Engineering
Simplicity

Junos® OS

Transport and Internet Protocols User Guide

Published
2025-12-12

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 Transport and Internet Protocols User 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

About This Guide

Use this guide to configure the common transport and Internet protocol options on your device running Junos OS or Junos OS Evolved.

1
CHAPTER

Understanding IP Support on Junos OS

IN THIS CHAPTER

Junos OS Support for IPv4, IPv6, and MPLS Routing Protocols

Junos OS implements full IP routing functionality, providing support for IP version 4 and IP version 6 (IPv4 and IPv6, respectively). The routing protocols are fully interoperable with existing IP routing protocols, and they have been developed to provide the scale and control necessary for the Internet core.

Junos OS supports the following unicast routing protocols:

- BGP—Border Gateway Protocol version 4 is an *EGP* that guarantees loop-free exchange of routing information between routing domains (also called autonomous systems). BGP, in conjunction with Junos OS routing policies, provides a system of administrative checks and balances that can be used to implement peering and transit agreements.

- ICMP—Internet Control Message Protocol router discovery enables hosts to discover the addresses of operational routers on the subnet.

- IS-IS—Intermediate System to Intermediate System is a link-state *IGP* for IP networks that uses the *SPF* algorithm, which also is referred to as the *Dijkstra* algorithm, to determine routes. The Junos OS supports a new and complete implementation of the protocol, addressing issues of scale, convergence, and resilience.

- OSPF—Open Shortest Path First is an IGP that was developed for IP networks by the Internet Engineering Task Force (*IETF*). OSPF is a link-state protocol that makes routing decisions based on the *SPF* algorithm.

  OSPF Version 2 supports IPv4. OSPF Version 3 supports IPv6. The fundamental mechanisms of OSPF such as flooding, designated router (*DR*) election, area-based topologies, and the *SPF* calculations remain unchanged in OSPF Version 3. Some differences exist either because of changes in protocol semantics between IPv4 and IPv6, or because of the need to handle the increased address size of IPv6.

- RIP—Routing Information Protocol version 2 is a distance-vector IGP for IP networks based on the *Bellman-Ford* algorithm. RIP dynamically routes packets between a subscriber and a service provider without the subscriber having to configure BGP or to participate in the service provider's *IGP* discovery process.

Junos OS also provides the following routing and Multiprotocol Label Switching (*MPLS*) applications protocols:

- *Unicast* routing protocols:

  - BGP

  - ICMP
  - IS-IS
  - OSPF Version 2
  - RIP Version 2
- Multicast routing protocols:
  - DVMRP—Distance Vector Multicast Routing Protocol is a *dense-mode* (*flood-and-prune*) multicast routing protocol.
  - IGMP—Internet Group Management Protocol versions 1 and 2 are used to manage membership in multicast groups.
  - MSDP—Multicast Source Discovery Protocol enables multiple Protocol Independent Multicast (*PIM*) *sparse mode* domains to be joined. A rendezvous point (*RP*) in a PIM sparse mode domain has a peer relationship with an RP in another domain, enabling it to discover multicast sources from other domains.
  - PIM sparse mode and dense mode—Protocol-Independent Multicast is a multicast routing protocol. PIM sparse mode routes to multicast groups that might span wide-area and interdomain internets. PIM dense mode is a flood-and-prune protocol.
  - SAP/SDP—Session Announcement Protocol and Session Description Protocol handle conference session announcements.
- MPLS applications protocols:
  - LDP—The Label Distribution Protocol provides a mechanism for distributing labels in non-traffic-engineered applications. LDP enables routers to establish label-switched paths (LSPs) through a network by mapping network layer routing information directly to data-link layer switched paths. LSPs created by LDP can also traverse LSPs created by the Resource Reservation Protocol (*RSVP*).
  - MPLS—Multiprotocol Label Switching, formerly known as tag switching, enables you to manually or dynamically configure LSPs through a network. It lets you direct traffic through particular paths rather than rely on the IGP least-cost algorithm to choose a path.
  - RSVP—The Resource Reservation Protocol version 1 provides a mechanism for engineering network traffic patterns that is independent of the shortest path decided upon by a routing protocol. RSVP itself is not a routing protocol; it operates with current and future unicast and multicast routing protocols. The primary purpose of RSVP is to support dynamic signaling for MPLS LSPs.

RELATED DOCUMENTATION

Junos OS Overview

2

CHAPTER

Configure Transport and Internet Protocol Features

IN THIS CHAPTER

ARP Learning and Aging Options

IN THIS SECTION

Address Resolution Protocol (ARP) is a protocol used by IPv4 and IPv6 to map IP network addresses to MAC addresses. Use this topic to set passive ARP learning and ARP aging options for network devices. In these situations, a switch operates as a virtual router.

Configuring Passive ARP Learning for Backup VRRP Devices

By default, the backup Virtual Router Redundancy Protocol (VRRP) device drops ARP requests for the VRRP-IP to VRRP-MAC address translation. The backup device does not learn the ARP (IP-to-MAC address) mappings for the hosts sending the requests. When it detects a failure of the primary device and becomes the new primary, the backup device must learn all the entries that were present in the ARP cache of the primary device. In environments with many directly attached hosts, such as metro Ethernet environments for a router, the backup device may have to learn a large number of ARP entries. This can cause a significant transition delay, during which traffic transmitted to some of the hosts might be dropped.

Passive ARP learning enables the ARP cache in the backup device to hold approximately the same contents as the ARP cache in the primary device. When a backup device becomes the primary device, the new primary device will already know the entries in the ARP cache of what used to be the primary device, reducing the transition delay.

To enable passive ARP learning, include the passive-learning statement at the [edit system arp] hierarchy level:

```
[edit system arp]
passive-learning;
```

While a device is operating as the primary, the passive learning configuration has no operational impact. The primary (or a standalone) device always learns ARP entries from incoming requests. The configuration takes effect only when the device is operating as a backup device.

We recommend setting passive learning on both the backup and primary VRRP device. Otherwise, you will need to remember to configure passive learning on a primary device after it becomes a backup device.

Configuring a Delay in Gratuitous ARP Requests

By default, the Junos OS sends gratuitous ARP requests immediately after you make network-related configuration changes on an interface, like a VLAN ID, MAC address, or IP address change. It also sends gratuitous ARP requests if a failover occurs and the device becomes the new primary device.

The Packet Forwarding Engine may drop some initial request packets if the IP address configuration updates have not been fully processed by the time a gratuitous ARP request is sent. To avoid dropping request packets, you can configure a delay in gratuitous ARP requests.

To configure a delay in gratuitous ARP requests, include the gratuitous-arp-delay *seconds* statement at the [edit system arp] hierarchy level:

```
[edit system arp]
gratuitous-arp-delay seconds;
```

We recommend that you configure a value in the range of 3 through 6 seconds.

Sending a Gratuitous ARP Request When an Interface is Online

To configure the device to automatically send a gratuitous ARP request when an interface is online, include the `gratuitous-arp-on-ifup` statement at the `[edit system arp]` hierarchy level:

```
[edit system arp]
gratuitous-arp-on-ifup;
```

Purging ARP Entries

To configure a device to purge obsolete ARP entries in the cache when an interface goes offline, include the `purging` statement at the `[edit system arp]` hierarchy level:

```
[edit system arp]
purging;
```

Purging is configured to delete ARP entries immediately after an interface that has gone offline is detected. If purging is not configured, ARP entries in the ARP table are retried after they have expired and are deleted if there is no ARP response within the default timeout value of 20 minutes. The default timeout value can be changed to other values using the `aging-timer` statement, as explained below.

Adjusting the ARP Aging Timer

By default, the ARP aging timer is set at 20 minutes. In environments with many directly attached hosts, such as metro Ethernet environments, increasing the amount of time between ARP updates by configuring the ARP aging timer can improve performance. Thousands of clients timing out at the same time might impact packet forwarding performance. In environments where there are devices connected with lower ARP aging timers (less than 20 minutes), decreasing the ARP aging timer can improve performance by preventing the flooding of traffic toward next hops with expired ARP entries. In most environments, the default ARP aging timer value does not need to be adjusted.

The range of the ARP aging timer is 1 through 240 minutes. To configure a system-wide ARP aging timer, include the `aging-timer` statement at the `[edit system arp]` hierarchy level:

```
[edit system arp]
aging-timer minutes;
```

You can also configure the ARP aging timer for each logical interface of family type `inet`. To configure the ARP aging timer at the logical interface level, specify the `aging-timer` statement and the timer value in minutes at the `[edit system arp interfaces interface-name]` hierarchy level:

```
[edit system arp interfaces interface-name]
aging-timer minutes;
```

To configure the ARP aging timer for a specific interface in a logical system, include the `aging-timer` statement and the timer value in minutes at the `[edit logical-systems logical-system-name system arp interfaces interface-name]` hierarchy level:

```
[edit logical-systems logical-system-name system arp interfaces interface-name]
aging-timer minutes;
```

NOTE: If the aging timer value is configured both at the system and the logical interface levels, the value configured at the logical interface level takes precedence for the specific logical interface.

The timer value you configure takes effect as ARP entries expire. Each refreshed ARP entry receives the new timer value. The new timer value does not apply to ARP entries that exist at the time you commit the configuration.

Disabling Neighbor Discovery

You can prevent the device from learning the MAC addresses of its neighbors through ARP or neighbor discovery for IPv4 and IPv6 neighbors. To disable ARP address learning by not sending ARP requests and not learning from ARP replies, use the `no-neighbor-learn` configuration statement.

To disable neighbor discovery for IPv4 neighbors:

```
[edit interfaces interface-name unit interface-unit-number family inet]
no-neighbor-learn;
```

To disable neighbor discovery for IPv6 neighbors:

```
[edit interfaces interface-name unit interface-unit-number family inet6]
no-neighbor-learn;
```

Configure ARP Cache Protection

Sending IP packets on a multiaccess network requires mapping from an IP address to a media access control (MAC) address (the physical or hardware address). In an Ethernet environment, ARP is used to map a MAC address to an IP address. Hosts that use ARP maintain a cache of discovered Internet-to-Ethernet address mappings to minimize the number of ARP broadcast messages.

To keep the cache from growing too large, by default, an entry is removed from the cache if it is not used within a certain period of time. In addition to this, you can manage the number of ARP cache entries by configuring a limit on the resolved and unresolved next-hop entries.

The ARP cache feature supports two types of limits:

- Count—Count limit is the maximum number of next hops that can be created in the ARP cache.
- Hold—Hold limit is the maximum number of hold routes pointing to a particular interface that can be retained before getting added to the ARP cache.

The default limit differs by platform. For the default cache limit on your platform, see arp-system-cache-limit. You can modify the default limit by configuring the ARP next-hop cache protection feature.

The ARP cache limits are executed at two levels:

- Local—Local limits are configured per interface and are defined for resolved and unresolved entries in the ARP cache.
- Global—Global limits apply system-wide. A global limit is further defined separately for the public interfaces and management interfaces (for example, fxp0). The management interface has a single global limit and no local limit. The global limit enforces a system-wide cap on entries for the ARP cache, including private Internal routing interfaces (IRIs) for internal routing instances (for example, em0 and em1).

- To configure the ARP cache count limit for resolved and unresolved next-hop entries globally, include the `arp-system-cache-limit` statement at the `[edit system]` hierarchy level.
- To configure the ARP cache count limit for resolved and unresolved next-hop entries locally, include the `arp-system-cache-limit` statement at the `[edit interfaces interface-name unit interface-unit-number family inet]` hierarchy level.
- To configure the ARP cache hold limit for unresolved next-hop entries locally, include the `arp-new-hold-limit` statement at the `[edit interfaces interface-name unit interface-unit-number family inet]` hierarchy level.

  NOTE: The ARP cache hold limit is configured on a per-interface basis only, and cannot be configured at the system level.

The ARP cache next-hop entries get allotted to different types of interfaces differently, irrespective of the ARP cache protection feature configuration.

1. By default, 200 entries get allotted to IRIs.
2. 80 percent of the remaining entries get allotted to public interfaces.
3. 20 percent of the remaining entries get allotted to management interfaces.

When the ARP next-hop entries exceed the configured count limit, new entries are either discarded, or kept under the hold counter, if a hold limit is configured for that interface. The ARP next-hop hold limit specifies the maximum number of hold entries or hold routes that point to a particular interface. When the number of hold entries exceeds the configured hold limit, the drop counter for that interface is affected drastically, as the new hold entries create a loop and continue to increment until there is bandwidth to accommodate them.

NOTE: After modifying the default ARP next-hop cache limit on an interface, the interface must be deactivated and reactivated for the newly configured values to take effect.

Example: Configuring ARP Cache Protection

IN THIS SECTION

You can configure an ARP cache limit for resolved and unresolved next-hop entries in the cache. This example shows how to configure ARP cache protection by specifying a maximum count and hold limit for resolved and unresolved next-hop entries in the ARP cache. This limit can be specified globally for all interfaces, or locally on a particular interface of the device. The benefit of configuring such a limit on the ARP cache is to protect the device from denial-of-service (DoS) attacks.

Requirements

This example uses the following hardware and software components:

- Two routers that can be a combination of MX Series routers.
- Two host devices connected to the routers.
- Junos OS Release running on the routers.

Overview

IN THIS SECTION

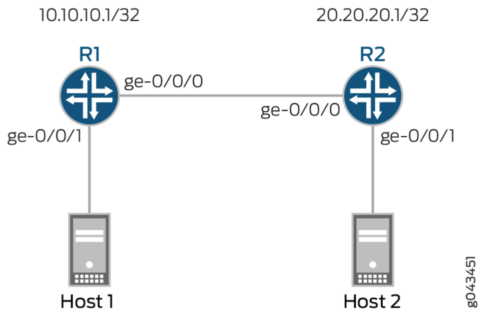
Sending IP packets on a multiaccess network requires mapping from an IP address to a media access control (MAC) address (the physical or hardware address). In an Ethernet environment, ARP is used to map a MAC address to an IP address. Hosts that use ARP maintain a cache of discovered Internet-to-Ethernet address mappings to minimize the number of ARP broadcast messages.

To keep the cache from growing too large, by default, an entry is removed from the cache if it is not used within a certain period of time. In addition to this, you can manage the number of ARP cache entries by configuring a limit on the resolved and unresolved next-hop entries.

Topology

Figure 1 on page 13 illustrates a simple two-router topology with ARP cache protection enabled. Routers R1 and R2 are each connected to hosts, Host1 and Host2, respectively.

Figure 1: ARP Cache Protection



The ARP cache next-hop entries get allotted to different types of interfaces differently, irrespective of the ARP cache protection feature configuration.

1. By default, 200 entries get allotted to IRIs.

2. 80 percent of the remaining entries get allotted to public interfaces.

3. 20 percent of the remaining entries get allotted to management interfaces.

For example, if Router R1 is configured with an `arp-system-cache-limit` of 220 globally, and it receives 230 ARP entries, on the first interface receiving the entries (say, ge-0/0/0), the following actions are performed:

1. When 230 entries are received, the global limit of 220 entries is applied to the system, where the configured limit is divided among the different types of interfaces, and the remaining entries received on a particular interface get discarded.

2. Out of the 220 cached entries, by default, 200 entries are allocated for IRI interfaces.

3. Out of the remaining 20 entries, 80 percent of the entries (16 entries) are sent to public interfaces and 20 percent of the entries (4 entries) are sent to the management interface. If the 230 ARP entries are received on the public interface, only the cache limit of 16 entries is retained, and the remaining 214 entries get discarded.

In addition, if ge-0/0/0 on Router R1 is configured with an `arp-new-hold-limit` value of 8, the following actions are performed:

1. Out of the 230 received entries, only 220 entries are cached in the ARP table. However, instead of discarding the remaining entries, the hold entries are sent to the hold counter of ge-0/0/0, and then the remaining entries are sent to the drop counter of ge-0/0/0.

2. Depending on availability of bandwidth, the eight hold entries are cached in the ARP table of ge-0/0/0 before taking any newly received entries into account.

3. The drop counter of ge-0/0/0, however, does not increment by single entries. The discarded hold entries in the drop counter form a loop and add to the entries count until there is bandwidth on the interface to accommodate all the entries. Therefore, additions to the drop counter have a drastic effect on the interface performance.

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, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter commit from configuration mode.

R1

```
set interfaces ge-0/0/0 unit 0 family inet address 192.0.2.1/24
set interfaces ge-0/0/0 unit 0 family inet arp-new-hold-limit 8
set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.1/24
set interfaces lo0 unit 0 family inet address 10.10.10.1/32
set system arp-system-cache-limit 220
```

R2

```
set interfaces ge-0/0/0 unit 0 family inet address 192.0.2.2/24
set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.1/24
set interfaces lo0 unit 0 family inet address 10.20.20.1/32
```

Procedure

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 Router R1 with ARP cache protection:

1. Configure the interfaces of Router R1.

```
[edit interfaces]
user@R1# set ge-0/0/0 unit 0 family inet address 192.0.2.1/24
user@R1# set ge-0/0/1 unit 0 family inet address 192.0.2.1/24
user@R1# set lo0 unit 0 family inet address 10.10.10.1/32
```

2. Configure ARP cache protection globally for all the interfaces of Router R1.

```
[edit system]
user@R1# set arp-system-cache-limit 220
```

3. Configure a hold limit on the ARP cache entries of interface ge-0/0/0 of Router R1.

```
[edit interfaces]
user@R1# set ge-0/0/0 unit 0 family inet arp-new-hold-limit 8
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show system` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 192.0.2.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.0.2.1/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.10.10.1/32;
        }
    }
}
```

```
user@R1# show system
arp-system-cache-limit 220 ;
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying Global ARP Next-Hop Cache Limit

Purpose

Verify the system-wide ARP next-hop cache limits and the allocation of next-hop entries for different interfaces.

Action

From operational mode, run the **show system statistics arp** command.

```
user@R1> show system statistics arp
arp:
         717253 datagrams received
         47 ARP requests received
         31 ARP replies received
         285 resolution request  received
         0 unrestricted proxy requests
         0 restricted proxy requests
         0 received proxy requests
         0 unrestricted proxy requests not proxied
        *****
         220 Max System ARP nh cache limit
         16 Max Public ARP nh cache limit
         200 Max IRI ARP nh cache limit
         4 Max Management intf ARP nh cache limit
         16 Current Public ARP next-hops present
         1 Current IRI ARP next-hops present
         2 Current Management ARP next-hops present
         2457 Total ARP next-hops creation failed as limit reached
         2454 Public ARP next-hops creation failed as public limit reached
         3 IRI ARP next-hops creation failed as iri limit reached
         0 Management ARP next-hops creation failed as mgt limit reached
```

Meaning

The global ARP next-hop cache limits are displayed in the output, along with the allocation of next-hop entries for IRI, public, and management interfaces.

Verifying Local ARP Next-Hop Cache Limit

Purpose

Verify the interface ARP next-hop cache limit.

Action

From operational mode, run the **show interfaces *interface-name*** command.

```
user@R1> show interface fxp0
fxp0
Physical interface: fxp0, Enabled, Physical link is Up
  Interface index: 1, SNMP ifIndex: 1
  Type: Ethernet, Link-level type: Ethernet, MTU: 1514, Speed: 100mbps
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  Link type      : Full-Duplex
  Current address: 00:a0:a5:62:8e:39, Hardware address: 00:a0:a5:62:8e:39
  Last flapped   : 2014-10-16 10:23:29 PDT (16:27:21 ago)
    Input packets : 0
    Output packets: 0

  Logical interface fxp0.0 (Index 3) (SNMP ifIndex 13)
    Flags: Up SNMP-Traps Encapsulation: ENET2
    Bandwidth: 0
    Input packets : 23
    Output packets: 4
    Protocol inet, MTU: 1500
    Max nh cache: 220 New hold nh limit: 8, Curr nh cnt: 2, Curr new hold cnt: 0, NH drop cnt: 0
    Flags: Sendbcast-pkt-to-re, Is-Primary
    Addresses, Flags: Is-Default Is-Preferred Is-Primary
      Destination: 10.209.0/18, Local: 10.209.3.69, Broadcast: 10.209.63.255
```

Meaning

The local ARP next-hop cache count and hold limits for the management interface is displayed in the output.

Troubleshooting

IN THIS SECTION

To troubleshoot the ARP cache protection configuration, see:

Troubleshooting System Log Messages

Problem

System log messages are generated to record events when the ARP cache limits are exceeded.

Solution

To interpret the system log messages, refer to the following:

- **Feb 08 17:12:39 [TRACE] [R1]: Public intf soft (80%) arp nh cache limit reached**—Router R1 has reached 80 percent of the allowed ARP next-hop cache limit for public interfaces.
- **Feb 08 17:07:43 [TRACE] [R1]: Public intf hard arp nh cache limit reached**—Router R1 has reached the maximum allowed limit for ARP next-hop cache entries on the public interface.
- **Feb 08 17:15:14 [TRACE] [R1]: Max cache soft (80%) arp nh cache limit for intf idx 325 reached**—Router R1 has reached 80 percent of the configured global ARP next-hop cache limit for all its interfaces.

- **Feb 08 17:19:41 [TRACE] [R1]: Max cache hard arp nh cache limit for intf idx 325 reached**—Router R1 has reached the maximum configured global ARP next-hop cache limit for all its interfaces.

RELATED DOCUMENTATION

arp-system-cache-limit

arp-new-hold-limit

ICMP Features

SUMMARY

Use Internet Control Message Protocol (ICMP) features to diagnose network issues and check device reachability.

IN THIS SECTION

Use Feature Explorer to confirm platform and release support for specific features.

Protocol Redirect Messages

IN THIS SECTION

ICMP redirect, also known as protocol redirect, is a mechanism used by switches and routers to convey routing information to hosts. Devices use protocol redirect messages to notify the hosts on the same data link of the best route available for a given destination.

Understanding Protocol Redirect Messages

Protocol redirect messages inform a host to update its routing information and to send packets on an alternate route. Suppose a host tries to send a data packet through a switch S1 and S1 sends the data packet to another switch, S2. Also, suppose that a direct path from the host to S2 is available (that is, the host and S2 are on the same Ethernet segment). S1 then sends a protocol redirect message to inform the host that the best route for the destination is the direct route to S2. The host should then send packets directly to S2 instead of sending them through S1. S2 still sends the original packet that it received from S1 to the intended destination.

Refer to RFC-1122 and RFC-4861 for more details on protocol redirecting.

NOTE:

- Switches do not send protocol redirect messages if the data packet contains routing information.
- All EX series switches support sending protocol redirect messages for both IPv4 and IPv6 traffic.

Disable Protocol Redirect Messages

By default, devices send protocol redirect messages for both IPv4 and IPv6 traffic. For security reasons, you may want to disable the device from sending protocol redirect messages.

To disable protocol redirect messages for the entire device, include the `no-redirects` or `no-redirects-ipv6` statement at the `[edit system]` hierarchy level.

- For IPv4 traffic:

```
[edit system]
user@host# set no-redirects
```

- For IPv6 traffic:

```
[edit system]
user@host# set no-redirects-ipv6
```

To re-enable the sending of redirect messages on the device, delete the `no-redirects` statement (for IPv4 traffic) or the `no-redirects-ipv6` statement (for IPv6 traffic) from the configuration.

To disable protocol redirect messages on a per-interface basis, include the `no-redirects` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family` *`family`*`]` hierarchy level.

- For IPv4 traffic:

```
[edit interfaces interface-name unit logical-unit-number]
user@host# set family inet no-redirects
```

- For IPv6 traffic:

```
[edit interfaces interface-name unit logical-unit-number]
user@host# set family inet6 no-redirects
```

Pings

IN THIS SECTION

Pings use ICMP. A successful ping is when a device sends an ICMP echo request to a target and the target responds with an ICMP echo reply. However, there might be situations where you do not want your device to respond to ping requests.

Disable the Routing Engine Response to Multicast Ping Packets

By default, the Routing Engine responds to ICMP echo requests sent to multicast group addresses. By configuring the Routing Engine to ignore multicast ping packets, you can prevent unauthorized persons from discovering the list of provider edge (PE) devices in the network.

To disable the Routing Engine from responding to these ICMP echo requests, include the `no-multicast-echo` statement at the `[edit system]` hierarchy level:

```
[edit system]
user@host# set no-multicast-echo
```

Disable Reporting IP Address and Timestamps in Ping Responses

When you issue the `ping` command with the `record-route` option, the Routing Engine displays the path of the ICMP echo request packets and the timestamps in the ICMP echo responses by default. By configuring the `no-ping-record-route` and `no-ping-timestamp` options, you can prevent unauthorized persons from discovering information about the provider edge (PE) device and its loopback address.

You can configure the Routing Engine to disable the setting of the `record-route` option in the IP header of the ping request packets. Disabling the `record-route` option prevents the Routing Engine from recording and displaying the path of the ICMP echo request packets in the response.

To configure the Routing Engine to disable the setting of the `record route` option, include the `no-ping-record-route` statement at the `[edit system]` hierarchy level:

```
[edit system]
user@host# set no-ping-record-route
```

To disable the reporting of timestamps in the ICMP echo responses, include the `no-ping-time-stamp` option at the `[edit system]` hierarchy level:

```
[edit system]
user@host# set no-ping-time-stamp
```

Source Quench Messages

When a device is receiving too many or undesired datagrams, it can send a source quench message to the originating device. The source quench message signals the originating device to reduce the amount of traffic it is sending.

By default, the device reacts to ICMP source quench messages. To ignore ICMP source quench messages, include the `no-source-quench` statement at the `[edit system internet-options]` hierarchy level:

```
[edit system internet-options]
no-source-quench;
```

To stop ignoring ICMP source quench messages, use the `source-quench` statement:

```
[edit system internet-options]
source-quench;
```

Time-to-Live (TTL) Expiration

The time-to-live (TTL) value in a packet header determines how long the packet remains traveling through the network. The TTL value decrements with each device (or hop) the packet travels through. When a device receives a packet with a TTL value of 0, it discards the packet. The TTL expiry message is sent using ICMP.

You can configure your device to use an IPv4 address as the source address for ICMP time-to-live (TTL) expiry error messages. This means you can configure the loopback address as the source address in response to ICMP error packets. Doing this is useful when you cannot use the device address for traceroute purposes because you have duplicate IPv4 addresses in your network.

The source address must be an IPv4 address. To specify the source address, use the `ttl-expired-source-address` *`source-address`* option at the `[edit system icmp (System)]` hierarchy level:

```
[edit system icmp]
user@host# set ttl-expired-source-address source-address
```

This configuration only applies to ICMP TTL expiry messages. Other ICMP error reply messages continue to use the address of the ingress interface as the source address.

Rate Limit ICMP Traffic

To limit the rate at which ICMPv4 or ICMPv6 messages can be generated by the Routing Engine and sent to the Routing Engine, include the appropriate rate limiting statement at the `[edit system internet-options]` hierarchy level.

- For IPv4:

```
[edit system internet-options]
icmpv4-rate-limit bucket-size bucket-size packet-rate packet-rate
```

- For IPv6:

```
[edit system internet-options]
icmpv6-rate-limit bucket-size bucket-size packet-rate packet-rate
```

Rate Limit ICMP Error Messages

IN THIS SECTION

By default, ICMP error messages for non-TTL-expired IPv4 and IPv6 packets are generated at the rate of 1 packet per second (pps). You can adjust this rate to a value that you decide provides sufficient information for your network without causing network congestion.

NOTE: For TTL-expired IPv4 or IPv6 packets, the rate for ICMP error messages is not configurable. It is fixed at 500 pps.

Why to Rate Limit ICMPv4 and ICMPv6 Error Messages

An example use case for adjusting the rate limit is a data center providing web services. Suppose this data center has many servers on the network that use jumbo frames with an MTU of 9100 bytes when they communicate to hosts over the Internet. These other hosts require an MTU of 1500 bytes. Unless maximum segment size (MSS) is enforced on both sides of the connection, a server might reply with a packet that is too large to be transmitted across the Internet without being fragmented when it reaches the edge router in the data center.

Because TCP/IP implementations often have Path MTU Discovery enabled by default with the `dont-fragment` bit set to 1, a transit device will drop a packet that is too big rather than fragmenting it. The device will return an ICMP error message indicating the destination was unreachable because the packet was too big. The message will also provide the MTU that is required where the error occurred. The sending host should adjust the sending MSS for that connection and resend the data in smaller packet sizes to avoid the fragmentation issue.

At high core interface speeds, the default rate limit of 1 pps for the error messages may not be enough to notify all the hosts when there are many hosts in the network that require this service. The consequence is that outbound packets are silently dropped. This action can trigger additional retransmissions or back-off behaviors, depending on the volume of requests that the data center edge router is handling on each core-facing interface.

In this situation, you can increase the rate limit to enable a higher volume of oversized packets to reach the sending hosts. (Adding more core-facing interfaces can also help resolve the problem.)

How to Rate Limit ICMPv4 and ICMPv6 Error Messages

Although you configure the rate limit at the `[edit chassis]` hierarchy level, it is not a chassis-wide limit. Instead, the rate limit applies per interface family. This means, for example, that multiple physical interfaces configured with `family inet` can simultaneously generate the ICMP error messages at the configured rate.

> **NOTE**: This rate limit takes effect only for traffic that lasts 10 seconds or longer. The rate limit is not applied to traffic with a shorter duration, such as 5 seconds or 9 seconds.

- To configure the rate limit for ICMPv4, use the `icmp` statement:

  ```
  [edit chassis]
  user@host# set icmp rate-limit rate-limit
  ```

  Starting in Junos OS Release 19.1R1, the maximum rate increased from 50 pps to 1000 pps.

- To configure the rate limit for ICMPv6, use the `icmp6` statement:

  ```
  [edit chassis]
  user@host# set icmp6 rate-limit rate-limit
  ```

You must also consider that the rate limit value can interact with your DDoS protection configuration. The default bandwidth value for exceptioned packets that exceed the MTU is 250 pps. DDoS protection flags a violation when the number of packets exceeds that value. If you set the rate limit higher than the current `mtu-exceeded` bandwidth value, then you must configure the bandwidth value to match the rate limit.

For example, suppose you set the ICMP rate limit to 300 pps:

```
user@host# set chassis icmp rate-limit 300
```

You must configure the DDoS protection `mtu-exceeded` `bandwidth` to match that value.

```
user@host# set system ddos-protection protocols exceptions mtu-exceeded bandwidth 300
```

IPv6 Features

IN THIS SECTION

Configure IPv6 Duplicate Address Detection Attempts

To set the number of attempts the device makes to detect IPv6 duplicate addresses, use the `ipv6-duplicate-addr-detection-transmits` statement at the `[edit system internet-options]` hierarchy level:

```
[edit system internet-options]
ipv6-duplicate-addr-detection-transmits;
```

Accept IPv6 Packets with a Zero Hop Limit

By default, incoming IPv6 packets that have a zero hop limit value in their header are rejected both when they are addressed to the local host and when they are transiting the device. To accept zero hop-limit packets addressed to the local host, include the `no-ipv6-reject-zero-hop-limit` statement at the `[edit system internet-options]` hierarchy level. Transit packets are still dropped.

```
[edit system internet-options]
no-ipv6-reject-zero-hop-limit;
```

To re-enable rejection of these packets, use the following statement:

```
[edit system internet-options]
ipv6-reject-zero-hop-limit;
```

Process IPv4-mapped IPv6 Addresses

By default, the Junos OS disables the processing of IPv4-mapped IPv6 packets to protect against malicious packets from entering the network. You may want to enable IPv4-mapped IPv6 packets:

- To ensure smooth packet flow in a mixed routing environment of IPv4 and IPv6 networks.
- So that IPv6 packets aren’t dropped in a pure IPv4 routing environment.
- When you are transitioning your routing environment from IPv4 to IPv6 networks.

To enable the processing of IPv4-mapped IPv6 packets, use the `allow-v4mapped-packets` statement:

```
[edit system]
allow-v4mapped-packets;
```

NOTE: We recommend that you configure this statement only after fully understanding the security implications of allowing IPv4-mapped IPv6 packets in your network.

Process 6PE Traceroutes

In a dual-stack IPv6 network connected over an IPv4 MPLS network, the P routers in the IPv4 MPLS backbone do not have an IPv6 family. Consequently, the transit P routers are not shown in the output when you do an IPv6 traceroute. To generate an ICMPv6 echo request and a TTL expired response packet to and from the intermediate transit routers in the 6PE network, use the `allow-6pe-traceroute` statement:

```
[edit system]
allow-6pe-traceroute;
```

RELATED DOCUMENTATION

Understanding IPv6

IPv6 Neighbor Discovery Overview

Path MTU Discovery | 30

Junos OS Support for IPv4, IPv6, and MPLS Routing Protocols

Path MTU Discovery

IN THIS SECTION

Learn how to configure path maximum transmission unit (MTU) discovery on different types of connections.

Path MTU Discovery Overview

The maximum transmission unit (MTU) size of a node is the largest packet the node can transmit. For a packet to successfully traverse the path from the source node to the destination node, the packet must be no larger than the smallest MTU size of all nodes on the path between the source and destination. This value is referred to as the path MTU. If a packet is larger than a link's MTU size, it is likely that the link will drop it. Path MTU discovery checks the MTU on a network path between two nodes.

NOTE: Path MTU discovery is enabled by default. We recommended leaving path MTU discovery enabled. Disabling path MTU discovery can have adverse affects on network stability.

Configure Path MTU Discovery on Outgoing TCP Connections

By default, path MTU discovery on outgoing TCP connections is enabled. To disable path MTU discovery for IPv4, include the no-path-mtu-discovery statement at the [edit system internet-options] hierarchy level:

```
[edit system internet-options]
no-path-mtu-discovery;
```

To reenable path MTU discovery on outgoing TCP connections, include the path-mtu-discovery statement at the [edit system internet-options] hierarchy level:

```
[edit system internet-options]
path-mtu-discovery;
```

To disable and reenable path MTU discovery for IPv6, use the ipv6-path-mtu-discovery and no-ipv6-path-mtu-discovery statements.

Configure IP-IP Path MTU Discovery on IP-IP Tunnel Connections

By default, MTU discovery on outgoing IP-IP tunnel connections is enabled.

To disable IP-IP path MTU discovery, include the no-ipip-path-mtu-discovery statement at the [edit system internet-options] hierarchy level:

```
[edit system internet-options]
no-ipip-path-mtu-discovery;
```

To reenable IP-IP path MTU discovery, include the ipip-path-mtu-discovery statement at the [edit system internet-options] hierarchy level:

```
[edit system internet-options]
ipip-path-mtu-discovery;
```

Configure Path MTU Discovery on Outgoing GRE Tunnel Connections

By default, path MTU discovery on outgoing GRE tunnel connections is enabled. To disable GRE path MTU discovery, include the `no-gre-path-mtu-discovery` statement at the `[edit system internet-options]` hierarchy level:

```
[edit system internet-options]
no-gre-path-mtu-discovery;
```

To re-enable GRE path MTU discovery, include the `gre-path-mtu-discovery` statement at the `[edit system internet-options]` hierarchy level:

```
[edit system internet-options]
gre-path-mtu-discovery;
```

To verify details of the path MTU on outgoing GRE tunnels, use the command `show interfaces (GRE)`.

RELATED DOCUMENTATION

ICMP Features | 20

TCP | 32

TCP

IN THIS SECTION

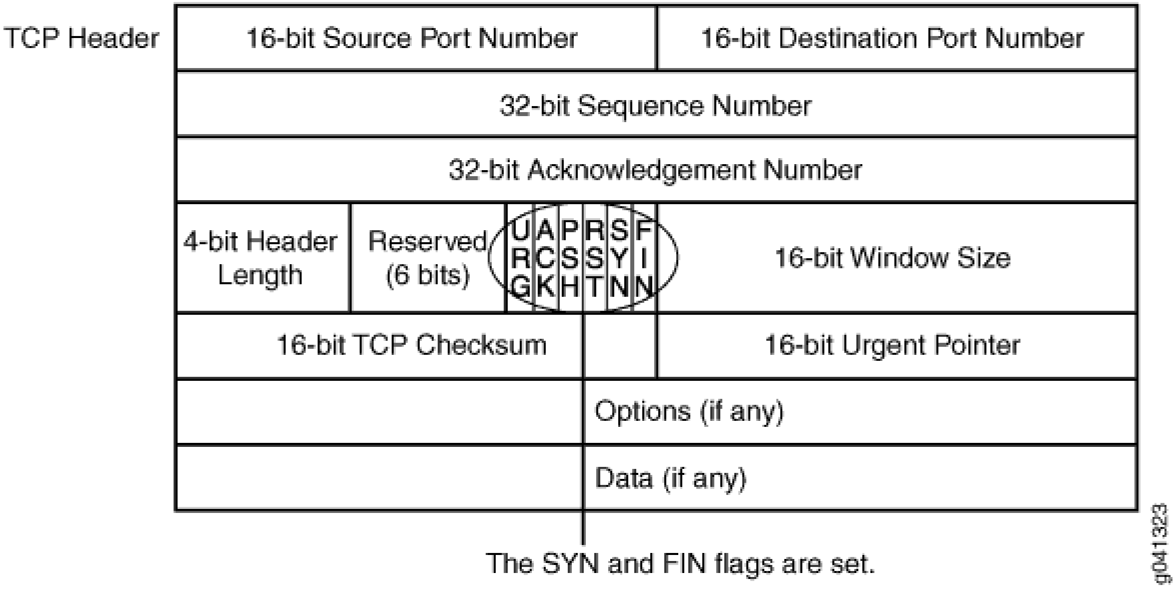
Many applications and services use TCP to communicate. Configure TCP options to improve link quality and security.

Security for TCP Headers with SYN and FIN Flags Set

By default, your device accepts packets that have both the SYN and FIN bits set in the TCP flag. Configure your device to drop packets with both the SYN and FIN bits set to reduce security vulnerabilities.

Both the SYN and FIN control flags are not normally set in the same TCP segment header. The SYN flag synchronizes sequence numbers to initiate a TCP connection. The FIN flag indicates the end of data transmission to finish a TCP connection. Their purposes are mutually exclusive. A TCP header with the SYN and FIN flags set is anomalous TCP behavior, causing various responses from the recipient, depending on the OS. See Figure 2 on page 33.

Figure 2: TCP Header with SYN and FIN Flags Set



An attacker can send a segment with both flags set to see what kind of system reply is returned and thereby determine what kind of OS is on the receiving end. The attacker can then use any known system vulnerabilities for further attacks. When you enable the `tcp-drop-synfin-set` statement, Junos OS checks if the SYN and FIN flags are set in TCP headers. If it discovers such a header, it drops the packet.

```
[edit system internet-options]
tcp-drop-synfin-set;
```

Disable TCP RFC 1323 Extensions

To disable RFC 1323 TCP extensions, include the `no-tcp-rfc1323` statement at the `[edit system internet-options]` hierarchy level:

```
[edit system internet-options]
no-tcp-rfc1323;
```

To disable the Protection Against Wrapped Sequence (PAWS) number extension (described in RFC 1323, *TCP Extensions for High Performance*), include the `no-tcp-rfc1323-paws` statement at the `[edit system internet-options]` hierarchy level:

```
[edit system internet-options]
no-tcp-rfc1323-paws;
```

Configure TCP MSS for Session Negotiation

IN THIS SECTION

During session connection establishment, two peers agree in negotiations to determine the IP segment size of packets that they will exchange during their communication. The TCP maximum segment size (MSS) value in TCP SYN packets specifies the maximum number of bytes that a TCP packet's data field, or segment, can contain. An MSS value that is set too high can result in an IP datagram that is too large to send and that must be fragmented. Fragmentation can incur additional overhead cost and packet loss.

To diminish the likelihood of fragmentation and to protect against packet loss, you can specify a lower TCP MSS value. This custom TCP MSS value applies to all IPv4 TCP SYN packets traversing all the router's ingress interfaces whose MSS value is higher than the one you specify. You cannot exempt particular ports from its effects.

The following sections describe how to configure a custom TCP MSS value.

Configure TCP MSS on MX Series Routers Using a Service Card

To specify a TCP MSS value on MX Series routers using a service card:

1. Configure the MSS value.

   Include the `tcp-mss` *`mss-value`* statement at the `[edit services service-set` *`service-set-name`*`]` hierarchy level. The range of the `tcp-mss` *`mss-value`* parameter is from 536 through 65535 bytes.

   ```
   [edit services service-set service-set-name]
   tcp-mss mss-value;
   ```

2. Add the service set to any interface for which you want to adjust the TCP-MSS value:

   ```
   [edit interfaces interface-name unit 0 family family service]
   input service-set service-set-name;
   output service-set service-set-name;
   ```

3. To view statistics of SYN packets received and SYN packets whose MSS value is modified, issue the `show services service-sets statistics tcp-mss` operational mode command.

Configure TCP MSS Inline on MX Series Routers Using MPC Line Cards

To specify a TCP MSS value on MX Series routers that use MPC line cards, specify the MSS value. Include the `tcp-mss` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family` *`family`*`]` hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family]
tcp-mss mss-value;
```

The range of the *mss-value* parameter is from 64 through 65,535 bytes. The TCP MSS value must be lower than the MTU of the interface.

This statement is supported on the following interfaces: gr- (GRE), ge- (Gigabit Ethernet), xe- (10-Gigabit Ethernet), and et- (40-Gigabit and 100-Gigabit Ethernet). Families supported are inet and inet6.

> **NOTE**: Configuring TCP MSS inline on MX Series routers using MPC line cards works only for traffic exiting/egressing the interface, not traffic entering/ingressing the interface.

Select a Fixed Source Address for Locally Generated TCP/IP Packets

Locally generated IP packets are the packets that are produced by applications running on the Routing Engine. Junos OS chooses a source address for these packets so that the application peers can respond. It also enables you to specify the source address on a per application basis. To serve this purpose, the Telnet CLI command contains the source-address argument.

This section introduces the default-address-selection statement:

```
[edit system]
default-address-selection;
```

If you specifically choose the source address, as in the case of Telnet, default-address-selection does not influence the source address selection. The source address becomes the one that is specified with the source-address argument (provided the address is a valid address specified on the interface of a router). If the source address is not specified or if the specified address is invalid, default-address-selection influences the default source address selection.

If the source address is not explicitly specified as in the case of Telnet, then by default (when default-address-selection is not specified) the source address chosen for locally generated IP packets is the IP address of the outgoing interface. This indicates that depending on the chosen outgoing interface, the source address might be different for different invocations of a given application.

If the interface is unnumbered (no IP address is specified on an interface), Junos OS uses a predictable algorithm to determine the default source address. If default-address-selection is specified, Junos OS uses the algorithm to choose the source address irrespective of whether the outgoing interface is numbered. This indicates that with default-address-selection, you can influence Junos OS to provide the same source address in locally generated IP packets regardless of the outgoing interface.

The behavior of source address selection by Junos OS can be summed up as shown in the following table:

Table 1: Source Address Selection

Outgoing Interface	When `default-address-selection` Is Specified	When `default-address-selection` Is Not Specified
Unnumbered	Use `default-address-selection`	Use `default-address-selection`
Numbered	Use `default-address-selection`	Use IP address of outgoing interface

See Configuring Default, Primary, and Preferred Addresses and Interfaces for more information about the default address source selection algorithm.

NOTE: For IP packets sent by IP routing protocols (including OSPF, RIP, RSVP, and the multicast protocols, but not including IS-IS), the local address selection is often constrained by the protocol specification so that the protocol operates correctly. When this constraint exists in the routing protocol, the packet's source address is unaffected by the presence of the `default-address-selection` statement in the configuration. For protocols in which the local address is unconstrained by the protocol specification like IBGP and multihop EBGP, if you do not configure a specific local address when configuring the protocol, the local address is chosen using the same method as other locally generated IP packets.

TCP Authentication

IN THIS SECTION

Enabling a TCP authentication method enhances the security and ensures the authenticity of TCP segments exchanged during BGP and LDP sessions. Junos devices support three main types of TCP authentication: TCP MD5, TCP Authentication Option (TCP-AO), and TCP keychain-based authentication. For more information about TCP-AO, see "TCP Authentication Option (TCP-AO)" on page 43.

NOTE: Although Junos devices support both the TCP-AO and TCP MD5 authentication methods, you cannot use both at the same time for a given connection.

IP Subnet Support

IN THIS SECTION

TCP-AO and TCP MD5 authentication support IP subnets for LDP and BGP sessions. When you configure TCP authentication with a network address and a prefix length, your chosen TCP authentication method authenticates TCP connections to the entire range of addresses under that subnet. This means you can authenticate TCP connections without needing to know the exact IP addresses of the destination devices.

When IP subnets overlap, the authentication method uses the longest prefix match (LPM) to determine the exact authentication key for a specific TCP session.

BGP

To configure prefix-based authentication for BGP sessions, include the allow (all | *prefix-list*) statement at either of the following hierarchies:

- [edit protocols bgp group *group-name*]
- [edit protocols bgp group *group-name* dynamic-neighbor *dyn-name*]

You can use IPV4 or IPV6 addresses for the subnet.

In this example, TCP MD5 authenticates TCP connections to devices in the 10.0.3.0/24 subnet for all BGP sessions:

```
[edit protocols]
bgp {
    group one {
        authentication-key "$ABC123";
        allow 10.0.3.0/24;
        dynamic-neighbor dyn_one {
            allow 10.0.3.0/24;
            authentication-key "$ABC123";
        }
    }
}
```

LDP

To configure prefix-based authentication for LDP, configure TCP authentication under the `session-group` *`ip-prefix`* hierarchy. You must use an IPv4 address.

In this example, LDP uses TCP-AO to authenticate any TCP connection with a device that has an address in the 10.0.0.0/24 subnet:

```
[edit protocols ldp]
session-group 10.0.0.0/24 {
        authentication-algorithm ao;
        authentication-key-chain tcpao;
}
```

For how to configure your TCP-AO keychain, see "TCP Authentication Option (TCP-AO)" on page 43.

VRF Support

IN THIS SECTION

TCP-AO and TCP MD5 authentication are VRF aware in BGP and LDP sessions. This means you can configure TCP-AO and TCP MD5 under non-default routing instances. The TCP authentication method you configure under a routing instance is only applied to the TCP sessions inside that VRF instance. If a TCP connection in a different VRF instance has the same destination IP address, the TCP authentication method does not get applied to that TCP connection if the VRF instance does not have TCP authentication configured for the peer.

Configure VRF-based TCP authentication as you normally would, but under a `routing-instances` hierarchy level. To use TCP MD5 authentication, include the `authentication-key` *authentication-key* statement. To use TCP-AO, include the following statements:

```
user@device# set authentication-algorithm ao
user@device# set authentication-key-chain keychain
```

For how to configure your TCP-AO keychain, see "TCP Authentication Option (TCP-AO)" on page 43.

You can combine VRF-aware configurations with IP subnets. This enables you to authenticate connections to a range of addresses inside the VRF instance.

BGP

Configure VRF-based TCP authentication for BGP sessions at any of the following hierarchy levels:

- [edit routing-instances *vrf-instance* protocols bgp]
- [edit routing-instances *vrf-instance* protocols bgp group *group-name*]
- [edit routing-instances *vrf-instance* protocols bgp group *group-name* neighbor *neighbor-ip*]
- [edit routing-instances *vrf-instance* protocols bgp group *group-name* dynamic-neighbor *dyn-name*]

If you configure VRF-based authentication at the `dynamic-neighbor` level, include the `allow` statement along with your chosen authentication method configuration. For example, to use TCP-AO with a dynamic neighbor:

```
[edit routing-instances vrf-instance protocols bgp group group-name dynamic-neighbor dyn-name]
user@device# set allow (all | prefix-list)
user@device# set authentication-algorithm ao
user@device# set authentication-key-chain keychain
```

In the following example, BGP uses TCP authentication to ensure the security of TCP connections in a VRF instance called `vrf-one`. In group one, BGP uses TCP MD5 to authenticate connections to the neighbor with the IP address 10.0.1.1. It uses TCP-AO to authenticate connections to the neighbor with the IP address 10.0.1.2.

In group two, BGP uses TCP-AO to authenticate connections to any device in the 10.0.0.0/24 subnet.

```
[edit routing-instances]
vrf-one {
    protocols {
        bgp {
            group one {
                peer-as 22;
                neighbor 10.0.1.1 {
                    authentication-key "$ABC123"; ## SECRET-DATA
                }
                neighbor 10.0.1.2 {
                    authentication-algorithm ao;
                    authentication-key-chain tcpao;
                }
            }
            group two {
                peer-as 22;
                dynamic-neighbor dyn_two {
                    allow 10.0.0.0/24;
                    authentication-algorithm ao;
                    authentication-key-chain tcpao;
                }
            }
        }
    }
}
```

LDP

Configure VRF-based authentication for LDP sessions at any of the following hierarchy levels:

- `[edit routing-instances` *`vrf-instance`* `protocols ldp]`
- `[edit routing-instances` *`vrf-instance`* `protocols ldp session` *`session-ip`*`]`
- `[edit routing-instances` *`vrf-instance`* `protocols ldp session-group` *`ip-prefix`*`]`

In this example, TCP-AO authenticates TCP connections in a VRF instance called `vrf-two`. It authenticates TCP connections to the address 10.0.1.1 as well as any address in the 10.0.0.0/24 subnet.

```
[edit routing-instances]
vrf-two {
    protocols {
        ldp {
            session 10.0.1.1 {
                authentication-algorithm ao;
                authentication-key-chain tcpao;
            }
            session-group 10.0.0.0/24 {
                authentication-algorithm ao;
                authentication-key-chain tcpao;
            }
        }
    }
}
```

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
22.4R1	Starting in Junos OS Evolved Release 22.4R1, you can configure TCP-AO or TCP MD5 authentication with an IP subnet to include the entire range of addresses under that subnet. Prior to Junos OS Evolved Release 22.4R1, Junos devices only permit you to use TCP authentication with a specific address. This means you can only authenticate TCP connections to remote peers with known IP addresses.
22.4R1	Starting in Junos OS Evolved Release 22.4R1, TCP authentication is VRF aware. In releases prior to Junos OS Evolved Release 22.4R1, TCP MD5 and TCP-AO ignore virtual routing and forwarding (VRF) instances. The device ignores TCP MD5 and TCP-AO configurations under non-default routing instances. When you configure TCP MD5 or TCP-AO under the default VRF instance, the device applies that authentication method to all TCP sessions that have destinations inside the IP address range for that VRF instance. If a TCP session belonged to non-default VRF instance but had the same destination IP address as the default VRF instance, TCP MD5 and TCP-AO would apply the same authentication key to two TCP connections with the same destination IP address.

RELATED DOCUMENTATION

Extend the Default Port Address Range | 76

Protocol Redirect Messages | 20

TCP Authentication Option (TCP-AO)

SUMMARY

Learn about TCP Authentication Option (TCP-AO) for BGP and LDP sessions.

IN THIS SECTION

TCP-AO for BGP and LDP Sessions

IN THIS SECTION

The BGP and LDP protocols use TCP for transport. TCP-AO is a new authentication method proposed through *RFC5925, The TCP Authentication Option* to enhance the security and authenticity of TCP segments exchanged during BGP and LDP sessions. It also supports both IPv4 and IPv6 traffic.

Benefits of TCP-AO

TCP-AO provides the following benefits over TCP MD5:

- **Stronger algorithms**—Supports multiple stronger authentication algorithms such as HMAC-SHA-1-96 and AES-128-CMAC-96 (mandated by *RFC5925, The TCP Authentication Option*). HMAC-

SHA-1-96 is a hash-based MAC and AES-128-CMAC-96 is a cipher-based MAC, thus making the message digest more complex and secure than the digest created by using the MD5 algorithm.

- **Two-Fold security**—In the TCP-AO method, the configured Authentication algorithm is used in two stages: Once to generate an internal traffic key from a user-configured key and then to generate a message digest using the generated traffic key, whereas in the TCP MD5 method, the MD5 algorithm generates a message digest using its user-configured key.

- **Better Key Management and Agility**—You can configure up to 64 keys for a session and you can add them at any time during the lifetime of a session. It provides a simple key coordination mechanism by giving the ability to change keys (move from one key to another) within the same connection without causing any TCP connection closure. Changing TCP MD5 keys during an established connection might cause a flap or restart in the connection.

- **Suitable for long-lived connections**—More suitable for long-lived connections for routing protocols such as BGP and LDP and across repeated instances of a single connection.

What is TCP-AO?

TCP-AO provides a framework to:

- Support multiple stronger algorithms, such as HMAC-SHA1 and AES-128 to create an internal traffic key and message digest.

- Add a new user-configured key to re-generate internal traffic keys for an established connection and a mechanism to synchronize key change between BGP or LDP peers.

In earlier releases, Junos devices only supported the TCP MD5 authentication method for BGP and LDP sessions. The MD5 method supports only the MD5 algorithm, which is less secure than TCP-AO. In addition, changing a MD5 key normally disrupts the TCP session, unlike TCP-AO. TCP MD5 is defined in *RFC2385, Protection of BGP Sessions via the TCP MD5 Signature Option*. For more information about TCP MD5, see "TCP" on page 32.

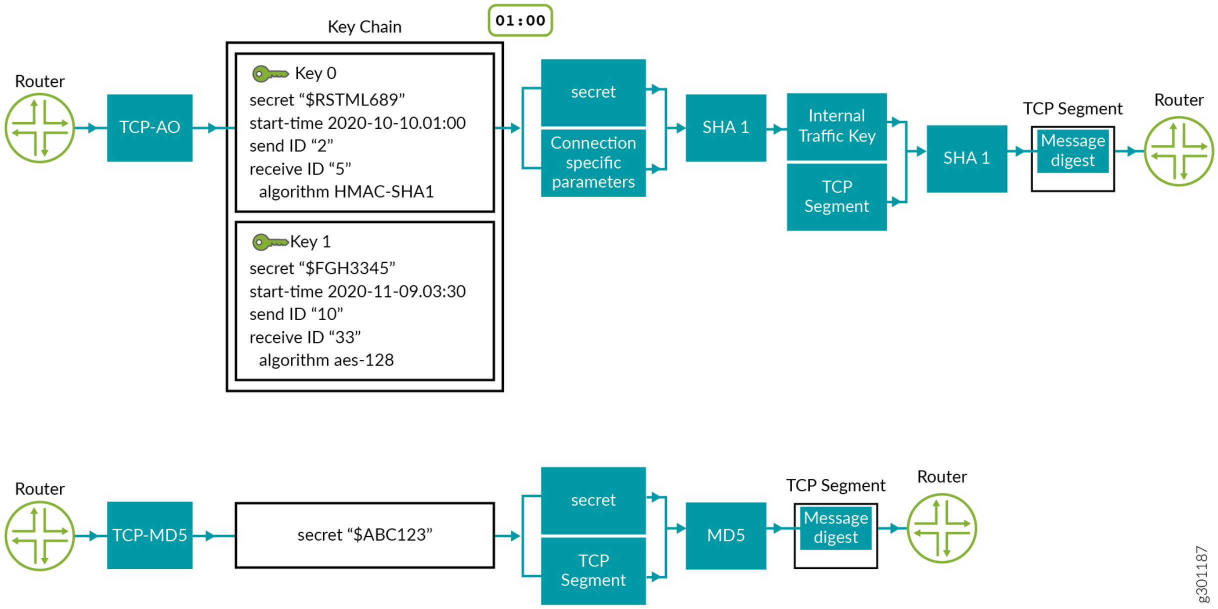
NOTE:

- While Junos devices support both the TCP-AO and TCP MD5 authentication methods, you cannot use both at the same time for a given connection.

- TCP-AO supports Nonstop Active Routing.

The following diagram explains the difference between TCP-AO and TCP MD5 authentication. The first flow shows the configuration and processing flow for TCP-AO and the second flow shows the configuration and processing flow for TCP-MD5.

Figure 3: TCP-AO in comparison with TCP MD5



Below is an explanation of the processing flows shown in Figure 1:

- **TCP-AO**—The user has configured two keys in the keychain (key 0 and key 1) with all required parameters. The keychain supports two algorithms: HMAC SHA1 and AES-128 (mandated per RFC5925). TCP fetches key 0, which is the key that is currently active, as shown by the timestamp in the figure. In the example, key 0 is configured with HMAC-SHA1.

  SHA1 takes the "secret" (from the key 0 configuration) and connection specific parameters for encryption and generates an internal traffic key.

  SHA1 again encrypts the internal traffic key and the TCP segment to generate the message digest. The digest is copied to the TCP-AO MAC field of the TCP-AO option in the TCP segment. The segment is then sent to the receiving device.

- **TCP-MD5**—The user has configured a single key because TCP MD5 option supports only one key for a connection. Further, it only supports the MD5 algorithm. The MD5 algorithm takes the "secret" from the key and the TCP segment for encryption and generates a message digest. This message digest is then copied to MD5 digest field in the TCP segment and is sent to the receiving device.

Configuration

First, configure a keychain. Then apply TCP-AO to the BGP or LDP session.

To configure a keychain for TCP-AO (with one key), configure the following statement at the [edit security] hierarchy level.

```
[edit security]
user@router# set authentication-key-chains key-chain key-chain key id secret secretpassword
start-time YYYY-MM-DD.HH:MM algorithm ao ao-attribute send-id send-id recv-id recv-id
cryptographic-algorithm cryptographic-algorithm tcp-ao-option enabled
```

To apply TCP-AO to a BGP session (with the configured keychain), configure the following statement at the [edit protocols] hierarchy level.

```
[edit protocols]
user@router# set bgp group group neighbor neighbor authentication-algorithm ao
user@router# set bgp group group neighbor neighbor authentication-key-chain key-chain
```

To apply TCP-AO to an LDP session (with the configured keychain), configure the following statement at the [edit protocols] hierarchy level.

```
[edit protocols]
user@router# set ldp session session authentication-algorithm ao
user@router# set ldp session session authentication-key-chain key-chain
```

Example: Configure a Keychain (TCP-AO)

SUMMARY

This example shows you how to create a TCP-AO keychain to authenticate a BGP or LDP session.

This example uses the following hardware and software components:

- MX Series or PTX Series routers.
- Junos OS Release 20.3R1 or later version.

This example shows you how to create a TCP-AO keychain to authenticate a BGP or LDP session.

In this example, you can create a keychain `new_auth_key` with two keys, `key 0` and `key 1` on devices R1 and R2.

1. To create a keychain `new_auth_key` with the first key, (`key 0`):

   NOTE: Copy the following commands, paste them into a text file, remove any line breaks and change any details necessary to match your network configuration, copy and paste the commands into the CLI.

   R1

   ```
   [edit security]
   user@R1# set authentication-key-chains key-chain new_auth_key key 0 secret secretpassword
   start-time 2020-10-10.03:00 algorithm ao ao-attribute send-id 3 recv-id 8 cryptographic-
   algorithm hmac-sha-1-96 tcp-ao-option enabled
   ```

   R2 (with `send-id` and `recv-id` values reversed)

   ```
   [edit security]
   user@R2# set authentication-key-chains key-chain new_auth_key key 0 secret secretpassword
   start-time 2020-10-10.03:00 algorithm ao ao-attribute send-id 8 recv-id 3 cryptographic-
   algorithm hmac-sha-1-96 tcp-ao-option enabled
   ```

   Consider the following parameters while configuring a keychain:

   Table 2: Keychain Parameters

Parameter	Description
`key-chain`	Enter a unique name.
`key`	Enter a unique key ID.
`secret`	Enter a unique password.
`start-time`	Enter a unique time in *YYYY-MM-DD.HH:MM* format to specify the start time of the key.

Table 2: Keychain Parameters *(Continued)*

Parameter	Description
algorithm	Enter algorithm ao.
send-id and recv-id	Enter any two numbers between 0 and 255. You must not use these numbers for any other key within that keychain.
cryptographic-algorithm	Choose either hmac-sha-1-96 or aes-128-cmac-96.
tcp-ao-option	Choose enabled to enable the TCP-AO option.

2. To add another key (key 1), after creating key 0:

   R1

```
[edit security authentication-key-chains key-chain new_auth_key]
user@R1# set key 1 secret password start-time 2020-11-11.04:00 algorithm ao ao-attribute send-
id 1 recv-id 2 cryptographic-algorithm aes-128-cmac-96 tcp-ao-option enabled
```

   R2 (with send-id and recv-id values reversed)

```
[edit security authentication-key-chains key-chain new_auth_key]
user@R2# set key 1 secret password start-time 2020-11-11.04:00 algorithm ao ao-attribute send-
id 2 recv-id 1 cryptographic-algorithm aes-128-cmac-96 tcp-ao-option enabled
```

3. Enter commit from configuration mode on both devices to activate your changes.
4. To verify the keychain new_auth_key with the 2 keys configured, use the show security authentication-key-chains command from configuration mode.

   The following is sample output based on this example:

```
user@R1# show security authentication-key-chains key-chain new_auth_key {
    key 0 {
        secret "$RSTML689"; ## SECRET-DATA
        start-time "2020-10-10.03:00:00 -0700";
        algorithm ao;
        ao-attribute {
            send-id 3;
            recv-id 8;
            tcp-ao-option enabled;
            cryptographic-algorithm hmac-sha-1-96;
        }
    }
    key 1 {
        secret "$FFGH3345"; ## SECRET-DATA
        start-time "2020-11-11.04:00:00 -0800";
        algorithm ao;
        ao-attribute {
            send-id 1;
            recv-id 2;
            tcp-ao-option enabled;
            cryptographic-algorithm aes-128-cmac-96;
        }
    }
}
```

You have successfully created a keychain!

To delete a keychain, use the `delete security authentication-key-chains key-chain` *`key-chain-name`* command from configuration mode.

NOTE:

- You can associate only one TCP-AO keychain with a BGP or LDP session during its life-time. You cannot point another keychain to the session in its life-time.
- We recommend a minimum interval of 30 minutes between the start-time of any two subsequent keys within a keychain.
- Once a keychain is configured and in use by a TCP connection, you cannot change the `send-id` or `recv-id` values of its active key. However, you can change the other parameters in the key, and any new connection associated with the updated keychain will take the updated parameters for its connection establishment.
- Starting in Junos OS Release 21.2R1, you can use the `tcpao-auth-mismatch allow-without-tcpao` to allow the connection establishment without TCP-AO if any one TCP endpoint does not have TCP-AO configured on it.

To display information about existing keychains (if any) from the operational mode, use the `show security keychain` command. Here is sample output:

```
user@R1> show security keychain
```

```
Keychain                 Active-ID        Next-ID       Transition  Tolerance
                       Send    Receive  Send    Receive
 new_auth_key            1       1        None    None     None        3600 (secs)
```

Example: Use TCP-AO to Authenticate a BGP Session

SUMMARY

This example shows you how to authenticate a BGP session using a TCP Authentication Option (TCP-AO) keychain.

IN THIS SECTION

Requirements

This example uses the following hardware and software components:

- MX Series or PTX Series routers.
- Junos OS Release 20.3R1 or later version.
- Configure a keychain `new_auth_key`. See "Configure a Keychain (TCP-AO)" on page 46.

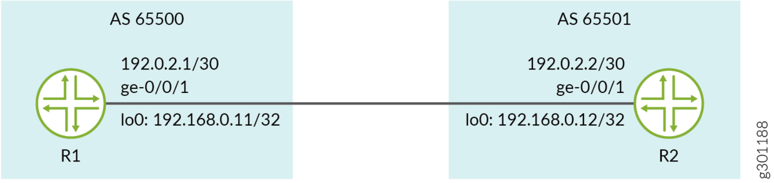
Overview

IN THIS SECTION

BGP uses TCP as its transport protocol. TCP-AO is a method you can use to authenticate BGP sessions. You can apply a TCP-AO keychain at the BGP neighbor or at BGP group levels of the configuration hierarchy.

Topology

Figure 4: Topology for BGP Authentication



Configuration

IN THIS SECTION

In this example, you associate the TCP-AO authentication keychain `new_auth_key` and authentication algorithm `ao` on both devices to authenticate a BGP session.

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.

R1

```
[edit]
set interfaces ge-0/0/1 description R1-to-R2-Link
set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.11/32
set routing-options router-id 192.168.0.11
set routing-options autonomous-system 65500
set protocols bgp group ebgp_grp type external
set protocols bgp group ebgp_grp peer-as 65501
set protocols bgp group ebgp_grp neighbor 192.0.2.2
set protocols bgp group ebgp_grp neighbor 192.0.2.2 authentication-key-chain new_auth_key
set protocols bgp group ebgp_grp neighbor 192.0.2.2 authentication-algorithm ao
```

R2

```
[edit]
set interfaces ge-0/0/1 description R2-to-R1-Link
set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.2/30
set interfaces lo0 unit 0 family inet address 192.168.0.12/32
set routing-options router-id 192.168.0.12
set routing-options autonomous-system 65501
set protocols bgp group ebgp_grp type external
set protocols bgp group ebgp_grp peer-as 65500
set protocols bgp group ebgp_grp neighbor 192.0.2.1
set protocols bgp group ebgp_grp neighbor 192.0.2.1 authentication-key-chain new_auth_key
set protocols bgp group ebgp_grp neighbor 192.0.2.1 authentication-algorithm ao
```

Step-By-Step Procedure

1. Enter configuration mode.

2. Configure basic settings such as the interface IP address, interface description, a loopback address, router-ID, AS number on both devices.

   R1

   ```
   [edit]
   user@R1# set interfaces ge-0/0/1 description R1-to-R2-Link
   user@R1# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.1/30
   user@R1# set interfaces lo0 unit 0 family inet address 192.168.0.11/32
   user@R1# set routing-options router-id 192.168.0.11
   user@R1# set routing-options autonomous-system 65500
   ```

R2

```
[edit]
user@R2# set interfaces ge-0/0/1 description R2-to-R1-Link
user@R2# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.2/30
user@R2# set interfaces lo0 unit 0 family inet address 192.168.0.12/32
user@R2# set routing-options router-id 192.168.0.12
user@R2# set routing-options autonomous-system 65501
```

3. Configure an EBGP between R1 and R2.

   R1

```
[edit]
user@R1# set protocols bgp group ebgp_grp type external
user@R1# set protocols bgp group ebgp_grp peer-as 65501
user@R1# set protocols bgp group ebgp_grp neighbor 192.0.2.2
```

   R2

```
[edit]
user@R2# set protocols bgp group ebgp_grp type external
user@R2# set protocols bgp group ebgp_grp peer-as 65500
user@R2# set protocols bgp group ebgp_grp neighbor 192.0.2.1
```

4. Associate the authentication keychain `new_auth_key` and the authentication algorithm `ao` to the BGP session on both devices.

   R1

```
[edit]
user@R1# set protocols bgp group ebgp_grp neighbor 192.0.2.2 authentication-key-chain
new_auth_key
user@R1# set protocols bgp group ebgp_grp neighbor 192.0.2.2 authentication-algorithm ao
```

R2

```
[edit]
user@R2# set protocols bgp group ebgp_grp neighbor 192.0.2.1 authentication-key-chain
new_auth_key
user@R2# set protocols bgp group ebgp_grp neighbor 192.0.2.1 authentication-algorithm ao
```

5. Enter `commit` from configuration mode on both devices.

   Once you commit the configurations statements on both devices the BGP session should establish using the TCP-AO authentication method.

Results

Confirm your configurations by using the `show interfaces`, `show routing-options`, and `show protocols` commands from configuration mode.

user@R1# **show interfaces**

```
ge-0/0/1 {
     description R1-to-R2-Link;
    unit 0 {
        family inet {
            address 192.0.2.1/30;
        }
    }
}
lo0 {
     unit 0 {
        family inet {
            address 192.168.0.11/32;
        }
    }
}
```

user@R1# **show routing-options**

```
autonomous-system 65500;
```

```
user@R1# show protocols
```

```
bgp {
    group ebgp_grp {
        type external;
        peer-as 65500;
        neighbor 192.0.2.1 {
            authentication-algorithm ao;
            authentication-key-chain new_auth_key;
        {
    {
{
```

```
bgp {
    group ebgp_grp {
        type external;
        peer-as 65551;
    neighbor 192.0.2.2 {
        authentication-algorithm ao;
        authentication-key-chain new_auth_key;
        }
    }
}
```

Verification

IN THIS SECTION

Verify BGP Session Establishment

Purpose

Confirm BGP session establishment output after enabling TCP-AO.

Action

View a BGP summary of BGP session state with the `show bgp summary` operational mode command.

```
user@R1> show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                      0          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damaged...
192.0.2.2             65501          6          4       0       0        1:19 Establ
  inet.0: 0/0/0/0
```

Meaning

The highlighted output values indicate that BGP has successfully established a session with the TCP-AO authentication method 1:19 minute ago.

Verify BGP Session is Using TCP-AO

Purpose

Verify a BGP neighbor is authenticated with the TCP-AO keychain.

Action

Use the show bgp neighbor *neighbor* command to view configuration details for BGP peers. To filter only authentication-specific details in the output, use the pipe (|) function and match on authentication, as shown:

```
user@R1> show bgp neighbor 192.0.2.2 | match authentication
Authentication key chain: new_auth_key
  Authentication algorithm: ao
```

Meaning

The output indicates that authentication keychain new_auth_key and Authentication algorithm ao is applied to the BGP neighbor 192.0.2.2.

Example: Use TCP-AO to Authenticate an LDP Session

SUMMARY

This example shows you how to authenticate an LDP session using a TCP Authentication Option (TCP-AO) keychain.

IN THIS SECTION

Requirements

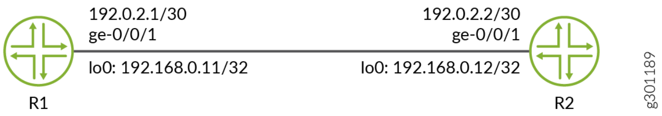
This example uses the following hardware and software components:

- MX Series or PTX Series routers.
- Junos OS Release 20.3R1 or later version.
- Configure a keychain new_auth_key. See "Configure a Keychain (TCP-AO)" on page 46.

Overview

Label Distribution Protocol (LDP) is an MPLS signaling protocol. It allows routers to establish label-switched paths (LSPs) through a network. TCP-AO helps enhance the security of sessions created among LDP peers.

Figure 5: Topology for LDP Configuration



Configuration

IN THIS SECTION

In this example, you associate the TCP-AO authentication keychain `new_auth_key` and authentication algorithm `ao` to both devices to authenticate their LDP session.

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.

R1

```
                    [edit]

set interfaces ge-0/0/1 description R1-to-R2-Link
set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.1/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.11/32
set routing-options router-id 192.168.0.11
set protocols ldp interface ge-0/0/1.0
set protocols ldp interface lo0.0
set protocols ldp session 192.168.0.12 authentication-algorithm ao
set protocols ldp session 192.168.0.12 authentication-key-chain new_auth_key
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

R2

```
                  [edit]

                  set interfaces ge-0/0/1 description R2-to-R1-Link
set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.2/30
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.12/32
set routing-options router-id 192.168.0.12
set protocols ldp interface ge-0/0/1.0
set protocols ldp interface lo0.0
set protocols ldp session 192.168.0.11 authentication-algorithm ao
set protocols ldp session 192.168.0.11 authentication-key-chain new_auth_key
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
```

Step-By-Step Procedure

1. Enter configuration mode.

2. Configure basic setup such as device interface, loopback, interface description, router ID, AS number on R1 and R2.

R1

```
[edit]
user@R1# set interfaces ge-0/0/1 description R1-to-R2-Link
user@R1# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.1/30
user@R1# set interfaces lo0 unit 0 family inet address 192.168.0.11/32
user@R1# set routing-options router-id 192.168.0.11
```

R2

```
[edit]
user@R2# set interfaces ge-0/0/1 description R2-to-R1-Link
user@R2# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.2/30
user@R2# set interfaces lo0 unit 0 family inet address 192.168.0.12/32
user@R2# set routing-options router-id 192.168.0.12
```

3. Configure MPLS and LDP on both devices.

   R1

   ```
   [edit]
   user@R1# set interfaces ge-0/0/1 unit 0 family mpls
   user@R1# set protocols ldp interface ge-0/0/1.0
   user@R1# set protocols ldp interface lo0.0
   ```

   R2

   ```
   [edit]
   user@R2# set interfaces ge-0/0/1 unit 0 family mpls
   user@R2# set protocols ldp interface ge-0/0/1.0
   user@R2# set protocols ldp interface lo0.0
   ```

4. Configure an interior gateway protocol (IGP) to advertise loopback address reachability. In this example, we configure OSPF.

R1

```
[edit protocols]
user@R1# set ospf area 0.0.0.0 interface ge-0/0/1.0
user@R1# set ospf area 0.0.0.0 interface lo0.0 passive
```

R2

```
[edit protocols]
user@R2# set ospf area 0.0.0.0 interface ge-0/0/1.0
user@R2# set ospf area 0.0.0.0 interface lo0.0 passive
```

5. Associate `authentication-key-chain new_auth_key` and `authentication-algorithm ao` with the label space ID of R1 and R2.

   R1

```
[edit protocols]
user@R1# set ldp session 192.168.0.12 authentication-algorithm ao
user@R1# set ldp session 192.168.0.12 authentication-key-chain new_auth_key
```

   R2

```
[edit protocols]
user@R2# set ldp session 192.168.0.11 authentication-algorithm ao
user@R2# set ldp session 192.168.0.11 authentication-key-chain new_auth_key
```

6. Enter `commit` from the configuration mode on both devices.

Results

Confirm your configuration by using the `show interfaces`, `show routing-options` and `show protocols` commands.

user@R1# **show interfaces**

```
ge-0/0/1 {
     description R1-to-R2-Link;
    unit 0 {
        family inet {
            address 192.0.2.1/30;
        }
    }
}
lo0 {
     unit 0 {
        family inet {
            address 192.168.0.11/32;
        }
    }
}
```

user@R1# **show routing-options**

```
router-id 192.168.0.11;
```

user@R1# **show protocols**

```
ldp {
     interface ge-0/0/1.0;
    interface lo0.0 passive;
            authentication-algorithm ao;
            authentication-key-chain new_auth_key;
        {
{
ospf {
    area 0.0.0.0 {
        interface ge-0/0/1.0;
        interface lo0.0;
    {
{
```

Verification

IN THIS SECTION

Verify LDP Session

Purpose

Verify LDP session Establishment with TCP-AO.

Action

Use the `show ldp session detail` operational mode command to verify the LDP session is correctly established.

```
user@R1> show ldp session detail



                     Address: 192.168.0.12, State: Operational, Connection: Open, Hold time: 22
  Session ID: 192.168.0.11:0--192.168.0.12:0
  Next keepalive in 2 seconds
  Passive, Maximum PDU: 4096, Hold time: 30, Neighbor count: 1
  Neighbor types: discovered
  Keepalive interval: 10, Connect retry interval: 1
  Local address: 192.168.0.11, Remote address: 192.168.0.12
  Up for 01:11:59
  Last down 01:13:12 ago; Reason: authentication key was changed
  Number of session flaps: 2
  Capabilities advertised: none
  Capabilities received: none
  Protection: disabled
  Session flags: none
  Authentication type: new_auth_key(ao key-chain, 192.168.0.12/32)
  Local - Restart: disabled, Helper mode: enabled
  Remote - Restart: disabled, Helper mode: enabled
  Local maximum neighbor reconnect time: 120000 msec
  Local maximum neighbor recovery time: 240000 msec
  Local Label Advertisement mode: Downstream unsolicited
  Remote Label Advertisement mode: Downstream unsolicited
  Negotiated Label Advertisement mode: Downstream unsolicited
  MTU discovery: disabled
  Nonstop routing state: Not in sync
  Next-hop addresses received:
    192.0.2.2
    192.168.0.12
    128.49.110.110
```

Meaning

The output indicates that LDP session is established.

Example: Use TCP-AO to Authenticate RPKI Validation Sessions

IN THIS SECTION

Overview

Resource Public Key Infrastructure (RPKI) is a public key infrastructure framework that is designed to secure the Internet's routing infrastructure, specifically the BGP. RPKI provides a way to connect Internet number resource information, such as IP Addresses, to a trust anchor. By using RPKI, legitimate holders of number resources are able to control the operation of Internet routing protocols to prevent route hijacking and other attacks.

Starting in Junos OS Release 22.2R1, you can authenticate RPKI sessions by using TCP Authentication Option (TCP-AO) and keychain.

This example shows you how to authenticate an RPKI validation session using a TCP-AO keychain. We'll be establishing an authenticated RPKI session between a client device (R1) and a server (R2).

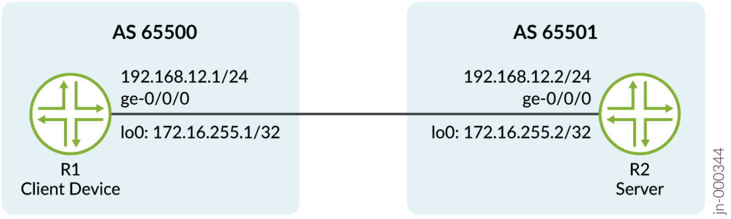
Requirements

This example uses the following hardware and software components:

- 2 MX Series routers
- Junos OS Release 22.2R1 or later version.

Topology

Figure 6: Topology for Authenticated RPKI Session



Configuration

IN THIS SECTION

In this example, you must associate the TCP-AO authentication keychain `new_auth_key` and authentication algorithm `ao` on both devices to authenticate an RPKI connection.

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.

R1

```
[edit]
set system host-name R1
set interfaces ge-0/0/0 description R1-to-R2-Link
set interfaces ge-0/0/0 unit 0 family inet address 192.168.12.1/24
set interfaces lo0 unit 0 family inet address 172.16.255.1/32
set routing-options router-id 172.16.255.1
set routing-options autonomous-system 65500
set security authentication-key-chains key-chain new_auth_key key 0 secret "$ABC123"
set security authentication-key-chains key-chain new_auth_key key 0 start-time
"2022-5-18.04:00:00 -0700"
set security authentication-key-chains key-chain new_auth_key key 0 algorithm ao
set security authentication-key-chains key-chain new_auth_key key 0 ao-attribute send-id 3
set security authentication-key-chains key-chain new_auth_key key 0 ao-attribute recv-id 8
set security authentication-key-chains key-chain new_auth_key key 1 secret "$ABC123"
set security authentication-key-chains key-chain new_auth_key key 1 start-time
"2022-6-18.04:00:00 -0700"
set security authentication-key-chains key-chain new_auth_key key 1 algorithm ao
set security authentication-key-chains key-chain new_auth_key key 1 ao-attribute send-id 1
set security authentication-key-chains key-chain new_auth_key key 1 ao-attribute recv-id 2
set routing-options validation group to_servers session 192.168.12.2 port 8282
set routing-options validation group to_servers session 192.168.12.2 authentication-algorithm ao
set routing-options validation group to_servers session 192.168.12.2 authentication-key-chain
new_auth_key
```

R2

```
[edit]
set system host-name R2
set logical-systems rv_server_1 interfaces ge-0/0/0 unit 0 family inet address 192.168.12.2/24
set interfaces lo0 unit 0 family inet address 172.16.255.2/32
set routing-options router-id 172.16.255.2
set routing-options autonomous-system 65501
set logical-systems rv_server_1 routing-options validation local-cache listen-port 8282
set logical-systems rv_server_1 routing-options validation group to_dut session 192.168.12.1
local-cache
set logical-systems rv_server_1 routing-options validation group to_dut session 192.168.12.1
local-address 192.168.12.2
set security authentication-key-chains key-chain new_auth_key key 0 secret "$ABC123"
set security authentication-key-chains key-chain new_auth_key key 0 start-time
"2022-5-18.04:00:00 -0700"
set security authentication-key-chains key-chain new_auth_key key 0 algorithm ao
set security authentication-key-chains key-chain new_auth_key key 0 ao-attribute send-id 8
set security authentication-key-chains key-chain new_auth_key key 0 ao-attribute recv-id 3
set security authentication-key-chains key-chain new_auth_key key 1 secret "$ABC123"
set security authentication-key-chains key-chain new_auth_key key 1 start-time
"2022-6-18.04:00:00 -0700"
set security authentication-key-chains key-chain new_auth_key key 1 algorithm ao
set security authentication-key-chains key-chain new_auth_key key 1 ao-attribute send-id 2
set security authentication-key-chains key-chain new_auth_key key 1 ao-attribute recv-id 1
set logical-systems rv_server_1 routing-options validation group to_dut session 192.168.12.1
authentication-algorithm ao
set logical-systems rv_server_1 routing-options validation group to_dut session 192.168.12.1
authentication-key-chain new_auth_key
```

Step-By-Step Procedure

1. Configure basic settings such as, interfaces, a loopback address, router-ID, and AS number on both devices. On R2, we configure logical systems interface for the server.

   R1

   ```
   [edit]
   user@R1# set system host-name R1
   user@R1# set interfaces ge-0/0/0 description R1-to-R2-Link
   user@R1# set interfaces ge-0/0/0 unit 0 family inet address 192.168.12.1/24
   user@R1# set interfaces lo0 unit 0 family inet address 172.16.255.1/32
   user@R1# set routing-options router-id 172.16.255.1
   user@R1# set routing-options autonomous-system 65500
   ```

   R2

   ```
   [edit]
   user@R2# set system host-name R2
   user@R2# set logical-systems rv_server_1 interfaces ge-0/0/0 unit 0 family inet address
   192.168.12.2/24
   user@R2# set interfaces lo0 unit 0 family inet address 172.16.255.2/32
   user@R2# set routing-options router-id 172.16.255.2
   user@R2# set routing-options autonomous-system 65501
   ```

2. Configure a TCP session on the client device (R1) with the RPKI server (R2) with an alternative TCP port number.

R1

```
[edit]
user@R1# set routing-options validation group to_servers session 192.168.12.2 port 8282
```

3. On the server R2, configure an RPKI session with the client R1 for origin validation.

```
[edit]
user@R2# set logical-systems rv_server_1 routing-options validation group to_dut session
192.168.12.1 local-cache
user@R2# set logical-systems rv_server_1 routing-options validation group to_dut session
192.168.12.1 local-address 192.168.12.2
```

4. Create a keychain `new_auth_key` with the first key, (key 0):

R1

```
[edit security]
user@R1# set authentication-key-chains key-chain new_auth_key key 0 secret secretpassword
start-time 2022-5-18.04:00 algorithm ao ao-attribute send-id 3 recv-id 8
```

R2 (with send-id and recv-id values reversed)

```
[edit security]
user@R2# set authentication-key-chains key-chain new_auth_key key 0 secret secretpassword
start-time 2022-5-18.04:00 algorithm ao ao-attribute send-id 8 recv-id 3
```

5. To add another key (key 1), after creating key 0:

R1

```
[edit security authentication-key-chains key-chain new_auth_key]
user@R1# set key 1 secret secretpassword start-time 2022-6-18.04:00 algorithm ao ao-attribute
send-id 1 recv-id 2
```

R2 (with send-id and recv-id values reversed)

```
[edit security authentication-key-chains key-chain new_auth_key]
user@R2# set key 1 secret secretpassword start-time 2022-6-18.04:00 algorithm ao ao-attribute
send-id 2 recv-id 1
```

6. Apply the configured keychain `new_auth_key` and authentication algorithm `ao` on both R1 and R2.

   R1

```
[edit]
user@R1# set routing-options validation group to_servers session 192.168.12.2 authentication-
algorithm ao
user@R1# set routing-options validation group to_servers session 192.168.12.2 authentication-
key-chain new_auth_key
```

   R2

```
[edit]
user@R2# set logical-systems rv_server_1 routing-options validation group to_dut session
192.168.12.1 authentication-algorithm ao
user@R2# set logical-systems rv_server_1 routing-options validation group to_dut session
192.168.12.1 authentication-key-chain new_auth_key
```

7. Enter `commit` from configuration mode on both devices to activate your changes.

8. To verify the keychain `new_auth_key` with the two keys configured, use the `show security authentication-key-chains` command from configuration mode.

Results

Check the results of the keychain configuration on R1:

`user@R1#` **show security authentication-key-chains**

```
key-chain new_auth_key {
    key 0 {
        secret "$ABC123"; ## SECRET-DATA
        start-time "2022-5-18.04:00:00 -0700";
        algorithm ao;
        ao-attribute {
            send-id 3;
            recv-id 8;
        }
    }
    key 1 {
        secret "$ABC123"; ## SECRET-DATA
        start-time "2022-6-18.04:00:00 -0700";
        algorithm ao;
        ao-attribute {
            send-id 1;
            recv-id 2;
        }
    }
}
```

Confirm the remaining configurations applied on R1 by using the following commands:

`user@R1#` **show interfaces**

```
ge-0/0/0 {
    description R1-to-R2-Link;
    unit 0 {
        family inet {
            address 192.168.12.1/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 172.16.255.1/32;
        }
    }
}
```

`user@R1#` **show routing-options**

```
router-id 172.16.255.1;
autonomous-system 65500;
validation {
    group to_servers {
        session 192.168.12.2 {
            authentication-algorithm ao;
            authentication-key-chain new_auth_key;
            port 8282;
        }
    }
}
```

Verification

IN THIS SECTION

Purpose

Verify the session is established with TCP-AO keychain and algorithm configured on both the peers.

Action

View a validated session by using the `show validation session 192.168.12.2 detail` operational mode command.

```
user@R1> show validation session 192.168.12.2 detail
Session 192.168.12.2, State: up, Session index: 2
  Group: to_servers, Preference: 100
  Port: 8282
  Refresh time: 300s
  Hold time: 600s
  Record Life time: 3600s
  Serial (Full Update): 6
  Serial (Incremental Update): 6
  Authentication key-chain: new_auth_key
    Session flaps: 1
    Session uptime: 2d 01:40:05
    Last PDU received: 00:04:59
    IPv4 prefix count: 0
    IPv6 prefix count: 0
```

Meaning

The output indicates the session is up with the configured keychain `new_auth_key`.

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
22.4R1	Starting in Junos OS Evolved Release 22.4R1, you can configure TCP-AO or TCP MD5 authentication with an IP subnet to include the entire range of addresses under that subnet.
22.4R1	Starting in Junos OS Evolved Release 22.4R1, TCP authentication is VRF aware.

RELATED DOCUMENTATION

authentication-key-chains (TCP-AO)

3

CHAPTER

Configure Port Security

IN THIS CHAPTER

System Settings

SUMMARY

Configure the system settings on your device.

IN THIS SECTION

Review the "Platform-Specific System Settings Behavior" on page 78 section for notes related to your platform.

Specify the Physical Location of the Switch

To specify the physical location of the switch, specify the following options for the `location` statement at the `[edit system]` hierarchy level:

- `altitude` *feet*—Number of feet above sea level.
- `building` *name*—Name of the building, 1 to 28 characters in length. If the string contains spaces, enclose it in quotation marks (" ").
- `country-code` *code*—Two-letter country code.
- `floor` *number*—Floor in the building.
- `hcoord` *horizontal-coordinate*—Bellcore Horizontal Coordinate.
- `lata` *service-area*—Long-distance service area.
- `latitude` *degrees*—Latitude in degree format.

- longitude *degrees*—Longitude in degree format.
- npa-nxx *number*—First six digits of the phone number (area code and exchange).
- postal-code *postal-code*—Postal code.
- rack *number*—Rack number.
- vcoord *vertical-coordinate*—Bellcore Vertical Coordinate.

SEE ALSO

> *location (System)*
>
> *Example: Configuring the Name of the Switch, IP Address, and System ID*

Modify the Default Time Zone on the Device

To modify the local time zone, include the time-zone statement at the [edit system] hierarchy level:

```
[edit]
user@host# set system time-zone (GMT hour-offset | time-zone)
```

You can use the GMT *hour-offset* option to set the time zone relative to UTC (GMT) time. By default, *hour-offset* is 0. You can configure this to be a value from -14 to +12.

You can also specify the *time-zone* value as a string such as PDT (Pacific Daylight Time) or WET (Western European Time), or specify the continent and major city.

NOTE: Junos OS and Junos OS Evolved comply with the POSIX time-zone standard, which is counter-intuitive to the way time zones are generally indicated relative to UTC. A time zone ahead of UTC (east of the Greenwich meridian) is commonly indicated as GMT +*n*; for example, the Central European Time (CET) zone is indicated as GMT +1. However, this is not true for POSIX time zone designations. POSIX indicates CET as GMT-1. If you include the set system time-zone GMT+1 statement for a router in the CET zone, your router time will be set to one hour behind GMT, or two hours behind the actual CET time. For this reason, you might find it easier to use the POSIX time-zone strings, which you can list by entering set system time-zone ?.

For the time zone change to take effect for all processes running on the router or switch, you must reboot the router or switch.

The following example shows how to change the current time zone to `America/New_York`:

```
[edit]
user@host# set system time-zone America/New_York
[edit]
user@host# show
system {
    time-zone America/New_York;
}
```

SEE ALSO

NTP Time Servers

Configure Time Zones

Extend the Default Port Address Range

By default, the upper range of a port address is 5000. You can increase the range from which the port number can be selected to decrease the probability that someone can determine your port number.

- To configure the operating system to extend the default port address range, include the `source-port` statement at the `[edit system internet-options]` hierarchy level:

```
[edit]
user@host# set system internet-options source-port upper-limit upper-limit
```

The statement `upper-limit` *`upper-limit`* is the upper limit of a source port address and can be a value from 5000 through 65,355.

SEE ALSO

internet-options

Select a Fixed Source Address for Locally Generated TCP/IP Packets

By default, the source address included in locally generated Transmission Control Protocol/IP (TCP/IP) packets, such as FTP traffic, and in User Datagram Protocol (UDP) and IP packets, such as Network Time Protocol (NTP) requests, is chosen as the local address for the interface on which the traffic is transmitted. This means that the local address chosen for packets to a particular destination might change from connection to connection based on the interface that the routing protocol has chosen to reach the destination when the connection is established. If multiple equal-cost next hops are present for a destination, locally generated packets use the `lo0` address as a source.

To configure the software to select a fixed address to use as the source for locally generated IP packets, include the *default-address-selection* statement at the `[edit system]` hierarchy level:

```
[edit system]
user@host# set system default-address-selection
```

If you include the `default-address-selection` statement in the configuration, the operating system chooses the system default address as the source for most locally generated IP packets. The default address is usually an address configured on the `lo0` loopback interface. For example, if you specified that SSH and telnet use a particular address, but you also have `default-address selection` configured, the system default address is used.

Rebooting and Halting a Device

To reboot the switch, issue the `request system reboot` command.

```
user@switch> request system reboot ?
Possible completions:
  <[Enter]>            Execute this command
  all-members          Reboot all virtual chassis members
  at                   Time at which to perform the operation
  both-routing-engines Reboot both the Routing Engines
  fast-boot            Enable fast reboot
  hypervisor           Reboot Junos OS, host OS,  and Hypervisor
  in                   Number of minutes to delay before operation
  local                Reboot local virtual chassis member
  member               Reboot specific virtual chassis member (0..9)
  message              Message to display to all users
  other-routing-engine Reboot the other Routing Engine
  |                    Pipe through a command
{master:0}
user@switch> request system reboot
Reboot the system ? [yes,no] (no) yes
Rebooting switch
```

Similarly, to halt the switch, issue the `request system halt` command.

CAUTION: Before entering this command, you must have access to the switch's console port in order to bring up the Routing Engine.

```
user@switch> request system halt ?
Possible completions:
  <[Enter]>            Execute this command
  all-members          Halt all virtual chassis members
  at                   Time at which to perform the operation
  backup-routing-engine  Halt backup Routing Engine
  both-routing-engines  Halt both Routing Engines
  in                   Number of minutes to delay before operation
  local                Halt local virtual chassis member
  member               Halt specific virtual chassis member (0..9)
  message              Message to display to all users
  other-routing-engine  Halt other Routing Engine
  |                    Pipe through a command
```

Issuing the `request system halt` command on the switch halts the Routing Engine. To reboot a Routing Engine that has been halted, you must connect through the console.

SEE ALSO

- clear system reboot
- request system halt
- request system power-off

Platform-Specific System Settings Behavior

Use the following table to review platform-specific behavior for your platform:

Platform	Difference
QFX Series	• QFX10000 Series: When you issue the `request system reboot hypervisor` command on QFX10000 Series switches, the reboot takes longer than a standard reboot.

RELATED DOCUMENTATION

Disable Reporting IP Address and Timestamps in Ping Responses

4

CHAPTER

Configuration Statements and Operational Commands

IN THIS CHAPTER

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