```

Results

Check the results of the configuration:

```
user@switch> show vlans employee-vlan forwarding-options
employee-vlan {
    forwarding-options {
        dhcp-security {
            arp-inspection;
            ip-source-guard;
        }
    }
}
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying That DHCP Snooping Is Working Correctly on the Switch

Purpose

Verify that DHCP snooping is working on the switch.

Action

Send some DHCP requests from network devices (here they are DHCP clients) connected to the switch.

Display the DHCP snooping information when the port on which the DHCP server connects to the switch is trusted. The following output results when requests are sent from the MAC addresses and the server has provided the IP addresses and leases:

```
user@switch> show dhcp-security binding


IP Address    MAC Address            Vlan      Expires         State      Interface

192.0.2.17    00:05:85:3A:82:77    employee-vlan   86265        BOUND            ge-0/0/1.0

192.0.2.18    00:05:85:3A:82:79    employee-vlan   86265        BOUND            ge-0/0/1.0

192.0.2.19    00:05:85:3A:82:80    employee-vlan   86287        BOUND           ge-0/0/2.0

192.0.2.20    00:05:85:3A:82:81    employee-vlan   86287        BOUND           ge-0/0/2.0

192.0.2.21    00:05:85:3A:82:83    employee-vlan   86287        BOUND           ge-0/0/2.0

192.0.2.22    00:05:85:27:32:88    employee-vlan   86254        BOUND           ge-0/0/3.0
```

Meaning

When the interface on which the DHCP server connects to the switch has been set to trusted, the output (see preceding sample) shows, for the assigned IP address, the device's MAC address, the VLAN name, and the time, in seconds, remaining before the lease expires.

Verifying That IP Source Guard is Working on the VLAN

Purpose

Verify that IP source guard is enabled and working on the VLAN.

Action

Send some DHCP requests from network devices (here they are DHCP clients) connected to the switch. View the IP source guard information for the data VLAN.

```
user@switch>                              show dhcp-security binding ip-source-
guard


IP Address    MAC Address           Vlan      Expires         State      Interface

192.0.2.17    00:05:85:3A:82:77   employee-vlan   86265        BOUND            ge-0/0/1.0

192.0.2.18    00:05:85:3A:82:79   employee-vlan   86265        BOUND            ge-0/0/1.0

192.0.2.19    00:05:85:3A:82:80   employee-vlan   86287        BOUND           ge-0/0/2.0

192.0.2.20    00:05:85:3A:82:81   employee-vlan   86287        BOUND           ge-0/0/2.0

192.0.2.21    00:05:85:3A:82:83   employee-vlan   86287        BOUND           ge-0/0/2.0

192.0.2.22    00:05:85:27:32:88   employee-vlan   86254        BOUND           ge-0/0/3.0
```

Meaning

The IP source guard database table contains the VLANs enabled for IP source guard.

Verifying That DAI Is Working Correctly on the Switch

Purpose

Verify that DAI is working on the switch.

Action

Send some ARP requests from network devices connected to the switch.

Display the DAI information:

```
user@switch> show dhcp-security arp inspection statistics
ARP inspection statistics:

Interface         Packets received  ARP inspection pass   ARP inspection failed
ge-0/0/1.0                       7                    5                       2
ge-0/0/2.0                      10                   10                       0
ge-0/0/3.0                      12                   12                       0
```

Meaning

The sample output shows the number of ARP packets received and inspected per interface, with a listing of how many packets passed and how many failed the inspection on each interface. The switch compares the ARP requests and replies against the entries in the DHCP snooping database. If a MAC address or IP address in the ARP packet does not match a valid entry in the database, the packet is dropped.

RELATED DOCUMENTATION

Configuring IP Source Guard (ELS) | 545

Enabling Dynamic ARP Inspection (ELS) | 531

Enabling Dynamic ARP Inspection (J-Web Procedure)

Example: Configuring IPv6 Source Guard and Neighbor Discovery Inspection to Protect a Switch from IPv6 Address Spoofing

IN THIS SECTION

NOTE: This example uses Junos OS with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Example: Protecting Against ARP Spoofing Attacks" on page 482. For ELS details, see Using the Enhanced Layer 2 Software CLI.

This example describes how to enable IPv6 source guard and neighbor discovery inspection on a specified VLAN to protect the switch against IPv6 address spoofing attacks. When you enable either IPv6 source guard or neighbor discovery inspection, DHCPv6 snooping is automatically enabled on the same VLAN.

Requirements

This example uses the following hardware and software components:

NOTE: This example also applies to QFX5100, QFX5110, and QFX5200 switches.

- One EX Series switch that supports the Enhanced Layer 2 Software configuration style.
- Junos OS Release 13.2X51-D20 or later for EX Series switches
- A DHCPv6 server to provide IPv6 addresses to network devices on the switch

Before you configure IPv6 source guard and neighbor discovery inspection to prevent IPv6 address spoofing attacks, be sure you have:

- Connected the DHCPv6 server to the switch.
- Configured the VLAN to which you are adding DHCPv6 security features. See the documentation that describes setting up basic bridging and a VLAN for your switch.

Overview and Topology

IN THIS SECTION

Ethernet LAN switches are vulnerable to attacks on security that involve spoofing (forging) of source MAC addresses or source IPv6 addresses. These spoofed packets are sent from hosts connected to untrusted access interfaces on the switch. For more information on IPv6 address spoofing attacks, see "IPv6 Neighbor Discovery Inspection" on page 605.

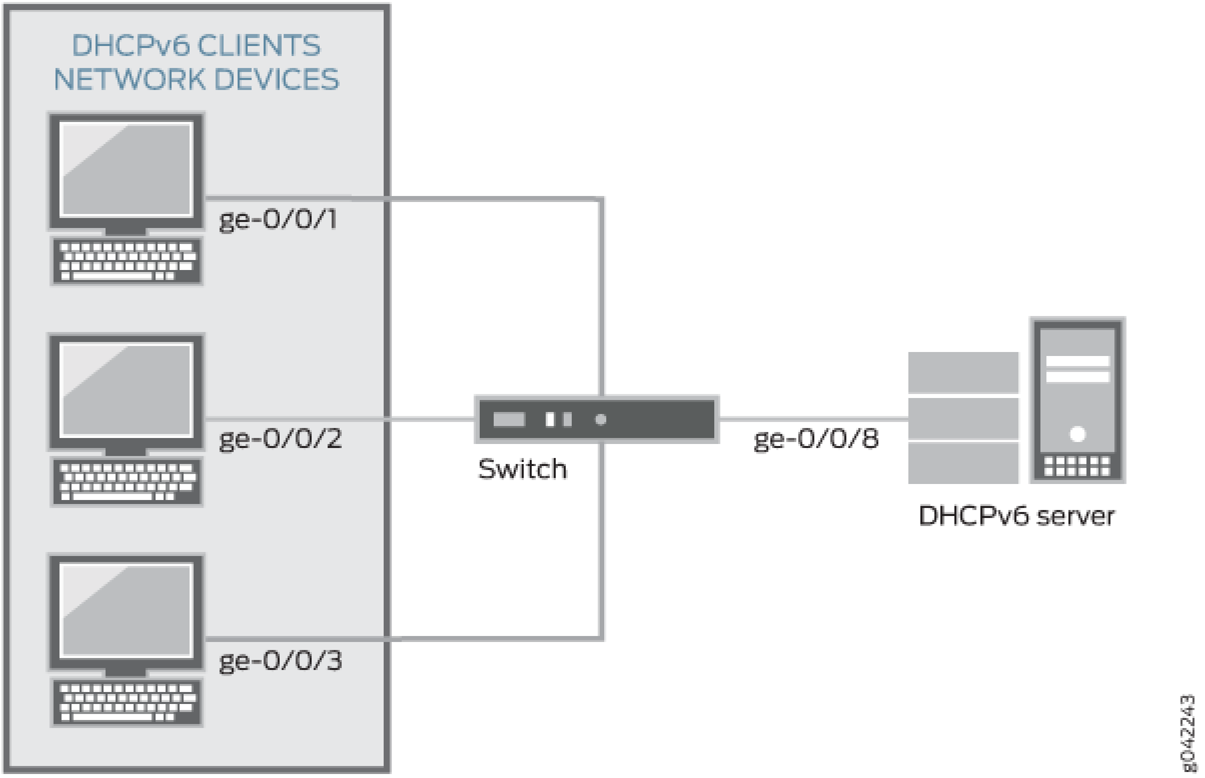
By using the DHCPv6 snooping table, also known as the binding table, IPv6 source guard and neighbor discovery inspection mitigate the risk of IPv6 spoofing attacks. The DHCPv6 snooping table contains the IP address, MAC address, VLAN and interface ID for each host associated with the VLAN. When a packet is sent from a host attached to an untrusted access interface on the switch, IPv6 source guard checks it against the entries in the DHCPv6 snooping table. If there is no match in the table, the switch does not forward the packet—that is, the packet is discarded. Neighbor discovery inspection verifies neighbor discovery messages sent between IPv6 nodes on the same network link against the DHCPv6 snooping table, and also discards the packet if no match is found.

This example shows how to configure these important port security features on a switch that is connected to a DHCPv6 server. The setup for this example includes the VLAN sales on the switch. Figure 34 on page 584 illustrates the topology for this example.

NOTE: The trunk interface connecting to the DHCPv6 server interface is a trusted port by default.

Topology

Figure 34: Network Topology for Basic Port Security



The components of the topology for this example are shown in Table 28 on page 584.

Table 28: Components of the Port Security Topology

Properties	Settings
Switch hardware	One EX Series switch that supports the Enhanced Layer 2 Software configuration style.
VLAN name and ID	sales, tag `20`
VLAN subnets	192.0.2.16/28 192.0.2.17 through 192.0.2.30 192.0.2.31 is the subnet's broadcast address

Table 28: Components of the Port Security Topology *(Continued)*

Properties	Settings
Interfaces in sales	ge-0/0/1, ge-0/0/2, ge-0/0/3, ge-0/0/8
Interface connecting to DHCPv6 server	ge-0/0/8

In this example, the switch has already been configured as follows:

- All access ports are untrusted, which is the default setting.
- The trunk port (ge-0/0/8) is trusted, which is the default setting.
- The VLAN (sales) has been configured to include the specified interfaces.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure IPv6 source guard and neighbor discovery inspection (and thereby, also automatically configure DHCPv6 snooping), copy the following commands and paste them into the switch terminal window:

```
[edit]
set vlans sales forwarding-options dhcp-security ipv6-source-guard
set vlans sales forwarding-options dhcp-security neighbor-discovery-inspection
```

Step-by-Step Procedure

Configure IPv6 source guard and neighbor discovery inspection (and thereby, also automatically configure DHCPv6 snooping) on the VLAN:

1. Configure IPv6 source guard on the VLAN:

```
[edit vlans sales forwarding-options dhcp-security]
user@switch# set ipv6-source-guard
```

2. Enable neighbor discovery inspection on the VLAN:

```
[edit vlans sales forwarding-options dhcp-security]
user@switch# set neighbor-discovery-inspection
```

Results

Check the results of the configuration:

```
user@switch> show vlans sales forwarding-options
dhcp-security {
    neighbor-discovery-inspection;
    ipv6-source-guard;
    }
}
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying That DHCPv6 Snooping Is Working Correctly on the Switch

Purpose

Verify that DHCPv6 snooping is working on the switch.

Action

Send DHCPv6 requests from network devices (in this example, these are DHCPv6 clients) connected to the switch.

Display the DHCPv6 snooping information when the port on which the DHCPv6 server connects to the switch is trusted. The following output results when requests are sent from the MAC addresses and the server has provided the IPv6 addresses and leases:

```
user@switch> show dhcp-security ipv6 binding
IPv6 address             MAC address         Vlan     Expires State   Interface
2001:db8:fe10::          00:10:94:00:55:0b   vlan20   3456    BOUND   ge-0/0/1.0
fe80::210:94ff:fe00:1    00:10:94:00:55:0b   vlan20   3456    BOUND   ge-0/0/1.0
2001:db8:fe12::          00:10:94:00:00:34   vlan20   3456    BOUND   ge-0/0/2.0
fe80::210:94ff:fe00:2    00:10:94:00:00:34   vlan20   3456    BOUND   ge-0/0/2.0
2001:db8:fe14::          00:10:94:00:00:55   vlan20   3456    BOUND   ge-0/0/3.0
fe80::210:94ff:fe00:3    00:10:94:00:00:55   vlan20   3456    BOUND   ge-0/0/3.0
```

Meaning

The output shows the assigned IPv6 addresses, the MAC address, the VLAN name, and the time, in seconds, remaining before the lease expires. Because IPv6 hosts usually have more than one IPv6 address assigned to each of their IPv6-enabled network interfaces, there are two entries added for each client: one with the link-local IPv6 address, which is used by the client for DHCP transactions, and another with the IPv6 address assigned by the server. The link-local address always has the prefix `fe80::/10`.

Verifying That Neighbor Discovery Inspection Is Working Correctly on the Switch

Purpose

Verify that neighbor discovery inspection is working on the switch.

Action

Send neighbor discovery packets from network devices connected to the switch.

Display the neighbor discovery information:

```
user@switch> show dhcp-security neighbor-discovery-inspection statistics
ND inspection statistics:
```

```
Interface     ND Packets received   ND inspection pass    ND inspection failed
ge-0/0/1.0                      7                    5                       2
ge-0/0/2.0                     10                   10                       0
ge-0/0/3.0                     12                   12                       0
```

Meaning

The sample output shows the number of neighbor discovery packets received and inspected per interface, with a list of the number of packets that passed and the number of packets that failed the inspection on each interface. The switch compares the neighbor discovery requests and replies against the entries in the DHCPv6 snooping database. If a MAC address or IPv6 address in the neighbor discovery packet does not match a valid entry in the database, the packet is dropped.

RELATED DOCUMENTATION

Configuring IP Source Guard (ELS) | 545

Configuring Port Security (ELS) | 9

Configuring IP Source Guard to Mitigate the Effects of Source IP Address Spoofing and Source MAC Address Spoofing

You can use the IP source guard access port security feature on MX Series routers to mitigate the effects of source IP address spoofing and source MAC address spoofing. If IP source guard determines that a host connected to an access interface has sent a packet with an invalid source IP address or source MAC address in the packet header, then IP source guard ensures that the switching device does not forward the packet—that is, the packet is discarded.

To configure IP source guard on a specific bridge domain by using the CLI:

- Configure the IP source guard on a bridge domain:

```
[edit bridge-domains bridge-domain-name forwarding-options dhcp-security]
user@device# set  ip-source-guard (MX Series)
```

To configure IP source guard at the routing instance level by using the CLI:

- Configure the IP source guard at the routing instance level:

```
[edit routing-instances ri-name bridge-domains bridge-domain-name forwarding-options dhcp-security]
user@device# set  ip-source-guard (MX Series)
```

RELATED DOCUMENTATION

ip-source-guard (MX Series)

Example: Configuring IP Source Guard and Dynamic ARP Inspection on a Specified Bridge Domain to Protect the Devices Against Attacks

IN THIS SECTION

This example describes how to enable IP source guard and Dynamic ARP inspection (DAI) on a specified bridge domain to protect the device against spoofed IP/MAC addresses and ARP spoofing attacks. When you enable either IP source guard or DAI, the configuration automatically enables DHCP snooping for the same bridge domain.

Requirements

This example uses the following hardware and software components:

- One MX Series router
- Junos OS Release 14.1
- A DHCP server to provide IP addresses to network devices on the device

Before you configure IP source guard to prevent IP/MAC spoofing or DAI to mitigate ARP spoofing attacks, be sure you have:

- Connected the DHCP server to the device.
- Configured the bridge domain to which you are adding DHCP security features. See Configuring the Bridge Domain for MX Series Router Cloud CPE Services.

Overview and Topology

IN THIS SECTION

Ethernet LAN devices are vulnerable to attacks on security that involve spoofing (forging) of source MAC addresses or source IP addresses. These spoofed packets are sent from hosts connected to untrusted access interfaces on the device. IP source guard checks the IP source address and MAC source address in a packet sent from a host attached to an untrusted access interface on the device against entries stored in the DHCP snooping database. If IP source guard determines that the packet header contains an invalid source IP address or source MAC address, it ensures that the device does not forward the packet—that is, the packet is discarded.

Another type of security attack is ARP spoofing (also known as ARP poisoning or ARP cache poisoning). ARP spoofing is a way to initiate man-in-the-middle attacks. The attacker sends an ARP packet that spoofs the MAC address of another device on the bridge domain. Instead of the device sending traffic to the proper network device, it sends it to the device with the spoofed address that is impersonating the proper device. If the impersonating device is the attacker's machine, the attacker receives all the traffic from the device that should have gone to another device. The result is that traffic from the device is misdirected and cannot reach its proper destination.

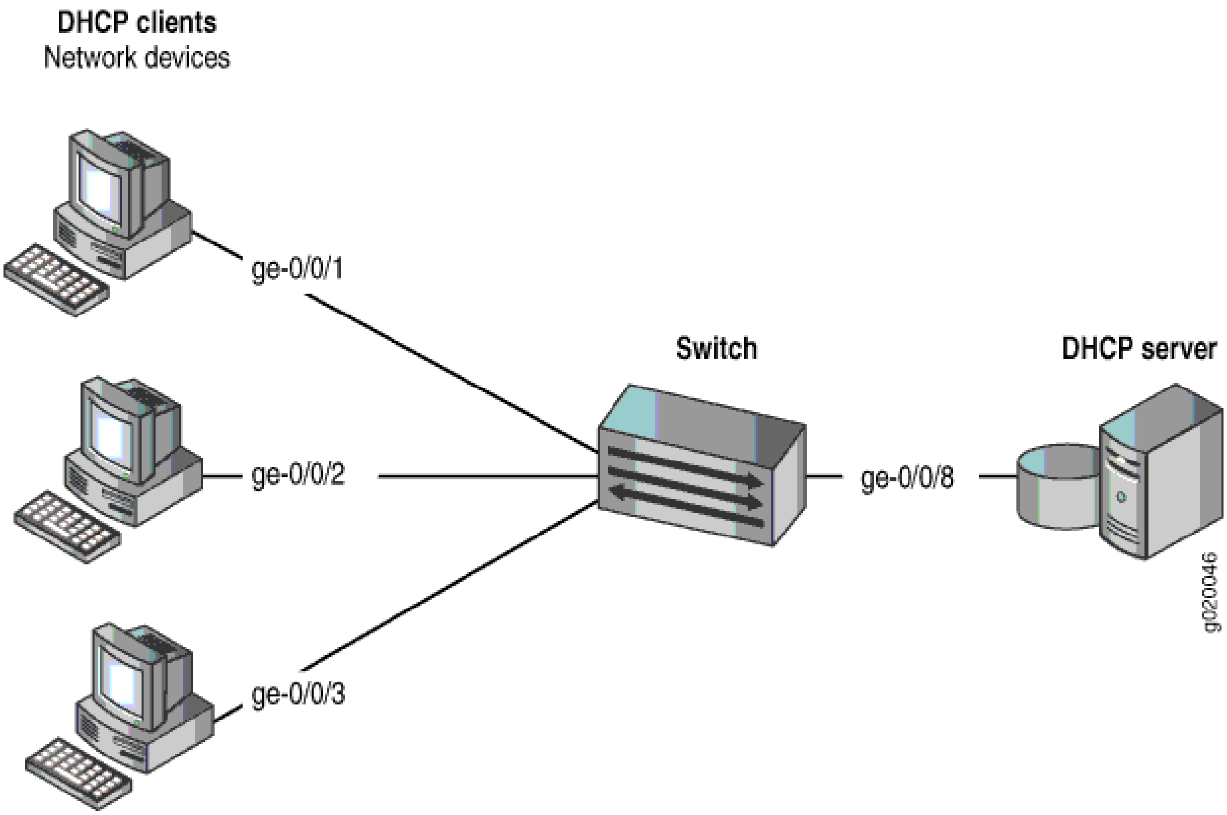
NOTE: When DAI is enabled, the device logs the number of invalid ARP packets that it receives on each interface, along with the sender's IP and MAC addresses. You can use these log messages to discover ARP spoofing on the network.

This example shows how to configure these important port security features on a device that is connected to a DHCP server. The setup for this example includes the bridge domain `employee-bdomain` on the switching device. Figure 35 on page 591 illustrates the topology for this example.

NOTE: The trunk interface connecting to the DHCP server interface is a trusted port by default.

Topology

Figure 35: Switching Device Network Topology for Basic Port Security



The components of the topology for this example are shown in Table 29 on page 591.

Table 29: Components of the Port Security Topology

Properties	Settings
Device hardware	One MX Series router
Bridge domain name and ID	`employee-bdomain`, tag `20`

Table 29: Components of the Port Security Topology *(Continued)*

Properties	Settings
Bridge domain subnets	192.0.2.16/28 192.0.2.17 through 192.0.2.30 192.0.2.31 is the subnet's broadcast address
Interfaces in employee-bdomain	ge-0/0/1, ge-0/0/2, ge-0/0/3, ge-0/0/8
Interface connecting to DHCP server	ge-0/0/8

In this example, the device has already been configured as follows:

- All access ports are untrusted, which is the default setting.
- The trunk port (ge-0/0/8) is trusted, which is the default setting.
- The bridge-domain (employee-bdomain) has been configured to include the specified interfaces.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure IP source guard and DAI (and thereby, also automatically configure DHCP snooping to protect the device against IP spoofing and ARP attacks), copy the following commands and paste them into the device terminal window:

```
[edit]
set bridge-domains employee-bdomain forwarding-options dhcp-security ip-source-guard
set bridge-domains employee-bdomain forwarding-options dhcp-security arp-inspection
```

Step-by-Step Procedure

To configure IP source guard and DAI (and thereby, also automatically configure DHCP snooping) on the bridge domain:

1. Configure IP source guard on the bridge domain:

   ```
   [edit bridge-domains employee-bdomain forwarding-options dhcp-security]
   user@device# set ip-source-guard
   ```

2. Enable DAI on the bridge domain:

   ```
   [edit bridge-domains employee-bdomain forwarding-options dhcp-security]
   user@device# set arp-inspection
   ```

Results

Check the results of the configuration:

```
user@device> show bridge-domains employee-bdomain forwarding-options
employee-bdomain {
    forwarding-options {
        dhcp-security {
            arp-inspection;
            ip-source-guard;
        }
    }
}
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying That DHCP Snooping Is Working Correctly on the Device

Purpose

Verify that DHCP snooping is working on the device.

Action

Send some DHCP requests from network devices (here they are DHCP clients) connected to the device.

Display the DHCP snooping information when the port on which the DHCP server connects to the device is trusted. The following output results when requests are sent from the MAC addresses and the server has provided the IP addresses and leases:

```
user@device> show dhcp-security binding


IP Address    MAC Address            Vlan      Expires         State      Interface

192.0.2.17    00:05:85:3A:82:77    employee-vlan   86265        BOUND          ge-0/0/1.0

192.0.2.18    00:05:85:3A:82:79    employee-vlan   86265        BOUND          ge-0/0/1.0

192.0.2.19    00:05:85:3A:82:80    employee-vlan   86287        BOUND          ge-0/0/2.0

192.0.2.20    00:05:85:3A:82:81    employee-vlan   86287        BOUND          ge-0/0/2.0

192.0.2.21    00:05:85:3A:82:83    employee-vlan   86287        BOUND          ge-0/0/2.0

192.0.2.22    00:05:85:27:32:88    employee-vlan   86254        BOUND          ge-0/0/3.0
```

Meaning

When the interface on which the DHCP server connects to the device has been set to trusted, the output (see the preceding sample) shows, for the assigned IP address, the device's MAC address, the VLAN name, and the time, in seconds, remaining before the lease expires.

Verifying That IP Source Guard Is Working on the Bridge Domain

Purpose

Verify that IP source guard is enabled and working on the bridge domain.

Action

Send some DHCP requests from network devices (here they are DHCP clients) connected to the device. View the IP source guard information for the data bridge domain.

```
user@device>                              show dhcp-security binding ip-source-
guard


IP Address    MAC Address           Vlan      Expires         State      Interface

192.0.2.17    00:05:85:3A:82:77    employee-vlan   86265        BOUND          ge-0/0/1.0

192.0.2.18    00:05:85:3A:82:79    employee-vlan   86265        BOUND          ge-0/0/1.0

192.0.2.19    00:05:85:3A:82:80    employee-vlan   86287        BOUND          ge-0/0/2.0

192.0.2.20    00:05:85:3A:82:81    employee-vlan   86287        BOUND          ge-0/0/2.0

192.0.2.21    00:05:85:3A:82:83    employee-vlan   86287        BOUND          ge-0/0/2.0

192.0.2.22    00:05:85:27:32:88    employee-vlan   86254        BOUND          ge-0/0/3.0
```

Meaning

The IP source guard database table contains the VLANS and bridge domains enabled for IP source guard.

Verifying That DAI Is Working Correctly on the Device

Purpose

Verify that DAI is working on the device.

Action

Send some ARP requests from network devices connected to the device.

Display the DAI information:

```
user@device> show dhcp-security arp inspection statistics
ARP inspection statistics:

Interface        Packets received  ARP inspection pass   ARP inspection failed
ge-0/0/1.0                      7                    5                       2
ge-0/0/2.0                     10                   10                       0
ge-0/0/3.0                     12                   12                       0
```

Meaning

The sample output shows the number of ARP packets received and inspected per interface, with a listing of how many packets passed and how many failed the inspection on each interface. The device compares the ARP requests and replies against the entries in the DHCP snooping database. If a MAC address or IP address in the ARP packet does not match a valid entry in the database, the packet is dropped.

RELATED DOCUMENTATION

Configuring IP Source Guard (ELS) | 545

Enabling Dynamic ARP Inspection (ELS) | 531

Example: Configuring IPv6 Source Guard and Neighbor Discovery Inspection to Protect a Switch from IPv6 Address Spoofing

IN THIS SECTION

This example describes how to enable IPv6 source guard and neighbor discovery inspection on a specified VLAN to protect an EX Series switch against IPv6 address spoofing attacks. IPv6 source guard and neighbor discovery inspection support introduced on EX2200 and EX3300 switches in Junos OS Release 14.1X53-D10.

Requirements

This example uses the following hardware and software components:

- One EX2200 or EX3300 switch
- Junos OS Release 14.1X53-D10 or later for EX Series switches
- A DHCPv6 server to provide IPv6 addresses to network devices on the switch

Before you configure IPv6 source guard and neighbor discovery inspection to prevent IPv6 address spoofing attacks, be sure you have:

- Connected the DHCPv6 server to the switch.
- Configured the VLAN to which you are adding DHCPv6 security features. See Configuring VLANs for EX Series Switches.

Overview and Topology

IN THIS SECTION

Ethernet LAN switches are vulnerable to attacks on security that involve spoofing (forging) of source MAC addresses or source IPv6 addresses. These spoofed packets are sent from hosts connected to untrusted access interfaces on the switch. For more information on IPv6 address spoofing attacks, see "IPv6 Neighbor Discovery Inspection" on page 605.

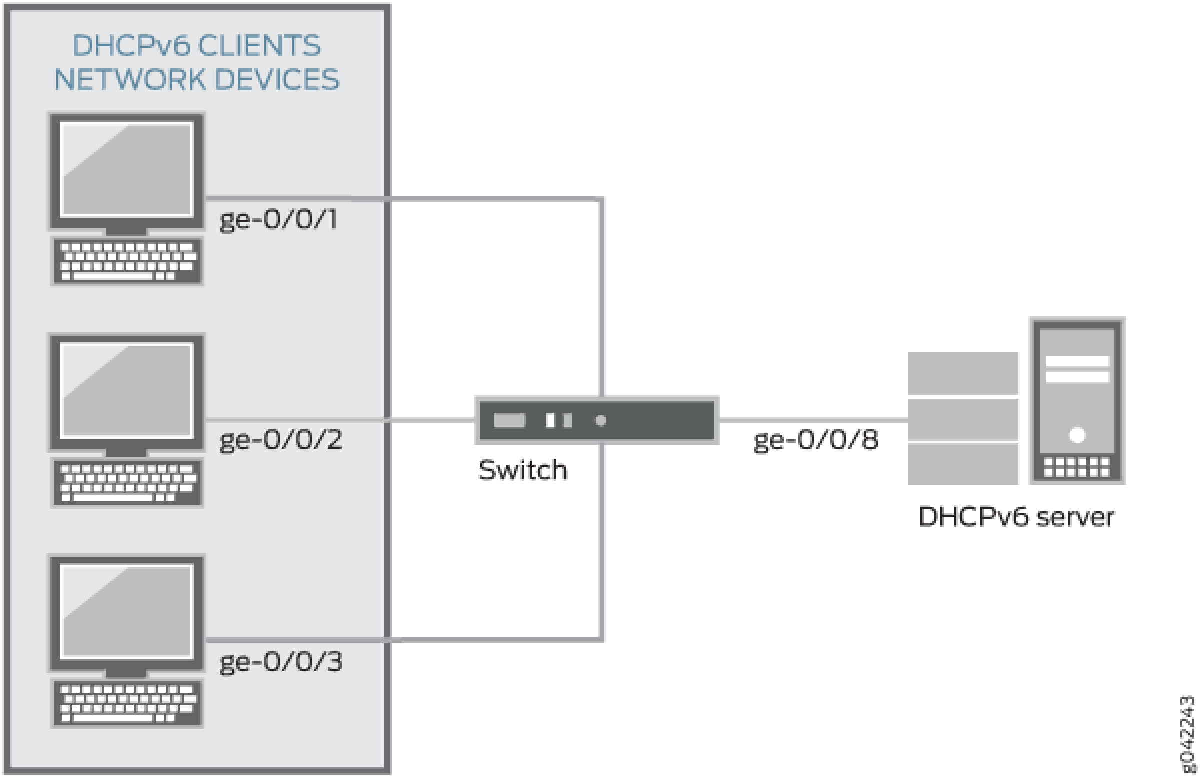
IPv6 source guard and neighbor discovery inspection mitigate the risk of IPv6 spoofing attacks by using the DHCPv6 snooping table. Also known as the binding table, the DHCPv6 snooping table contains the valid bindings of IPv6 addresses to MAC addresses. When a packet is sent from a host attached to an untrusted access interface on the switch, IPv6 source guard verifies the source IPv6 address and MAC address of the packet against the DHCPv6 snooping table. If there is no match in the table, the switch does not forward the packet—that is, the packet is discarded. Neighbor discovery inspection verifies neighbor discovery messages sent between IPv6 nodes on the same network link against the DHCPv6 snooping table, and also discards the packet if no match is found.

This example shows how to configure these important port security features on a switch that is connected to a DHCPv6 server. The setup for this example includes the VLAN `sales` on the switch. Figure 36 on page 598 illustrates the topology for this example.

NOTE: The trunk interface connecting to the DHCPv6 server interface is a trusted port by default.

Topology

Figure 36: Network Topology for Basic Port Security



The components of the topology for this example are shown in Table 30 on page 599.

Table 30: Components of the Port Security Topology

Properties	Settings
Switch hardware	One EX2200 or EX3300 switch
VLAN name and ID	sales, tag
VLAN subnets	192.0.2.16/28 192.0.2.17 through 192.0.2.30 192.0.2.31 is the subnet's broadcast address
Interfaces in `sales`	ge-0/0/1, ge-0/0/2, ge-0/0/3, ge-0/0/8
Interface connecting to DHCPv6 server	ge-0/0/8

In this example, the switch has already been configured as follows:

- All access ports are untrusted, which is the default setting.
- The trunk port (ge-0/0/8) is trusted, which is the default setting.
- The VLAN (sales) has been configured to include the specified interfaces.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure IPv6 source guard and neighbor discovery inspection, copy the following commands and paste them into the switch terminal window:

```
[edit]
set ethernet-switching-options secure-access-port vlan sales examine-dhcpv6
set ethernet-switching-options secure-access-port vlan sales ipv6-source-guard
set ethernet-switching-options secure-access-port vlan sales neighbor-discovery-inspection
```

Step-by-Step Procedure

Configure IPv6 source guard and neighbor discovery inspection (and thereby, also automatically configure DHCPv6 snooping) on the VLAN:

1. Enable DHCPv6 snooping on the VLAN:

   ```
   [edit ethernet-switching-options secure-access-port vlan sales]
   user@switch# set examine-dhcpv6
   ```

2. Configure IPv6 source guard on the VLAN:

   ```
   [edit ethernet-switching-options secure-access-port vlan sales]
   user@switch# set ipv6-source-guard
   ```

3. Configure neighbor discovery inspection on the VLAN:

   ```
   [edit ethernet-switching-options secure-access-port vlan sales]
   user@switch# set neighbor-discovery-inspection
   ```

Results

Check the results of the configuration:

```
user@switch> show ethernet-switching-options secure-access-port
vlan sales {
    examine-dhcpv6;
    ipv6-source-guard;
    neighbor-discovery-inspection;
    }
}
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying That DHCPv6 Snooping Is Working Correctly on the Switch

Purpose

Verify that DHCPv6 snooping is working on the switch.

Action

Send DHCPv6 requests from network devices (in this example, these are DHCPv6 clients) connected to the switch.

Display the DHCPv6 snooping information when the port on which the DHCPv6 server connects to the switch is trusted. The following is the output when requests are sent from the MAC addresses and the server has provided the IPv6 addresses and leases:

```
user@switch> show dhcpv6 snooping binding
DHCP Snooping Information:
MAC address        IP address              Lease (seconds)  Type  VLAN       Interface
00:10:94:00:00:01  2001:db8::10:0:3        3599992         dynamic  sales      ge-0/0/1.0
00:10:94:00:00:01  fe80::210:94ff:fe00:1  3599992         dynamic  sales      ge-0/0/1.0
00:10:94:00:00:02  2001:db8::10:0:5        3599992         dynamic  sales      ge-0/0/2.0
00:10:94:00:00:02  fe80::210:94ff:fe00:2  3599992         dynamic  sales      ge-0/0/2.0
00:10:94:00:00:03  2001:db8::10:0:7        3599992         dynamic  sales      ge-0/0/3.0
00:10:94:00:00:03  fe80::210:94ff:fe00:3  3599992         dynamic  sales      ge-0/0/3.0
```

Meaning

The output shows the assigned IP address, the MAC address, the VLAN name, and the time, in seconds, leased to the IP address. Because IPv6 hosts usually have more than one IP address assigned to each of their IPv6-enabled network interfaces, there are two entries added for each client: one with the link-local IP address, which is used by the client for DHCP transactions, and another with the IP address assigned by the server. The link-local address always has the prefix `fe80::/10`.

Verifying That Neighbor Discovery Inspection Is Working Correctly on the Switch

Purpose

Verify that neighbor discovery inspection is working on the switch.

Action

Send neighbor discovery packets from network devices connected to the switch.

Display the neighbor discovery information:

```
user@switch> show neighbor-discovery-inspection statistics
ND inspection statistics:

Interface        Packets received   ND inspection pass    ND inspection failed
ge-0/0/1.0                      7                    5                     2
```

```
ge-0/0/2.0                     10                   10                      0
ge-0/0/3.0                     12                   12                      0
```

Meaning

The sample output shows the number of neighbor discovery packets received and inspected per interface, and lists the number of packets passed and the number that failed the inspection on each interface. The switch compares the neighbor discovery requests and replies against the entries in the DHCPv6 snooping database. If a MAC address or IPv6 address in the neighbor discovery packet does not match a valid entry in the database, the packet is dropped.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
14.1X53-D10	IPv6 source guard and neighbor discovery inspection support introduced on EX2200 and EX3300 switches in Junos OS Release 14.1X53-D10.

RELATED DOCUMENTATION

Configuring IP Source Guard (non-ELS) | **541**

Enabling DHCP Snooping (non-ELS) | **456**

Configuring Port Security (non-ELS) | **11**

9

PART

IPv6 Access Security

CHAPTER 20

Neighbor Discovery Protocol

IN THIS CHAPTER

IPv6 Neighbor Discovery Inspection

IN THIS SECTION

IPv6 Neighbor Discovery Protocol Overview

IPv6 nodes (hosts and routers) use Neighbor Discovery Protocol (NDP) to discover the presence and link-layer addresses of other nodes residing on the same link. Hosts use NDP to find neighboring routers that are willing to forward packets on their behalf, while routers use it to advertise their presence. Nodes also use NDP to maintain reachability information about the paths to active neighbors. When a router or the path to a router fails, a host can search for alternate paths.

The NDP process is based on the exchange of neighbor solicitation and advertisement messages. NDP messages are unsecured, which makes NDP susceptible to attacks that involve the spoofing (or forging) of link-layer addresses. An attacking node can cause packets for legitimate nodes to be sent to some other link-layer address by either sending a neighbor solicitation message with a spoofed source MAC address, or by sending a neighbor advertisement address with a spoofed target MAC address. The spoofed MAC address is then associated with a legitimate network IPv6 address by the other nodes.

Neighbor Discovery (ND) Inspection

IPv6 neighbor discovery inspection mitigates NDP security vulnerabilities by inspecting neighbor discovery messages and verifying them against the DHCPv6 snooping table. The DHCPv6 snooping table, which is built by snooping DHCPv6 message exchanges, includes the IPv6 address, MAC address, VLAN and interface for each host associated with the VLAN. When a neighbor discovery message is received on an untrusted interface, neighbor discovery inspection discards the packet unless the source IPv6 and MAC addresses, VLAN, and interface can be matched to an entry in the DHCPv6 snooping table. Entries can be added to the DHCPv6 snooping table by configuring the `static-ipv6` CLI statement.

NOTE: Neighbor discovery messages are always allowed on trusted interfaces.

Neighbor discovery inspection verifies five different ICMPv6 message types: Router Solicitation, Router Advertisement, Neighbor Solicitation, Neighbor Advertisement, and Redirect. By discarding message packets that can not be verified against the DHCPv6 snooping table, neighbor discovery inspection can prevent the following types of attacks:

- Cache poisoning attacks—Neighbor discovery cache poisoning is the IPv6 equivalent of ARP spoofing, in which an attacker uses a forged address to send an unsolicited advertisement to other hosts on the network, for associating its own MAC address with a legitimate network IP address. These bindings between IPv6 addresses and MAC addresses are stored by each node in its neighbor cache. Once the caches are updated with the malicious bindings, the attacker can initiate a man-in-the-middle attack, intercepting traffic that was intended for a legitimate host.

- Routing denial-of-service (DoS) attacks—An attacker could cause a host to disable its first-hop router by spoofing the address of a router and sending a neighbor advertisement message with the *router* flag cleared. The victim host assumes that the device that used to be its first-hop router is no longer a router.

- Redirect attacks—Routers use ICMPv6 redirect requests to inform a host of a more efficient route to a destination. Hosts can be redirected to a better first-hop router, but can also be informed by a Router Redirect message that the destination is in fact a neighbor. An attacker using this provision can achieve an effect similar to cache poisoning and intercept all traffic from the victim host. Neighbor discovery inspection checks that Router Redirect messages are sent only by trusted routers.

Enabling ND Inspection

NOTE: DHCPv6 snooping is enabled automatically when neighbor discovery inspection is configured. There is no explicit configuration required for DHCPv6 snooping.

To enable neighbor discovery inspection on a VLAN:

```
[edit vlans vlan-name forwarding-options dhcp-security]
user@switch# set  neighbor-discovery-inspection
```

RELATED DOCUMENTATION

Understanding DHCP Snooping (ELS) | 435

CHAPTER 21

SLAAC Snooping

IN THIS CHAPTER

IPv6 Stateless Address Auto-configuration (SLAAC) Snooping

IN THIS SECTION

Understanding SLAAC Snooping

IN THIS SECTION

Dynamic address assignment is an important feature of IPv6 due to the vast increase in address space over IPv4. In addition to static addressing, IPv6 provides two options for clients to obtain addresses dynamically: DHCPv6 (stateful) and stateless address auto-configuration (SLAAC).

SLAAC simplifies IPv6 address management by providing plug-and-play IP connectivity with no manual configuration of hosts. SLAAC enables an IPv6 client to generate its own addresses using a combination of locally-available information and information advertised by routers through Neighbor Discovery Protocol (NDP).

NDP messages are unsecured, which makes SLAAC susceptible to attacks that involve the spoofing (or forging) of link-layer addresses. You must configure SLAAC snooping to validate IPv6 clients using SLAAC before allowing them to access the network.

SLAAC Process

The client begins auto-configuration by generating a link-local address for the IPv6-enabled interface. This is done by combining the advertised link-local prefix (first 64 bits) with the interface identifier (last 64 bits). The address is generated according to the following format: [fe80 (10 bits) + 0 (54 bits)] + *interface ID* (64 bits).

Before assigning the link-local address to its interface, the client verifies the address by running Duplicate Address Detection (DAD). DAD sends a Neighbor Solicitation message destined to the new address. If there is a reply, then the address is a duplicate and the process stops. If the address is unique, it is assigned to the interface.

To generate a global address, the client sends a Router Solicitation message to prompt all routers on the link to send Router Advertisement (RA) messages. Routers that are enabled to support SLAAC send an RA that contains a subnet prefix for use by neighboring hosts. The client appends the interface identifier to the subnet prefix to form a global address, and again runs DAD to confirm its uniqueness.

SLAAC Snooping

SLAAC is subject to the same security vulnerabilities found in NDP. You can configure SLAAC snooping to secure traffic from IPv6 clients using SLAAC for dynamic address assignment. For more information on NDP, see "IPv6 Neighbor Discovery Inspection" on page 605.

SLAAC snooping is similar to DHCP snooping, in that it snoops packets to build a table of IP-MAC address bindings. SLAAC snooping extracts address information from DAD packets exchanged during the SLAAC process to build the SLAAC snooping table. The address bindings in this table are used to inspect and validate NDP/IP packets sent by IPv6 clients using SLAAC.

Configuring SLAAC Snooping

SLAAC snooping is enabled on a per-VLAN basis. By default, SLAAC snooping is disabled for all VLANs.

To enable SLAAC, use the following commands:

- To enable SLAAC on a specific VLAN:

```
[edit]
user@switch# set forwarding-options access-security slaac-snooping vlans vlan-name
```

- To enable SLAAC on all VLANs:

```
[edit]
user@switch# set forwarding-options access-security slaac-snooping vlans all
```

Configuring Auto-DAD

If DAD is disabled on the client side, or DAD packets are dropped due to traffic congestion, SLAAC snooping will perform auto-DAD on behalf of the client. The client-generated address is in a tentative state until the DAD process is completed.

Auto-DAD sends a Neighbor Solicitation message with the client-generated address as a target, and waits for a Neighbor Advertisement in response. If there is a response, then the address is a duplicate and cannot be assigned to the client. If there is no response, then the address is confirmed.

The amount of time that auto-DAD waits for a response is 1 second by default, with no retries. You can configure the number of retries and the length of the interval between transmissions.

NOTE: During a MAC move, the first Neighbor Solicitation packet will result in a SLAAC entry flush from the old port and the second will result in the creation of a SLAAC entry for the new port.

To configure the number of retries for auto-DAD parameters, use the following commands:

- For a specific interface:

```
[edit]
user@switch# set forwarding-options access-security slaac-snooping interface interface-name
auto-dad retries retry-count
```

- For all interfaces:

```
[edit]
user@switch# set forwarding-options access-security slaac-snooping interface all auto-dad
retries retry-count
```

To configure the interval between auto-DAD transmissions, use the following commands:

- For a specific interface:

```
[edit]
user@switch# set forwarding-options access-security slaac-snooping interface interface-name
auto-dad retrans-interval seconds
```

- For all interfaces:

```
[edit]
user@switch# set forwarding-options access-security slaac-snooping interface all auto-dad
retrans-interval seconds
```

Configuring the Link-Local Address Expiration

The link-local address learned by SLAAC has a default expiration period of 1 day. When the lease for the address expires, the snooping device sends a DAD message with the client address as the target. If the client is still reachable, the lease is renewed.

To configure the length of the expiration period, use the following command:

```
[edit]
user@switch# set forwarding-options access-security slaac-snooping link-local expiry interval
seconds
```

Configuring the Allowed DAD Contentions

You can configure the maximum number of DAD contentions (Neighbor Solicitation or Neighbor Advertisement) messages for an interface. If the maximum number of contentions is exceeded during the allowed time interval, the interface is considered invalid and the SLAAC snooping table is not updated with any bindings for that client.

NOTE: Maximum allowed contentions is configured on a per-interface basis, to allow for interfaces that belong to more than one VLAN.

To configure the maximum number of DAD contentions and the allowed time interval, use the following command:

```
[edit]
user@switch# set forwarding-options access-security slaac-snooping interface interface-name max-
allowed-contention count integer duration seconds
```

Configuring an Interface as Trusted for SLAAC Snooping

When you configure an interface as trusted, the binding entry for the interface is added to the SLAAC snooping table using the same process as for untrusted interfaces.

When a DAD request is received on a trusted port with an IP/MAC entry that already exists on an untrusted port, SLAAC snooping sends a unicast DAD towards the untrusted port to see whether the host is live.

- If the host responds with an NA message on the untrusted port, the lease time is renewed for the existing binding entry.
- If there is no response (NA) on the untrusted port, the corresponding binding entry is deleted.

If the entry for the untrusted port is deleted, the binding for the trusted port is not created immediately. When the trusted port starts to send data traffic, it will send an NS message. At that time, SLAAC snooping adds the new binding on the trusted port.

Router advertisement packets received on a trusted port are flooded to all the ports in that VLAN irrespective of the SLAAC entry for the receiving port.

NOTE: Maximum number of DAD contentions is not applicable to trusted interfaces.

To configure an interface as trusted for SLAAC snooping, use the following command:

```
[edit]
user@switch# set forwarding-options access-security slaac-snooping interface interface-name mark-
interface trusted
```

Configuring Persistent SLAAC Snooping Bindings

The IP-MAC bindings in the DHCP snooping database file are not persistent. If the switch is rebooted, the bindings are lost. You can configure persistent bindings by specifying a local pathname or a remote URL for the storage location of the SLAAC snooping database file.

To configure persistent bindings for SLAAC snooping, use the following command:

```
[edit]
user@switch# set system processes slaac-snooping persistent-file (local-pathname | remote-url)
write-interval seconds
```

CHAPTER 22

Router Advertisement Guard

IN THIS CHAPTER

Understanding IPv6 Router Advertisement Guard

In an IPv6 deployment, routers periodically multicast Router Advertisement (RA) messages to announce their availability and convey information to neighboring nodes that enable them to be automatically configured on the network. RA messages are used by Neighbor Discovery Protocol (NDP) to detect neighbors, advertise IPv6 prefixes, assist in address provisioning, and share link parameters such as maximum transmission unit (MTU), hop limit, advertisement intervals, and lifetime. Hosts listen for RA messages for IPv6 address autoconfiguration and discovery of link-local addresses of the neighboring routers, and can also send a Router Solicitation (RS) message to request immediate advertisements.

RA messages are unsecured, which makes them susceptible to attacks on the network that involve the spoofing (or forging) of link-layer addresses. Also, unintended misconfiguration by users or administrators might lead to the presence of unwanted, or rogue, RA messages, which can cause operational problems for neighboring hosts. You can configure IPv6 Router Advertisement (RA) guard to protect your network against rogue RA messages generated by unauthorized or improperly configured routers connecting to the network segment.

RA guard works by validating RA messages on the basis of whether they meet certain criteria, configured on the switch using policies. RA guard inspects RA messages and compares the information contained in the message attributes to the configured policy. Depending on the policy, RA guard either drops or forwards the RA messages that match the conditions.

The following information contained in RA message attributes can be used by RA guard to validate the source of the RA message:

- Source MAC address
- Source IPv6 address

- Source IPv6 address prefix
- Hop-count limit
- Router preference priority
- *Managed* configuration flag
- *Other* configuration flag

You can configure RA guard to operate in either stateless or stateful mode. In stateless mode, in the default state, an RA message that is received on an interface is examined and filtered on the basis of whether it matches the conditions configured in the policy attached to that interface. If the content of the RA message is validated, it forwards the RA message to its destination; otherwise, the RA message is dropped. The state of an interface operating in stateless mode can be changed by configuration. If the interface is configured as *trusted*, all RA messages are forwarded without being validated against the policy. If the interface is configured as *blocked*, all RA messages are dropped without being validated against the policy.

In stateful mode, an interface can dynamically transition from one state to another based on information gathered during a learning period. During this period, known as the *learning* state, ingress RA messages are validated against a policy to determine which interfaces are attached to links with valid IPv6 routers. At the end of the learning period, interfaces attached to legitimate senders of RA messages transition dynamically to the *forwarding* state, in which RA messages are forwarded if they can be validated against a policy. Interfaces that do not receive valid RA messages during the learning period transition dynamically to the *blocked* state, in which all ingress RA messages are dropped.

Table 31 on page 615 summarizes the states of IPv6 RA guard for both stateless and stateful mode.

Table 31: IPv6 RA guard states

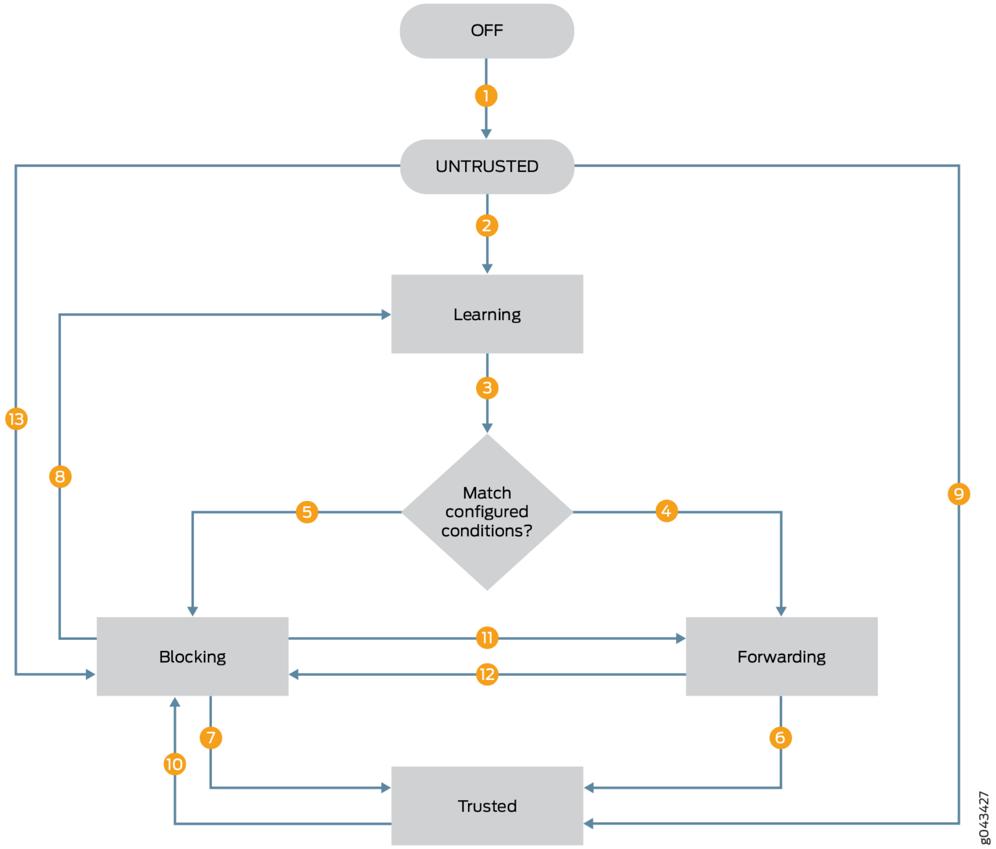
State	Description	Mode
Off	The interface operates as if RA guard is not available.	Stateless/stateful
Untrusted	The interface forwards ingress RA messages if received RA messages are validated against the configured policy rules; otherwise, it drops the RA message. Untrusted state is the default state of an interface enabled for RA guard.	Stateless/stateful
Blocked	The interface blocks ingress RA messages.	Stateless/stateful

Table 31: IPv6 RA guard states *(Continued)*

State	Description	Mode
Forwarding	The interface forwards ingress RA messages if received RA messages are validated against the configured policy rules; otherwise, it drops the RA messages.	Stateful
Learning	The switch actively acquires information about the IPv6 routing device connected to the interface. The learning process takes place over a predefined period of time.	Stateful
Trusted	The interface forwards all RA messages directly, without validating them against the policy.	Stateless/stateful

Figure 37 on page 617 illustrates the transition of states when stateful RA guard is enabled. The numbers shown on the illustrations are described in the text that follows; these are not sequential steps.

Figure 37: Stateful RA Guard State Transitions



1. When RA guard is enabled on an interface it moves to the *untrusted* state from the *off* state. The *untrusted* state is the default state of an interface that is enabled for RA guard.

2. When the command requesting the learning state is issued, the interface is moved from the *off* state to the *learning* state.

3. RA messages received during the learning state are compared to the configured policy.

4. If RA messages are validated against the configured policy, the interface moves to *forwarding* state.

5. If RA messages are not validated against the configured policy, the interface moves to *blocked* state.

6. If `mark-interface trust` is configured on the validated interface, then it moves from *forwarding* state to *trusted* state.

7. If `mark-interface trust` is configured on the blocked interface, then it moves from *blocked* state to *trusted* state.

8. If learning is requested on a blocked interface, then the interface moves from the *blocked* state to the *learning* state.

9. If an interface in the default *untrusted* state is configured as `mark-interface trust`, it moves directly to the trusted state. In this case a policy can not be applied on that interface.

10. If the `mark-interface trust` configuration is deleted, and no valid RAs are received on the interface, then the interface moves to the *blocked* state.

11. If the command requesting the forwarding state is issued, then the interface moves directly from *blocked* to *forwarding* state.

12. If the command requesting the blocking state is issued, then the interface moves directly from *forwarding* to *blocked*.

13. If an interface in the default *untrusted* state is configured as `mark-interface block`, it moves directly to the *blocked* state. In this case a policy can not be applied on that interface.

RELATED DOCUMENTATION

IPv6 Neighbor Discovery Protocol Overview
Port Security Features | 2
Configuring Port Security (non-ELS) | 11

Configuring Stateful IPv6 Router Advertisement Guard

IN THIS SECTION

Stateful IPv6 Router Advertisement (RA) guard enables a switch to learn about the sources of RA messages for a certain period of time. During this period, during which the switch is known to be in the learning state, the information contained in received RA message attributes is stored and compared to the policy. At the end of the learning period, the switch has a record of which interfaces are attached to links with valid IPv6 routers. If there is no valid IPv6 router attached to an interface, the switch dynamically transitions the interface from the learning state into the blocking state. Subsequent RA messages received after the transition to blocking state are dropped. If there is a valid IPv6 router attached to the interface, the interface transitions to the forwarding state. In the forwarding state, RA messages that can be validated against the configured policy are forwarded.

You can override the dynamic state transitions by statically configuring the forwarding or blocking states on an interface. When you statically configure the state on an interface, the state can be changed only through configuration. For example, if you configure the forwarding state on an interface, the interface remains in the forwarding state until you configure a different state on that interface.

Before you can enable IPv6 RA guard on an interface or a VLAN, you must configure a policy. Stateful RA guard uses the policy to determine whether the RA messages received on an interface are from valid senders. You can configure the policy to either accept or discard RA messages that meet the predefined criteria. If the criteria for the policy includes source addresses or address prefixes, you must configure a list of the addresses before configuring the policy.

Enabling Stateful RA Guard on an Interface

You can enable stateful RA guard on an interface. You must first configure a policy, which is used to validate incoming RA messages during the learning period. After you apply an RA guard policy to an interface, you must enable RA guard on the corresponding VLAN.

To enable stateful RA guard on an interface:

1. Apply a policy to an interface.

   ```
   [edit]
   user@switch# set forwarding-options access-security router-advertisement-guard interface
   interface-name policy policy-name
   ```

2. Configure the `stateful` option on the interface:

   ```
   [edit forwarding-options access-security router-advertisement-guard interface interface-name
   policy policy-name]
   user@switch# set stateful
   ```

3. Enable stateful RA guard on the corresponding VLAN:

```
[edit]
user@switch# set forwarding-options access-security router-advertisement-guard vlans vlan-
name policy policy-name stateful
```

Enabling Stateful RA Guard on a VLAN

You can enable stateful RA guard on a per-VLAN basis or for all VLANs. You must first configure a policy, which used to validate incoming RA messages during the learning state.

To enable stateful RA guard on a specific VLAN:

1. Apply a policy to a VLAN.

```
[edit]
user@switch# set forwarding-options access-security router-advertisement-guard vlans vlan-
name policy policy-name
```

2. Configure the `stateful` option on the VLAN:

```
[edit forwarding-options access-security router-advertisement-guard vlans vlan-name policy
policy-name]
user@switch# set stateful
```

To enable stateful RA guard on all VLANs:

1. Apply a policy to all VLANs.

```
[edit]
user@switch# set forwarding-options access-security router-advertisement-guard vlans all
policy policy-name
```

> **NOTE**: If a policy has been configured for a specific VLAN using the command `set forwarding-options access-security router-advertisement-guard vlans` *`vlan-name`* `policy` *`policy-name`*, that policy takes priority over the policy applied globally to all VLANs.

2. Configure the `stateful` option on all VLANs:

```
[edit forwarding-options access-security router-advertisement-guard vlans all policy policy-
name]
user@switch# set stateful
```

Configuring the Learning State on an Interface

When stateful RA guard is first enabled, the default state is *off*. An interface in the off state operates as if RA guard is not available. To transition an interface to the learning state, you must request learning on the interface. An interface in the learning state actively acquires information from the RA messages that it receives.

To configure stateful RA guard learning on an interface:

1. Request learning on the interface.

```
[edit]
user@switch# request access-security router-advertisement-guard-learn interface interface-name
```

2. Configure the learning period in seconds.

```
[edit]
user@switch# request access-security router-advertisement-guard-learn interface interface-
name  duration seconds
```

3. Configure the action to take on ingress RA messages received during the learning period. To forward RA messages received during the learning period, configure forwarding on the interface.

   - To forward RA messages during the learning period:

   ```
   [edit]
   user@switch# request access-security router-advertisement-guard-learn interface interface-
   name  duration seconds forward
   ```

   - To block RA messages during the learning period:

   ```
   [edit]
   user@switch# request access-security router-advertisement-guard-learn interface interface-
   name  duration seconds block
   ```

Configuring the Forwarding State on an Interface

An interface in the forwarding state accepts ingress RA messages that can be validated against the configured policy and forwards them to their destination. An interface can dynamically transition to the forwarding state directly from the learning state, or the forwarding state can be statically configured on the interface.

- To configure the forwarding state on an interface:

```
[edit]
user@switch# request access-security router-advertisement-guard-forward interface interface-
name
```

Configuring the Blocking State on an Interface

An interface in the blocking state blocks ingress RA messages. An interface can dynamically transition to the blocking state directly from the learning state, or the blocking state can be statically configured on the interface. An interface that has been statically configured to be in the blocking state will remain in the blocking state until another state is configured on that interface.

- To configure the blocking state on an interface:

```
[edit]
user@switch# request access-security router-advertisement-guard-block interface interface-name
```

RELATED DOCUMENTATION

Understanding IPv6 Router Advertisement Guard | 614

Configuring Stateless IPv6 Router Advertisement Guard

Configuring Stateless IPv6 Router Advertisement Guard

IN THIS SECTION

Stateless IPv6 Router Advertisement (RA) guard enables the switch to examine incoming RA messages and filter them based on a predefined set of criteria. If the switch validates the content of the RA message, it forwards the RA message to its destination; otherwise, the RA message is dropped.

Before you can enable IPv6 RA guard, you must configure a policy with the criteria to be used for validating RA messages received on an interface. You can configure the policy to either accept or discard RA messages on the basis of whether they meet the criteria. The criteria are compared to information included in the RA messages. If the criteria for the policy includes source addresses or address prefixes, you must configure a list of the addresses before configuring the policy.

Configuring a Discard Policy for RA Guard

You can configure a discard policy to drop RA messages from predefined sources. You must first configure a list or lists of the source addresses or address prefixes, and then associate them with a policy. The following lists can be associated with discard policy:

- `source-ip-address-list`
- `source-mac-address-list`
- `prefix-list-name`

NOTE: You can include more than one type of list in a discard policy. If the information contained in a received RA message matches any one of the list parameters, then that RA message is discarded.

To configure a discard policy for RA guard:

1. Define one or more lists of disallowed source addresses or address prefixes that RA guard will use to filter incoming RA messages. Add one address or address prefix per line in the configuration.

   - To define a list of IPv6 source addresses:

     ```
     [edit]
     user@switch# set policy-options prefix-list address-list-name ipv6-address
     ```

- To define a list of IPv6 address prefixes:

  ```
  [edit]
  user@switch# set policy-options prefix-list prefix-list-name ipv6-prefix
  ```

- To define a list of MAC source addresses:

  ```
  [edit]
  user@switch# set policy-options mac-list address-list-name mac-address
  ```

2. Configure the policy name:

   ```
   [edit]
   user@switch# set forwarding-options access-security router-advertisement-guard policy policy-
   name
   ```

3. Specify the discard action:

   ```
   [edit forwarding-options access-security router-advertisement-guard policy policy-name]
   user@switch# set discard
   ```

4. Associate the policy with the list or lists defined in Step 1. For example, to discard RA messages that match a source MAC address in the list:

   ```
   [edit forwarding-options access-security router-advertisement-guard policy policy-name
   discard]
   user@switch# set source-mac-address-list address-list-name
   ```

Configuring an Accept Policy for RA Guard

You can configure an accept policy to forward RA messages on the basis of certain criteria. You can configure either match lists of source address or address prefixes as the criteria, or you can configure other match conditions, such as hop limit, configuration flags, or router preference as the criteria.

The following lists can be associated with an accept policy by using the `match-list` option:

- `source-ip-address-list`
- `source-mac-address-list`
- `prefix-list-name`

NOTE: You can associate more than one type of match list with an accept policy. If the match-all suboption is configured, then a received RA message must match all configured match lists in order to be forwarded; otherwise, it is discarded. If the match-any option is configured, then a received RA message must match any one of the configured match lists in order to be forwarded; if it does not match any of the configured lists, then it is discarded.

The following match conditions can be configured using the match-option option:

- hop-limit—Configure the RA guard policy to verify the minimum or maximum hop count for an incoming RA message.
- managed-config-flag—Configure the RA guard policy to verify that the managed address configuration flag of an incoming RA message is set.
- other-config-flag—Configure the RA guard policy to verify that the other configuration flag of an incoming RA message is set.
- router-preference-maximum—Configure the RA guard policy to verify that the default router preference parameter value of an incoming RA message is lower than or equal to a specified limit.

NOTE: The match-list and match-option options are used only in accept policies, not in discard policies.

To configure an accept policy for RA guard by using the match-list option:

1. Define one or more lists of authorized source addresses or address prefixes that RA guard will use to filter incoming RA messages. Add one address or address prefix per line in the configuration.

   - To define a list of IPv6 source addresses:

     ```
     [edit]
     user@switch# set policy-options prefix-list address-list-name ipv6-address
     ```

   - To define a list of IPv6 address prefixes:

     ```
     [edit]
     user@switch# set policy-options prefix-list prefix-list-name ipv6-prefix
     ```

- To define a list of MAC source addresses:

  ```
  [edit]
  user@switch# set policy-options mac-list address-list-name mac-address
  ```

2. Specify the policy name:

   ```
   [edit]
   user@switch# set forwarding-options access-security router-advertisement-guard policy policy-name
   ```

3. Specify the accept action:

   ```
   [edit forwarding-options access-security router-advertisement-guard policy policy-name]
   user@switch# set accept
   ```

4. Specify whether RA guard must meet the criteria in all lists or in any of the lists configured in 1:

   - To match on all lists:

     ```
     [edit forwarding-options access-security router-advertisement-guard policy policy-name accept]
     user@switch# set match-list match-criteria match-all
     ```

   - To match on any of the lists:

     ```
     [edit forwarding-options access-security router-advertisement-guard policy policy-name accept]
     user@switch# set match-list match criteria match-any
     ```

5. Associate the accept policy with the list or lists configured in Step 1. For example:

   ```
   [edit forwarding-options access-security router-advertisement-guard policy policy-name accept
   user@switch# set match-list source-mac-address-list address-list-name
   ```

To configure an accept policy for RA guard using the `match-option` option:

1. Specify the policy name:

```
[edit]
user@switch# set forwarding-options access-security router-advertisement-guard policy policy-
name
```

2. Specify the accept action:

```
[edit forwarding-options access-security router-advertisement-guard policy policy-name]
user@switch# set accept
```

3. Specify the match conditions by using the `match-option` option. For example, to specify a match on the maximum number of hops:

```
[edit forwarding-options access-security router-advertisement-guard policy policy-name accept]
user@switch# set match-option hop-limit maximum value
```

Enabling Stateless RA Guard on an Interface

You can enable stateless RA guard on an interface. You must first configure a policy, which is applied to incoming RA messages on the interface or interfaces. After you apply a policy to an interface, you must also enable RA guard on the corresponding VLAN; otherwise, the policy applied to the interface does not have any impact on received RA packets.

To enable stateless RA guard on an interface:

1. Apply a policy to an interface:

```
[edit]
user@switch# set forwarding-options access-security router-advertisement-guard interface
interface-name policy policy-name
```

2. Configure the `stateless` option on the interface:

```
[edit forwarding-options access-security router-advertisement-guard interface interface-name
policy policy-name]
user@switch# set stateless
```

3. Enable stateless RA guard on the corresponding VLAN:

```
[edit]
user@switch# set forwarding-options access-security router-advertisement-guard vlans vlan-
name  policy policy-name stateless
```

Enabling Stateless RA Guard on a VLAN

You can enable stateless RA guard on a per-VLAN basis or for all VLANs. You must first configure a policy, which is used to validate incoming RA messages in the learning state.

To enable stateless RA guard on a specific VLAN:

1. Apply a policy to a VLAN.

```
[edit]
user@switch# set forwarding-options access-security router-advertisement-guard vlans vlan-
name policy policy-name
```

2. Configure the `stateless` option on the VLAN:

```
[edit forwarding-options access-security router-advertisement-guard vlans vlan-name policy
policy-name]
user@switch# set stateless
```

To enable stateless RA guard on all VLANs:

1. Apply a policy to all VLANs.

```
[edit]
user@switch# set forwarding-options access-security router-advertisement-guard vlans all
policy policy-name
```

NOTE: If a policy has been configured for a specific VLAN using the command `set forwarding-options access-security router-advertisement-guard vlans` *`vlan-name`* `policy` *`policy-name`*, that policy takes priority over the policy applied globally to all VLANs.

2. Configure the `stateful` option on all VLANs:

```
[edit forwarding-options access-security router-advertisement-guard vlans all policy policy-
name]
user@switch# set stateful
```

Configuring an Interface as Trusted or Blocked to Bypass Inspection by RA Guard

You can configure an interface as trusted or blocked to bypass inspection of RA messages by RA guard. When an RA message is received on a trusted or blocked interface, it is not subject to validation against the configured policy. A trusted interface forwards all RA messages. A blocked interface discards all RA messages.

- To configure an interface as trusted:

```
[edit]
user@switch# set forwarding-options access-security router-advertisement-guard interface
interface-name mark-interface trusted
```

- To configure an interface as blocked:

```
[edit]
user@switch# set forwarding-options access-security router-advertisement-guard interface
interface-name mark-interface block
```

RELATED DOCUMENTATION

Understanding IPv6 Router Advertisement Guard | **614**

Configuring Stateful IPv6 Router Advertisement Guard | **618**

10

PART

Control Plane Distributed Denial-of-Service (DDoS) Protection and Flow Detection

CHAPTER 23

Control Plane DDoS Protection

IN THIS CHAPTER

Control Plane Distributed Denial-of-Service (DDoS) Protection Overview

IN THIS SECTION

A denial-of-service (DoS) attack is any attempt to deny valid users access to network or server resources by using up all the resources of the network element or server. Distributed denial-of-service (DDoS) attacks involve an attack from multiple sources, enabling a much greater amount of traffic to attack the network. The attacks typically use network protocol control packets to trigger a large number of exceptions to the device's control plane. This results in an excessive processing load that disrupts normal network operations.

With a single point of DDoS protection management, network administrators can customize profiles for their network control traffic. For routers, protection and monitoring persists across *graceful Routing Engine switchover* (GRES) and unified in-service-software-upgrade (ISSU) switchovers. Protection is not diminished as the number of subscribers increases.

Host-bound Traffic Policers for DDoS Violations

To protect the control plane against DDoS attacks, devices have policers enabled by default for host-bound traffic. If needed, you can modify many policer default values. Host-bound traffic is traffic destined to the Routing Engine, including protocol control packets for routing protocols, such as OSPF and BGP. Traffic destined to router IP addresses is also considered host-bound traffic.

The policers specify rate limits for all control traffic for a given protocol, or, in some cases, for specific control packet types for a protocol. You can monitor policer actions for packet types and protocol groups at the level of the device, Routing Engine, and line cards. You can also control logging of policer events.

Devices drop control traffic when it exceeds default or configured policer values. When a DDoS violation occurs, the device will not stop processing packets; it only limits their rate. Each violation immediately generates a notification to alert operators about a possible attack. The device counts the violation and notes the time that the violation starts and the time of the last observed violation. When the traffic rate drops below the bandwidth violation threshold, a recovery timer determines when the traffic flow is considered to have returned to normal. If no further violation occurs before the timer expires, the device clears the violation state and generates a notification.

NOTE: On PTX routers and QFX Series switches, the timer is set to 300 seconds and cannot be modified.

The first line of protection is the policer on the Packet Forwarding Engine (PFE). On devices with multiple line cards, policer states and statistics from each line card are relayed to the Routing Engine and aggregated. The policer states are maintained during a switchover. Although line card statistics and violation counts are preserved during a switchover, Routing Engine policer statistics are not. Control traffic arriving from all ports of the line card converges on the Packet Forwarding Engine, where it is policed, dropping excess packets before they reach the Routing Engine and ensuring the Routing Engine receives only the amount of traffic it can process.

ACX Series routers that support this feature only support aggregate policers, and don’t support policing at the line card level. You can change the default policer values at the Routing Engine level globally or for specific protocol groups, which propagates down to the PFE chipset level. However, you can’t apply additional scaling parameters at the line card level like on other devices. You can disable policing at the Routing Engine level globally or for specific protocol groups. Disabling policing globally effectively disables control plane DDoS protection on the device.

QFX10000 Series switches and PTX Series routers enforce DDoS protection limits at three levels, in the PFE chipset, line card, and the Routing Engine.

In addition to providing notification of violations through event logging, control plane DDoS protection allows you to monitor policers, obtaining information such as policer configuration, number of violations encountered, date and time of violations, packet arrival rates, and number of packets received or dropped.

> **NOTE**: Control plane DDoS protection policers act on the system's traffic queues. The QFX5100 and QFX5200 lines of switches manage traffic for more protocols than the number of queues, so the system often must map more than one protocol to the same queue. When traffic for one protocol shares a queue with other protocols and violates DDoS protection policer limits, these devices report a violation on that queue for all mapped protocols because the system doesn't distinguish which protocol's traffic specifically caused the violation. You can use what you know about the types of traffic flowing through your network to identify which of the reported protocols actually triggered the violation.

Platform Support

In Junos OS Release 14.2 and later releases, control plane DDoS protection is supported on specific platforms. In general, some models of the following platforms have control plane DDoS protection enabled by default and support configuration options to change default policer parameters:

- ACX Series routers.
- EX9200 switches.
- MX Series routers that have only MPCs installed.
- MX Series routers with a built-in MPC.

  > **NOTE**: For simplicity, where the text refers to line cards or line card policers, for these routers that means the built-in MPC.
  >
  > Because these routers do not have FPC slots, information displayed in `FPC` fields by `show` commands actually refers to TFEB.

- PTX Series routers that have only PE-based FPCs installed (PTX3000, PTX5000, PTX1000, and PTX10000) support control plane DDoS protection starting in Junos OS Release 17.4R1.

  PTX10002 routers support control plane DDoS protection starting in Junos OS Release 18.2R1.

PTX10003 routers support control plane DDoS protection starting in Junos OS Evolved Release 19.3R1.

PTX10004 routers support control plane DDoS protection starting in Junos OS Evolved Release 20.3R1.

PTX10008 routers support control plane DDoS protection starting in Junos OS Evolved Release 20.1R1.

- QFX Series switches, including the QFX5100 line, QFX5200 line, and the QFX10000 line of switches.

  QFX10002-60C switches support control plane DDoS protection starting in Junos OS Release 18.1R1.

- T4000 routers that have only Type 5 FPCs installed.

NOTE:

- On Junos Evolved platforms you must configure the `inet` and/or `inet6` protocol family on the device's lo0 interface for DDoS protection to work for those protocol families, respectively.
- Some EX Series switches might have control plane DDoS protection but don’t support CLI options to show or change the default policer parameters.
- For router platforms that have other line cards in addition to MPCs (MX Series), Type 5 FPCs (T4000), or PE-based FPCs (PTX3000, PTX5000, PTX1000, and PTX10000), the CLI accepts the configuration but the other line cards are not protected, so the router is not protected.
- Control plane DDoS protection support for Enhanced Subscriber Management was added in Junos OS Release 17.3R1 on routing platforms.
- To change default-configured control plane DDoS protection parameters for supported protocol groups and packet types, supporting ACX Series routers, PTX Series routers and QFX Series switches have CLI configuration options that differ significantly from the options available for MX Series and T4000 routers. See the following configuration statements for the available configuration options on different devices:
  - For routing devices except PTX Series routers, see *protocols (DDoS)*.

- For ACX Series routers, PTX Series routers and QFX Series switches, see *protocols (DDoS) (ACX Series, PTX Series, and QFX Series)*.

Policer Types and Packet Priorities

Control plane DDoS protection includes two types of policers:

- An *aggregate policer* is applied to the complete set of packet types that belong to a protocol group. For example, you can configure an aggregate policer that applies to all PPPoE control packet types or to all DHCPv4 control packet types. You can specify bandwidth (packets per second [pps]) and burst (packets in a burst) limits, scale the bandwidth and burst limits, and set a traffic priority for aggregate policers. An aggregate policer is supported by all protocol groups.

- An *individual policer*, also referred to as a *packet-type policer*, is allocated for each control packet type within a protocol group. For example, you can configure a policer for one or more types of PPPoE control packets, RADIUS control packets, or multicast snooping packets. You can specify bandwidth (pps) and burst (packets) limit values, scale the bandwidth and burst limits, and set a traffic priority for packet-type policers. Individual policers are available for some protocol groups.

Protocol group and packet type support varies across platforms and Junos OS releases, as follows:

- For routing devices except PTX Series routers, see *protocols (DDoS)*.

- For ACX Series routers, PTX Series router, and QFX Series switches, see *protocols (DDoS) (ACX Series, PTX Series, and QFX Series)*.

A control packet is policed first by its individual policer (if supported) and then by its aggregate policer. A packet dropped by the individual policer never reaches the aggregate policer. A packet that passes the individual policer can subsequently be dropped by the aggregate policer.

NOTE: ACX Series routers only support the aggregate policer for any supported protocol groups.

Packet types within a protocol group have a default, configurable priority: low, medium, or high. Each control packet competes with other packets for the bandwidth within the packet rate limit imposed by its aggregate policer based on the priority configured for each packet type in the protocol group.

The priority mechanism is absolute. High-priority traffic gets bandwidth in preference to medium-priority and low-priority traffic. Medium-priority traffic gets bandwidth in preference to low-priority traffic. Low-priority traffic can use only the bandwidth left by high-priority and medium-priority traffic. If higher priority traffic takes all of the bandwidth, then all the lower priority traffic is dropped.

In releases before Junos OS Release 23.2R1, on MX Series devices, the type of line card in the device drives the distributed denial of service (DDoS) priority of incoming protocols. Starting in Junos OS

Release 23.2R1, the device determines the DDoS priority of a protocol based on the DDoS parameters table. This enhancement enables the device to treat all packets of a particular protocol the same by default, regardless of the device's line card. You can modify the DDoS parameters table using CLI. This feature improves consistency in the way devices in the network prioritize protocols to protect against DDoS attacks.

Policer Priority Behavior Example

For example, on a device that supports control plane DDoS protection for the PPPoE protocol group, consider how you might configure packet types within this protocol group. Ignoring other PPPoE packet types for this example, suppose you configure individual policers for PADI and PADT packets, as well as a PPPoE aggregate policer for all those packets. You prioritize PADT packets over PADI packets because PADT packets enable the PPPoE application to release resources to accept new connections. Therefore, you assign high priority to the PADT packets and low priority to the PADI packets.

The aggregate policer imposes a total packet rate limit for the protocol group. PADT packets passed by their individual policer have access to that bandwidth before PADI packets passed by their individual policer, because the PADT packets have a higher priority. If so many PADT packets are passed that they use all the available bandwidth, then all the PADI packets are dropped, because there is no bandwidth remaining at the aggregate policer.

Policer Hierarchy Example

Control plane DDoS policers are organized to match the hierarchical flow of protocol control traffic. Control traffic arriving from all ports of a line card converges on the Packet Forwarding Engine. Control traffic from all line cards on the router converges on the Routing Engine. Similarly, the DDoS policers are placed hierarchically along the control paths so that excess packets are dropped as early as possible on the path. This design preserves system resources by removing excess, malicious traffic so that the Routing Engine receives only the amount of traffic that it can process.

To implement this design on MX Series routers, for example, five DDoS policers are present: One on the Packet Forwarding Engine (the chipset), two at the line card, and two at the Routing Engine. An aggregate policer is also present on the Packet Forwarding Engine for some protocol groups, for a total of six policers; for simplicity, the text follows the general case. For example, Figure 38 on page 637 shows the policer process for PPPoE traffic. Figure 39 on page 638 shows the policer process for DHCPv4 traffic. (The same process applies to DHCPv6 traffic as well.)

NOTE: Recall that PTX Series routers and QFX Series switches have a simpler design with policers in the Packet Forwarding Engine only. PTX10003 and PTX10008 routers enforce control plane DDoS protection limits at three levels, two at the Packet

Forwarding Engine chipset and line card levels and one at the Routing Engine level. However, packet type and aggregate policers operate similarly on all of these platforms.

Figure 38: Policer Hierarchy for PPPoE Packets

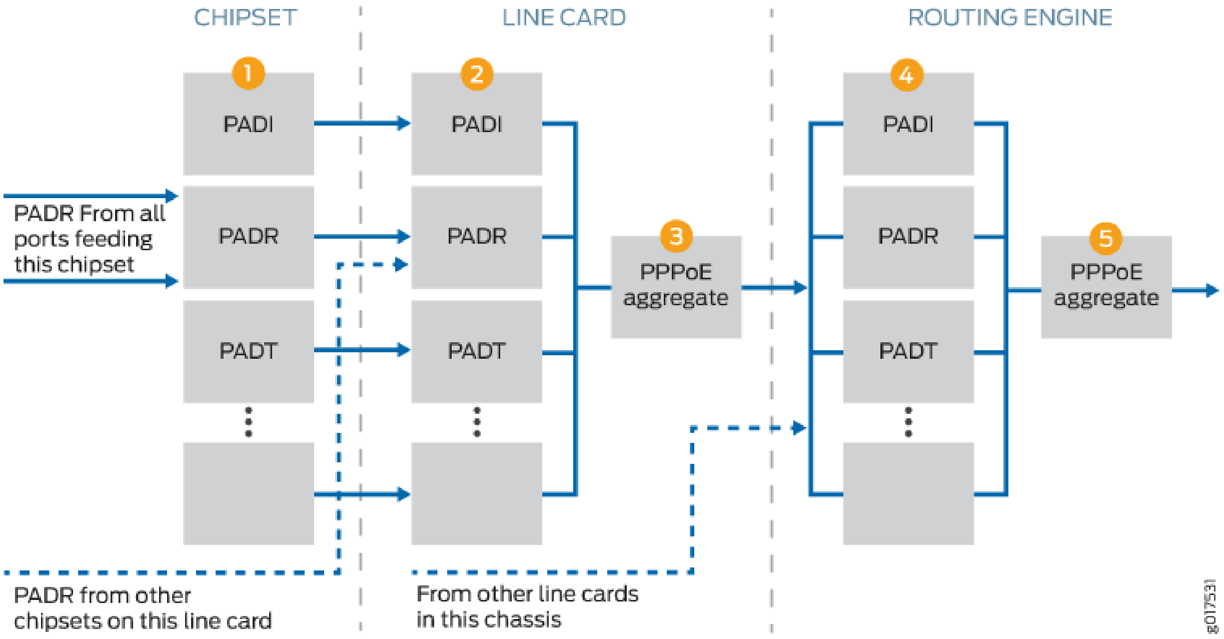
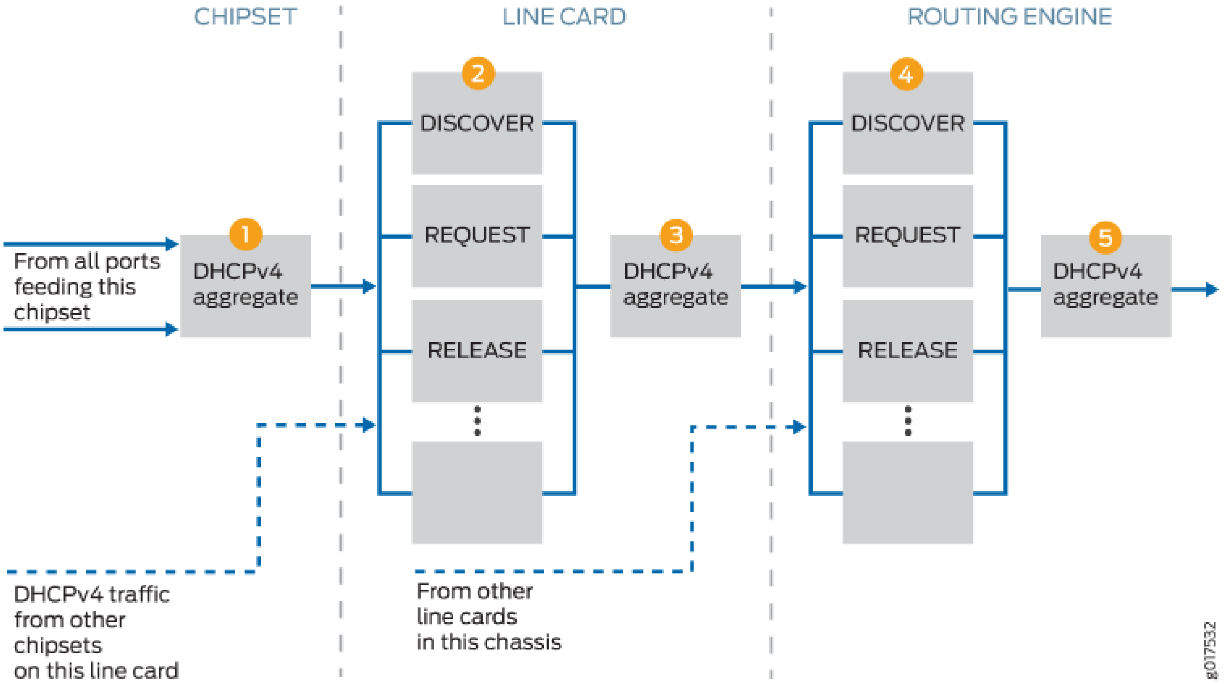

CHIPSET
LINE CARD
ROUTING ENGINE
1
PADI
PADR
PADT
2
PADI
PADR
PADT
3
PPPoE aggregate
4
PADI
PADR
PADT
5
PPPoE aggregate
PADR From all ports feeding this chipset
PADR from other chipsets on this line card
From other line cards in this chassis
g017531

Figure 39: Policer Hierarchy for DHCPv4 Packets



Control packets arrive at the Packet Forwarding Engine for processing and forwarding. The first policer (1) is either an individual policer (Figure 38 on page 637) or an aggregate policer (Figure 39 on page 638).

- The first policer is an individual policer for protocol groups that support individual policers, with two exceptions. For DHCPv4 and DHCPv6 traffic, the first policer is an aggregate policer.

- The first policer is an aggregate policer for protocol groups that support only aggregate policers.

Traffic that passes the first policer is monitored by one or both of the line card policers. If the card has more than one Packet Forwarding Engine, traffic from all Packet Forwarding Engines converges on the line card policers.

- When the traffic belongs to a protocol group that supports individual policers, it passes through the line card individual policer (2) and then the line card aggregate policer (3). Traffic that passes the individual policer can be dropped by the aggregate policer. Although DHCPv4 and DHCPv6 traffic was monitored by an aggregate policer at the Packet Forwarding Engine, at the line card it is handled like other protocols that support individual policers.

- When the traffic belongs to a protocol group that supports only aggregate policers, only the line card aggregate policer monitors the traffic.

Traffic that passes the line card policers is monitored by one or both of the Routing Engine policers. Traffic from all the line cards converges on the Routing Engine policers.

- When the traffic belongs to a protocol group that supports individual policers, it passes through the Routing Engine individual policer (4) and then the Routing Engine aggregate policer (5). Traffic that passes the individual policer can be dropped by the aggregate policer. As it was at the line card level, DHCPv4 and DHCPv6 traffic at the Routing Engine is handled like other protocols that support individual policers.

- When the traffic belongs to a protocol group that supports only aggregate policers, only the aggregate policer monitors the traffic.

With this design, three policers evaluate the traffic for protocol groups that support only aggregate policers. Among other groups, this includes ANCP, dynamic VLAN, FTP, and IGMP traffic. Traffic for protocol groups that support both aggregate and individual policers is evaluated by all five policers. Among other groups, this includes DHCPv4, MLP, PPP, PPPoE, and *virtual chassis* traffic.

Figure 38 on page 637 shows how control plane DDoS protection polices PPPoE control packets:

1. PADR packets, for example, are evaluated at the first policer on the Packet Forwarding Engine to determine whether they are within the packet rate limits. PADR packets that exceed the limit are dropped.

2. All PADR packets that pass the policer on all Packet Forwarding Engines on the line card are next evaluated by the line card individual policer. PADR packets that exceed the limit are dropped.

3. All PADR packets that pass the line card individual policer proceed to the line card aggregate policer. PADR packets that exceed the limit are dropped.

4. All PADR packets that are passed by the line card aggregate policers on all line cards on the router proceed to the Routing Engine individual policer. PADR packets that exceed the limit are dropped.

5. Finally, all PADR packets that are passed by the Routing Engine individual policer proceed to the Routing Engine aggregate policer. PADR packets that exceed the limit are dropped. PADR packets that are not dropped here are passed along as safe, normal traffic.

By default, all three individual policers (Packet Forwarding Engine, line card, and Routing Engine) have the same packet rate limit for a given packet type. With this design, all the control traffic from a Packet Forwarding Engine and line card can reach the Routing Engine as long as there is no competing traffic of the same type from other Packet Forwarding Engines or line cards. When competing traffic is present, excess packets are dropped at the convergence points. That is, they are dropped at the line card for all competing Packet Forwarding Engines and at the Routing Engine for all competing line cards.

Example of Policer Behavior to Limit Packet Rate

For example, suppose you set the policer `bandwidth` option for PADI packets to 1000 packets per second. This value applies to the individual PADI policers at the Packet Forwarding Engine, the line card, and the Routing Engine. If only the card in slot 5 is receiving PADI packets, then up to 1000 PADI pps can reach the Routing Engine (if the PPPoE aggregate policer is not exceeded). However, suppose the card in slot 9 is also receiving PADI packets at 1000 pps and that its PPPoE aggregate policer is not exceeded. The traffic passes the individual and aggregate policers at both line cards and proceeds to the Routing Engine. At the Routing Engine, the combined packet rate is 2000 pps. Because the PADI policer at the Routing Engine allows only 1000 PADI pps to pass, it drops the excess 1000 packets. It continues to drop the excess packets for as long as the bandwidth (pps) limit is exceeded.

You can apply a scaling factor for both the bandwidth (pps) limit and the burst (packets in a burst) limit at the line card to fine-tune the traffic limits for each slot. For example, suppose the individual policer sets the PADI packet rate to 1000 pps and the burst size to 50,000 packets. You can reduce the traffic limit for PADI packets on any line card by specifying the slot number and scaling factor. A bandwidth scaling factor of 20 for slot 5 reduces the traffic in this example to 20 percent of 1000 pps, or 200 pps for the line card in that slot. Similarly, a burst scaling factor of 50 for that slot reduces the burst size by 50 percent to 25,000 packets. By default, scaling factors are set to 100 so traffic can pass through at 100 percent of the rate limit.

Control Plane DDoS Protection Compared to Subscriber Login Packet Overload Protection

In addition to the control plane DDoS protection capability, MX Series routers also have a built-in subscriber login overload protection mechanism. The login overload protection mechanism (also called a load-throttling mechanism) monitors the incoming subscriber login packets and admits only what the system is capable of handling in accordance with the prevailing load on the system. Packets in excess of what the system can handle are discarded. By shedding this excess load, the system is able to maintain optimal performance and prevent any degradation of login-completion rate under overload conditions. This mechanism uses minimal resources and is enabled by default; no user configuration is required.

The protection provided by this mechanism is secondary to what control plane DDoS protection provides as a first level of defense against high rates of incoming packets. Control plane DDoS protection operates on the Packet Forwarding Engine and protects against all packet types of all protocols. In contrast, the login overload protection mechanism is located on the Routing Engine and specifically operates only on incoming connection-initiation packets such as DHCPv4 DHCPDISCOVER, DHCPv6 SOLICIT, and PPPoE PADI packets.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
20.3R1	PTX10004 routers support control plane DDoS protection starting in Junos OS Evolved Release 20.3R1.
19.4R1-S1	PTX10008 routers support control plane DDoS protection starting in Junos OS Evolved Release 20.1R1.
19.3R1	PTX10003 routers support control plane DDoS protection starting in Junos OS Evolved Release 19.3R1.
18.2R1	PTX10002 routers support control plane DDoS protection starting in Junos OS Release 18.2R1.
18.2R1	QFX10002-60C switches support control plane DDoS protection starting in Junos OS Release 18.1R1.
17.4R1	PTX Series routers that have only PE-based FPCs installed (PTX3000, PTX5000, PTX1000, and PTX10000) support control plane DDoS protection starting in Junos OS Release 17.4R1.
17.3R1	Control plane DDoS protection support for Enhanced Subscriber Management was added in Junos OS Release 17.3R1 on routing platforms.
14.2	In Junos OS Release 14.2 and later releases, control plane DDoS protection is supported on specific platforms.

RELATED DOCUMENTATION

Configuring Control Plane DDoS Protection | 642

Control Plane DDoS Protection Flow Detection Overview | 678

Configuring Control Plane DDoS Protection

IN THIS SECTION

Control plane DDoS protection is enabled by default for all supported protocol groups and packet types. Devices have default values for bandwidth (packet rate in pps), bandwidth scale, burst (number of packets in a burst), burst scale, priority, and recover time. To see the default policer values for all supported protocol groups and packet types, run the `show ddos-protection protocols` CLI command before modifying any configurable DDoS protection values.

NOTE: EX2300 and EX2300-C switches might have control plane DDoS protection but don't support CLI options to show or change the default policer parameters.

Starting in Junos OS Release 24.2R1 the EX4100 and EX4400 devices and starting in Junos OS Release 24.4R1 the EX3400 and EX4300-MP devices support CLI options to show or change the default policer parameters.

You can change the control plane DDoS configuration parameters as follows:

- For individual packet types supported within a protocol group, you can change bandwidth (pps), burst (packets), and priority policer values.

  NOTE: On PTX10003 and PTX10008 routers, you can change default bandwidth (pps) and burst (packets) values for aggregate or packet type policers, but not priority values.

- For the aggregate policer for a protocol group, you can change bandwidth (pps) and burst (packets) policer values.

- When you set bandwidth (pps), burst (packets), and priority values for a protocol group or packet type policer, the same values apply at all policer levels. Change the scaling configuration options to tune those values at the Packet Forwarding Engine level.

NOTE: ACX Series routers with control plane DDoS protection support changing policer values at the Routing Engine level, which propagates down to the PFE chipset level. They don't support the line card scaling configuration options at the `[edit system ddos-protection protocols` *`protocol-group`* `aggregate fpc` statement hierarchy.

You can disable control plane DDoS protection as follows:

- On most routing devices that have policers at the Routing Engine level, you can disable control plane DDoS protection at the Routing Engine and for all line cards either globally or for individual packet types within a protocol group.

- PTX10003, PTX10008 and PTX10016 routers include policers at the Routing Engine level, but like other PTX Series routers, you can only disable line-card policers.

- On other PTX Series routers and QFX Series switches, policers are supported only at the line cards, so on these devices you can disable control plane DDoS protection for all line cards either globally or for individual packet types within a protocol group.

Control plane DDoS logging is enabled by default, but you can disable it globally for all control plane DDoS events or for individual packet types within a protocol group. You can also configure tracing operations for monitoring control plane DDoS events.

NOTE: MX Series routers with MPCs and T4000 routers with FPC5s support control plane DDoS protection. The CLI accepts the configuration if other line cards are also installed on either of these types of routers, but the other line cards are not protected so the router is essentially not protected.

To change default-configured control plane DDoS protection parameters:

1. (Optional) Configure global control plane DDoS protection settings or disable control plane DDoS protection.

2. (Optional) Configure control plane DDoS protection settings for the aggregate policer or individual packet types for the desired protocol groups.

3. (Optional) Configure tracing for control plane DDoS protection operations.

Disabling Control Plane DDoS Protection Policers and Logging Globally

Control plane DDoS protection policers are enabled by default for all supported protocol groups and packet types.

On ACX Series routers, you can disable policers globally or for individual protocol groups at the Routing Engine level. Disabling policers globally essentially disables control plane DDoS protection on the device.

On MX Series routers, T4000 routers, and EX9200 switches, policers are established at the level of the individual line card and the Routing Engine. You can disable the line card policers globally for all MPCs or FPC5s. You can also disable the Routing Engine policer. When you disable either of these policers, the policers at that level for all protocol groups and packet types are disabled.

On PTX Series routers and QFX Series switches, policers are established at the level of individual line cards only. If you disable line-card policers globally, control plane DDoS protection is disabled on the switch.

PTX10003, PTX10008 and PTX10016 routers include policers at the Routing Engine level, but like other PTX Series routers, you can only disable line-card policers.

Control plane DDoS protection logging is also enabled by default. You can disable all control plane DDoS event logging (including flow detection event logging) for all protocol groups and packet types across the router or switch.

> **NOTE**: The global configuration for disabling policers and logging overrides any local configuration for packet types.

To configure global control plane DDoS protection settings:

1. (Optional) (Not available on ACX Series routers) To disable line card policers:

   ```
   [edit system ddos-protection global]
   user@host# set disable-fpc
   ```

2. (Optional) (Not available on PTX Series Routers or QFX Series switches) To disable Routing Engine policers:

   ```
   [edit system ddos-protection global]
   user@host# set disable-routing-engine
   ```

3. (Optional) To disable event logging:

   ```
   [edit system ddos-protection global]
   user@host# set disable-logging
   ```

Configuring Control Plane DDoS Protection Aggregate or Individual Packet Type Policers

Control plane DDoS policers are applied to control packet traffic and are enabled by default for all supported protocol groups and packet types. You can change default policer parameters to configure different values for the maximum allowed traffic rate, maximum burst size, traffic priority, and how much time must pass since the last violation before the traffic flow is considered to have recovered from the attack. You can also scale the bandwidth and burst values for individual line cards so that the policers at this level trigger at lower thresholds than the overall protocol or packet thresholds.

Protocol group and packet type support varies across platforms and Junos OS releases, as follows:

- For most routing devices, see *protocols (DDoS)*.
- For ACX Series routers, PTX Series routers and QFX Series switches, see *protocols (DDoS) (ACX Series, PTX Series, and QFX Series)*.

You can configure aggregate policer values for any protocol group. The aggregate policer applies to the combination of all types of control packet traffic for that group.

NOTE: ACX Series routers only support the aggregate policer for any supported protocol groups.

For some protocol groups, you can also configure policer values for individual packet types. When you configure aggregate policer values for certain protocol groups, you can optionally bypass that policer for one or more particular packet types in that group.

BEST PRACTICE: Although all policers have default parameter values, these values might not accurately reflect the control traffic pattern of your network. We recommend that you model your network to determine the best values for your situation. Before you configure policers for your network, you can quickly view the default values for all packet types from operational mode using the `show ddos-protection protocols parameters brief` command. You can also use the command to specify a single protocol group of interest. For example, to see default values for the `dhcpv4` protocol group, use the `show ddos-protection protocols dhcpv4 parameters brief` command.

You can disable a packet type policer at either the Routing Engine level (if supported) or at the Packet Forwarding Engine level (if supported) for a specified line card or for all line cards. You can also disable logging of all control plane DDoS protection events for individual packet types within a protocol group.

To configure the desired aggregate or packet-type DDoS protection policer settings:

1. Specify the protocol group.

```
[edit system ddos-protection protocols]
user@host# edit protocol-group
```

For example, to specify the DHCPv4 protocol group on MX Series, PTX10003 or PTX10008 routers:

```
[edit system ddos-protection protocols]
user@host# edit dhcpv4
```

or, on ACX Series, PTX Series, and QFX Series devices, control plane DDoS protection support has a combined DHCPv4 and DHCPv6 option that allows only aggregate policer configuration:

```
[edit system ddos-protection protocols]
user@host# edit dhcpv4v6
```

2. Specify a supported individual packet type or the `aggregate` option to encompass all packet types in the protocol group.

```
[edit system ddos-protection protocols protocol-group]
user@host# set packet-type
```

or

```
[edit system ddos-protection protocols protocol-group]
user@host# set aggregate
```

For example, to specify only DHCPv4 release packets on devices that support individual DHCPv4 packet types:

```
[edit system ddos-protection protocols dhcpv4]
user@host# edit release
```

3. (Optional) Configure the maximum traffic rate the policer allows for the packet type (or aggregate).

```
[edit system ddos-protection protocols protocol-group (aggregate | packet-type)]
user@host# set bandwidth packets-per-second
```

For example, to set a bandwidth of 600 packets per second for DHCPv4 release packets:

```
[edit system ddos-protection protocols dhcpv4 release]
user@host# set bandwidth 600
```

4. (Optional) Configure the maximum number of packets of this packet type (or aggregate) that the policer allows in a burst of traffic.

```
[edit system ddos-protection protocols protocol-group (aggregate | packet-type)]
user@host# set burst size
```

For example, to set a maximum of 5000 DHCPv4 release packets:

```
[edit system ddos-protection protocols dhcpv4 release]
user@host# set burst 5000
```

5. (Optional) Set the traffic priority.

NOTE: You can’t change default priority values on PTX10003 or PTX10008 routers.

```
[edit system ddos-protection protocols protocol-group (aggregate | packet-type)]
user@host# set priority level
```

For example, to specify a medium priority for DHCPv4 release packets:

```
[edit system ddos-protection protocols dhcpv4 release]
user@host# set priority medium
```

6. (Optional) Configure how much time must pass since the last violation before the traffic flow is considered to have recovered from the attack.

```
[edit system ddos-protection protocols protocol-group (aggregate | packet-type)]
user@host# set recover-time seconds
```

For example, to specify that 600 seconds must have passed since the last violation of the DHCPv4 release packet policer:

```
[edit system ddos-protection protocols dhcpv4 release]
user@host# set recover-time 600
```

7. (Optional, supported on some devices) Bypass the aggregate policer configuration. This is applicable only when an aggregate policer and an individual policer is configured for the protocol group.

```
[edit system ddos-protection protocols protocol-group packet-type]
user@host# set bypass-aggregate
```

For example, to bypass the aggregate policer for DHCPv4 renew packets:

```
[edit system ddos-protection protocols dhcpv4 renew]
user@host# set bypass-aggregate
```

8. (Optional, supported on some devices) Disable line card policers for the packet type (or aggregate) on all line cards.

```
[edit system ddos-protection protocols protocol-group (aggregate | packet-type)]
user@host# set disable-fpc
```

NOTE: When you disable line card policers globally at the `[edit system ddos-protection global]` hierarchy level, the global setting overrides the per-packet type setting shown in this step. If you subsequently remove the global configuration, then the per-packet type configuration takes effect.

For example, to disable the line card policer for DHCPv4 bootp packets:

```
[edit system ddos-protection protocols dhcpv4 bootp]
user@host# set disable-fpc
```

9. (Optional) Disable control plane DDoS protection event logging for only one packet type (or aggregate).

```
[edit system ddos-protection protocols protocol-group (aggregate | packet-type)]
user@host# set disable-logging
```

NOTE: Events disabled for the packet are associated with policer violations; logging of flow detection culprit flow events is not affected by this statement.

NOTE: When you disable control plane DDoS protection event logging globally at the `[edit system ddos-protection global]` hierarchy level, the global setting overrides the per-packet type setting shown in this step. If you subsequently remove the global configuration, then the per-packet type configuration takes effect.

For example, to disable control plane DDoS protection event logging on the line card policer for DHCPv4 discover packets:

```
[edit system ddos-protection protocols dhcpv4 discover]
user@host# set disable-logging
```

10. (Optional, not available on PTX Series Routers or QFX Series switches) Disable the Routing Engine policer for only this packet type.

```
[edit system ddos-protection protocols protocol-group packet-type]
user@host# set disable-routing-engine
```

NOTE: When you disable the Routing Engine policer globally at the `[edit system ddos-protection global]` hierarchy level, the global setting overrides the per-packet type setting shown in this step. If you subsequently remove the global configuration, then the per-packet type configuration takes effect.

For example, to disable the Routing Engine policer for DHCPv4 discover packets:

```
[edit system ddos-protection protocols dhcpv4 discover]
user@host# set disable-routing-engine
```

11. (Optional, not supported on ACX Series routers) Configure packet-level settings for the packet type (or aggregate) on a single line card. On switches with a single, fixed line card (a single FPC considered to be in slot 0 and labeled `fpc0`), scaling the policer values affects the entire switch.

    ```
    [edit system ddos-protection protocols protocol-group (aggregate | packet-type)]
    user@host# edit fpc slot-number
    ```

    For example, to access DHCPv4 discover packet settings on the line card in slot 3:

    ```
    [edit system ddos-protection protocols dhcpv4 discover]
    user@host# edit fpc 3
    ```

12. (Optional, not supported on ACX Series routers) Scale the policer bandwidth for the packet type (or aggregate) on the line card.

    ```
    [edit system ddos-protection protocols protocol-group (aggregate | packet-type) fpc slot-
    number]
    user@host# set bandwidth-scale percentage
    ```

    For example, to scale the bandwidth to 80 percent of the all-line-card setting configured for DHCPv4 discover packets on the line card in slot 3:

    ```
    [edit system ddos-protection protocols dhcpv4 discover fpc 3]
    user@host# edit bandwidth-scale 80
    ```

13. (Optional, not supported on ACX Series routers) Scale the policer burst size for the packet type (or aggregate) on the line card.

    ```
    [edit system ddos-protection protocols protocol-group (aggregate | packet-type) fpc slot-
    number]
    user@host# set burst-scale percentage
    ```

    For example, to scale the maximum bandwidth to 75 percent of the all-line-card setting configured for DHCPv4 discover packets on the line card in slot 3:

    ```
    [edit system ddos-protection protocols dhcpv4 discover fpc 3]
    user@host# edit burst-scale 75
    ```

14. (Optional, not supported on ACX Series routers) Disable the line card policer for the packet type (or aggregate) on a particular line card.

```
[edit system ddos-protection protocols protocol-group (aggregate | packet-type) fpc slot-
number]
user@host# set disable-fpc
```

For example, to disable the line card policer for DHCPv4 discover packets on the line card in slot 3:

```
[edit system ddos-protection protocols dhcpv4 discover fpc 3]
user@host# edit disable-fpc
```

SEE ALSO

Control Plane Distributed Denial-of-Service (DDoS) Protection Overview | 631
Example: Configuring Control Plane DDoS Protection | 657
Example: Configuring Control Plane DDoS Protection on QFX Series Switches | 671

Verifying and Managing Control Plane DDoS Protection

IN THIS SECTION

Purpose

View or clear information about control plane DDoS protection configurations, states, and statistics.

Action

- To display the control plane DDoS protection policer configuration, violation state, and statistics for all packet types in all protocol groups:

```
user@host> show ddos-protection protocols
```

Run this command before you make any configuration changes to see the default policer values.

- To display the control plane DDoS protection policer configuration, violation state, and statistics for a particular packet type in a particular protocol group:

```
user@host> show ddos-protection protocols protocol-group packet-type
```

- To display only the number of control plane DDoS protection policer violations for all protocol groups:

```
user@host> show ddos-protection protocols violations
```

- To display a table of the control plane DDoS protection configuration for all packet types in all protocol groups:

```
user@host> show ddos-protection protocols parameters brief
```

- To display a complete list of packet statistics and control plane DDoS protection violation statistics for all packet types in all protocol groups:

```
user@host> show ddos-protection protocols statistics detail
```

- To display global control plane DDoS protection violation statistics:

```
user@host> show ddos-protection statistics
```

- To display the control plane DDoS protection version number:

```
user@host> show ddos-protection version
```

- To clear control plane DDoS protection statistics for all packet types in all protocol groups:

```
user@host> clear ddos-protection protocols statistics
```

- To clear control plane DDoS protection statistics for all packet types in a particular protocol group:

  ```
  user@host> clear ddos-protection protocols protocol-group statistics
  ```

- To clear control plane DDoS protection statistics for a particular packet type in a particular protocol group:

  ```
  user@host> clear ddos-protection protocols protocol-group packet-type  statistics
  ```

- To clear control plane DDoS protection violation states for all packet types in all protocol groups:

  ```
  user@host> clear ddos-protection protocols states
  ```

- To clear control plane DDoS protection violation states for all packet types in a particular protocol group:

  ```
  user@host> clear ddos-protection protocols protocol-group states
  ```

- To clear control plane DDoS protection violation states for a particular packet type in a particular protocol group:

  ```
  user@host> clear ddos-protection protocols protocol-group packet-type  states
  ```

SEE ALSO

Verifying and Managing Flow Detection | 691

Tracing Control Plane DDoS Protection Operations

IN THIS SECTION

The Junos OS trace feature tracks control plane DDoS protection operations and records events in a log file. The error descriptions captured in the log file provide detailed information to help you solve problems.

By default, nothing is traced. When you enable the tracing operation, the default tracing behavior is as follows:

1. Important events are logged in a file located in the `/var/log` directory. By default, the router uses the filename `jddosd`. You can specify a different filename, but you cannot change the directory in which trace files are located.

2. When the trace log file *filename* reaches 128 kilobytes (KB), it is compressed and renamed *filename*`.0.gz`. Subsequent events are logged in a new file called *filename*, until it reaches capacity again. At this point, *filename*`.0.gz` is renamed *filename*`.1.gz` and *filename* is compressed and renamed *filename*`.0.gz`. This process repeats until the number of archived files reaches the maximum file number. Then the oldest trace file—the one with the highest number—is overwritten.

   You can optionally specify the number of trace files to be from 2 through 1000. You can also configure the maximum file size to be from 10 KB through 1 gigabyte (GB). (For more information about how log files are created, see the System Log Explorer.)

By default, only the user who configures the tracing operation can access log files. You can optionally configure read-only access for all users.

This topic describes how you can configure all aspects of control plane DDoS protection tracing operations.

Configuring the Control Plane DDoS Protection Trace Log Filename

By default, the name of the file that records trace output for control plane DDoS protection is `jddosd`. You can specify a different name with the `file` option.

To configure the filename for subscriber management database tracing operations:

- Specify the name of the file used for the trace output.

```
[edit system ddos-protection traceoptions]
user@host# set file ddos_logfile_1
```

Configuring the Number and Size of Control Plane DDoS Protection Log Files

You can optionally specify the number of compressed, archived trace log files to be from 2 through 1000. You can also configure the maximum file size to be from 10 KB through 1 gigabyte (GB); the default size is 128 kilobytes (KB).

The archived files are differentiated by a suffix in the format .*number*.gz. The newest archived file is .0.gz and the oldest archived file is .*(maximum number)-1*.gz. When the current trace log file reaches the maximum size, it is compressed and renamed, and any existing archived files are renamed. This process repeats until the maximum number of archived files is reached, at which point the oldest file is overwritten.

For example, you can set the maximum file size to 2 MB, and the maximum number of files to 20. When the file that receives the output of the tracing operation, *filename*, reaches 2 MB, *filename* is compressed and renamed *filename*.0.gz, and a new file called *filename* is created. When the new *filename* reaches 2 MB, *filename*.0.gz is renamed *filename*.1.gz and *filename* is compressed and renamed *filename*.0.gz. This process repeats until there are 20 trace files. Then the oldest file, *filename*.19.gz, is simply overwritten when the next oldest file, *filename*.18.gz is compressed and renamed to *filename*.19.gz.

To configure the number and size of trace files:

- Specify the name, number, and size of the file used for the trace output.

```
[edit system ddos-protection traceoptions]
user@host# set file ddos_1_logfile_1 files 20 size 2097152
```

Configuring Access to the Control Plane DDoS Protection Log File

By default, only the user who configures the tracing operation can access the log files. You can enable all users to read the log file and you can explicitly set the default behavior of the log file.

To specify that all users can read the log file:

- Configure the log file to be world-readable.

```
[edit system ddos-protection traceoptions]
user@host# set file ddos_1 _logfile_1 world-readable
```

To explicitly set the default behavior, only the user who configured tracing can read the log file:

- Configure the log file to be no-world-readable.

```
[edit system ddos-protection traceoptions]
user@host# set file ddos_1 _logfile_1 no-world-readable
```

Configuring a Regular Expression for Control Plane DDoS Protection Messages to Be Logged

By default, the trace operation output includes all messages relevant to the logged events.

You can refine the output by including regular expressions to be matched.

To configure regular expressions to be matched:

- Configure the regular expression.

```
[edit system ddos-protection traceoptions]
user@host# set file ddos_1 _logfile_1 match regex
```

Configuring the Control Plane DDoS Protection Tracing Flags

By default, only important events are logged. You can specify which events and operations are logged by specifying one or more tracing flags.

To configure the flags for the events to be logged:

- Configure the flags.

```
[edit system ddos-protection traceoptions]
user@host# set flag flag
```

Configuring the Severity Level to Filter Which Control Plane DDoS Protection Messages Are Logged

The messages associated with a logged event are categorized according to severity level. You can use the severity level to determine which messages are logged for the event type. The severity level that you configure depends on the issue that you are trying to resolve. In some cases you might be interested in seeing all messages relevant to the logged event, so you specify `all` or `verbose`. Either choice generates a large amount of output. You can specify a more restrictive severity level, such as `notice` or `info` to filter the messages . By default, the trace operation output includes only messages with a severity level of `error`.

To configure the type of messages to be logged:

- Configure the message severity level.

```
[edit system ddos-protection traceoptions]
user@host# set level severity
```

Example: Configuring Control Plane DDoS Protection

IN THIS SECTION

This example shows how to configure control plane DDoS protection that enables the router to quickly identify an attack and prevent a flood of malicious control packets from exhausting system resources.

Requirements

Control plane DDoS protection requires the following hardware and software:

- MX Series routers that have only MPCs installed, T4000 Core Routers that have only FPC5s installed, EX9200 switches.

> **NOTE**: If a router has other cards in addition to MPCs or FPC5s, the CLI accepts the configuration but the other cards are not protected and therefore the router is not protected.

- Junos OS Release 11.2 or later

No special configuration beyond device initialization is required before you can configure this feature.

Overview

IN THIS SECTION

Distributed denial-of-service attacks use multiple sources to flood a network or router with protocol control packets. This malicious traffic triggers a large number of exceptions in the network and attempts exhaust the system resources to deny valid users access to the network or server.

This example describes how to configure rate-limiting policers that identify excess control traffic and drop the packets before the router is adversely affected. Sample tasks include configuring policers for particular control packet types within a protocol group, configuring an aggregate policer for a protocol group and bypassing that policer for a particular control packet type, and specifying trace options for DDoS operations.

This example does not show all possible configuration choices.

Topology

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure control plane DDoS protection for protocol groups and particular control packet types, copy the following commands, paste them in a text file, remove any line breaks, and then copy and paste the commands into the CLI.

```
[edit]
edit system
```

```
set ddos-protection protocols dhcpv4 aggregate bandwidth 669
set ddos-protection protocols dhcpv4 aggregate burst 6000
set ddos-protection protocols dhcpv4 discover bandwidth 100
set ddos-protection protocols dhcpv4 discover recover-time 200
set ddos-protection protocols dhcpv4 discover burst 300
set ddos-protection protocols dhcpv4 offer priority medium
set ddos-protection protocols dhcpv4 offer bypass-aggregate
set ddos-protection protocols dhcpv4 offer fpc 1 bandwidth-scale 80
set ddos-protection protocols dhcpv4 offer fpc 1 burst-scale 75
set ddos-protection protocols pppoe aggregate bandwidth 800
set ddos-protection traceoptions file ddos-trace size 10m
set ddos-protection traceoptions flag all
top
```

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 DDoS protection:

1. Specify a protocol group.

   ```
   [edit system ddos-protection protocols]
   user@host# edit dhcpv4
   ```

2. Configure the maximum traffic rate (in packets per second [pps]) for the DHCPv4 aggregate policer; that is, for the combination of all DHCPv4 packets.

   > **NOTE**: You change the traffic rate using the `bandwidth` option. Although the term bandwidth usually refers to bits per second (bps), this feature’s `bandwidth` option represents a packets per second (pps) value.

   ```
   [edit system ddos-protection protocols dhcpv4]
   user@host# set aggregate bandwidth 669
   ```

3. Configure the maximum burst size (number of packets) for the DHCPv4 aggregate policer.

```
[edit system ddos-protection protocols dhcpv4]
user@host# set aggregate burst 6000
```

4. Configure the maximum traffic rate (in pps) for the DHCPv4 policer for discover packets.

```
[edit system ddos-protection protocols dhcpv4]
user@host# set discover bandwidth 100
```

5. Decrease the recover time for violations of the DHCPv4 discover policer.

```
[edit system ddos-protection protocols dhcpv4]
user@host# set discover recover-time 200
```

6. Configure the maximum burst size (number of packets) for the DHCPv4 discover policer.

```
[edit system ddos-protection protocols dhcpv4]
user@host# set discover burst 300
```

7. Increase the priority for DHCPv4 offer packets.

```
[edit system ddos-protection protocols dhcpv4]
user@host# set offer priority medium
```

8. Prevent offer packets from being included in the aggregate bandwidth (pps); that is, offer packets do not contribute towards the combined DHCPv4 traffic to determine whether the aggregate bandwidth (pps) is exceeded. However, the offer packets are still included in traffic rate statistics.

```
[edit system ddos-protection protocols dhcpv4]
user@host# set offer bypass-aggregate
```

9. Reduce the bandwidth (pps) and burst size (packets) allowed before violation is declared for the DHCPv4 offer policer on the MPC or FPC5 in slot 1.

```
[edit system ddos-protection protocols dhcpv4]
user@host# set offer fpc 1 bandwidth-scale 80
user@host# set offer fpc 1 burst-scale 75
```

10. Configure the maximum traffic rate for the PPPoE aggregate policer, that is, for the combination of all PPPoE packets.

```
[edit system ddos-protection protocols dhcpv4]
user@host# up
[edit system ddos-protection protocols]
user@host# set pppoe aggregate bandwidth 800
```

11. Configure tracing for all DDoS protocol processing events.

```
[edit system ddos-protection traceoptions]
user@host# set file ddos-log
user@host# set file size 10m
user@host# set flag all
```

Results

From configuration mode, confirm your configuration by entering the `show ddos-protection` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit system]
user@host# show ddos-protection
traceoptions {
    file ddos-trace size 10m;
    flag all;
}
    protocols {
        pppoe {
            aggregate {
                bandwidth 800;
            }
        }
        dhcpv4 {
            aggregate {
                bandwidth 669;
                burst 6000;
            }
            discover {
                bandwidth 100;
                burst 300;
                recover-time 200;
            }
            offer {
                priority medium;
                fpc 1 {
                    bandwidth-scale 80;
                    burst-scale 75;
                }
                bypass-aggregate;
            }
        }
    }
```

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

Verification

IN THIS SECTION

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

Verifying the DHCPv4 DDoS Protection Configuration and Operation

Purpose

Verify that the DHCPv4 aggregate and protocol policer values have changed from the default. With DHCPv4 and PPPoE traffic flowing, verify that the policers are working correctly. You can enter commands to display the individual policers you are interested in, as shown here, or you can enter the `show ddos-protection protocols dhcpv4` command to display this information for all DHCPv4 packet types.

Action

From operational mode, enter the `show ddos-protection protocols dhcpv4 aggregate` command.

```
user@host> show ddos-protection protocols dhcpv4 aggregate
Protocol Group: DHCPv4

  Packet type: aggregate (aggregate for all DHCPv4 traffic)
    Aggregate policer configuration:
      Bandwidth:        669 pps
      Burst:            6000 packets
      Priority:         medium
      Recover time:     300 seconds
      Enabled:          Yes
    System-wide information:
      Aggregate bandwidth is no longer being violated
        No. of FPCs currently receiving excess traffic: 0
        No. of FPCs that have received excess traffic:  1
        Violation first detected at: 2011-03-10 06:27:47 PST
        Violation last seen at:      2011-03-10 06:28:57 PST
        Duration of violation: 00:01:10 Number of violations: 1
      Received:  71064               Arrival rate:     0 pps
      Dropped:   23115               Max arrival rate: 1000 pps
    Routing Engine information:
      Bandwidth: 669 pps, Burst: 6000 packets, enabled
      Aggregate policer is never violated
      Received:  36130               Arrival rate:     0 pps
      Dropped:   0                   Max arrival rate: 671 pps
        Dropped by aggregate policer: 0
    FPC slot 1 information:
      Bandwidth: 100% (669 pps), Burst: 100% (5000 packets), enabled
      Aggregate policer is no longer being violated
        Violation first detected at: 2011-03-10 06:27:48 PST
        Violation last seen at:      2011-03-10 06:28:58 PST
        Duration of violation: 00:01:10 Number of violations: 1
      Received:  71064               Arrival rate:     0 pps
      Dropped:   34934               Max arrival rate: 1000 pps
```

```
        Dropped by individual policers: 11819
        Dropped by aggregate policer: 23115
```

From operational mode, enter the `show ddos-protection protocols dhcpv4 discover` command.

```
user@host> show ddos-protection protocols dhcpv4 discover
Protocol Group: DHCPv4

  Packet type: discover (DHCPv4 DHCPDISCOVER)
    Individual policer configuration:
      Bandwidth:        100 pps
      Burst:            300 packets
      Priority:         low
      Recover time:     200 seconds
      Enabled:          Yes
      Bypass aggregate: No
    System-wide information:
      Bandwidth is no longer being violated
        No. of FPCs currently receiving excess traffic: 0
        No. of FPCs that have received excess traffic:  1
        Violation first detected at: 2011-03-10 06:28:34 PST
        Violation last seen at:      2011-03-10 06:28:55 PST
        Duration of violation: 00:00:21 Number of violations: 1
      Received:  47949               Arrival rate:     0 pps
      Dropped:   11819               Max arrival rate: 671 pps
    Routing Engine information:
      Bandwidth: 100 pps, Burst: 300 packets, enabled
      Policer is never violated
      Received:  36130               Arrival rate:     0 pps
      Dropped:   0                   Max arrival rate: 0 pps
        Dropped by aggregate policer: 0
    FPC slot 1 information:
      Bandwidth: 100% (100 pps), Burst: 100% (300 packets), enabled
      Policer is no longer being violated
        Violation first detected at: 2011-03-10 06:28:35 PST
        Violation last seen at:      2011-03-10 06:28:55 PST
        Duration of violation: 00:00:20 Number of violations: 1
      Received:  47949               Arrival rate:     0 pps
      Dropped:   11819               Max arrival rate: 671 pps
        Dropped by this policer: 11819
        Dropped by aggregate policer: 0
```

From operational mode, enter the `show ddos-protection protocols dhcpv4 offer` command.

```
user@host> show ddos-protection protocols dhcpv4 offer
Protocol Group: DHCPv4

  Packet type: offer (DHCPv4 DHCPOFFER)
    Individual policer configuration:
      Bandwidth:        1000 pps
      Burst:            1000 packets
      Priority:         medium
      Recover time:     300 seconds
      Enabled:          Yes
      Bypass aggregate: Yes
    System-wide information:
      Bandwidth is never violated
      Received:  0                   Arrival rate:     0 pps
      Dropped:   0                   Max arrival rate: 0 pps
    Routing Engine information:
      Policer is never violated
      Received:  0                   Arrival rate:     0 pps
      Dropped:   0                   Max arrival rate: 0 pps
        Dropped by aggregate policer: 0
    FPC slot 1 information:
      Bandwidth: 80% (800 pps), Burst: 75% (750 packets), enabled
      Policer is never violated
      Received:  0                   Arrival rate:     0 pps
      Dropped:   0                   Max arrival rate: 0 pps
        Dropped by aggregate policer: 0
```

Meaning

The output of these commands lists the policer configuration and traffic statistics for the DHCPv4 aggregate, discover, and offer policers respectively.

The `Aggregate policer configuration` section in the first output example and `Individual policer configuration` sections in the second and third output examples list the configured values for bandwidth, burst, priority, recover time, and bypass-aggregate.

The `System-wide information` section shows the total of all DHCPv4 traffic statistics and violations for the policer recorded across all line cards and at the Routing Engine. The `Routing engine information` section shows the traffic statistics and violations for the policer recorded at the Routing Engine. The `FPC slot 1` `information` section shows the traffic statistics and violations for the policer recorded only at the line card in slot 1.

The output for the aggregate policer in this example shows the following information:

- The `System-wide information` section shows that 71,064 DHCPv4 packets of all types were received across all line cards and the Routing Engine. The section shows a single violation with a time stamp and that the aggregate policer at a line card dropped 23,115 of these packets.
- The `FPC slot 1 information` section shows that this line card received all 71,064 DHCPv4 packets, but its aggregate policer experienced a violation and dropped the 23,115 packets shown in the other section. The line card individual policers dropped an additional 11,819 packets.
- The `Routing Engine information` section shows that the remaining 36,130 packets all reached the Routing Engine and that its aggregate policer dropped no additional packets.

  The difference between the number of DHCPv4 packets received and dropped at the line card [71,064 - (23,115 + 11,819)] matches the number received at the Routing Engine. That might not always be the case, because packets can be received and dropped at more than one line card. In this example, only the line card in slot 1received any DHCPv4 packets.

The output for the DHCPv4 discover packet policer in this example shows the following information:

- The `System-wide information` section shows that 47,949 DHCPv4 discover packets were received across all line cards and the Routing Engine. The section shows a single violation with a time stamp and that the aggregate policer at a line card dropped 11,819 of these packets.
- The `FPC slot 1 information` section shows that this line card received all 47,949 DHCPv4 discover packets, but its individual policer experienced a violation and dropped the 11,819 packets shown in the other section.
- The `Routing Engine information` section shows that only 36,130 DHCPv4 discover packets reached the Routing Engine and that it dropped no additional packets.

  The difference between the number of DHCPv4 discover packets received and dropped at the line card (47,949 - 11,819) matches the number received at the Routing Engine. That might not always be the case, because packets can be received and dropped at more than one line card. In this example, only the line card in slot 1received any DHCPv4 discover packets.

The output for the DHCPv4 offer packet policer in this example shows the following information:

- This individual policer has never been violated at any location.
- No DHCPv4 offer packets have been received at any location.

Verifying the PPPoE DDoS Configuration

Purpose

Verify that the PPPoE policer values have changed from the default.

Action

From operational mode, enter the `show ddos-protection protocols pppoe parameters brief` command.

```
user@host> show ddos-protection protocols pppoe parameters brief
Number of policers modified: 1
Protocol    Packet      Bandwidth Burst  Priority Recover   Policer Bypass FPC
group       type        (pps)     (pkts)          time(sec) enabled aggr.  mod
pppoe       aggregate   800*      2000   medium   300       yes     --     no
pppoe       padi        500       500    low      300       yes     no     no
pppoe       pado        0         0      low      300       yes     no     no
pppoe       padr        500       500    medium   300       yes     no     no
pppoe       pads        0         0      low      300       yes     no     no
pppoe       padt        1000      1000   high     300       yes     no     no
pppoe       padm        0         0      low      300       yes     no     no
pppoe       padn        0         0      low      300       yes     no     no
```

From operational mode, enter the `show ddos-protection protocols pppoe padi` command, and enter the command for `padr` as well.

```
user@host> show ddos-protection protocols pppoe padi
Protocol Group: PPPoE

  Packet type: padi (PPPoE PADI)
    Individual policer configuration:
      Bandwidth:        500 pps
      Burst:            500 packets
      Priority:         low
      Recover time:     300 seconds
      Enabled:          Yes
      Bypass aggregate: No
    System-wide information:
      Bandwidth for this packet type is being violated!
        Number of slots currently receiving excess traffic: 1
        Number of slots that have received excess traffic:  1
        Violation first detected at: 2011-03-09 11:26:33 PST
        Violation last seen at:      2011-03-10 12:03:44 PST
        Duration of violation: 1d 00:37 Number of violations: 1
      Received:  704832908           Arrival rate:     8000 pps
      Dropped:   660788548           Max arrival rate: 8008 pps
    Routing Engine information:
      Bandwidth: 500 pps, Burst: 500 packets, enabled
      Policer is never violated
      Received:  39950330            Arrival rate:     298 pps
      Dropped:   0                   Max arrival rate: 503 pps
        Dropped by aggregate policer: 0
    FPC slot 3 information:
      Bandwidth: 100% (500 pps), Burst: 100% (500 packets), enabled
      Policer is currently being violated!
        Violation first detected at: 2011-03-09 11:26:35 PST
        Violation last seen at:      2011-03-10 12:03:44 PST
        Duration of violation: 1d 00:37 Number of violations: 1
      Received:  704832908           Arrival rate:     8000 pps
      Dropped:   664882578           Max arrival rate: 8008 pps
        Dropped by this policer: 660788548
        Dropped by aggregate policer: 4094030
```

```
user@host> **show ddos-protection protocols pppoe padr**
Protocol Group: PPPoE

  Packet type: padr (PPPoE PADR)
    Individual policer configuration:
      Bandwidth:        500 pps
      Burst:            500 packets
      Priority:         medium
      Recover time:     300 seconds
      Enabled:          Yes
      Bypass aggregate: No
    System-wide information:
      Bandwidth for this packet type is being violated!
        Number of slots currently receiving excess traffic: 1
        Number of slots that have received excess traffic:  1
        Violation first detected at: 2011-03-10 06:21:17 PST
        Violation last seen at:      2011-03-10 12:04:14 PST
        Duration of violation: 05:42:57 Number of violations: 1
      Received:  494663595           Arrival rate:     24038 pps
      Dropped:   484375900           Max arrival rate: 24062 pps
    Routing Engine information:
      Bandwidth: 500 pps, Burst: 500 packets, enabled
      Policer is never violated
      Received:  10287695            Arrival rate:     500 pps
      Dropped:   0                   Max arrival rate: 502 pps
        Dropped by aggregate policer: 0
    FPC slot 1 information:
      Bandwidth: 100% (500 pps), Burst: 100% (500 packets), enabled
      Policer is currently being violated!
        Violation first detected at: 2011-03-10 06:21:18 PST
        Violation last seen at:      2011-03-10 12:04:14 PST
        Duration of violation: 05:42:56 Number of violations: 1
      Received:  494663595           Arrival rate:     24038 pps
      Dropped:   484375900           Max arrival rate: 24062 pps
        Dropped by this policer: 484375900
        Dropped by aggregate policer: 0
```

Meaning

The output from the `show ddos-protection protocols pppoe parameters brief` command lists the current configuration for each of the individual PPPoE packet policers and the PPPoE aggregate policer. A change from a default value is indicated by an asterisk next to the modified value. The only change made to PPPoE policers in the configuration steps was to the aggregate policer bandwidth limit (pps); this change is confirmed in the output. Besides the configuration values, the command output also reports whether a policer has been disabled, whether it bypasses the aggregate policer (meaning that the traffic for that packet type is not included for evaluation by the aggregate policer), and whether the policer has been modified for one or more line cards.

The output of the `show ddos-protection protocols pppoe padi` command in this example shows the following information:

- The `System-wide information` section shows that 704,832,908 PPPoE PADI packets were received across all line cards and the Routing Engine. The section shows a single violation on a line card that is still in progress, and that the aggregate policer at the line card dropped 660,788,548 of the PADI packets.

- The `FPC slot 3 information` section shows that this line card received all 704,832,908 PADI packets. Its individual policer dropped 660,788,548 of those packets and its aggregate policer dropped the other 4,094,030 packets. The violation is ongoing and has lasted more than a day.

- The `Routing Engine information` section shows that only 39,950,330 PADI packets reached the Routing Engine and that it dropped no additional packets.

The difference between the number of PADI packets received and dropped at the line card [704,832,908 - (660,788,548 + 4,094030)] matches the number received at the Routing Engine. That might not always be the case, because packets can be received and dropped at more than one line card. In this example, only the line card in slot 3 received any PADI packets.

The output of the `show ddos-protection protocols pppoe padr` command in this example shows the following information:

- The `System-wide information` section shows that 494,663,595 PPPoE PADR packets were received across all line cards and the Routing Engine. The section shows a single violation on a line card that is still in progress, and that the policer at the line card dropped 484,375,900 of the PADR packets.

- The `FPC slot 1 information` section shows that this line card received all 494,663,595 PADR packets. Its individual policer dropped 484,375,900 of those packets. The violation is ongoing and has lasted more than five hours.

- The `Routing Engine information` section shows that only 10,287,695 PADR packets reached the Routing Engine and that it dropped no additional packets.

  The difference between the number of PADR packets received and dropped at the line card (494,663,595 - 484,375,900) matches the number received at the Routing Engine. That might not always be the case, because packets can be received and dropped at more than one line card. In this example, only the line card in slot 1 received any PADR packets.

> **NOTE**: This scenario is unrealistic in showing all PADI packets received on one line card and all PADR packets on a different line card. The intent of the scenario is to illustrate how policer violations are reported for individual line cards.

RELATED DOCUMENTATION

Control Plane Distributed Denial-of-Service (DDoS) Protection Overview | **631**

Configuring Control Plane DDoS Protection | **642**

Example: Configuring Control Plane DDoS Protection on QFX Series Switches

IN THIS SECTION

This example shows how to configure control plane DDoS protection so a switch can quickly identify an attack and prevent a flood of malicious control packets from exhausting system resources.

Requirements

Control plane DDoS protection requires the following hardware and software:

- QFX Series switch that supports control plane DDoS protection
- Junos OS Release 15.1X53-D10 or later

No special configuration beyond device initialization is required before you can configure this feature.

Overview

IN THIS SECTION

Distributed denial-of-service (DDoS) attacks use multiple sources to flood a network with protocol control packets. This malicious traffic triggers a large number of exceptions in the network and attempts to exhaust the system resources to deny valid users access to the network or server.

Control plane DDoS protection is enabled by default on a supported QFX Series switch. This example describes how you can modify the default configuration for the rate-limiting policers that identify excess control traffic and drop the packets before the switch is adversely affected. Sample tasks include configuring an aggregate policer for a protocol group, configuring policers for particular control packet types within a protocol group, and specifying trace options for control plane DDoS protection operations.

This example show how to change some of the default policer parameters and behavior for the `radius` protocol group and the Radius `accounting` packet type. You can use the same commands to change policer limits for other supported protocol groups and packet types. See the *ddos-protection* configuration statement at the `[edit system]` hierarchy level for all available configuration options.

Topology

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure control plane DDoS protection for protocol groups and particular control packet types, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
[edit]
edit system
set ddos-protection protocols radius aggregate bandwidth 150
set ddos-protection protocols radius aggregate burst 2000
set ddos-protection protocols radius accounting bandwidth 100 burst 150
set ddos-protection protocols radius accounting priority low
set ddos-protection protocols radius server bypass-aggregate
set ddos-protection traceoptions file ddos-trace size 10m
set ddos-protection traceoptions flag all
top
```

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 control plane DDoS protection:

1. Specify a protocol group.

   ```
   [edit system ddos-protection protocols]
   user@host# edit radius
   ```

2. Configure the maximum traffic rate for the RADIUS aggregate policer; that is, for the combination of all RADIUS packets.

   > **NOTE**: You change the traffic rate using the `bandwidth` option. Although the term bandwidth usually refers to bits per second (bps), this feature's `bandwidth` option represents a packets per second (pps) value.

   ```
   [edit system ddos-protection protocols radius]
   user@host# set aggregate bandwidth 150
   ```

3. Configure the maximum burst size (number of packets) for the RADIUS aggregate policer.

   ```
   [edit system ddos-protection protocols radius]
   user@host# set aggregate burst 2000
   ```

4. Configure a different maximum traffic rate (pps) and burst size (packets) for RADIUS accounting packets.

   ```
   [edit system ddos-protection protocols radius]
   user@host# set accounting bandwidth 100 burst 1500
   ```

5. Decrease the priority for RADIUS accounting packets.

   ```
   [edit system ddos-protection protocols radius]
   user@host# set accounting priority low
   ```

6. Prevent RADIUS server control packets from being included in the aggregate bandwidth (pps); that is, server packets do not contribute toward the combined RADIUS traffic to determine whether the aggregate bandwidth is exceeded. However, the server packets are still included in traffic rate statistics.

```
[edit system ddos-protection protocol radius]
user@host# set server bypass-aggregate
```

7. (On switches with multiple line cards only) Reduce the bandwidth (pps) and burst size (packets) allowed before a violation is declared for the RADIUS policer on the FPC in slot 1.

```
[edit system ddos-protection protocols radius]
user@host# set aggregate fpc 1 bandwidth-scale 80
user@host# set aggregate fpc 1 burst-scale 75
```

8. Configure tracing for all control plane DDoS protection protocol processing events.

```
[edit system ddos-protection traceoptions]
user@host# set file ddos-log
user@host# set file size 10m
user@host# set flag all
```

Results

From configuration mode, confirm your configuration by entering the `show ddos-protection` command at the `system` hierarchy level.

```
[edit system]
user@host# show ddos-
protection

traceoptions {
    file ddos-log size 10m;
    flag all;
}
protocols {
    radius {
        aggregate {
            bandwidth 150;
            burst 2000;
        }
        server {
            bypass-aggregate;
        }
        accounting {
            bandwidth 100;
            burst 1500;
            priority low;
        }
    }
}
```

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

Verification

IN THIS SECTION

To confirm that the control plane DDoS protection configuration is working properly, perform these tasks:

Verifying the control plane DDoS Protection Configuration

Purpose

Verify that the RADIUS policer values have changed from the default.

Action

From operational mode, enter the `show ddos-protection protocols radius parameters` command.

```
user@host> show ddos-protection protocols radius parameters
Packet types: 5, Modified: 3
* = User configured value
```

```
Protocol Group: Radius

  Packet type: aggregate (Aggregate for all Radius traffic)
    Aggregate policer configuration:
      Bandwidth:        150 pps*
      Burst:            2000 packets*
      Recover time:     300 seconds
      Enabled:          Yes
    Routing Engine information:
      Bandwidth: 150 pps, Burst: 2000 packets, enabled
    FPC slot 0 information:
      Bandwidth: 100% (150 pps), Burst: 100% (2000 packets), enabled

  Packet type: server (Radius server traffic)
    Individual policer configuration:
      Bandwidth:        200 pps
      Burst:            2048 packets
      Priority:         High
      Recover time:     300 seconds
      Enabled:          Yes
      Bypass aggregate: Yes*
    Routing Engine information:
      Bandwidth: 200 pps, Burst: 2048 packets, enabled
    FPC slot 0 information:
      Bandwidth: 100% (200 pps), Burst: 100% (2048 packets), enabled

  Packet type: accounting (Radius accounting traffic)
    Individual policer configuration:
      Bandwidth:        100 pps*
      Burst:            1500 packets*
      Priority:         Low*
      Recover time:     300 seconds
      Enabled:          Yes
      Bypass aggregate: No
    Routing Engine information:
      Bandwidth: 100 pps, Burst: 1500 packets, enabled
    FPC slot 0 information:
      Bandwidth: 100% (100 pps), Burst: 100% (1500 packets), enabled

  Packet type: authorization (Radius authorization traffic)
    Individual policer configuration:
      Bandwidth:        200 pps
      Burst:            2048 packets
```

```
     Priority:         High
     Recover time:     300 seconds
     Enabled:          Yes
     Bypass aggregate: No
  Routing Engine information:
    Bandwidth: 200 pps, Burst: 2048 packets, enabled
  FPC slot 0 information:
    Bandwidth: 100% (200 pps), Burst: 100% (2048 packets), enabled
```

Meaning

The command output shows the current configuration of the RADIUS aggregate policer and the RADIUS accounting, server, and authorization control packet policers. Policer values that have been modified from the default values are marked with an asterisk. The output shows that the RADIUS policer configuration has been modified correctly.

RELATED DOCUMENTATION

Control Plane Distributed Denial-of-Service (DDoS) Protection Overview | **631**

Configuring Control Plane DDoS Protection | **642**

CHAPTER 24

Flow Detection and Culprit Flows

IN THIS CHAPTER

Control Plane DDoS Protection Flow Detection Overview

IN THIS SECTION

Flow detection is an enhancement to control plane DDoS protection that supplements the DDoS policer hierarchies; it is part of a complete control plane DDoS protection solution. Flow detection uses a limited amount of hardware resources to monitor the arrival rate of host-bound flows of control traffic. Flow detection is much more scalable than a solution based on filter policers. Filter policers track all flows, which consumes a considerable amount of resources. In contrast, flow detection only tracks flows it identifies as suspicious, using far fewer resources to do so.

The flow detection application has two interrelated components, detection and tracking. Detection is the process where flows suspected of being improper are identified and subsequently controlled. Tracking is the process where flows are tracked to determine whether they are truly hostile and when these flows recover to within acceptable limits.

Flow Detection and Control

Flow detection is disabled by default. When you enable it at the `[edit system ddos-protection global]` hierarchy level, the application begins monitoring control traffic flows when a control plane DDoS protection policer is violated for almost all protocol groups and packet types. In addition to enabling flow detection globally, you can configure its operation mode—that is, whether it is automatically triggered by the violation of a DDoS protection policer (the default) or is always on—for almost all protocol groups and packet types. You can override the global configuration settings for individual protocol groups and packet types. Other than event report rates, all other characteristics of flow detection are configurable only at the level of individual packet types.

Enhanced Subscriber Management supports flow detection for control plane DDoS protection as of Junos OS Release 17.3R1.

NOTE: You cannot enable flow detection globally for the following groups and packet type because they do not have typical Ethernet, IP, or IPv6 headers:

- Protocol groups: `fab-probe`, `frame-relay`, `inline-ka`, `isis`, `jfm`, `mlp`, `pfe-alive`, `pos`, and `services`.
- Packet type: `unclassified` in the `ip-options` protocol group.

Control flows are aggregated at three levels. The *subscriber level* is the finest grained of the three and consists of flows for individual subscriber sessions. The *logical interface level* aggregates multiple subscriber flows, so it is coarser grained and does not provide discrimination into individual subscriber flows. The *physical interface level* aggregates multiple logical interface flows, so it provides the coarsest view of traffic flows.

You can turn flow detection off or on at any of the three control flow levels. You can set flow detection to be automatically triggered by a DDoS protection policer violation or to remain always on. In automatic mode, flow detection activates only after a DDoS protection policer violation occurs. Flow detection initiates at the finest-grained level where detection is configured to be `on` or `automatic`.

When a flow arrives, flow detection checks whether the flow is already listed in a table of *suspicious* flows. A suspicious flow is one that exceeds the bandwidth allowed by default or configuration. If the flow is not in the table and the aggregation level flow detection mode is `on`, then flow detection lists the flow in the table. If the flow is not in the table and the flow detection mode is `automatic`, flow detection checks whether this flow is suspicious.

If the flow is suspicious, then it goes in the flow table. If the flow is not suspicious, then it is processed the same way at the next coarser aggregation level that has flow detection set to `on`. If none of the higher levels have detection on, then the flow continues to the DDoS protection packet policer for action, where it can be passed or dropped.

When the initial check finds the flow in the table, then the flow is dropped, policed, or kept, depending on the control mode setting for that aggregation level. All packets in dropped flows are dropped. In policed flows, packets are dropped until the flow is within the acceptable bandwidth for the aggregation level. Kept flows are passed along to the next aggregation level for processing.

For more details, see "Configuring How Flow Detection Operates Globally" on page 692.

Flow Tracking

The flow detection application tracks flows that have been listed in the suspicious flow table. It periodically checks each entry in the table to determine whether the listed flow is still suspicious (violating the bandwidth). If a suspicious flow has continuously violated the bandwidth since it was inserted in the table for a period greater than the configurable flow detection period, then it is considered to be a *culprit* flow rather than merely suspicious. However, if the bandwidth has been violated for less than the detection period, the violation is treated as a false positive. Flow detection considers the flow to be safe and stops tracking it (deletes it from the table).

You can enable a timeout feature that suppresses culprit flows for a configurable timeout period, during which the flow is kept in the flow table. (Suppression is the default behavior, but the flow detection action can be changed by the flow level control configuration.) If the check of listed flows finds one for which the timeout is enabled and the timeout period has expired, then the flow has timed out and it is removed from the flow table.

If the timeout has not yet expired or if the timeout feature is not enabled, then the application performs a recovery check. If the time since the flow last violated the bandwidth is longer than the configurable recovery period, the flow has recovered and is removed from the flow table. If the time since last violation is less than the recovery period, the flow is kept in the flow table.

Notifications

By default, flow detection automatically generates system logs for a variety of events that occur during flow detection. The logs are referred to as *reports* in the flow detection CLI. All protocol groups and packet types are covered by default, but you can disable automatic logging for individual packet types. You can also configure the rate at which reports are sent, but this applies globally to all packet types.

Each report belongs to one of the following two types:

- Flow reports—These reports are generated by events associated with the identification and tracking of culprit flows. Each report includes identifying information for the flow that experienced the event. This information is used to accurately maintain the flow table; flows are deleted or retained in the table based on the information in the report. Table 32 on page 681 describes the event that triggers each flow report.

Table 32: Triggering Event for Flow Detection Reports

Name	Description
DDOS_SCFD_FLOW_FOUND	A suspicious flow is detected.
DDOS_SCFD_FLOW_TIMEOUT	The timeout period expires for a culprit flow. Flow detection stops suppressing (or monitoring) the flow.
DDOS_SCFD_FLOW_RETURN_NORMAL	A culprit flow returns to within the bandwidth limit.
DDOS_SCFD_FLOW_CLEARED	A culprit flow is cleared manually with a `clear` command or automatically as the result of suspicious flow monitoring shifting to a different aggregation level.
DDOS_SCFD_FLOW_AGGREGATED	Control flows are aggregated to a coarser level. This event happens when the flow table nears capacity or when the flow cannot be found at a particular flow level and the next coarser level has to be searched.
DDOS_SCFD_FLOW_DEAGGREGATED	Control flows are deaggregated to a finer level. This event happens when the flow table is not very full or when flow control is effective and the total arrival rate for the flow at the policer for the packet type is below its bandwidth for a fixed, internal period.

- Bandwidth violation reports—These reports are generated by events associated with the discovery of suspicious flows. Each report includes identifying information for the flow that experienced the event. This information is used to track the suspicious flow and identify flows that are placed in the flow table. Table 33 on page 681 describes the event that triggers each violation report.

Table 33: Triggering Event for Bandwidth Violation Reports

Name	Description
DDOS_PROTOCOL_VIOLATION_SET	The incoming traffic for a control protocol exceeded the configured bandwidth.

Table 33: Triggering Event for Bandwidth Violation Reports *(Continued)*

Name	Description
DDOS_PROTOCOL_VIOLATION_CLEAR	The incoming traffic for a violated control protocol returned to normal.

A report is sent only when triggered by an event; that is, there are no null or empty reports. Because the reports are made periodically, the only events of interest are ones that occur during the interval since the last report.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.3R1	Enhanced Subscriber Management supports flow detection for control plane DDoS protection as of Junos OS Release 17.3R1.

RELATED DOCUMENTATION

Setting Up and Using Flow Detection | 682

Setting Up and Using Flow Detection

IN THIS SECTION

Flow detection monitors the flows of control traffic for violation of the bandwidth allowed for each flow and manages traffic identified as a culprit flow. Suppression of the traffic is the default management option. Flow detection is typically implemented as part of an overall control plane DDoS protection strategy, but it is also useful for troubleshooting and understanding traffic flow in new configurations. Flow detection is disabled by default.

Enhanced Subscriber Management supports flow detection for control plane DDoS protection as of Junos OS Release 17.3R1.

Before you begin, ensure you have configured control plane DDoS protection appropriately for you network. See "Configuring Control Plane DDoS Protection" on page 642 for detailed information about DDoS protection.

Configuring the Detection Period for Suspicious Flows

DDoS protection flow detection considers a monitored flow to be a suspicious flow whenever the flow exceeds its allowed bandwidth, based on a crude test that eliminates obviously good flows from consideration. A closer examination of a suspicious flow requires the flow to remain in violation of the bandwidth for a period of time before flow detection considers it to be a culprit flow against which it must take action. You can include the `flow-detect-time` statement to configure the duration of this detection period or you can rely on the default period of three seconds.

Enhanced Subscriber Management supports flow detection for DDoS protection as of Junos OS Release 17.3R1.

> **BEST PRACTICE**: We recommend that you use the default value for the detection period.

To specify how long a flow must be in violation before flow detection declares it to be a culprit flow:

- Set the detection period.

```
[edit system ddos-protection protocols protocol-group packet-type]
user@host# set flow-detect-time seconds
```

For example, include the following statement to require the DHCPv4 discover packet flow to be in violation of its allowed bandwidth for 30 seconds before it is considered to be a culprit flow:

```
[edit system ddos-protection protocols dhcpv4 discover]
user@host# set flow-detect-time 30
```

Configuring the Recovery Period for a Culprit Flow

After DDoS protection flow detection has identified a suspicious flow as a culprit flow, it has to determine when that flow no longer represents a threat to the router. When the traffic flow rate drops back to within the allowed bandwidth, the rate must remain within the bandwidth for a recovery period. Only then does flow detection consider the flow to be normal and stop the traffic handling action enacted against the culprit flow. You can include the `flow-recover-time` statement to configure the duration of this recovery period or you can rely on the default period of 60 seconds.

To specify how long a flow must be within its allowed bandwidth after a violation before flow detection declares it to be a normal flow:

- Set the recovery period.

```
[edit system ddos-protection protocols protocol-group packet-type]
user@host# set flow-recover-time seconds
```

For example, include the following statement to require the DHCPv4 discover packet flow to be in recovery for five minutes (300 seconds):

```
[edit system ddos-protection protocols dhcpv4 discover]
user@host# set flow-recover-time 300
```

Configuring the Timeout Period for a Culprit Flow

When DDoS protection flow detection identifies a suspicious flow as a culprit flow, by default it suppresses traffic for that flow for as long as the traffic flow exceeds the bandwidth limit. Suppression stops and the flow is removed from the flow table when the time since the last violation by the flow is greater than the recovery period.

Alternatively, you can include the `timeout-active-flows` statement to enable flow detection to suppress a culprit flow for a configurable timeout period. When the timeout period expires, suppression stops and the flow is removed from the flow table. You can either include the `flow-timeout-time` statement to configure the duration of the timeout period or rely on the default timeout of 300 seconds.

To enable flow detection to suppress a culprit flow for a timeout period:

1. Enable the timeout.

```
[edit system ddos-protection protocols protocol-group packet-type]
user@host# set timeout-active-flows
```

2. Specify the timeout period.

```
[edit system ddos-protection protocols protocol-group packet-type]
user@host# setflow-timeout-time seconds
```

For example, include the following statements to suppress the DHCPv4 discover packet flow for 10 minutes (600 seconds):

```
[edit system ddos-protection protocols dhcpv4 discover]
user@host# set timeout-active-flows
user@host# setflow-timeout-time 600
```

Configuring How Flow Detection Operates at Each Flow Aggregation Level

When flow detection is turned on, traffic flows are monitored by default for all protocol groups and packet types. When a policer violation occurs, each suspicious flow is examined to determine whether it is the culprit flow that caused the violation. You can include the `flow-level-detection` statement to configure how flow detection works at each flow aggregation level for a packet type: subscriber, logical interface, or physical interface.

NOTE: The flow detection mode at the packet level must be either `automatic` or `on` for flow detection to operate at individual flow aggregation levels.

Like flow detection at the protocol group and packet level, flow detection at the flow aggregation level supports three modes:

- automatic—When a control plane DDoS protection policer is violated, traffic flows at this flow aggregation level are monitored for suspicious behavior only until flow detection determines that the suspect flow is not at this aggregation level and instead must be at a coarser level of aggregation. Flows at this level are subsequently not searched again until the policer is no longer violated at the coarser level.

- off—Traffic flows are never monitored at this flow aggregation level.

- on—Traffic flows at this flow aggregation level are monitored for suspicious flows even when no DDoS protection policer is currently being violated, if flow detection at the packet level is configured to on. Monitoring continues at this level regardless of whether a suspect flow is identified at this level. However, If the packet level mode is automatic, then the policer must be in violation for traffic flows to be checked at this level.

Flows are examined first at the finest-grained (lowest bandwidth) flow aggregation level, subscriber. If the suspect flow is not found at the subscriber level, then flows are checked at the logical interface level. Finally, if the suspect is not found there, then flows are checked at the physical interface level; barring some misconfiguration, the culprit flow must be found at this level.

To configure how flow detection operates at each flow aggregation level:

1. (Optional) Specify the detection mode at the subscriber level.

```
[edit system ddos-protection protocols protocol-group packet-type flow-level-detection]
user@host# set subscriber flow-detection-mode
```

2. (Optional) Specify the detection mode at the logical interface level.

```
[edit system ddos-protection protocols protocol-group packet-type flow-level-detection]
user@host# set logical-interface flow-detection-mode
```

3. (Optional) Specify the detection mode at the physical interface level.

```
[edit system ddos-protection protocols protocol-group packet-type flow-level-detection]
user@host# set physical-interface flow-detection-mode
```

For example, include the following statements to configure flow detection to check for suspicious flows at the subscriber level only when the policer is being violated, to never check at the logical interface level, and to always check at the physical interface level:

```
[edit system ddos-protection protocols dhcpv4 discover]
user@host# edit flow-level-detection
user@host# set subscriber automatic
user@host# set logical-interface off
user@host# set physical-interface on
```

Configuring How Traffic in a Culprit Flow Is Controlled at Each Flow Aggregation Level

When flow detection is enabled, all traffic in a culprit flow is dropped by default for all protocol groups and packet types and at all flow aggregation levels. You can include the flow-level-control statement to configure flow detection to control traffic differently for individual packet types. You have to specify the control behavior at a particular flow aggregation level: subscriber, logical interface, or physical interface.

You can configure flow detection flow control to employ one of the following modes for a packet type:

- Drop all traffic—Configure flow control to drop all traffic when you think the flow that is violating a bandwidth limit is malicious. This behavior is the default at all flow aggregation levels.
- Police traffic—Configure flow control to police a flow that is violating bandwidth, forcing the rate below the bandwidth limit. Flow control acts as a simple policer in this case.
- Keep all traffic—Configure flow control to keep all traffic whether the flow is in violation or below the bandwidth limit. This mode is helpful when you need to debug traffic flow for your network.

Flow control mode enables great flexibility in how you manage control traffic in your network. For example, if you only want to ensure that control flows for a packet type at all aggregation levels are within their limits, you can configure flow control to police the traffic at each level. Or if you want to detect culprit flows and suppress them at one level but only restrain traffic to the allowed bandwidth at another level, you can configure one level to drop all traffic and the other to police traffic.

To configure how flow detection controls traffic in a culprit flow:

1. (Optional) Specify the control mode at the subscriber level.

   ```
   [edit system ddos-protection protocols protocol-group packet-type flow-level-control]
   user@host# set subscriber flow-control-mode
   ```

2. (Optional) Specify the control mode at the logical interface level.

   ```
   [edit system ddos-protection protocols protocol-group packet-type flow-level-control]
   user@host# set logical-interface flow-control-mode
   ```

3. (Optional) Specify the control mode at the physical interface level.

   ```
   [edit system ddos-protection protocols protocol-group packet-type flow-level-control]
   user@host# set physical-interface flow-control-mode
   ```

For example, to configure flow detection to keep all traffic for a physical interface under the configured bandwidth, but detect and suppress culprit flows at the subscriber level:

```
[edit system ddos-protection protocols dhcpv4 discover]
user@host# edit flow-level-control
user@host# set subscriber drop
```

```
user@host# set physical-interface police
user@host# edit flow-level-detection
user@host# set logical-interface off
```

In this example, you do not care about the logical interface, so flow detection is turned off for that level. Because flow detection is disabled, the state of flow control for that level does not matter.

Enabling Flow Detection for All Protocol Groups and Packet Types

By default, flow detection is disabled for all protocol groups and packet types. You must enable flow detection globally by including the `flow-detection` statement. If you subsequently disable flow detection for individual packet types, you cannot use this global statement to override all such individual configurations; you must re-enable detection at the packet configuration level.

To enable flow detection globally:

- Set flow detection.

  ```
  [edit system ddos-protection global]
  user@host# set flow-detection
  ```

> **NOTE**: You cannot enable flow detection globally for the following groups and packet type because they do not have typical Ethernet, IP, or IPv6 headers:
>
> - Protocol groups: `fab-probe`, `frame-relay`, `inline-ka`, `isis`, `jfm`, `mlp`, `pfe-alive`, `pos`, and `services`.
> - Packet type: `unclassified` in the `ip-options` protocol group.

Configuring the Culprit Flow Reporting Rate for All Protocol Groups and Packet Types

When flow detection confirms that a suspicious flow it is tracking on a line card is indeed a culprit flow, it sends a report to the Routing Engine. Flow detection also reports each culprit flow that subsequently recovers to within the allowed bandwidth or is cleared. You can include the `flow-report-rate` statement to limit how many flows per second on each line card can be reported. Culprit flow events are reported for all protocol groups and packet types by default. When too many flows are reported, congestion can occur on the host path to the Routing Engine flow.

To globally configure the maximum report rate for culprit flows:

- Set the reporting rate.

```
[edit system ddos-protection global]
user@host# set flow-report-rate rate
```

Configuring the Violation Reporting Rate for All Protocol Groups and Packet Types

By default, flow detection reports to the Routing Engine all violations of bandwidth at the FPC for all protocol groups and packet types. You can include the `violation-report-rate` statement to limit how many violations per second flow detection reports from the line cards, thus reducing the load on the router. We recommend that you configure a report rate that is suitable for your network rather than rely on the default value.

To globally configure the maximum bandwidth violation reporting rate:

- Set the reporting rate.

```
[edit system ddos-protection global]
user@host# set violation-report-rate rate
```

Disabling Automatic Logging of Culprit Flow Events for a Packet Type

By default, flow detection automatically logs policer violation events associated with suspicious flows (violation reports) and culprit flow events (flow reports) for all protocol groups and packet types. You can include the `no-flow-logging` statement to prevent automatic logging of culprit flow events for individual packet types. Automatic logging of suspicious flow violation events is disabled with the `disable-logging` statement at the `[edit system ddos-protection global` hierarchy level.

To disable automatic culprit flow event logging for a packet type:

- Disable logging.

```
[edit system ddos-protection protocols protocol-group packet-type]
user@host# set no-flow-logging
```

To disable automatic suspicious flow violation event logging for a packet type:

- Disable logging.

```
[edit system ddos-protection protocols protocol-group packet-type]
user@host# set disable-logging
```

For example, include the following statement to disable automatic logging for DHCPv4 DISCOVER packet flows:

```
[edit system ddos-protection protocols dhcpv4 discover]
user@host# set no-flow-logging
```

Configuring the Maximum Flow Bandwidth at Each Flow Aggregation Level

You can include the `flow-level-bandwidth` statement to configure the maximum acceptable bandwidth for traffic flows for individual packet types. You have to specify the bandwidth behavior at a particular flow aggregation level: subscriber, logical interface, or physical interface. We recommend that you tune the bandwidth values for your network rather than rely on the defaults.

To configure the maximum bandwidth for traffic flows each flow aggregation level:

1. (Optional) Configure the bandwidth for flows at the subscriber level.

   ```
   [edit system ddos-protection protocols protocol-group packet-type flow-level-bandwidth]
   user@host# set subscriber flow-bandwidth
   ```

2. (Optional) Configure the bandwidth for flows at the logical interface level.

   ```
   [edit system ddos-protection protocols protocol-group packet-type flow-level-bandwidth]
   user@host# set logical-interface flow-bandwidth
   ```

3. (Optional) Configure the bandwidth for flows at the physical interface level.

   ```
   [edit system ddos-protection protocols protocol-group packet-type flow-level-bandwidth]
   user@host# set physical-interface flow-bandwidth
   ```

For example, to configure the flow bandwidth to 1000 pps at the subscriber level, 5000 pps at the logical interface level, and 30,000 at the physical interface level:

```
[edit system ddos-protection protocols dhcpv4 discover]
user@host# edit flow-level-bandwidth
user@host# set subscriber 1000
user@host# set logical-interface 5000
user@host# set physical-interface 30000
```

Verifying and Managing Flow Detection

IN THIS SECTION

Purpose

View or clear information about flow detection as part of a control plane DDoS protection configuration.

Enhanced Subscriber Management supports flow detection for control plane DDoS protection as of Junos OS Release 17.3R1.

Action

- To display configuration information for flow detection:

```
user@host> show ddos-protection protocols flow-detection
```

- To display information about culprit flows identified by flow detection, including number of flows detected and tracked, source address of the flow, arriving interface, and rates:

```
user@host> show ddos-protection protocols culprit-flows
```

- To clear culprit flows for all packet types in all protocol groups:

```
user@host> clear ddos-protection protocols culprit-flows
```

- To clear culprit flows for all packet types in a particular protocol group:

```
user@host> clear ddos-protection protocols protocol-group culprit-flows
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.3R1	Enhanced Subscriber Management supports flow detection for control plane DDoS protection as of Junos OS Release 17.3R1.
17.3R1	Enhanced Subscriber Management supports flow detection for DDoS protection as of Junos OS Release 17.3R1.
17.3R1	Enhanced Subscriber Management supports flow detection for control plane DDoS protection as of Junos OS Release 17.3R1.

Configuring How Flow Detection Operates Globally

Flow detection is disabled globally for all protocol groups and packet types by default. After you have turned on flow detection globally with the `flow-detection` statement at the `[edit system ddos-protection global]` hierarchy level, you can include the `flow-detection-mode` statement to configure *how* flow detection operates globally for all protocol groups and packet types. By default, flow detection operates in automatic mode for all packet types, meaning that it monitors control traffic for suspicious flows only after a DDoS policer has been violated. You can also configure flow detection either to never monitor flows or to always monitor flows.

When flow detection is turned on, traffic flows are monitored by default for all protocol groups and packet types. You can override the global configuration by including the `flow-detection-mode` statement at the `[edit system ddos-protection protocols` *`protocol-group packet-type`*`]` hierarchy level to configure how flow detection works for a protocol group or a packet type. You can also use the `flow-level-detection` statement to specify the behavior for one or more traffic flow aggregation levels (subscriber, logical interface, or physical interface).

⚠ **CAUTION**: In a virtual chassis configuration, we recommend that you override flow detection for all Virtual Chassis control packets. The flow is based on the MAC address of the module in the FPC slot. If the `virtual-chassis control-low` flow is in violation, then all control traffic is lost, resulting in unexpected behavior. This behavior can include DHCP and PPPoE control traffic loss, loss of ARP requests, routing protocol flaps, and more.

To override flow detection for Virtual Chassis control packets when you have enabled global flow detection:

- Disable flow detection for each packet type.

  ```
  [edit]
  user@host# set system ddos-protection protocols virtual-chassis control-low flow-
  detection-mode off
  user@host# set system ddos-protection protocols virtual-chassis control-high flow-
  detection-mode off
  user@host# set system ddos-protection protocols virtual-chassis unclassified flow-
  detection-mode off
  user@host# set system ddos-protection protocols virtual-chassis vc-packets flow-
  detection-mode off
  user@host# set system ddos-protection protocols virtual-chassis vc-ttl-errors flow-
  detection-mode off
  ```

Flow detection supports the following three modes:

- automatic—When a control plane DDoS protection policer is violated, traffic flows where the violation occurred are monitored for suspicious behavior. Each suspicious flow is examined to determine whether it is the culprit flow that caused the violation.
- off—Traffic flows are never monitored for any protocol group or packet type.
- on—Traffic flows for all protocol groups and packet types are monitored for suspicious flows even when no DDoS protection policer is currently being violated.

NOTE: The detection mode is set to `automatic` by default. This means that if you enable global flow-detection and do not specify a mode, then flows are detected only when the policer is being violated.

To configure how flow detection operates at each flow aggregation level:

- Specify the detection mode.

  ```
  [edit system ddos-protection protocols global]
  user@host# set flow-detection-mode flow-detection-mode
  ```

For example, to configure flow detection to always monitor and detect flows for all protocol groups and packet types at all flow aggregation levels:

```
[edit system ddos-protection global]
user@host# set flow-detection-mode on
```

Configuring How Traffic in a Culprit Flow Is Controlled Globally

When flow detection is enabled, all traffic in a culprit flow is dropped by default for all protocol groups and packet types and at all flow aggregation levels. You can include the `flow-level-control` statement to configure how flow detection controls traffic for all traffic flow aggregation levels globally for all protocol groups and packet types. You cannot specify the control behavior globally for a particular flow aggregation level: subscriber, logical interface, or physical interface. To do that, you must override the global configuration with the `flow-level-control` statement at the `[edit system ddos-protection protocols` *`protocol-group packet-type`*`]` hierarchy level.

You can configure flow detection flow control to employ one of the following modes:

- Drop all traffic—Configure flow control to drop all traffic when you think the flow that is violating a bandwidth limit is malicious. This behavior is the default at all flow aggregation levels for all protocol groups and packet types.

- Police traffic—Configure flow control to police a flow that is violating bandwidth, forcing the rate below the bandwidth limit. Flow control acts as a simple policer in this case.

- Keep all traffic—Configure flow control to keep all traffic whether the flow is in violation or below the bandwidth limit. This mode is helpful when you need to debug traffic flow for your network.

To configure how flow detection controls traffic in a culprit flow for all flow aggregation levels for all protocol groups and packet types:

- Specify the control mode.

  ```
  [edit system ddos-protection global]
  user@host# set flow-level-control flow-control-mode
  ```

Flow control mode enables great flexibility in how you manage control traffic in your network. For example, if you only want to ensure that control flows for all packet types at all aggregation levels are within their limits, you can configure flow control globally to police the traffic.

```
[edit system ddos-protection global]
user@host# set flow-level-control police
```

Or, suppose you want to detect culprit flows and suppress them for DHCP discover packets at the physical interface flow aggregation level, but only restrain all traffic to the allowed bandwidth at the other levels. You can configure the police action globally, then override it for the packet type and physical level by configuring that level to drop all traffic.

```
[edit system ddos-protection global]
user@host# set flow-level-control police
[edit system ddos-protection protocols dhcpv4 discover ]
user@host# set flow-level-control physical-interface drop
```

11

PART

Unicast Forwarding

CHAPTER 25

Unicast Reverse Path Forwarding

IN THIS CHAPTER

Understanding Unicast RPF (Switches)

IN THIS SECTION

To protect against IP spoofing, and some types of denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks, unicast reverse-path-forwarding (RPF) verifies that packets are arriving from a legitimate path. It does this by checking the source address of each packet that arrives on an untrusted ingress interface and, comparing it to the forwarding-table entry for its source address. If the packet is from a valid path, that is, one that the sender would use to reach the destination, the device forwards the packet to the destination address. If it is not from a valid path, the device discards the packet. Unless it is protected against, IP spoofing can be an effective way for intruders to pass IP packets to a destination as genuine traffic, when in fact the packets are not actually meant for the destination.

Unicast RPF is supported for the IPv4 and IPv6 protocol families, as well as for the virtual private network (VPN) address family. Unicast RPF is not supported on interfaces configured as tunnel sources. This affects only the transit packets exiting the tunnel.

NOTE: RPF check is not supported on vxlan-enabled interface on QFX Series and EX Series switches.

There are two modes of unicast RPF, *strict mode*, and *loose mode*. The default is strict mode, which means the switch forwards a packet only if the receiving interface is the best return path to the packet's unicast source address. Strict mode is especially useful on untrusted interfaces (where untrusted users or processes can place packets on the network segment), and for symmetrically routed interfaces (see "When to Enable Unicast RPF" on page 699.) For more information about strict unicast RPF, see RFC 3704, *Ingress Filtering for Multihomed Networks* at http://www.ietf.org/rfc/rfc3704.txt.

To enable strict mode unicast RPF on a selected customer-edge interface:

```
[edit interfaces]user@switch# set interface-name unit 0 family inet rpf-check
```

The other mode is loose mode, which means the system checks to see if the packet has a source address with a corresponding prefix in the routing table, but it does not check whether the receiving interface is the best return path to the packet's unicast source address.

To enable unicast RPF loose mode, enter:

```
[edit interfaces]user@switch# set interface-name unit 0 family inet rpf-check mode loose
```

NOTE: On Juniper Networks EX3200, EX4200, and EX4300 Ethernet Switches, the switch applies unicast RPF *globally* to all interfaces when unicast RPF is configured on any interface. For additional information, see "Limitations of the Unicast RPF Implementation on EX3200, EX4200, and EX4300 Switches" on page 701.

Unicast RPF for Switches Overview

Unicast RPF functions as an ingress filter that reduces the forwarding of IP packets that might be spoofing an address. By default, unicast RPF is disabled on the switch interfaces. The switch supports only the active paths method of determining the best return path back to a unicast source address. The active paths method looks up the best reverse path entry in the forwarding table. It does not consider alternate routes specified using routing-protocol-specific methods when determining the best return path.

If the forwarding table lists the receiving interface as the interface to use to forward the packet back to its unicast source, it is the best return path interface.

Unicast RPF Implementation

Unicast RPF Packet Filtering

When you enable unicast RPF on the switch, the switch handles traffic in the following manner:

- If the switch receives a packet on the interface that is the best return path to the unicast source address of that packet, the switch forwards the packet.
- If the best return path from the switch to the packet's unicast source address is not the receiving interface, the switch discards the packet.
- If the switch receives a packet that has a source IP address that does not have a routing entry in the forwarding table, the switch discards the packet.

Bootstrap Protocol (BOOTP) and DHCP Requests

Bootstrap protocol (BOOTP) and DHCP request packets are sent with a broadcast MAC address and therefore the switch does not perform unicast RPF checks on them. The switch forwards all BOOTP packets and DHCP request packets without performing unicast RPF checks.

Default Route Handling

If the best return path to the source is the default route (`0.0.0.0`) and the default route points to `reject`, the switch discards the packets. If the default route points to a valid network interface, the switch performs a normal unicast RPF check on the packets.

NOTE: On the EX4300, the default route is not used when the switch is configured in unicast RPF strict mode.

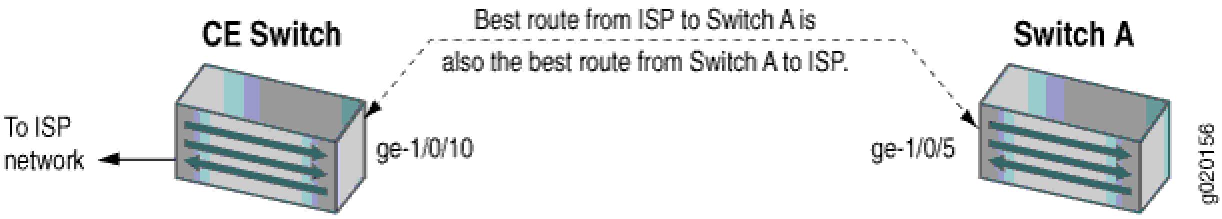
When to Enable Unicast RPF

Enable unicast RPF when you want to ensure that traffic arriving on a network interface comes from a source that resides on a network that interface can reach. You can enable unicast RPF on untrusted interfaces to filter spoofed packets. For example, a common application for unicast RPF is to help defend an enterprise network from DoS/DDoS attacks coming from the Internet.

Enable unicast RPF only on symmetrically routed interfaces, and as close as possible to the traffic source stops spoofed traffic before it can proliferate or reach interfaces that do not have unicast RPF enabled. Because unicast RPF is enabled globally on EX3200, EX4200, and EX4300 switches, ensure that *all* interfaces are symmetrically routed before you enable unicast RPF on these switches, as shown in Figure 40 on page 700. Enabling unicast RPF on asymmetrically routed interfaces results in packets from legitimate sources being filtered. A symmetrically routed interface uses the same route in both directions between the source and the destination.

Unicast RPF is enabled globally on EX3200, EX4200, and EX4300 switches, to with these devices, be sure that *all* interfaces are symmetrically routed before you enable unicast RPF on these switches. Enabling unicast RPF on asymmetrically routed interfaces results in packets from legitimate sources being filtered.

Figure 40: Symmetrically Routed Interfaces


CE Switch
Best route from ISP to Switch A is
also the best route from Switch A to ISP.
Switch A
To ISP network
ge-1/0/10
ge-1/0/5
g020156

The following switch interfaces are most likely to be symmetrically routed and thus are candidates for unicast RPF enabling:

- The service provider edge to a customer
- The customer edge to a service provider
- A single access point out of the network (usually on the network perimeter)
- A terminal network that has only one link

On EX3200, EX4200, and EX4300 switches, we recommend that you enable unicast RPF explicitly on either all interfaces or only one interface. To avoid possible confusion, do not enable it on only some interfaces:

- Enabling unicast RPF explicitly on only one interface makes it easier if you choose to disable it in the future because you must explicitly disable unicast RPF on every interface on which you explicitly enabled it. If you explicitly enable unicast RPF on two interfaces and you disable it on only one interface, unicast RPF is still implicitly enabled globally on the switch. The drawback of this approach is that the switch displays the flag that indicates that unicast RPF is enabled only on interfaces on which unicast RPF is explicitly enabled, so even though unicast RPF is enabled on all interfaces, this status is not displayed.

- Enabling unicast RPF explicitly on all interfaces makes it easier to know whether unicast RPF is enabled on the switch because every interface shows the correct status. (Only interfaces on which you explicitly enable unicast RPF display the flag that indicates that unicast RPF is enabled.) The drawback of this approach is that if you want to disable unicast RPF, you must explicitly disable it on every interface. If unicast RPF is enabled on any interface, it is implicitly enabled on all interfaces.

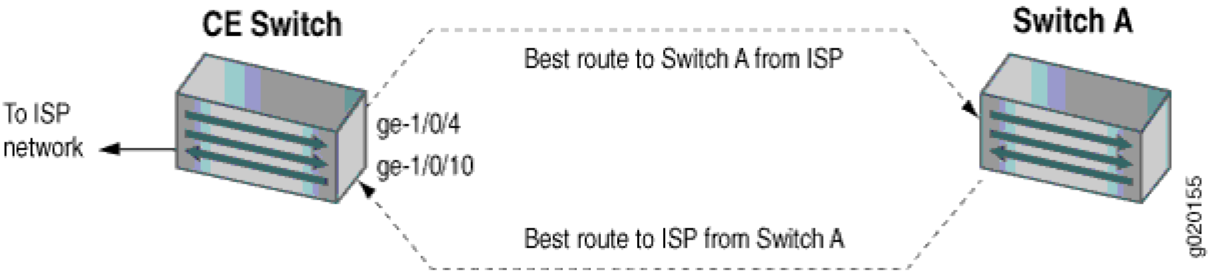
When Not to Enable Unicast RPF

Typically, you will not enable unicast RPF if:

- Switch interfaces are multihomed.
- Switch interfaces are trusted interfaces.
- BGP is carrying prefixes and some of those prefixes are not advertised or are not accepted by the ISP under its policy. (The effect in this case is the same as filtering an interface by using an incomplete access list.)
- Switch interfaces face the network core. Core-facing interfaces are usually asymmetrically routed.

An asymmetrically routed interface uses different paths to send and receive packets between the source and the destination, as shown in Figure 41 on page 701. This means that if an interface receives a packet, that interface does not match the forwarding table entry as the best return path back to the source. If the receiving interface is not the best return path to the source of a packet, unicast RPF causes the switch to discard the packet even though it comes from a valid source.

Figure 41: Asymmetrically Routed Interfaces


CE Switch
Switch A
Best route to Switch A from ISP
To ISP network
ge-1/0/4
ge-1/0/10
Best route to ISP from Switch A
g020155

NOTE: Do not enable unicast RPF on EX3200, EX4200, and EX4300 switches if any switch interfaces are asymmetrically routed, because unicast RPF is enabled globally on all interfaces of these switches. All switch interfaces must be symmetrically routed for you to enable unicast RPF without the risk of the switch discarding traffic that you want to forward.

Limitations of the Unicast RPF Implementation on EX3200, EX4200, and EX4300 Switches

On EX3200, EX4200, and EX4300 switches, the switch implements unicast RPF on a global basis. You cannot enable unicast RPF on a per-interface basis. Unicast RPF is globally disabled by default.

- When you enable unicast RPF on any interface, it is automatically enabled on all switch interfaces, including link aggregation groups (LAGs), integrated routing and bridging (IRB) interfaces, and routed VLAN interfaces (RVIs).
- When you disable unicast RPF on the interface (or interfaces) on which you enabled unicast RPF, it is automatically disabled on all switch interfaces.

> **NOTE**: You must explicitly disable unicast RPF on every interface on which it was explicitly enabled or unicast RPF remains enabled on all switch interfaces.

QFX switches, OCX switches, and EX3200 and EX4200 switches do not perform unicast RPF filtering on equal-cost multipath (ECMP) traffic. The unicast RPF check examines only one best return path to the packet source, but ECMP traffic employs an address block consisting of multiple paths. Using unicast RPF to filter ECMP traffic on these switches can result in the switch discarding packets that you want to forward because the unicast RPF filter does not examine the entire ECMP address block.

RELATED DOCUMENTATION

Example: Configuring Unicast RPF (On a Switch)

Troubleshooting Unicast RPF

Understanding Unicast RPF (Routers)

IN THIS SECTION

For interfaces that carry IPv4 or IPv6 traffic, you can reduce the impact of denial of service (DoS) attacks by configuring unicast reverse path forwarding (RPF). Unicast RPF helps determine the source of attacks and rejects packets from unexpected source addresses on interfaces where unicast RPF is enabled.

NOTE:

- You can protect a network by applying unicast RPF check feature at the edge (on customer facing interfaces) of the network. In an ISP environment, this can impact the network which can impose on a scaled setup. In case if you have already protected the edge of your network, a packet with a spoofed IP source address would not even appear in a core facing interface. In this case, unicast RPF check is not necessary. Enabling unicast RPF feature can impact the control plane performance, so use it where it is required. So it is strongly recommended not to enable this feature on the network core (internal) interfaces.

NOTE: Currently on PTX platforms, configuring BGP flow specification (flowspec) creates an implicit filter to set the VRF instance. On PTX platforms, the filter lookup precedes the source/destination IP lookup. Therefore, source and destination IP lookup happens within the context of the VRF instance.

NOTE: Currently on PTX platforms, when Unicast Reverse Path Forwrading (uRPF) and filter based forwarding (FBF) is configured on an interface, the default behavior is that the source IP address lookup of an incoming packet is done by uRPF in the routing instance that the interface points to and destination IP address lookup is done in the routing instance specified by the FBF filter. You can use `set forwarding-options no-rpf-fbf-handling` to override this default behavior and when you apply this configuration, the source IP address lookup of an incoming packet is done by uRPF in the routing instance specified by the FBF filter.

Unicast RPF and Default Route

IN THIS SECTION

When the active route cannot be chosen from the routes in a routing table, the router chooses a default route. A default route is equivalent to an IP address of 0.0.0.0/0. If you configure a default route, and you configure unicast RPF on an interface that the default route uses, unicast RPF behaves differently than it does otherwise.

To determine whether the default route uses an interface, enter the `show route` command:

```
user@host> show route address
```

address is the next-hop address of the configured default route. The default route uses the interfaces shown in the output of the `show route` command.

The following sections describe how unicast RPF behaves when a default route uses an interface and when a default route does not use an interface:

Unicast RPF Behavior with a Default Route

On all routers except those with MPCs and the MX80 router, unicast RPF behaves as follows if you configure a default route that uses an interface configured with unicast RPF:

- Loose mode—All packets are automatically accepted. For this reason, we recommend that you not configure unicast RPF loose mode on interfaces that the default route uses.

- Strict mode—The packet is accepted when the source address of the packet matches any of the routes (either default or learned) that can be reachable through the interface. Note that routes can have multiple destinations associated with them; therefore, if one of the destinations matches the incoming interface of the packet, the packet is accepted.

On all routers with MPCs and the MX80 router, unicast RPF behaves as follows if you configure a default route that uses an interface configured with unicast RPF:

- Loose mode—All packets except the packets whose source is learned from the default route are accepted. All packets whose source is learned from the default route are dropped at the Packet Forwarding Engine. The default route is treated as if the route does not exist.

- Strict mode—The packet is accepted when the source address of the packet matches any of the routes (either default or learned) that can be reachable through the interface. Note that routes can have multiple destinations associated with them; therefore, if one of the destinations matches the incoming interface of the packet, the packet is accepted.

On all routers, the packet is not accepted when either of the following is true:

- The source address of the packet does not match a prefix in the routing table.
- The interface does not expect to receive a packet with this source address prefix.

Unicast RPF Behavior Without a Default Route

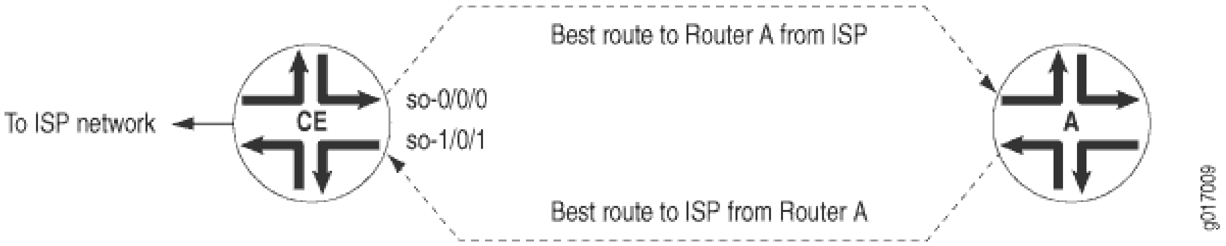
If you do not configure a default route, or if the default route does not use an interface configured with unicast RPF, unicast RPF behaves as described in "Configuring Unicast RPF Strict Mode" on page 706 and "Configuring Unicast RPF Loose Mode" on page 709. To summarize, unicast RPF without a default route behaves as follows:

- Strict mode—The packet is not accepted when either of the following is true:
  - The packet has a source address that does not match a prefix in the routing table.
  - The interface does not expect to receive a packet with this source address prefix.
- Loose mode—The packet is not accepted when the packet has a source address that does not match a prefix in the routing table.

Unicast RPF with Routing Asymmetry

In general, we recommend that you not enable unicast RPF on interfaces that are internal to the network because internal interfaces are likely to have *routing asymmetry*. Routing asymmetry means that a packet's outgoing and return paths are different. Routers in the core of the network are more likely to have asymmetric reverse paths than routers at the customer or provider edge. Figure 42 on page 705 shows unicast RPF in an environment with routing asymmetry.

Figure 42: Unicast RPF with Routing Asymmetry


Best route to Router A from ISP
To ISP network
CE
so-0/0/0
so-1/0/1
A
Best route to ISP from Router A
g017009

In Figure 42 on page 705, if you enable unicast RPF on interface `so-0/0/0`, traffic destined for Router A is not rejected. If you enable unicast RPF on interface `so-1/0/1`, traffic from Router A is rejected.

If you need to enable unicast RPF in an asymmetric routing environment, you can use fail filters to allow the router to accept incoming packets that are known to be arriving by specific paths. For an example of a fail filter that accepts packets with a specific source and destination address, see "Configuring Unicast RPF" on page 714.

Configuring Unicast RPF Strict Mode

In strict mode, unicast RPF checks whether the incoming packet has a source address that matches a prefix in the routing table, and whether the interface expects to receive a packet with this source address prefix.

When *active-path* mode is configured, outgoing interface-list of only active route is created. Any packet coming on these interfaces is considered valid and is processed.

If the incoming packet fails the unicast RPF check, the packet is not accepted on the interface. When a packet is not accepted on an interface, unicast RPF counts the packet and sends it to an optional fail filter. If the fail filter is not configured, the default action is to silently discard the packet.

The optional fail filter allows you to apply a filter to packets that fail the unicast RPF check. You can define the fail filter to perform any filter operation, including accepting, rejecting, logging, sampling, or policing.

When unicast RPF is enabled on an interface, Bootstrap Protocol (BOOTP) packets and Dynamic Host Configuration Protocol (DHCP) packets are not accepted on the interface. To allow the interface to accept BOOTP packets and DHCP packets, you must apply a fail filter that accepts all packets with a source address of `0.0.0.0` and a destination address of `255.255.255.255`. For a configuration example, see "Configuring Unicast RPF" on page 714.

For more information about defining fail filters, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

To configure unicast RPF, include the `rpf-check` statement:

```
rpf-check <fail-filter filter-name>;
```

You can include this statement at the following hierarchy levels:

- `[edit interfaces interface-name unit logical-unit-number family (inet | inet6)]`
- `[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number family (inet | inet6)]`

Using unicast RPF can have several consequences when implemented with traffic filters:

- RPF fail filters are evaluated after input filters and before output filters.

- If you configure a filter counter for packets dropped by an input filter, and you want to know the total number of packets dropped, you must also configure a filter counter for packets dropped by the RPF check.
- To count packets that fail the RPF check and are accepted by the RPF fail filter, you must configure a filter counter.
- If an input filter forwards packets anywhere other than the inet.0 or inet6.0 routing tables, the unicast RPF check is not performed.
- If an input filter forwards packets anywhere other than the routing instance the input interface is configured for, the unicast RPF check is not performed.

NOTE:
In the aforementioned bulleted list, the first, second-last, and last points are not applicable for MX platforms because on MX platforms uRPF is processed prior to the execution of firewall filters. uRPF check is processed for source address checking before any FBF (filter-based forwarding) actions are enabled for static and dynamic interfaces. This applies to both IPv4 and IPv6 families.

NOTE: On ACX and MX Series routers:

- The uRPF fail filter is supported on ACX1000, ACX2000, ACX4000, and ACX500, ACX5048, and ACX5096. The filter is not supported on ACX5448, ACX710, ACX7100-32C, ACX7100-48, ACX7509, and all routers of ACX7000 series.
- The uRPF fail filter cannot match packets failed at ingress port check (strict mode).
- The uRPF fail filter can match packets failing source IP lookup but cannot match packets failing the input interface check (strict mode).
- The uRPF fail filter applies only to interface-specific instances of the firewall filter.
- The uRPF fail filters do not support reject and routing-instance actions.

Configure unicast RPF strict mode, and apply a fail filter that allows the interface to accept BOOTP packets and DHCP packets. The filter accepts all packets with a source address of `0.0.0.0` and a destination address of `255.255.255.255`.

To configure unicast RPF in strict mode:

1. Configure the fail filter:

```
[edit firewall]
filter rpf-special-case-dhcp-bootp {
    term allow-dhcp-bootp {
        from {
            source-address {
                0.0.0.0/32;
            }
            address {
                255.255.255.255/32;
            }
        }
        then {
            count rpf-dhcp-bootp-traffic;
            accept;
        }
    }
    term default {
        then {
            log;
            reject;
        }
    }
}
```

2. Configure unicast RPF on interfaces:

```
[edit]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                rpf-check fail-filter rpf-special-case-dhcp-bootp;
            }
        }
    }
}
```

3. Commit the configuration.

```
[edit]
commit;
```

Configuring Unicast RPF Loose Mode

By default, unicast RPF uses strict mode. Unicast RPF loose mode is similar to unicast RPF strict mode and has the same configuration restrictions. The only check in loose mode is whether the packet has a source address with a corresponding prefix in the routing table; loose mode does not check whether the interface expects to receive a packet with a specific source address prefix. If a corresponding prefix is not found, unicast RPF loose mode does not accept the packet. As in strict mode, loose mode counts the failed packet and optionally forwards it to a fail filter, which either accepts, rejects, logs, samples, or polices the packet.

When *feasible-paths* mode is configured, outgoing interface list of active and inactive routes are created. Any packet coming on these interfaces is considered valid and is processed.

To configure unicast RPF loose mode, include the `mode`:

1.

   ```



                       mode loose;
   ```

   You can include this statement at the following hierarchy levels:

   - `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family (inet | inet6) rpf-check <fail-filter` *`filter-name`*`>]`
   - `[edit logical-systems` *`logical-system-name`* `interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family (inet | inet6) rpf-check <fail-filter` *`filter-name`*`>]`

2. For example:

   In this example, no special configuration beyond device initialization is required.

   Configure unicast RPF loose mode, and apply a fail filter that allows the interface to accept BOOTP packets and DHCP packets. The filter accepts all packets with a source address of `0.0.0.0` and a destination address of `255.255.255.255`.

   To configure unicast RPF in loose mode:

a. Configure the fail filter:

```
[edit firewall]
filter rpf-special-case-dhcp-bootp {
    term allow-dhcp-bootp {
        from {
            source-address {
                0.0.0.0/32;
            }
            address {
                255.255.255.255/32;
            }
        }
        then {
            count rpf-dhcp-bootp-traffic;
            accept;
        }
    }
    term default {
        then {
            log;
            reject;
        }
    }
}
```

b. Configure unicast RPF on interfaces:

```
[edit]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                rpf-check fail-filter rpf-special-case-dhcp-bootp;
                mode loose;
            }
        }
    }
}
```

c. Commit the configuration.

```
[edit]
commit;
```

Configuring Unicast RPF Loose Mode with Ability to Discard Packets

Unicast RPF loose mode has the ability to discard packets with the source address pointing to the discard interface. Using unicast RPF loose mode, along with Remote Triggered Null Route filtering, provides an efficient way to discard packets coming from known attack sources. BGP policies in edge routers ensure that packets with untrusted source addresses have their next hop set to a discard route. When a packet arrives at the router with an untrusted source address, unicast RPF performs a route lookup of the source address. Because the source address route points to a discard next hop, the packet is dropped and a counter is incremented. This feature is supported on both IPv4 (inet) and IPv6 (inet6) address families.

To configure unicast RPF loose mode with the ability to discard packets, include the `rpf-loose-mode-discard family (inet | inet6)` statement at the `[edit forwarding-options]` hierarchy level:

```
rpf-loose-mode-discard {
    family {
        inet;
    }
}
```

In this example, no special configuration beyond device initialization is required.

Configure unicast RPF loose mode, and apply a fail filter that allows the interface to accept BOOTP packets and DHCP packets. The filter accepts all packets with a source address of `0.0.0.0` and a destination address of `255.255.255.255`.

To configure unicast RPF loose mode with the ability to discard packets:

1. Configure the fail filter:

```
[edit firewall]
filter rpf-special-case-dhcp-bootp {
    term allow-dhcp-bootp {
        from {
            source-address {
                0.0.0.0/32;
            }
            address {
                255.255.255.255/32;
            }
        }
        then {
            count rpf-dhcp-bootp-traffic;
            accept;
        }
    }
    term default {
        then {
            log;
            reject;
        }
    }
}
```

2. Configure unicast RPF on interfaces:

```
[edit]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                rpf-check fail-filter rpf-special-case-dhcp-bootp;
                mode loose;
            }
        }
    }
}
```

3. Configure the ability to discard packets.

```
[edit]
forwarding-options{
    rpf-loose-mode-discard {
        family {
            inet;
            }
        }
    }
```

4. Commit the configuration.

```
[edit]
commit;
```

Configuring Unicast RPF on a VPN

You can configure unicast RPF on a VPN interface by enabling unicast RPF on the interface and including the `interface` statement at the `[edit routing-instances` *`routing-instance-name`*`]` hierarchy level.

You can configure unicast RPF only on the interfaces you specify in the routing instance. This means the following:

- For Layer 3 VPNs, unicast RPF is supported on the CE router interface.
- Unicast RPF is not supported on core-facing interfaces.
- For virtual-router routing instances, unicast RPF is supported on all interfaces you specify in the routing instance.
- If an input filter forwards packets anywhere other than the routing instance the input interface is configured for, the unicast RPF check is not performed.

Configure unicast RPF on a Layer 3 VPN interface:

```
[edit interfaces]
so-0/0/0 {
    unit 0 {
        family inet {
            rpf-check;
        }
    }
}
[edit routing-instance]
VPN-A {
    interface so-0/0/0.0;
}
```

Configuring Unicast RPF

Configure unicast RPF strict mode, and apply a fail filter that allows the interface to accept BOOTP packets and DHCP packets. The filter accepts all packets with a source address of `0.0.0.0` and a destination address of `255.255.255.255`.

```
[edit firewall]
filter rpf-special-case-dhcp-bootp {
    term allow-dhcp-bootp {
        from {
            source-address {
                0.0.0.0/32;
            }
            address {
                255.255.255.255/32;
            }
        }
        then {
            count rpf-dhcp-bootp-traffic;
            accept;
        }
    }
    term default {
        then {
            log;
            reject;
        }
    }
}
[edit]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                rpf-check fail-filter rpf-special-case-dhcp-bootp;
            }
        }
    }
}
```

SEE ALSO

unicast-reverse-path

Example: Configuring Unicast RPF (On a Switch)

IN THIS SECTION

This example shows how to help defend ingress interfaces against denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks by configuring unicast RPF (uRPF) to filter incoming traffic.

Requirements

This example uses two EX switches, referred to in this topic as Switch A and Switch B. Certain EX switch models allow you to configure uRPF on individual interfaces. Whereas on certain EX switch models you cannot configure individual interfaces for uRPF – the switch applies uRPF globally to all interfaces on the switch.

- Any Junos OS release for EX switches but not earlier than Junos OS Release 10.1
- Two EX switches that support uRPF configuration on individual interfaces.

Before you begin, ensure you have:

- Connected the two switches by symmetrically routed interfaces.
- Ensured that the interface on which you will configure unicast RPF is symmetrically routed. A symmetrically routed interface is an interface that uses the same route in both directions between the source and the destination. Do not enable unicast RPF on asymmetrically routed interfaces. An asymmetrically routed interface uses different paths to send and receive packets between the source and the destination.

- In this example, if you are using EX switches that apply uRPF globally to all interfaces, then ensure that all switch interfaces are symmetrically routed before you enable unicast RPF on an interface. When you enable unicast RPF on any interface, it is enabled globally on all switch interfaces. Do not enable unicast RPF on asymmetrically routed interfaces. An asymmetrically routed interface uses different paths to send and receive packets between the source and the destination.

Overview and Topology

In this example, an enterprise network's system administrator wants to protect Switch A against potential DoS and DDoS attacks from the Internet. The administrator configures unicast RPF on interface **xe-0/0/4** on Switch A. Packets arriving on interface **xe-0/0/4** on Switch A from the Switch B source also use incoming interface **xe-0/0/4** as the best return path to send packets back to the source. In this topology, Switch A and Switch B are both connected by symmetrically routed interfaces.

- Switch A is on the edge of an enterprise network. The interface **xe-0/0/4** on Switch A connects to the interface **xe-0/0/5** on Switch B.

- Switch B is on the edge of the service provider network that connects the enterprise network to the Internet.

Topology

Configuration

To enable unicast RPF, perform these tasks:

Procedure

CLI Quick Configuration

To quickly configure unicast RPF on Switch A, copy the following command and paste it into the switch terminal window:

```
[edit interfaces]
set xe-0/0/4 unit 0 family inet rpf-check
```

Step-by-Step Procedure

To configure unicast RPF on Switch A:

1. Enable unicast RPF on interface **xe-0/0/4**:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/4 unit 0 family inet rpf-check
   ```

Results

Check the results:

```
[edit interfaces]
user@switch# show
xe-0/0/4 {
    unit 0 {
        family inet {
            rpf-check;
        }
    }
}
```

Disabling Unicast RPF

IN THIS SECTION

Procedure

Step-by-Step Procedure

Verification

IN THIS SECTION

Unicast reverse-path forwarding (RPF) can help protect your LAN from denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks on untrusted interfaces. Unicast RPF filters traffic with source addresses that do not use the incoming interface as the best return path back to the source. If the network configuration changes so that an interface that has unicast RPF enabled becomes a trusted interface or becomes asymmetrically routed (the interface that receives a packet is not the best return path to the packet's source), disable unicast RPF.

NOTE: To disable uRPF on EX switches that apply uRPF globally to all interfaces, you must delete it from every interface on which you explicitly configured it. If you do not disable unicast RPF on every interface on which you explicitly enabled it, it remains implicitly enabled on all interfaces. If you attempt to delete unicast RPF from an interface on which it was not explicitly enabled, the `warning: statement not found` message appears. If you do not disable unicast RPF on every interface on which you explicitly enabled it, unicast RPF remains implicitly enabled on all interfaces.

On EX switch models that allow you to configure uRPF on individual interfaces, the switch does not apply unicast RPF to an interface unless you explicitly enable that interface for unicast RPF.

To disable unicast RPF, delete its configuration from the interface:

[edit interfaces]
user@switch# **delete xe-0/0/4 unit 0 family inet rpf-check**

Verifying That Unicast RPF Is Enabled on the Switch

Purpose

Verify that unicast RPF is enabled and working on the interface.

Action

Use one of the `show interfaces` *`interface-name`* commands with either the **extensive** or **detail** options to verify that unicast RPF is enabled and working on the switch. The example below displays output from the `show interfaces ge- extensive` command.

```
user@switch> show interfaces xe-0/0/4.0 extensive
Physical interface: xe-0/0/4, Enabled, Physical link is Up
  Interface index: 147, SNMP ifIndex: 659
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 10Gbps, BPDU Error: None, Loop
Detect PDU Error: None, Ethernet-Switching Error: None,
  MAC-REWRITE Error: None, Loopback: None, Source filtering: Disabled, Flow control: Enabled,
Speed Configuration: Auto
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: 84:c1:c1:7b:a8:04, Hardware address: 84:c1:c1:7b:a8:04
  Last flapped   : 2023-04-04 10:34:13 PDT (00:01:29 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics                      Seconds
    Bit errors                             2
    Errored blocks                         2
  Link Degrade :
    Link Monitoring                   :  Disable
  Interface transmit statistics: Disabled

  Logical interface xe-0/0/4.0 (Index 335) (SNMP ifIndex 696)
    Flags: Up SNMP-Traps 0x4004000 Encapsulation: ENET2
    Input packets : 0
    Output packets: 1
    Protocol inet, MTU: 1500
    Max nh cache: 100000, New hold nh limit: 100000, Curr nh cnt: 0, Curr new hold cnt: 0, NH 
drop cnt: 0
      Flags: Sendbcast-pkt-to-re, uRPF
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 10.0.1/24, Local: 10.0.1.1, Broadcast: 10.0.1.255
    Protocol multiservice, MTU: Unlimited
      Flags: Is-Primary
```

Meaning

The `show interfaces xe-0/0/4 extensive` command (and the `show interfaces xe-0/0/4 detail` command) displays in-depth information about the interface. The **Flags:** output field near the bottom of the display reports the unicast RPF status. If unicast RPF has not been enabled, the **uRPF** flag is not displayed.

On EX switches that apply uRPF globally to all interfaces, uRPF is implicitly enabled on *all* switch interfaces, including aggregated Ethernet interfaces (also referred to as link aggregation groups or LAGs) and routed VLAN interfaces (RVIs) when you enable uRPF on a single interface. However, the uRPF status is shown as enabled only on interfaces for which you have explicitly configured uRPF. Thus, the **uRPF** flag is not displayed on interfaces for which you have not explicitly configured uRPF even though uRPF is implicitly enabled on all interfaces.

Troubleshooting Unicast RPF

IN THIS SECTION

Legitimate Packets Are Discarded

Problem

The switch filters valid packets from legitimate sources, which results in the switch's discarding packets that should be forwarded.

Solution

The interface or interfaces on which legitimate packets are discarded are asymmetrically routed interfaces. An asymmetrically routed interface uses different paths to send and receive packets between the source and the destination, so the interface that receives a packet is not the same interface the switch uses to reply to the packet's source.

Unicast RPF works properly only on symmetrically routed interfaces. A symmetrically routed interface is an interface that uses the same route in both directions between the source and the destination. Unicast RPF filters packets by checking the forwarding table for the best return path to the source of an incoming packet. If the best return path uses the same interface as the interface that received the packet, the switch forwards the packet. If the best return path uses a different interface than the interface that received the packet, the switch discards the packet.

NOTE:
On EX switches that apply uRPF globally to all interfaces, uRPF works properly only if all switch interfaces—including aggregated Ethernet interfaces (also referred to as link aggregation groups or LAGs), integrated routing and bridging (IRB) interfaces, and routed VLAN interfaces (RVIs)—are symmetrically routed, because unicast RPF is enabled globally on all switch interfaces.

RELATED DOCUMENTATION

Understanding Unicast RPF (Switches)

Example: Configuring Unicast RPF (On a Router)

IN THIS SECTION

This example shows how to help defend ingress interfaces against denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks by configuring unicast RPF on a customer-edge interface to filter incoming traffic.

Requirements

No special configuration beyond device initialization is required.

Overview

IN THIS SECTION

In this example, Device A is using OSPF to advertise a prefix for the link that connects to Device D. Device B has unicast RPF configured. OSPF is enabled on the links between Device B and Device C and the links between Device A and Device C, but not on the links between Device A and Device B. Therefore, Device B learns about the route to Device D through Device C.

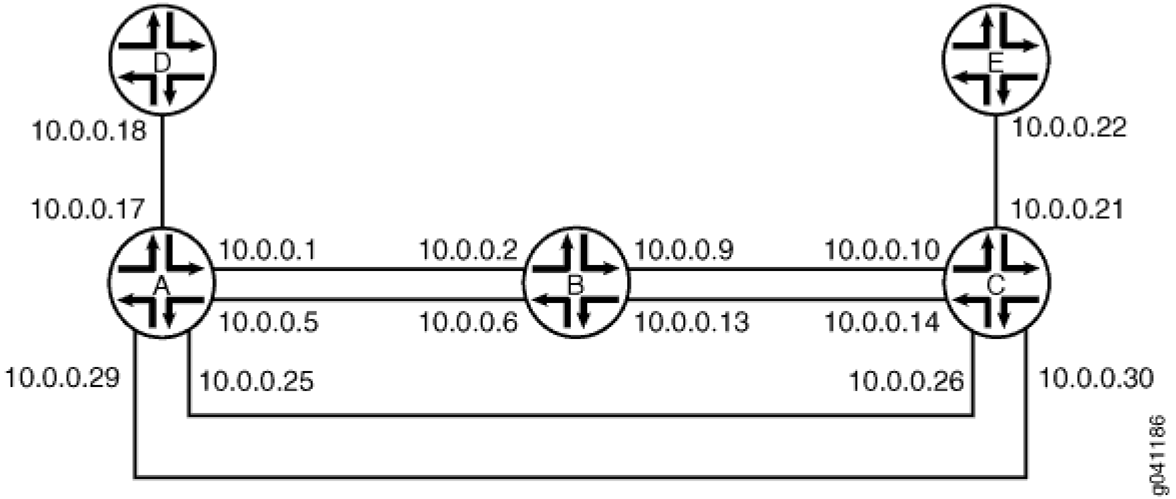
If ingress filtering is used in an environment where DHCP or BOOTP is used, it should be ensured that the packets with a source address of 0.0.0.0 and a destination address of 255.255.255.255 are allowed to reach the relay agent in routers when appropriate.

This example also includes a fail filter. When a packet fails the unicast RPF check, the fail filter is evaluated to determine if the packet should be accepted anyway. The fail filter in this example allows Device B’s interfaces to accept Dynamic Host Configuration Protocol (DHCP) packets. The filter accepts all packets with a source address of 0.0.0.0 and a destination address of 255.255.255.255.

Topology

Figure 43 on page 723 shows the sample network.

Figure 43: Unicast RPF Sample Topoolgy



Configuration

IN THIS SECTION

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

Device A

```
set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces fe-0/0/2 unit 5 family inet address 10.0.0.5/30
set interfaces fe-0/0/1 unit 17 family inet address 10.0.0.17/30
set interfaces fe-0/1/1 unit 25 family inet address 10.0.0.25/30
set interfaces fe-1/1/1 unit 29 family inet address 10.0.0.29/30
set protocols ospf export send-direct
set protocols ospf area 0.0.0.0 interface fe-0/1/1.25
set protocols ospf area 0.0.0.0 interface fe-1/1/1.29
set policy-options policy-statement send-direct from protocol direct
set policy-options policy-statement send-direct from route-filter 10.0.0.16/30 exact
set policy-options policy-statement send-direct then accept
```

Device B

```
set interfaces fe-1/2/0 unit 2 family inet rpf-check fail-filter rpf-special-case-dhcp
set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/1/1 unit 6 family inet rpf-check fail-filter rpf-special-case-dhcp
set interfaces fe-1/1/1 unit 6 family inet address 10.0.0.6/30
set interfaces fe-0/1/1 unit 9 family inet rpf-check fail-filter rpf-special-case-dhcp
set interfaces fe-0/1/1 unit 9 family inet address 10.0.0.9/30
set interfaces fe-0/1/0 unit 13 family inet rpf-check fail-filter rpf-special-case-dhcp
set interfaces fe-0/1/0 unit 13 family inet address 10.0.0.13/30
set protocols ospf area 0.0.0.0 interface fe-0/1/1.9
set protocols ospf area 0.0.0.0 interface fe-0/1/0.13
set routing-options forwarding-table unicast-reverse-path active-paths
set firewall filter rpf-special-case-dhcp term allow-dhcp from source-address 0.0.0.0/32
set firewall filter rpf-special-case-dhcp term allow-dhcp from destination-address
255.255.255.255/32
set firewall filter rpf-special-case-dhcp term allow-dhcp then count rpf-dhcp-traffic
set firewall filter rpf-special-case-dhcp term allow-dhcp then accept
set firewall filter rpf-special-case-dhcp term default then log
set firewall filter rpf-special-case-dhcp term default then reject
```

Device C

```
set interfaces fe-1/2/0 unit 10 family inet address 10.0.0.10/30
set interfaces fe-0/0/2 unit 14 family inet address 10.0.0.14/30
set interfaces fe-1/0/2 unit 21 family inet address 10.0.0.21/30
set interfaces fe-1/2/2 unit 26 family inet address 10.0.0.26/30
set interfaces fe-1/2/1 unit 30 family inet address 10.0.0.30/30
set protocols ospf area 0.0.0.0 interface fe-1/2/0.10
set protocols ospf area 0.0.0.0 interface fe-0/0/2.14
set protocols ospf area 0.0.0.0 interface fe-1/2/2.26
set protocols ospf area 0.0.0.0 interface fe-1/2/1.30
```

Device D

```
set interfaces fe-1/2/0 unit 18 family inet address 10.0.0.18/30
```

Device E

```
set interfaces fe-1/2/0 unit 22 family inet address 10.0.0.22/30
```

Configuring Device A

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 Device A:

1. Configure the interfaces.

   ```
   [edit interfaces]
   user@A# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30
   user@A# set fe-0/0/2 unit 5 family inet address 10.0.0.5/30
   user@A# set fe-0/0/1 unit 17 family inet address 10.0.0.17/30
   user@A# set fe-0/1/1 unit 25 family inet address 10.0.0.25/30
   user@A# set fe-1/1/1 unit 29 family inet address 10.0.0.29/30
   ```

2. Configure OSPF.

   ```
   [edit protocols ospf]
   user@A# set export send-direct
   user@A# set area 0.0.0.0 interface fe-0/1/1.25
   user@A# set area 0.0.0.0 interface fe-1/1/1.29
   ```

3. Configure the routing policy.

   ```
   [edit policy-options policy-statement send-direct]
   user@A# set from protocol direct
   ```

```
user@A# set from route-filter 10.0.0.16/30 exact
user@A# set then accept
```

4. If you are done configuring Device A, commit the configuration.

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

Configuring Device B

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 Device B:

1. Configure the interfaces.

```
[edit interfaces]
user@B# set fe-1/2/0 unit 2 family inet address 10.0.0.2/30
user@B# set fe-1/1/1 unit 6 family inet address 10.0.0.6/30
user@B# set fe-0/1/1 unit 9 family inet address 10.0.0.9/30
user@B# set fe-0/1/0 unit 13 family inet address 10.0.0.13/30
```

2. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@B# set interface fe-0/1/1.9
user@B# set interface fe-0/1/0.13
```

3. Configure unicast RPF, and apply the optional fail filter.

```
[edit interfaces]
user@B# set fe-1/2/0 unit 2 family inet rpf-check fail-filter rpf-special-case-dhcp
user@B# set fe-1/1/1 unit 6 family inet rpf-check fail-filter rpf-special-case-dhcp
user@B# set fe-0/1/1 unit 9 family inet rpf-check fail-filter rpf-special-case-dhcp
user@B# set fe-0/1/0 unit 13 family inet rpf-check fail-filter rpf-special-case-dhcp
```

4. (Optional) Configure the fail filter that gets evaluated if a packet fails the RPF check.

```
[edit firewall filter rpf-special-case-dhcp]
user@B# set term allow-dhcp from source-address 0.0.0.0/32
user@B# set term allow-dhcp from destination-address 255.255.255.255/32
user@B# set term allow-dhcp then count rpf-dhcp-traffic
user@B# set term allow-dhcp then accept
user@B# set term default then log
user@B# set term default then reject
```

5. (Optional) Configure only active paths to be considered in the RPF check.

   This is the default behavior.

```
[edit routing-options forwarding-table]
user@B# set unicast-reverse-path active-paths
```

6. If you are done configuring Device B, commit the configuration.

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

Results

Confirm your configuration by issuing the `show firewall`, `show interfaces`, `show protocols`, `show routing-options`, and `show policy-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Device A

```
user@A# show interfaces
fe-1/2/0 {
    unit 1 {
        family inet {
            address 10.0.0.1/30;
        }
    }
}
fe-0/0/2 {
    unit 5 {
        family inet {
            address 10.0.0.5/30;
        }
    }
}
fe-0/0/1 {
    unit 17 {
        family inet {
            address 10.0.0.17/30;
        }
    }
}
fe-0/1/1 {
    unit 25 {
        family inet {
            address 10.0.0.25/30;
        }
    }
}
fe-1/1/1 {
    unit 29 {
        family inet {
            address 10.0.0.29/30;
        }
    }
}
```

```
user@A# show protocols
ospf {
    export send-direct;
    area 0.0.0.0 {
        interface fe-0/1/1.25;
        interface fe-1/1/1.29;
    }
}
```

```
user@A# show policy-options
policy-statement send-direct {
    from {
        protocol direct;
        route-filter 10.0.0.16/30 exact;
    }
    then accept;
}
```

Device B

```
user@B# show firewall
filter rpf-special-case-dhcp {
    term allow-dhcp {
        from {
            source-address {
                0.0.0.0/32;
            }
            destination-address {
                255.255.255.255/32;
            }
        }
        then {
            count rpf-dhcp-traffic;
            accept;
        }
    }
    term default {
        then {
            log;
            reject;
        }
    }
}
user@B# show interfaces
fe-1/2/0 {
    unit 2 {
        family inet {
            rpf-check fail-filter rpf-special-case-dhcp;
            address 10.0.0.2/30;
        }
    }
}
fe-1/1/1 {
    unit 6 {
        family inet {
            rpf-check fail-filter rpf-special-case-dhcp;
            address 10.0.0.6/30;
        }
    }
}
fe-0/1/1 {
    unit 9 {
        family inet {
            rpf-check fail-filter rpf-special-case-dhcp;
            address 10.0.0.9/30;
        }
    }
}
fe-0/1/0 {
    unit 13 {
        family inet {
            rpf-check fail-filter rpf-special-case-dhcp;
            address 10.0.0.13/30;
        }
    }
}
```

```
user@B# show protocols
ospf {
    area 0.0.0.0 {
        interface fe-0/1/1.9;
        interface fe-0/1/0.13;
    }
}
```

```
user@B# show routing-options
forwarding-table {
    unicast-reverse-path active-paths;
}
```

Enter the configurations on Device C, Device D, and Device E, as shown in "CLI Quick Configuration" on page 723.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Confirm That Unicast RPF Is Enabled

Purpose

Make sure that the interfaces on Device B have unicast RPF enabled.

Action

```
user@B> show interfaces fe-0/1/0.13 extensive
 Logical interface fe-0/1/0.13 (Index 73) (SNMP ifIndex 553) (Generation 208)
    Flags: SNMP-Traps 0x4000 Encapsulation: ENET2
    Traffic statistics:
     Input  bytes  :               999390
     Output bytes  :              1230122
     Input  packets:                12563
     Output packets:                12613
    Local statistics:
     Input  bytes  :               998994
     Output bytes  :              1230122
     Input  packets:                12563
     Output packets:                12613
    Transit statistics:
     Input  bytes  :                  396                    0 bps
     Output bytes  :                    0                    0 bps
     Input  packets:                    0                    0 pps
     Output packets:                    0                    0 pps
    Protocol inet, MTU: 1500, Generation: 289, Route table: 22
      Flags: Sendbcast-pkt-to-re, uRPF
      RPF Failures: Packets: 0, Bytes: 0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 10.0.0.12/30, Local: 10.0.0.13, Broadcast: 10.0.0.15, Generation: 241
```

Meaning

The **uRPF** flag confirms that unicast RPF is enabled on this interface.

Confirm That the Source Addresses Are Blocked

Purpose

Use the `ping` command to make sure that Device B blocks traffic from unexpected source addresses.

Action

From Device A, ping Device B's interfaces, using 10.0.0.17 as the source address.

```
user@A> ping 10.0.0.6 source 10.0.0.17
PING 10.0.0.6 (10.0.0.6): 56 data bytes
^C
--- 10.0.0.6 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss
```

Meaning

As expected, the ping operation fails.

Confirm That the Source Addresses Are Unblocked

Purpose

Use the `ping` command to make sure that Device B does not block traffic when the RPF check is deactivated.

Action

1. Deactivate the RPF check on one of the interfaces.

2. Rerun the ping operation.

```
user@B> deactivate interfaces fe-1/1/1.6 family inet rpf-check

user@A> ping 10.0.0.6 source 10.0.0.17
PING 10.0.0.2 (10.0.0.2): 56 data bytes
64 bytes from 10.0.0.2: icmp_seq=0 ttl=63 time=1.316 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=63 time=1.263 ms
^C
--- 10.0.0.2 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.263/1.289/1.316/0.027 ms
```

Meaning

As expected, the ping operation succeeds.

CHAPTER 26

Unknown Unicast Forwarding

IN THIS CHAPTER

Understanding and Preventing Unknown Unicast Forwarding

IN THIS SECTION

Unknown unicast traffic consists of unicast packets with unknown destination MAC addresses. By default, the switch floods these unicast packets that traverse a VLAN to all interfaces that are members of that VLAN. Forwarding this type of traffic can create unnecessary traffic that leads to poor network performance or even a complete loss of network service. This flooding of packets is known as a traffic storm.

To prevent a traffic storm, you can disable the flooding of unknown unicast packets to all VLAN interfaces by configuring specific VLANs or all VLANs to forward all unknown unicast traffic traversing them to a specific interface. You can configure multiple VLANs to forward unknown unicast packets to the same interface or configure different interfaces for different VLANs. This channels the unknown unicast traffic traversing VLANs to specific interfaces instead of flooding all interfaces.

NOTE: The unknown-unicast-forwarding feature is not supported on QFX10000 Series platforms.

Verifying That Unknown Unicast Packets Are Forwarded to a Single Interface

IN THIS SECTION

Purpose

Verify that a VLAN is forwarding all unknown unicast packets (those with unknown destination MAC addresses) to a single interface instead of flooding unknown unicast packets across all interfaces that are members of that VLAN.

NOTE: This procedure uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. For ELS details see: Using the Enhanced Layer 2 Software CLI.

Action

(EX4300 Switches) Display the forwarding interface for unknown unicast packets for a VLAN (here, the VLAN name is v1):

```
user@switch> show configuration switch-options

unknown-unicast-forwarding {
    vlan v1 {
        interface ge-0/0/7.0;
    }
}
```

(EX9200 Switches) Display the forwarding interface for unknown unicast packets:

```
user@switch> show forwarding-options

next-hop-group uuf-nhg {
    group-type layer-2;
    interface ge-0/0/7.0;
}
```

Meaning

The sample output from the `show` commands show that the unknown unicast forwarding interface for VLAN `v1`is interface `ge-0/0/7`.

Configuring Unknown Unicast Forwarding (ELS)

IN THIS SECTION

NOTE: This task uses Junos OS for EX Series switches or QFX Series with support for the Enhanced Layer 2 Software (ELS) configuration style. For ELS details, see Using the Enhanced Layer 2 Software CLI

The unknown-unicast-forwarding feature is not supported on QFX10000 Series platforms.

Unknown unicast traffic consists of packets with unknown destination MAC addresses. By default, the switch floods these packets that traverse a VLAN to all interfaces associated with that VLAN. This flooding of packets is known as a traffic storm and can negatively impact network performance.

To prevent flooding unknown unicast traffic across the switch, configure unknown unicast forwarding to direct all unknown unicast packets within a VLAN to a specific interface. You can configure each VLAN to divert unknown unicast traffic to a different interface or use the same interface for multiple VLANs.

Configuring Unknown Unicast Forwarding on EX4300 Switches

To configure unknown unicast forwarding options on EX4300 switches:

- Configure unknown unicast forwarding for a specific VLAN and specify the interface to which all unknown unicast traffic will be forwarded:

  ```
  [edit switch-options]
  user@switch# set unknown-unicast-forwarding vlan vlan-name interface interface-name
  ```

- Configure unknown unicast forwarding for all VLANs and specify the interface to which all unknown unicast traffic will be forwarded:

  ```
  [edit switch-options]
  user@switch# set unknown-unicast-forwarding vlan all interface interface-name
  ```

Configuring Unknown Unicast Forwarding on EX9200 Switches

To configure unknown unicast forwarding on EX9200 switches, you must configure a flood filter and apply it to VLANs for which you want to configure unknown unicast forwarding. Flood filters are firewall filters that are applied only to broadcast, unknown unicast, and multicast (BUM) traffic. If a flood filter is configured, only traffic packets that are of the packet type `unknown-unicast` are forwarded to the interface on which unicast forwarding is configured. A next-hop group redirects the packets according to the action specified in the flood filter.

To configure the next-hop group that receives Layer 2 packets and then configure the interface to which these packets are forwarded:

1. Configure the `next-hop-group` action for the Layer 2 interface expected to receive unknown unicast packets:

   ```
   [edit forwarding-options]
   user@switch# set next-hop-group next-hop-group-name group-type layer-2
   [edit forwarding-options]
   user@switch# set next-hop-group next-hop-group-name interface interface-name
   ```

   For example:

   ```
   [edit forwarding-options]
   user@switch# set next-hop-group uuf-nhg group-type layer-2
   ```

```
[edit forwarding-options]
user@switch# set next-hop-group uuf-nhg interface ge-3/1/7.0
```

2. Configure a firewall filter with family address type `ethernet-switching`:

```
[edit firewall]
user@switch# set family ethernet-switching filter filter-name
```

For example:

```
[edit firewall]
user@switch# set family ethernet-switching filter uuf_filter
```

3. Configure a term in the firewall filter for the interface that receives unknown unicast packets (the interface specified in Step 1) to discard unknown unicast packets:

```
[edit firewall family ethernet-switching filter filter-name]
user@switch# set term term-name from interface interface-name
user@switch# set term term-name from traffic-type unknown-unicast
user@switch# set term term-name then discard
```

For example:

```
[edit firewall family ethernet-switching filter uuf_filter]
user@switch# set term source-drop from interface ge-3/1/7.0
user@switch# set term source-drop from traffic-type unknown-unicast
user@switch# set term source-drop then discard
```

4. Configure a term in the firewall filter for unknown unicast packets to be flooded to the interface enabled for unknown unicast forwarding by using `next-hop-group` (in step 1):

```
[edit firewall family ethernet-switching filter filter-name]

user@switch# set term term-name from traffic-type unknown-unicast
user@switch# set term term-name then next-hop-group group-name
```

For example:

```
[edit firewall family ethernet-switching filter uuf_filter]

user@switch# set term uuf-flood from traffic-type unknown-unicast
user@switch# set term uuf-flood then next-hop-group uuf-nhg
```

5. Configure a default term for the firewall filter to forward packets other than unknown unicast packets:

```
[edit firewall family ethernet-switching filter filter-name]

user@switch# set term term-name then accept
```

For example:

```
[edit firewall family ethernet-switching filter uuf_filter]

user@switch# set term fwd-default then accept
```

6. Apply the filter as a flood filter on the VLAN that includes the interface which will receive unknown unicast packets:

```
[edit vlans vlan-name]

user@switch# set forwarding-options flood input filter-name
```

For example:

```
[edit vlans v1]

user@switch# set forwarding-options flood input uuf_filter
```

Verifying That Unknown Unicast Packets Are Forwarded to a Trunk Interface

IN THIS SECTION

Purpose

Verify that a VLAN is forwarding all unknown unicast packets (those with unknown destination MAC addresses) to a single trunk interface instead of flooding unknown unicast packets across all interfaces that are members of the same VLAN.

Action

Display the forwarding interface for unknown unicast packets for a VLAN (here, the VLAN name is **v1**):

```
user@switch> show configuration ethernet-switching-options

unknown-unicast-forwarding {
    vlan v1 {
        interface ge-0/0/7.0;
    }
}
```

Display the Ethernet switching table:

```
user@switch> show ethernet-switching table vlan v1
Ethernet-switching table: 3 unicast entries
  VLAN              MAC address       Type         Age Interfaces
  v1                *                 Flood          - All-members
  v1                00:01:09:00:00:00 Learn         24 ge-0/0/7.0
  v1                00:11:09:00:01:00 Learn         37 ge-0/0/3.0
```

Meaning

The sample output from the `show configuration ethernet-switching-options` command shows that the unknown unicast forwarding interface for VLAN **v1** is interface **ge-0/0/7**. The `show ethernet-switching table` command shows that an unknown unicast packet is received on interface **ge-0/0/3** with the destination MAC address (DMAC) **00:01:09:00:00:00** and the source MAC address (SMAC) of **00:11:09:00:01:00**. This shows that the SMAC of the packet is learned in the normal way (through the interface **ge-0/0/3.0**), while the DMAC is learned on interface **ge-0/0/7**.

Configuring Unknown Unicast Forwarding (CLI Procedure)

Unknown unicast traffic consists of packets with unknown destination MAC addresses. By default, the switch floods these packets to all interfaces associated with a VLAN. Forwarding such traffic to interfaces on the switch can create a security issue.

To prevent flooding unknown unicast traffic across the switch, configure unknown unicast forwarding to direct all unknown unicast packets within a VLAN out to a specific trunk interface. From there, the destination MAC address can be learned and added to the Ethernet switching table. You can configure each VLAN to divert unknown unicast traffic to different trunk interfaces or use one trunk interface for multiple VLANs.

> **NOTE**: For Junos OS for EX Series switches or QFX Series with support for the Enhanced Layer 2 Software (ELS) configuration style, see "Configuring Unknown Unicast Forwarding (ELS)" on page 736.
>
> The unknown-unicast-forwarding feature is not supported on QFX10000 Series platforms.

To configure unknown unicast forwarding options:

> **NOTE**: Before you can configure unknown unicast forwarding within a VLAN, you must first configure that VLAN.

1. Configure unknown unicast forwarding for a specific VLAN (here, the VLAN name is **employee**):

   ```
   [edit ethernet-switching-options]
   user@switch# set unknown-unicast-forwarding vlan employee
   ```

2. Specify the trunk interface to which all unknown unicast traffic will be forwarded:

```
[edit ethernet-switching-options]
user@switch# set unknown-unicast-forwarding vlan employee interface ge-0/0/3.0
```

Platform-Specific Configuring Unknown Unicast Forwarding Behavior

Table 34: Platform-Specific Behavior

Platform	Difference
Junos Evolved ACX7000 series	ACX7000 series routers do not support configuring unknown unicast forwarding.

RELATED DOCUMENTATION

Understanding Storm Control

Configuring Autorecovery for Port Security Events

12

PART

Storm Control

CHAPTER 27

Understanding and Using Storm Control

IN THIS CHAPTER

Understanding Storm Control

IN THIS SECTION

A traffic storm is generated when messages are broadcast on a network and each message prompts a receiving node to respond by broadcasting its own messages on the network. This, in turn, prompts further responses, creating a snowball effect. The LAN is suddenly flooded with packets, creating unnecessary traffic that leads to poor network performance or even a complete loss of network service. Storm control enables the switch to monitor traffic levels and to drop broadcast, multicast, and unknown unicast packets when a specified traffic level—called the *storm control level*—is exceeded, thus preventing packets from proliferating and degrading the LAN. As an alternative to having the switch drop packets, you can configure it to shut down interfaces or temporarily disable interfaces (see the `action-shutdown` statement or the `port-error-disable` statement) when the storm control level is exceeded.

To recognize a storm, you must be able to identify when traffic has reached an abnormal level. Suspect a storm when operations begin timing out and network response times slow down. Users might be unable to access expected services. Monitor the percentage of broadcast and unknown unicast traffic in the network when it is operating normally. This data can then be used as a benchmark to determine when traffic levels are too high. You can then configure storm control to set the level at which you want to drop broadcast and unknown unicast traffic.

⚠ **CAUTION**: The Junos OS allows you to configure a storm control value that exceeds the bandwidth of the interface. If you configure an interface this way, storm control does not drop broadcast or unknown unicast packets even if they consume all the available bandwidth.

Storm control is enabled by default on ELS platforms and disabled by default on non-ELS platforms. If storm control is enabled, the default level is 80 percent of the available bandwidth for ingress traffic. You can change the storm control level by configuring it as a specific bandwidth value.

You can customize the storm control level for a specific interface by explicitly configuring either *bandwidth* or *level* (but not both at the same time for the same interface).

- bandwidth level—Configures the storm control level as the bandwidth in kilobits per second of the applicable traffic streams on that interface.

- Bandwidth percentage—Configures the storm control level as a percentage of the available bandwidth used by the combined applicable traffic streams that are subject to storm control on that interface.

When you configure storm control bandwidth or storm control level on an aggregated Ethernet interface, the storm control level for each member of the aggregated Ethernet interface is set to that bandwidth or level. For example, if you configure a storm control bandwidth of 15,000 Kbps on **ae1**, and **ae1** has two members, **ge-0/0/0** and **ge-0/0/1**, each member has a storm control level of 15,000 Kbps. Thus, the storm control level on **ae1** allows a traffic rate of up to 30,000 Kbps of combined traffic streams. Traffic might include broadcast, multicast, and unknown unicast traffic, depending upon the configuration.

The sending and receiving of broadcast, multicast, and unicast packets are part of normal LAN operation, so to recognize a storm, you must be able to identify when traffic has reached a level that is abnormal for your LAN. Suspect a storm when operations begin timing out and network response times slow down. As more packets flood the LAN, network users might be unable to access servers or e-mail.

Monitor the level of broadcast, multicast, and unknown unicast traffic in the LAN when it is operating normally. Use this data as a benchmark to determine when traffic levels are too high. Then configure storm control to set the level at which you want to drop broadcast traffic, multicast traffic, unknown unicast traffic, or two or all three of those traffic types.

You can change the storm control level for a specific interface by configuring the bandwidth value or the storm control level for the combined traffic streams that are subject to storm control on that interface. The type of traffic stream (broadcast, unknown unicast, and multicast) that is included within the bandwidth or storm control level consideration depends on which types of traffic are enabled for storm control monitoring on that interface.

You can disable the storm control selectively for broadcast, multicast, or unknown unicast traffic, or any combination of traffic types. When disabling storm control for multicast traffic, you can specify the traffic to be either registered multicast or unregistered multicast. Registered multicast MAC addresses are multicast MAC addresses that are within the range 01-00-5E-00-00-00 through 01-00-5E-7F-FF-FF (multicast MAC addresses outside this range are called unregistered multicast addresses).

NOTE: On an FCoE-FC gateway, storm control must be disabled on all Ethernet interfaces that belong to an FCoE VLAN to prevent FCoE traffic from being dropped. Configuring storm control on an Ethernet interface that is included in an FCoE-FC gateway may have undesirable effects, including FCoE packet loss. After disabling storm control on all interfaces, enable storm control on any interfaces that are not part of an FCoE-FC gateway on which you want to use storm control. However, on an FCoE transit switch, you can enable storm control on interfaces that carry FCoE traffic.

NOTE: On QFX/EX Junos platforms, storm control does not take effect on flows that are directed to the routing engine.

- You can enable storm control selectively for multicast traffic on a specific interface or on all interfaces.

- On all switches—You can disable storm control selectively for either broadcast streams, or multicast streams, or for unknown unicast streams.

Please note the following caveats for storm control:

- On EX4300 switches, storm control triggers on registered multicast packets even if you have configured no-registered-multicast.

- On EX9200 switches, storm control is not enabled by default.

- On QFX3500 switches, when you configure storm control bandwidth, the value you configure is rounded off internally to the closest multiple of 64 Kbps, and the rounded-off value represents the bandwidth that is actually enforced. For example, if you configure a bandwidth limit of 150 Kbps, storm control enforces a bandwidth limit of 128 Kbps.

- On QFX5000 switches that run on Junos OS, the IPv4 multicast entries are classified as known entries if the route is installed in the multicast routing table in the hardware. If the destination IP address is not programmed in the hardware, the packets are classified as unknown in the hardware. Therefore, storm control works differently when no-registered-multicast knob is enabled.

- On a QFX10002 switch, if storm control is configured on a VLAN port associated with an IRB interface, unregistered multicast traffic is classified as registered multicast traffic if IGMP snooping is enabled. If IGMP snooping is disabled, the traffic is classified as unknown unicast traffic.

- On switches other than QFX10000 switches, storm control is applied in aggregate per port. That is, if you set a storm control level of 100 megabits and the sum of the broadcast, unknown unicast, and multicast traffic exceeds 100 megabits, storm control is initiated. On QFX10000 switches, each traffic stream is measured independently per port, and storm control is initiated only if one of the streams exceeds the storm control level. For example, if you set a storm control level of 100 megabits and the broadcast and unknown unicast streams on the port are each flowing at 80 mbps, storm control is not triggered. In this case, storm control is initiated only if one of the streams exceeds 100 mbps.

- Storm control is not enabled by default on Juniper Networks MX Series routers.

- Starting in Junos OS release 17.4R1 for MX Series routers, you can also configure storm control on logical systems.

- In implementations of storm control prior to Junos version 17.3, rate limiting ingress traffic on a given port was based on PE trap-registers wherein the ingress traffic was rate limited per traffic type. As an example, in earlier implementations on applying a storm-control profile for BUM traffic at say x %; traffic would be rate limited per stream: broadcast, unknown unicast, multicast traffic individually to x% of link bandwidth. This behavior is different from rest of Junos implementation for storm-control where the net or aggregate traffic is rate limited to x% instead of per traffic type (broadcast, unknown unicast and multicast traffic). The implementation for Junos version 17.3 and later is based on policer resource per PE chip instead of the trap-registers and is coherent with the storm-control behavior across different Junos platforms.

Platform Specific Storm Control Behavior

Use Feature Explorer to confirm platform and release support for traffic storm control behavior.

Use the following table to review platform-specific behavior for your platform:

Platform	Difference
ACX Series routers	On ACX Series routers, storm control is only applicable at the physical interface level. No event is logged when a traffic storm hits an ACX Series router. Also interfaces are not bound to any default profile. The default action is to drop the packets exceeding the configured bandwidth.

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.4R1	Starting in Junos OS release 17.4R1 for MX Series routers, you can also configure storm control on logical systems.

RELATED DOCUMENTATION

Enabling and Disabling Storm Control (ELS) | 752

action-shutdown

port-error-disable

storm-control

Enabling and Disabling Storm Control (non-ELS)

IN THIS SECTION

NOTE: If your switching device is an EX Series switch and runs Junos OS with support for the Enhanced Layer 2 Software (ELS) configuration style, see "Enabling and Disabling Storm Control (ELS)" on page 752.

The factory default configuration enables storm control on all EX Series switch interfaces, with the storm control level set to 80 percent of the combined applicable traffic streams, as follows:

- On EX2200, EX3200, EX3300, and EX4200 switches—Storm control does not apply to multicast traffic by default. The factory default configuration enables storm control on all interfaces at 80 percent of the available bandwidth used by the combined unknown unicast and broadcast traffic streams.

- On EX4500 and EX8200 switches—The factory default configuration enables storm control on all interfaces at 80 percent of the combined broadcast, multicast, and unknown unicast streams.

- On EX6200 switches—Storm control does not apply to multicast traffic by default. The factory default configuration enables storm control on all interfaces at 80 percent of the available bandwidth used by the combined unknown unicast and broadcast traffic streams. Storm control can be disabled for each type of traffic individually.

You can disable storm control for all the applicable types of traffic on all interfaces or on a specified interface, as follows:

- On all switches—You can selectively disable storm control for broadcast streams, multicast streams, or for unknown unicast streams.

- On EX8200 switches—You can additionally selectively disable storm control on registered multicast traffic, on unregistered multicast traffic, or on both types of multicast traffic.

- On EX6200 switches—You can selectively disable storm control for each type of traffic individually.

You can enable storm control for multicast traffic (both registered and unregistered) on all interfaces or on a specific interface. This applies to all switches.

This topic describes:

Disabling Storm Control on Broadcast Traffic

To disable storm control on broadcast traffic:

- For all interfaces:

```
[edit ethernet-switching-options]
user@switch# set storm-control interface all no-broadcast
```

- For an individual interface:

```
[edit ethernet-switching-options]
user@switch# set storm-control interface interface-name no-broadcast
```

Disabling Storm Control on All Multicast Traffic

To disable storm control on all multicast traffic:

- For all interfaces:

```
[edit ethernet-switching-options]
user@switch# set storm-control interface all no-multicast
```

- For an individual interface:

```
[edit ethernet-switching-options]
user@switch# set storm-control interface interface-name no-multicast
```

Disabling Storm Control on Registered Multicast Traffic (EX8200 Switches Only)

To disable storm control only on registered multicast traffic (on EX8200 switches only):

- For all interfaces:

```
[edit ethernet-switching-options]
user@switch# set storm-control interface all no-registered-multicast
```

- For an individual interface:

```
[edit ethernet-switching-options]
user@switch# set storm-control interface interface-name no-registered-multicast
```

Disabling Storm Control on Unregistered Multicast Traffic (EX8200 Switches Only)

To disable storm control only on unregistered multicast traffic (on EX8200 switches only):

- For all interfaces:

```
[edit ethernet-switching-options]
user@switch# set storm-control interface all no-unregistered-multicast
```

- For an individual interface:

```
[edit ethernet-switching-options]
user@switch# set storm-control interface interface-name no-unregistered-multicast
```

Disabling Storm Control on Unknown Unicast Traffic

To disable storm control on unknown unicast traffic:

- For all interfaces:

```
[edit ethernet-switching-options]
user@switch# set storm-control interface all no-unknown-unicast
```

- For an individual interface:

```
[edit ethernet-switching-options]
user@switch# set storm-control interface interface-name no-unknown-unicast
```

Enabling Storm Control on Multicast Traffic

To enable storm control on multicast traffic:

- For all interfaces:

```
[edit ethernet-switching-options]
user@switch# set storm-control interface all multicast
```

- For an individual interface:

```
[edit ethernet-switching-options]
user@switch# set storm-control interface interface-name multicast
```

RELATED DOCUMENTATION

Configuring Autorecovery for Port Security Events | 758

Understanding Storm Control | 744

Enabling and Disabling Storm Control (ELS)

IN THIS SECTION

NOTE: This task uses Junos OS with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switching device is an EX Series switch and runs software that does not support ELS, see "Understanding Storm Control" on page 744. If your switching device is an EX Series switch and runs software that does support ELS, see Using the Enhanced Layer 2 Software CLI.

On EX4300 switches, the factory default configuration enables storm control on all Layer 2 switch interfaces. The default storm control level is set to 80 percent of the available bandwidth used by the combined broadcast, multicast, and unknown unicast traffic streams.

Storm control is not enabled by default on EX9200 switches or MX Series routers.

You can customize the storm control level for a specific interface. You specify the storm control level as the traffic rate in kilobits per second (Kbps) of the combined traffic streams or as the percentage of available bandwidth used by the combined traffic streams.

You can selectively disable storm control for broadcast, multicast, or unknown unicast traffic on all interfaces or on a specified interface. You can additionally disable storm control on registered or unregistered multicast traffic.

In the tasks described in this topic, you use the `[edit interfaces` *`interface-name`* `unit 0 family ethernet-switching]` hierarchy level to bind the storm control profile for EX Series switches and the `[edit interfaces` *`interface-name`* `unit 0 family bridge]` hierarchy level to bind the storm control profile for MX Series routers. Starting in Junos OS release 17.4R1 for MX Series routers, you can also configure storm control on logical systems.

Configuring Storm Control

You can configure storm control for a specific interface. The storm control level can be customized by explicitly configuring either the bandwidth level or the bandwidth percentage.

- `bandwidth-level`—Configures the storm control level as the bandwidth in kilobits per second of the combined traffic streams.

- `bandwidth-percentage`—Configures the storm control level as a percentage of the available bandwidth used by the combined traffic streams.

You can also configure a limit for `burst-size`. The burst size extends the function of the bandwidth limit to allow for bursts of traffic that exceed the configured bandwidth.

To configure storm control:

1. Create a storm control profile and set the storm control level as the traffic rate in kilobits per second of the combined traffic streams:

   ```
   [edit forwarding-options]
   user@device# set storm-control-profiles profile-name all bandwidth-level kbps
   ```

   NOTE: The name of the storm control profile can contain no more than 127 characters.

2. Bind the storm control profile to a logical interface:

   - **For EX Series Switches (Enterprise Style Configuration Only):**

     ```
     [edit]
     user@switch# set interfaces interface-name unit 0 family ethernet-switching storm-control
     profile-name
     ```

   - **For MX Series routers:**

     - **Enterprise Style Configuration:**

       ```
       [edit]
       user@device# set interfaces interface-name unit 0 family bridge storm-control profile-
       name
       ```

- **Service Provider Style Configuration: Starting in Junos OS release 18.3R1, you can configure storm control in the Service Provider Style configuration on MX Series devices.**

  ```
  [edit]
  ```

  ```
  user@device# set interfaces interface-name  flexible-vlan-tagging
  user@device# set interfaces interface-name  encapsulation flexible-ethernet-services
  user@device# set interfaces interface-name  unit logical-unit number encapsulation
  vlan-bridge
  user@device# see interfaces interface-name  unit logical-interface family bridge storm
  control profile-name
  ```

Disabling Storm Control on Broadcast Traffic

To disable storm control on broadcast traffic:

1. Create a storm control profile with the storm control level set as the traffic rate in kilobits per second of the combined traffic streams, and exclude broadcast traffic:

   ```
   [edit forwarding-options]
   user@device# set storm-control-profiles profile-name all bandwidth-level kbps no-broadcast
   ```

2. Bind the storm control profile to a logical interface:

   For EX Series switches:

   ```
   [edit]
   user@switch# set interfaces interface-name unit 0 family ethernet-switching storm-control
   profile-name
   ```

   For MX Series routers:

   ```
   [edit]
   user@device# set interfaces interface-name unit 0 family bridge storm-control profile-name
   ```

Disabling Storm Control on All Multicast Traffic

To disable storm control on all multicast traffic:

1. Create a storm control profile with the storm control level set as the traffic rate in kilobits per second of the combined traffic streams, but exclude multicast traffic:

```
[edit forwarding-options]
user@device# set storm-control-profiles profile-name all bandwidth-level kbps no-multicast
```

2. Bind the storm control profile to a logical interface:

   For EX Series switches:

```
[edit]
user@switch# set interfaces interface-name unit 0 family ethernet-switching storm-control profile-name
```

   For MX Series routers:

```
[edit]
user@device# set interfaces interface-name unit 0 family bridge storm-control profile-name
```

Disabling Storm Control on Registered Multicast Traffic

To disable storm control on only registered multicast traffic:

1. Create a storm control profile with the storm control level set as the traffic rate in kilobits per second of the combined traffic streams, but exclude registered multicast traffic:

```
[edit forwarding-options]
user@device# set storm-control-profiles profile-name all bandwidth-level kbps no-registered-multicast
```

2. Bind the storm control profile to a logical interface:

   For EX Series switches:

```
[edit]
user@switch# set interfaces interface-name unit 0 family ethernet-switching storm-control profile-name
```

For MX Series routers:

```
[edit]
user@device# set interfaces interface-name unit 0 family bridge storm-control profile-name
```

Disabling Storm Control on Unregistered Multicast Traffic

To disable storm control on only unregistered multicast traffic:

1. Create a storm control profile with the storm control level set as the traffic rate in kilobits per second of the combined traffic streams, but exclude unregistered multicast traffic:

```
[edit forwarding-options]
user@device# set storm-control-profiles profile-name all bandwidth-level kbps no-unregistered-multicast
```

2. Bind the storm control profile to a logical interface:

For EX Series switches:

```
[edit]
user@switch# set interfaces interface-name unit 0 family ethernet-switching storm-control profile-name
```

For MX Series routers:

```
[edit]
user@device# set interfaces interface-name unit 0 family bridge storm-control profile-name
```

Disabling Storm Control on Unknown Unicast Traffic

To disable storm control on only unknown unicast traffic:

1. Create a storm control profile with the storm control level set as the traffic rate in kilobits per second of the combined traffic streams, but exclude unregistered multicast traffic:

```
[edit forwarding-options]
user@device# set storm-control-profiles profile-name all bandwidth-level kbps no-unknown-unicast
```

2. Bind the storm control profile to a logical interface:

For EX Series switches:

```
[edit]
user@switch# set interfaces interface-name unit 0 family ethernet-switching storm-control
profile-name
```

For MX Series routers:

```
[edit]
user@device# set interfaces interface-name unit 0 family bridge storm-control profile-name
```

Disabling Storm Control on Multiple Types of Traffic

To disable storm control on multiple types of traffic; for example, broadcast and multicast traffic:

1. Create a storm control profile with the storm control level set as the traffic rate in kilobits per second of the combined traffic streams but exclude broadcast and multicast traffic:

```
[edit forwarding-options]
user@device# set storm-control-profiles profile-name all bandwidth-level kbps no-broadcast no-
multicast
```

2. Bind the storm control profile to a logical interface:

   For EX Series switches:

```
[edit]
user@switch# set interfaces interface-name unit 0 family ethernet-switching storm-control
profile-name
```

   For MX Series routers:

```
[edit]
user@device# set interfaces interface-name unit 0 family bridge storm-control profile-name
```

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
18.3R1	**Starting in Junos OS release 18.3R1, you can configure storm control in the Service Provider Style configuration on MX Series devices.**
17.4R1	Starting in Junos OS release 17.4R1 for MX Series routers, you can also configure storm control on logical systems.

RELATED DOCUMENTATION

Example: Configuring Storm Control to Prevent Network Outages on EX Series Switches

Example: Using Storm Control to Prevent Network (MX Routers) | **768**

Understanding Storm Control | **744**

Configuring Autorecovery for Port Security Events

You can have the device automatically restore interfaces that have been disabled due to MAC limiting, MAC move limiting, or storm control conditions by configuring the `recovery-timeout` statement.

- EX and QFX Series:

```
[edit interfaces interface-name unit 0 family ethernet-switching]
user@switch# set recovery-timeout 60
```

- MX Series:

```
[edit interfaces interface-name unit 0 family bridge]
user@switch# set recovery-timeout 60
```

An interface may shut down or be disabled as a result of one of the following port-security or storm-control configurations:

- Storm control—The `storm-control` statement is configured with the `action-shutdown` statement, or the action `shutdown`, depending on the platform you are using.

- MAC limiting—(Not supported on MX Series routers) The `mac-limit` statement is configured with the `action-shutdown` statement, or the action `shutdown`, depending on the platform you are using.
- MAC move limiting—(Not supported on MX Series routers) The `mac-move-limit` statement is configured with the `action-shutdown` statement, or the action `shutdown`, depending on the platform you are using.

There is no default, so unless the statement is explicitly configured, you will need to manually restore the interfaces by running a clear command.

- For EX Series switches, run: `clear ethernet-switching recovery-timeout`
- For MX Series routers, run: `clear bridge recovery-timeout`

RELATED DOCUMENTATION

Using the Enhanced Layer 2 Software CLI
Configuring MAC Limiting (ELS)
Configuring MAC Move Limiting (ELS) | 408
Enabling and Disabling Storm Control (ELS) | 752

Example: Using Storm Control to Prevent Network Outages

IN THIS SECTION

Using storm control can prevent problems caused by broadcast storms. You can configure storm control to rate-limit broadcast traffic, multicast traffic (on some devices), and unknown unicast traffic at a specified level so that the switch drops packets when the specified traffic level is exceeded, thus preventing packets from proliferating and degrading the LAN. You can also have the device shut down or temporarily disable an interface when the storm control limit is exceeded.

A traffic storm occurs when broadcast packets prompt receiving devices to broadcast packets in response. This prompts further responses, creating a knock-on effect that results in a broadcast storm that floods the device with packets, and causing poor performance or even a complete loss of service by some clients

Storm control monitors the level of applicable incoming traffic and compares it with the level that you specify. If the combined level of the applicable traffic exceeds the specified level, the switch drops packets for the controlled traffic types. As an alternative to having the switch drop packets, you can configure storm control to shut down interfaces or temporarily disable interfaces (see the `action-shutdown` statement or the `recovery-timeout` statement) when the storm control level is exceeded.

- On ELS systems, storm control is enabled by default on all interfaces at a level of 80 percent of the available bandwidth.

- On non-ELS systems, storm control is disabled by default on all interfaces. If you enable storm control, the default level is 80 percent of the available bandwidth.

NOTE: If you configure storm control on an aggregated Ethernet interface, the storm-control level is applies to each member interface individually. For example, if the aggregated interface has two members and you configure a storm-control level of 20 kbps, Junos will not detect a storm if one or both of the member interfaces receives traffic at 15 kbps because in neither of these cases does an individual member receive traffic at a rate greater than the configured storm-control level. In this example, Junos detects a storm only if at least one member interface receives traffic at greater than 20 Kbps.

- On EX2200, EX3200, EX3300, and EX4200 switches—Storm control is not enabled for multicast traffic by default. The factory default configuration enables storm control on all interfaces at 80 percent of the available bandwidth used by the combined unknown unicast and broadcast traffic streams.

- On EX4500 and EX8200 switches—The factory default configuration enables storm control on all interfaces at 80 percent of the available bandwidth used by the combined broadcast, multicast, and unknown unicast traffic streams.

- On EX6200 switches—Storm control is not enabled for multicast traffic by default. The factory default configuration enables storm control on all interfaces at 80 percent of the available bandwidth used by the combined unknown unicast and broadcast traffic streams. Storm control can be disabled for each type of traffic individually.

Example: Using Storm Control to Prevent Network Outages (ELS)

IN THIS SECTION

This example uses a Junos OS release that supports the Enhanced Layer 2 Software (ELS) configuration style.

Requirements

This example uses the following hardware and software components:

- One QFX Series switch running Junos OS with ELS
- Junos OS Release 13.2 or later

Overview and Topology

The topology used in this example consists of one switch connected to various network devices. This example shows how to configure the storm control level on interface xe-0/0/0 by setting the level to a traffic rate of 15,000 Kbps, based on the traffic rate of the combined applicable traffic streams. If the combined traffic exceeds this level, the switch drops packets for the controlled traffic types to prevent a network outage.

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure storm control based on the traffic rate in kilobits per second of the combined traffic streams, copy the following command and paste it into the switch terminal window:

```
[edit]
set forwarding-options storm-control-profiles sc-profile all bandwidth-level 15000
set interfaces xe-0/0/0 unit 0 family ethernet-switching storm-control sc-profile
```

Step-by-Step Procedure

To configure storm control:

1. Configure a storm control profile, `sc-profile`, and specify the traffic rate in kilobits per second of the combined traffic streams:

   ```
   [edit]
   user@switch> set forwarding-options storm-control-profiles sc-profile all bandwidth-level
   15000
   ```

2. Bind the storm control profile, `sc`, to a logical interface:

   ```
   [edit]
   user@switch> set interfaces xe-0/0/0 unit 0 family ethernet-switching storm-control sc-profile
   ```

Results

Display the results of the configuration:

```
[edit forwarding-options]
user@switch> show storm-control-profiles sc-profile
all {
```

```
    bandwidth 15000;
}
```

```
[edit]
user@switch> show interfaces xe-0/0/0
unit 0 {
    family ethernet-switching {
        vlan {
            members default;
        }
        storm-control sc-profile;
    }
}
```

SEE ALSO

- Understanding Storm Control | 744
- Example: Using Storm Control to Prevent Network Outages | 759

Example: Using Storm Control to Prevent Network Outages (non-ELS)

IN THIS SECTION

This example uses a Junos OS release that does not support the Enhanced Layer 2 Software (ELS) configuration style on a single EX Series switch. If your switch runs software that supports ELS, see "Example: Using Storm Control to Prevent Network Outages (ELS)" on page 761. For information about how to configure the switch to shut down or temporarily disable an interface when the storm control limit is exceeded, see "Example: Using Storm Control to Prevent Network Outages" on page 759

Requirements

This example uses the following hardware and software components:

- A switch
- Junos OS Release 11.1 or later

Overview and Topology

IN THIS SECTION

Topology

This example shows how to configure the storm control level on interface **xe-0/0/0** by setting the level to a traffic rate of 5000000 Kbps, based on the total of the combined broadcast and unknown unicast streams. If broadcast traffic and unknown unicast traffic exceed these levels, the switch drops packets for the controlled traffic types.

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To configure storm control for a 10-Gigabit Ethernet interface to the equivalent of 50 percent of the available bandwidth:

- Specify the level of allowed broadcast traffic and unknown unicast traffic on a specific interface:

```
[edit ethernet-switching-options]
user@switch# set storm-control interface xe-0/0/0 bandwidth 5000000
```

Results

Display the results of the configuration:

```
[edit ethernet-switching-options]
user@switch# show storm-control
interface xe-0/0/0 {
    bandwidth 5000000;
}
```

Verification

IN THIS SECTION

Verifying That the Storm Control Configuration Is in Effect

Purpose

Confirm that storm control is limiting the rate of traffic on the interface.

Action

Use the `show interfaces ge-0/0/0 detail` or `show interfaces ge-0/0/0 extensive` operational mode command to view traffic statistics on the storm controlled interface. The input rate (bps) must not exceed the storm control limit.

```
user@switch> show interfaces ge-0/0/0 extensive
Physical interface: ge-0/0/0, Enabled, Physical link is Up
```

```
Interface index: 160, SNMP ifIndex: 503, Generation: 163
Link-level type: Ethernet, MTU: 1514, Speed: Auto, Duplex: Auto,
BPDU Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
Remote fault: Online
Device flags   : Present Running
Interface flags: SNMP-Traps Internal: 0x0
Link flags     : None
CoS queues     : 8 supported, 8 maximum usable queues
Hold-times     : Up 0 ms, Down 0 ms
Current address: b0:c6:9a:67:90:84, Hardware address: b0:c6:9a:67:90:84
Last flapped   : 2013-05-16 22:46:42 UTC (14w3d 03:13 ago)
Statistics last cleared: Never
Traffic statistics:            5000000
 Input  bytes  :            312742788                  512 bps
 Output bytes  :            245552919                    0 bps
 Input  packets:              3550009                    1 pps
 Output packets:              2622101                    0 pps
 IPv6 transit statistics:
  Input  bytes  :                   0
  Output bytes  :                   0
  Input  packets:                   0
  Output packets:                   0
Dropped traffic statistics due to STP State:
 Input  bytes  :                    0
 Output bytes  :                    0
 Input  packets:                    0
 Output packets:                    0
Input errors:
  Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Policed discards: 0,
  L3 incompletes: 0, L2 channel errors: 0, L2 mismatch timeouts: 0,
  FIFO errors: 0, Resource errors: 0
Output errors:
  Carrier transitions: 0, Errors: 0, Drops: 0, Collisions: 0, Aged packets:
  FIFO errors: 0, HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
Egress queues: 8 supported, 4 in use
Queue counters:       Queued packets  Transmitted packets      Dropped packets
  0 best-effort                    0                    1                    0
  1 assured-forw                   0                    0                    0
  5 expedited-fo                   0                    0                    0
  7 network-cont                   0              2622100                    0
Queue number:         Mapped forwarding classes
  0                   best-effort
```

```
    1                   assured-forwarding
    5                   expedited-forwarding
    7                   network-control
  Active alarms  : None
  Active defects : None
  MAC statistics:                      Receive         Transmit
    Total octets                             0                0
    Total packets                            0                0
    Unicast packets                          0                0
    Broadcast packets                        0                0
    Multicast packets                        0                0
    CRC/Align errors                         0                0
    FIFO errors                              0                0
    MAC control frames                       0                0
    MAC pause frames                         0                0
    Oversized frames                         0
    Jabber frames                            0
    Fragment frames                          0
    VLAN tagged frames                       0
    Code violations                          0
  Autonegotiation information:
    Negotiation status: Incomplete
  Packet Forwarding Engine configuration:
    Destination slot: 0
  Interface transmit statistics: Disabled
```

Meaning

The traffic statistics `input bytes field` shows the ingress traffic rate at 512 bits per second (bps). This rate is within the storm control limit of 5000000 Kbps.

SEE ALSO

Understanding Storm Control | 744

Enabling and Disabling Storm Control (non-ELS) | 748

action-shutdown

interface (Storm Control)

Example: Using Storm Control to Prevent Network (MX Routers)

IN THIS SECTION

This example shows how to configure storm control on an pair of MX Series routers running Junos OS with Enhanced Layer 2 Software (ELS).

Requirements

This example uses the following hardware and software components:

- Two MX Series routers
- Junos OS Release 14.1 or later with ELS
- A traffic generator that can send broadcast and unknown unicast traffic at a rate that exceeds 100 Kbps
- A second host

Overview and Topology

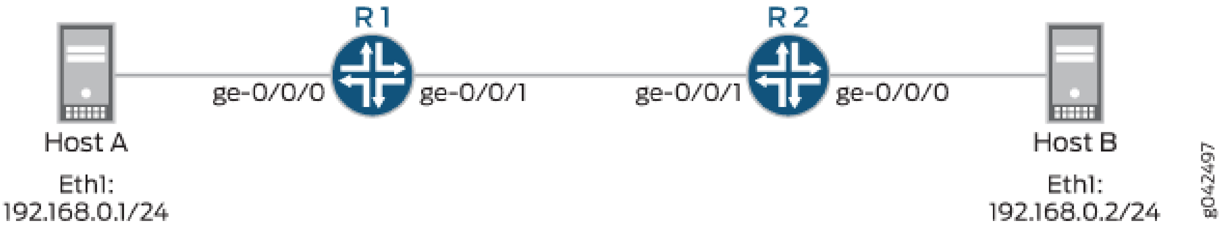
IN THIS SECTION

On MX Series routers, storm control is not enabled by default.

Topology

This example shows how to configure the storm control level on interface ge-0/0/1 by setting the level to a traffic rate of 100 Kbps. The topology used consists of two routers that could be connected to various network devices. If the combined traffic exceeds this level, the router drops packets for the controlled traffic types to prevent a network outage. (Starting in Junos OS release 17.4R1 for MX Series routers, you can also configure storm control on logical systems.)

Figure 44: Example Storm Control to Prevent Network Outages



Configuration

IN THIS SECTION

This example excludes multicast traffic from the storm traffic. Many protocols use multicast for control traffic, and for that reason network administrators and operators may want to keep multicast working to avoid obstructing protocol operation.

Procedure

CLI Quick Configuration

To quickly configure storm control based on the traffic rate in Kbps of the combined traffic streams, copy the following commands and paste them into the terminal window. The configurations of routers R1 and R2 are exactly the same:

```
set interfaces ge-0/0/0 unit 0 family bridge interface-mode access
set interfaces ge-0/0/0 unit 0 family bridge vlan-id 15
set interfaces ge-0/0/1 vlan-tagging
set interfaces ge-0/0/1 unit 0 family bridge interface-mode trunk
set interfaces ge-0/0/1 unit 0 family bridge vlan-id-list 15
set interfaces ge-0/0/1 unit 0 family bridge storm-control sc
set interfaces ge-0/0/1 unit 0 family bridge recovery-timeout 120
set bridge-domains bd1 domain-type bridge vlan-id 15
set forwarding-options storm-control-profiles sc all bandwidth-level 100 no multicast
set forwarding-options storm-control-profiles sc action-shutdown
```

Step-by-Step Procedure

To configure storm control:

1. Configure a storm control profile, `sc`, and specify the traffic rate in Kbps of the combined traffic streams. Exclude multicast traffic from the storm control profile.

   ```
   [edit]
   user@host# set forwarding-options storm-control-profiles sc all bandwidth-level 100 no-
   multicast
   user@host# set forwarding-options storm-control-profiles sc action-shutdown
   ```

2. Bind the storm control profile `sc` to a logical interface. Remember to do this for both interfaces between the routers.

   ```
   [edit]
   user@host#set interfaces ge-0/0/1 unit 0 family bridge storm-control sc
   ```

3. Configure interface ge-0/0/1 (the interface between routers). Do this for both interfaces between the routers.

   ```
   [edit]
   user@host# set interfaces ge-0/0/1 vlan-tagging
   user@host#set interfaces ge-0/0/1 unit 0 family bridge interface-mode trunk
   user@host#set interfaces ge-0/0/1 unit 0 family bridge vlan-id-list 15
   user@host#set interfaces ge-0/0/1 unit 0 family bridge recovery-timeout 120
   ```

4. Configure interface ge-0/0/0 (the interface from host to router). Remember to do this for both interfaces between the routers.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family bridge interface-mode access
user@host# set interfaces ge-0/0/0 unit 0 family bridge vlan-id 15
```

5. Set the bridge domain domain type and VLAN ID.

```
[edit]
user@host# set bridge-domains bd1 domain-type bridge vlan-id 15
```

Results

Display the results of the configuration:

```
[edit forwarding-options]
user@router> show storm-control-profiles sc
all {
    bandwidth-level 100;
    no-multicast;
}
action-shutdown;
```

```
[edit]
user@router> show interfaces ge-0/0/0
unit 0 {
    family bridge {
        interface-mode access;
        vlan-id 15;
    }
}
```

```
[edit]
user@router> show interfaces ge-0/0/1
vlan-tagging;
unit 0 {
    family bridge {
        interface-mode trunk;
        vlan-id-list 15;
        storm-control sc;
        recovery-timeout 120;
    }
}
```

```
[edit]
user@router> show bridge-domains bd1
domain-type bridge;
vlan-id 15;
```

Verification

IN THIS SECTION

Verifying That the Storm Control Configuration Is in Effect

Purpose

Confirm that storm control is limiting the rate of traffic on the interface.

Action

1. From Host A to Host B, use a traffic generator to send broadcast and unknown unicast traffic at a rate that exceeds 100 Kbps.

2. Verify on device R1’s ge-0/0/0 interface that traffic is entering at a rate that exceeds 100 Kbps.

```
user@R1# run show interfaces detail ge-0/0/0
Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 137, SNMP ifIndex: 513, Generation: 140
  Link-level type: Ethernet-Bridge, MTU: 1514, MRU: 1522, Speed: 1000mbps,
  BPDU Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled
  Pad to minimum frame size: Disabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x20004000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:05:86:71:6a:00, Hardware address: 00:05:86:71:6a:00
  Last flapped   : 2014-05-20 14:43:25 PDT (1w1d 01:20 ago)
  Statistics last cleared: 2014-05-28 15:59:39 PDT (00:04:02 ago)
  Traffic statistics:
   Input  bytes  :              830088               180432 bps
   Output bytes  :              0                    0 bps
   Input  packets:              8472                 230 pps
   Output packets:              0                    0 pps
   IPv6 transit statistics:
    Input  bytes  :                  0
    Output bytes  :                  0
    Input  packets:                  0
    Output packets:                  0
  Active alarms  : None
  Active defects : None
  Interface transmit statistics: Disabled
```

The Input bytes field shows the ingress traffic rate in bytes per second (bps). The input rate is within the storm control limit of 100 Kbps.

3. Verify that interface ge-0/0/1 on R1 is down (Admin down).

```
user@R1# run show interfaces ge-0/0/1.0 terse
Interface               Admin Link   Proto    Local                 Remote
ge-0/0/1.0              down    up   bridge
```

   Because the link remains up, control traffic continues to flow.

4. After the timeout period of 120 seconds (2 minutes), verify that the interface comes back up.

```
user@R1# run show interfaces ge-0/0/1.0 terse
Interface               Admin Link Proto    Local                 Remote
ge-0/0/1.0              up    up   bridge
```

SEE ALSO

Enabling and Disabling Storm Control (ELS) | **752**

Configuring Autorecovery for Port Security Events | **758**

Change History Table

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

Release	Description
17.4R1	(Starting in Junos OS release 17.4R1 for MX Series routers, you can also configure storm control on logical systems.)

13

PART

Malware Protection

CHAPTER 28

Juniper Malware Removal Tool

IN THIS CHAPTER

Juniper Malware Removal Tool

The Juniper Malware Removal Tool (JMRT) is a utility to scan for and remove malware running on Juniper Networks devices. JMRT is packaged with Junos OS and Junos OS Evolved by default. It is similar to antivirus software for desktop computers, except that JMRT runs on a Juniper device.

JMRT identifies malware present on the Routing Engine by using signature-based detection, but it can detect only malware with known signatures. If your device encounters new or unknown malware, JMRT might not be able to detect it.

You can use JMRT to perform two types of scans— quick scan and integrity check. You can also use the tool to spawn fake malware on your device and then run a test scan for it. Test scans show how JMRT behaves when it detects malware.

Quick Scan

JMRT scans the processes running on the system for malware. If it cannot find the executable for a process, it scans the process memory instead. By default, it scans all processes and, if malware is detected, it stops the processes and deletes the malware files. You can run this scan to quickly identify and remove malicious processes from your system. Alternatively, you can choose to scan a specific set of processes. You can also choose whether you want malware to be deleted or whether you simply want to be notified of it without any deletion.

Integrity Check

JMRT checks whether the system has the integrity mechanisms enabled. These mechanisms prevent arbitrary executable files that are not signed by Juniper from running. To enforce integrity mechanisms, Junos OS uses Verified Exec (Veriexec) and Junos OS Evolved uses Integrity Measurement Architecture (IMA).

RELATED DOCUMENTATION

request system malware scan

How to use the Juniper Malware Removal Tool

SUMMARY

You can use the Juniper Malware Removal Tool (JMRT) to scan for and remove malware running on Juniper Networks devices. You can run two types of scans— quick scan and integrity check. You can also run test scans that check for fake malware. Use Feature Explorer to confirm platform and release support for specific features.

IN THIS SECTION

Run a Quick Scan

You can use JMRT to run a quick scan to check for and remove malware on your system.

- To run a scan on all the processes currently running on the system, use the `request system malware-scan quick-scan` command.

  JMRT identifies processes and files containing malware and deletes them. Ideally, your device is free of malicious files and processes, and JMRT does not identify any process as potential malware, as seen in the following example:

  ```
  user@host> request system malware-scan quick-scan
  Found potential malware: No
  ```

  If JMRT identifies a file or process as potential malware, it displays the process ID and location of the malware and then deletes it.

  For example:

  ```
  user@host> request system malware-scan quick-scan
  Found potential malware: Yes
  ```

```
Scan Results:
Rule: tsb.auction-file/noclient
   pid:  95417
   file: /tmp/hidden/ssh
```

- To scan specific processes, use the `pids` option with `quick-scan` to specify the processes that need to be scanned.

  This method is faster than a general scan because JMRT does not scan every single process that is running on the system.

  In the following example, JMRT scans only processes with process IDs (PIDs) 42 and 97.

```
user@host> request system malware-scan quick-scan pids [ 42 97 ]
Found potential malware: No
```

- Use the `clean-action` option to indicate the action to take if malware is identified.

  The default is `clean`, which removes malicious files and processes. The `warn` action informs the user about malware but does not remove it.

  In this example, JMRT scans process 26329 and notifies the user if it is malware but does not delete the process.

```
user@host> request system malware-scan quick-scan pids 26329 clean-action warn
Found potential malware: No
```

  In this example, JMRT scans process 26315 and deletes it if it is malware.

```
user@host> request system malware-scan quick-scan pids 26315 clean-action clean
Found potential malware: No
```

Run an Integrity Check

You can use JMRT to check whether integrity mechanisms are enabled and working properly.

- Run the `request system malware-scan integrity-check` command.
  For example:

```
user@host> request system malware-scan integrity-check
Integrity is enforced: Yes
```

NOTE: From Junos OS Release 19.2 through Release 21.3, `integrity-check` was called `veriexec-check`. We changed the command name in Junos OS Release 21.4 to reflect that different integrity mechanisms might be used on different platforms (for instance, Junos OS uses Veriexec, whereas Junos OS Evolved uses Integrity Measurement Architecture, or IMA).

Run a Test Scan

Using JMRT, you can run fake malware processes on the system and use them for testing purposes. These processes are not actually malicious, but you can use them to observe how JMRT behaves when it identifies malware.

The test commands are available by default in Junos OS Evolved. To use these commands in Junos OS, you must install the optional `jmrt-test` package.

NOTE: Use the following commands to install the `jmrt-test` package:

- For Junos OS Release 20.1R1 or later:

  `request system software add optional://jmrt-test`

- For Junos OS releases before Release 20.1R1 (with 64-bit Routing Engine):

  `request system software add optional://jmrt-test-x86-64.tgz`

- For Junos OS releases before Release 20.1R1 (with 32-bit Routing Engine):

  `request system software add optional://jmrt-test-x86-32.tgz`

1. (Optional) Use JMRT to create a fake malware process.

```
user@host> request system malware-scan run-fake-malware
Fake malware PID: 25855
```

2. (Optional) View a list of the process IDs of all the fake malware that are currently running on the system.

```
user@host> request system malware-scan list-fake-malware
Example malware PIDs:
 25855
 25857
```

3. Run a test scan for fake malware by using the `test` option with the `quick-scan` statement.

The following example runs a test scan on processes 25855 and 25857, which are fake malware processes that were created earlier.

```
user@host> request system malware-scan quick-scan test pids [ 25855 25857 ]
Scan Results:
 Rule: test-malware/fake-jmrt-malware
 pid: 25855
 file: /packages/mnt/jmrt-test-x86-6464-74a7b298/opt/jmrt/example/fake-jmrt-malware
 Rule: test-malware/fake-jmrt-malware
 pid: 25857
 file: /packages/mnt/jmrt-test-x86-6464-74a7b298/opt/jmrt/example/fake-jmrt-malware
```

NOTE: You must use the `test` option because normal scans do not check for fake malware.

14

PART

Configuration Statements and Operational Commands

Security Services Configuration Statements

The following table lists the security services configuration statements available at the `[edit security]` hierarchy level:

Table 35: Security Services Configuration Statements

A-C	D-G	H-M	N-R	S-Z
algorithm (Junos FIPS)	*description (IKE policy)*	*identity*	*path-length*	*security-association (Junos OS)*
authentication (Security IPsec)	*dh-group*	*ike*	*perfect-forward-secrecy (Security)*	*security-association (Junos-FIPS Software)*
authentication-algorithm (Security IKE)	*direction (Junos OS)*	*internal*	*pki*	*spi (Junos OS)*
authentication-algorithm (Security IPsec)	*direction (Junos-FIPS Software)*	*ipsec (Security)*	*policy (Security IKE)*	*spi (Junos-FIPS Software)*
authentication-key-chains	*dynamic*	key (Authentication Keychain)	*policy (Security IPsec)*	*ssh-known-hosts*
authentication-method	*encoding*	*key (Junos FIPS)*	*pre-shared-key (Security)*	*traceoptions (Security)*
auto-re-enrollment	*encryption (Junos OS)*	key-chain (Authentication Keychain)	*proposal (Security IKE)*	*url*
auxiliary-spi	*encryption (Junos-FIPS Software)*	*ldap-url*	*proposal (Security IPsec)*	*validity-period*

Table 35: Security Services Configuration Statements *(Continued)*

A-C	D-G	H-M	N-R	S-Z
ca-identity	*encryption-algorithm*	*lifetime-seconds (Security)*	*proposals*	
ca-name	*enrollment*	*local*	*protocol (Junos OS)*	
ca-profile	*enrollment-retry*	*local-certificate (Security)*	*protocol (Junos-FIPS Software)*	
cache-size	*enrollment-url*	*local-key-pair*	*re-enroll-trigger-time-percentage*	
cache-timeout-negative	*file*	*manual (Junos OS)*	*re-generate-keypair*	
certificate-id		*manual (Junos-FIPS Software)*	*refresh-interval*	
certificates		*maximum-certificates*	*retry (Adaptive Services Interface)*	
certification-authority		*mode (IKE)*	*retry-interval*	
challenge-password		*mode (IPsec)*	*revocation-check*	
crl (Adaptive Services Interface)				
crl (Encryption Interface)				

RELATED DOCUMENTATION

[edit security] Hierarchy Level

Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Read this guide to learn about the syntax and options that make up the statements and commands. Also understand the contexts in which you'll use these CLI elements in your network configurations and operations.

- Junos CLI Reference

Click the links to access Junos OS and Junos OS Evolved configuration statement and command summary topics.

- Configuration Statements
- Operational Commands