Cisco N9300 Series Smart Switches NX-OS Feature Configuration Guide, Release 10.5(3s)

First Published: 2025-06-13

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Trademarks

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at: https://www.cisco.com/c/en/us/about/legal/cloud-and-software/software-terms.html. Cisco product warranty information is available at https://www.cisco.com/c/en/us/products/warranty-listing.html. US Federal Communications Commission Notices are found here https://www.cisco.com/c/en/us/products/us-fcc-notice.html.

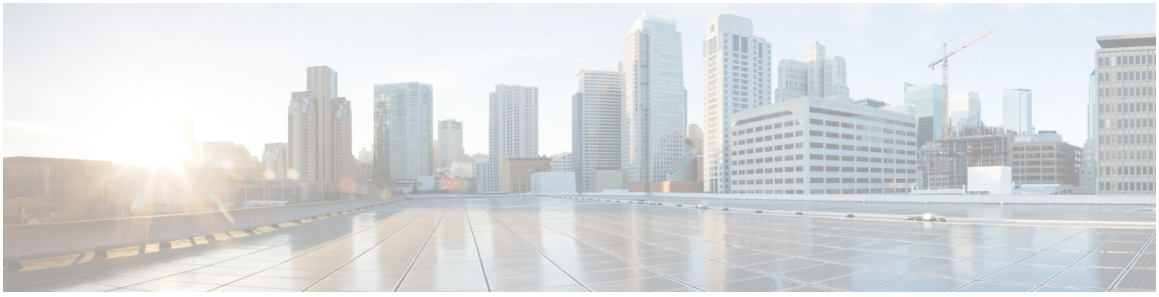
IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

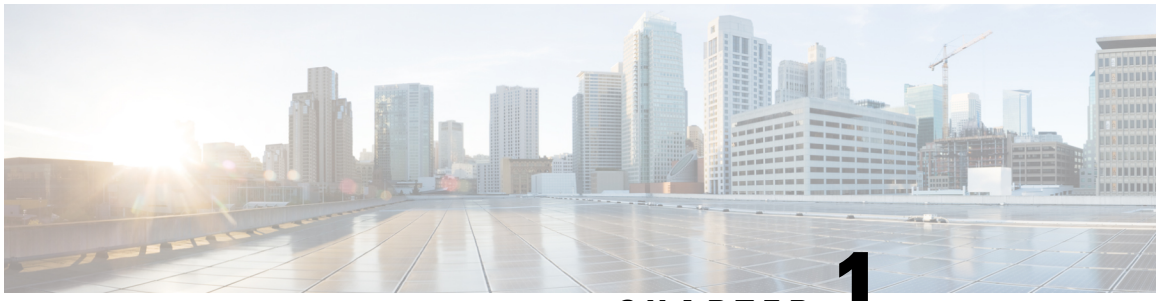
Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)



CONTENTS



CHAPTER 1

Upgrade N9300 Series Smart Switches

This chapter provides information regarding the software image for the N9300 series smart switches and the different ways in which the components such as Data Processing Unit (DPU) firmware can be upgraded.

N9300 series smart switches

Software image

Each Nexus switch is shipped with the Cisco NX-OS software preinstalled. The Cisco NX-OS software consists of an NX-OS software image, and this image is required to load the Cisco NX-OS operating system.

The Cisco N9300 Series Smart switches integrate Data Processing Units (DPUs) with networking ASICs to enhance data center networking and security. Hypershield manages the DPU to provide the security functions, while the NPU provides the N9000 routing and switching functions. At the core of the switch is the CPU, which runs the NXOS operating system. The CPU also hosts the Hypershield Agent, which connects to the external Hypershield system. For more information, refer to the Getting Started with Cisco N9300 Series Smart Switches document.

The Cisco NX-OS Release 10.5(3s) introduces Cisco N9324C-SE1U switch, a N9300 series smart switch, and a new NX-OS software image for the N9300 series smart switches. This 64-bit image has a file name with **nxos64-s1-dpu** as the prefix (for example, **nxos64-s1-dpu.10.5.3s.F.bin**) and is mandatory on the Cisco N9324C-SE1U switch. This image includes NX-OS component, DPU image, and Hypershield Agent.

The integration of Cisco NX-OS and Hypershield software into a single software image simplifies deployment and enhances operational flexibility.

Note The DPU image is available on the switch even when the DPU is powered off. To power on the DPU, enable feature service-acceleration. For more information about feature service-acceleration, refer to the Getting Started with Cisco N9300 Series Smart Switches document.

For more information about NX-OS upgrade and downgrade, refer to Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide, Release 10.5(x).

Upgrade DPU firmware and Hypershield Agent on N9324C-SE1U switch

DPU firmware and Hypershield Agent can be upgraded in different ways on Cisco N9324C-SE1U switch.

DPU firmware can be upgraded on a Cisco N9324C-SE1U switch in four ways that include:

- Through Hypershield Management Plane—On the Management Plane, ensure that the Hypershield Agent is connected. When a new DPU image is available on the Management Plane, the Agent downloads the DPU image and then performs the upgrade.
- Upgrade DPU firmware on NX-OS using RPM commands—Download the image to the switch bootflash and then use the **install add dpu_fw-**<*fw version*>**-10.5.3s.x86_64.rpm activate** command.

Note To downgrade, use the **install add dpu_fw-**<*fw version*>**-10.5.3s.x86_64.rpm activate downgrade** command.

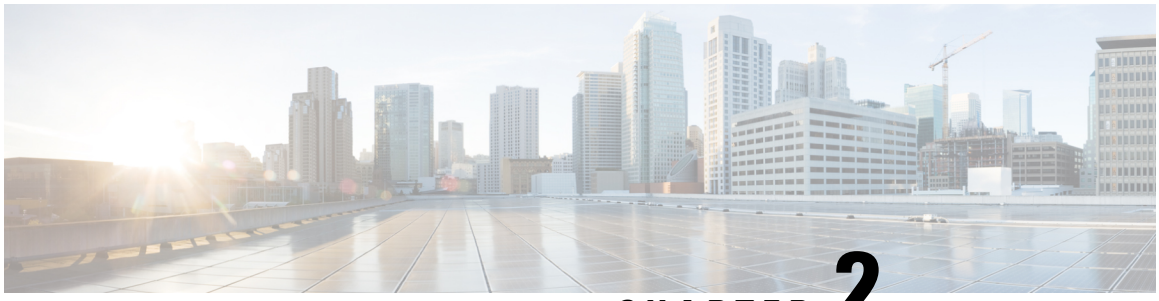
- Upgrade the entire NX-OS image and bundled DPU firmware—To upgrade the whole image, use the **install all nxos nxos64-s1-dpu.10.6.1.F.bin** command. After upgrade, the DPU that is bundled in the image must be used.
- Install RPMs during NX-OS upgrade—Use the **install all nxos nxos64-s1-dpu.10.6.1.F.bin package dpu_fw-**<*fw version*>**-10.6.1.x86_64.rpm** command.

Hypershield Agent can be upgraded on a Cisco N9324C-SE1U switch in two ways that include:

- Upgrade DPU firmware on NX-OS using RPM commands—Download the image to the switch bootflash and then use the **install add agent-**<*agent version*>**-10.5.3s.x86_64.rpm activate** command.

Note To downgrade, use the **install add agent-**<*agent version*>**-10.5.3s.x86_64.rpm activate downgrade** command.

- Install RPMs during NX-OS upgrade—Use the **install all nxos nxos64-s1-dpu.10.6.1.F.bin package agent-**<*agent version*>**-10.6.1.x86_64.rpm** command.



CHAPTER 2

Configure SPAN

This chapter describes how to configure an Ethernet switched port analyzer (SPAN) to analyze traffic between ports on Cisco NX-OS devices.

About SPAN

SPAN analyzes all traffic between source ports by directing the SPAN session traffic to a destination port with an external analyzer attached to it.

You can define the sources and destinations to monitor in a SPAN session on the local device.

SPAN Sources

The interfaces from which traffic can be monitored are called SPAN sources. Sources designate the traffic to monitor and whether to copy ingress (Rx), egress (Tx), or both directions of traffic. SPAN sources include the following:

- Ethernet ports (but not subinterfaces)
- The inband interface to the control plane CPU

Note When you specify the supervisor inband interface as a SPAN source, the device monitors all packets that are sent by the Supervisor CPU.

- VLANs

- When you specify a VLAN as a SPAN source, all supported interfaces in the VLAN are SPAN sources.
- VLANs can be SPAN sources only in the ingress direction.

Note This applies to all switches except Cisco Nexus 9300-EX/-FX/-FX2/-FX3/-GX platform switches, and Cisco Nexus 9500 series platform switches with -EX/-FX line cards.

- Satellite ports and host interface port channels on the Cisco Nexus 2000 Series Fabric Extender (FEX)
  - These interfaces are supported in Layer 2 access mode and Layer 2 trunk mode. They are not supported in Layer 3 mode, and Layer 3 subinterfaces are not supported.
  - Cisco Nexus 9300 and 9500 platform switches support FEX ports as SPAN sources in the ingress direction for all traffic and in the egress direction only for known Layer 2 unicast traffic flows through the switch and FEX. Routed traffic might not be seen on FEX HIF egress SPAN.

Note A single SPAN session can include mixed sources in any combination of the above.

Characteristics of Source Ports

SPAN source ports have the following characteristics:

- A port configured as a source port cannot also be configured as a destination port.
- If you use the supervisor inband interface as a SPAN source, all packets generated by the supervisor hardware (egress) are monitored.

Note Rx is from the perspective of the ASIC (traffic egresses from the supervisor over the inband and is received by the ASIC/SPAN).

SPAN Destinations

SPAN destinations refer to the interfaces that monitor source ports. Destination ports receive the copied traffic from SPAN sources. SPAN destinations include the following:

- Ethernet ports in either access or trunk mode
- Port channels in either access or trunk mode
- CPU as destination port
- Uplink ports on Cisco Nexus 9300 Series switches

Note FEX ports are not supported as SPAN destination ports.

Characteristics of Destination Ports

SPAN destination ports have the following characteristics:

- A port configured as a destination port cannot also be configured as a source port.
- The same destination interface cannot be used for multiple SPAN sessions. However, an interface can act as a destination for a SPAN and an ERSPAN session.
- Destination ports do not participate in any spanning tree instance. SPAN output includes bridge protocol data unit (BPDU) Spanning Tree Protocol hello packets.

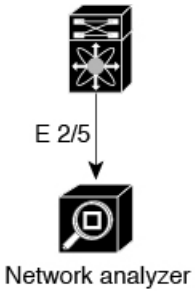
SPAN Sessions

You can create SPAN sessions to designate sources and destinations to monitor.

See the *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide* for information on the number of supported SPAN sessions.

This figure shows a SPAN configuration. Packets on three Ethernet ports are copied to destination port Ethernet 2/5. Only traffic in the direction specified is copied.

Figure 1: SPAN Configuration



Source Port	Direction	Destination Ports
E 2/1	Rx	E 2/5
E 2/2	Rx, Tx	
E 2/3	Tx	

503005

Localized SPAN Sessions

A SPAN session is localized when all of the source interfaces are on the same line card. A session destination interface can be on any line card.

Note A SPAN session with a VLAN source is not localized.

SPAN Truncation

Beginning with Cisco NX-OS Release 7.0(3)I7(1), you can configure the truncation of source packets for each SPAN session based on the size of the MTU. Truncation helps to decrease SPAN bandwidth by reducing

the size of monitored packets. Any SPAN packet that is larger than the configured MTU size is truncated to the given size. For example, if you configure the MTU as 300 bytes, the packets with greater than 300 bytes are truncated to 300 bytes.

SPAN truncation is disabled by default. To use truncation, you must enable it for each SPAN session.

ACL TCAM Regions

You can change the size of the ACL ternary content addressable memory (TCAM) regions in the hardware. For information on the TCAM regions used by SPAN sessions, see the Configuring IP ACLs chapter of the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

High Availability

The SPAN feature supports stateless and stateful restarts. After a reboot or supervisor switchover, the running configuration is applied. For more information on high availability, see the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide.

Prerequisites for SPAN

SPAN has the following prerequisites:

- You must first configure the ports on each device to support the desired SPAN configuration. For more information, see the Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide.

Guidelines and Limitations for SPAN

Note For scale information, see the release-specific *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.

SPAN has the following configuration guidelines and limitations:

- The **show monitor session** command displays incorrect statistics on the TX (egress) interface while mirroring traffic at line rate. This issue is seen in Cisco N93C64E-SG2-Q, Cisco N9364E-SG2-O switches with wide mode counters for polling statistics.
- Cisco N9336C-SE1 uses wide counters for statistics.
- A maximum of 48 source interfaces are supported per SPAN session (Rx and Tx, Rx, or Tx).
- Traffic that is denied by an ACL may still reach the SPAN destination port because SPAN replication is performed on the ingress side prior to the ACL enforcement (ACL dropping traffic).
- For SPAN session limits, see the *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.
- You can configure maximum of 32 source VLANs while configuring SPAN session.
- All SPAN replication is performed in the hardware. The supervisor CPU is not involved.

- You can configure a SPAN session on the local device only.
- Configuring two SPAN or ERSPAN sessions on the same source interface with only one filter is not supported. If the same source is used in multiple SPAN or ERSPAN sessions, either all the sessions must have different filters or no sessions should have filters.
- Packets with FCS errors are not mirrored in a SPAN session.
- The following guidelines apply to SPAN copies of access port dot1q headers:
  - When traffic ingresses from a trunk port or a routed port and egresses to an access port, an egress SPAN copy of an access port on a switch interface always has a dot1q header.
  - When traffic ingresses from an access port and egresses to a trunk port or a routed port, an ingress SPAN copy of an access port on a switch interface does not have a dot1q header.
  - When traffic ingresses from an access port and egresses to an access port, an ingress/egress SPAN copy of an access port on a switch interface does not have a dot1q header.
  - This behavior is applicable to Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9300-FX3, 9300-GX, 9300-GX2, 9500 platform switches with 9700-EX, 9700-FX, and 9700-GX line cards.
- You can configure only one destination port in a SPAN session.
- SPAN mirroring is not supported for PBR traffic.
- You cannot configure a port as both a source and destination port.
- Enabling UniDirectional Link Detection (UDLD) on the SPAN source and destination ports simultaneously is not supported. If UDLD frames are expected to be captured on the source port of such SPAN session, disable UDLD on the destination port of the SPAN session.
- SPAN is not supported for management ports.
- Statistics are not support for the filter access group.
- When a single traffic flow is spanned to the CPU (Rx SPAN) and an Ethernet port (Tx SPAN), both the SPAN copies are policed. Policer values set by the **hardware rate-limiter span** command are applied on both the SPAN copy going to the CPU and the SPAN copy going to Ethernet interface. This limitation applies to the following switches:
  - Cisco Nexus 92348GC-X, Cisco Nexus 9332C, and Cisco Nexus 9364C switches
  - Cisco Nexus 9300-EX, FX, FX2, FX3, GX platform switches
  - Cisco Nexus 9504, 9508, and 9516 platform switches with EX and FX line cards
- SPAN is supported in Layer 3 mode; however, SPAN is not supported on Layer 3 subinterfaces or Layer 3 port-channel subinterfaces.
- When a SPAN session contains source ports that are monitored in the transmit or transmit and receive direction, packets that these ports receive can be replicated to the SPAN destination port although the packets are not actually transmitted on the source ports. Some examples of this behavior on source ports are as follows:
  - Traffic that results from flooding
  - Broadcast and multicast traffic

- SPAN sessions cannot capture packets with broadcast or multicast MAC addresses that reach the supervisor, such as ARP requests and Open Shortest Path First (OSPF) protocol hello packets, if the source of the session is the supervisor Ethernet in-band interface. To capture these packets, you must use the physical interface as the source in the SPAN sessions.
- VLAN SPAN monitors only the traffic that enters Layer 2 ports in the VLAN.
- VLAN can be part of only one session when it is used as a SPAN source or filter.
- VLAN ACL redirects to SPAN destination ports are not supported.
- When using a VLAN ACL to filter a SPAN, only **action forward** is supported; **action drop** and **action redirect** are not supported.
- The combination of VLAN source session and port source session is not supported. If the traffic stream matches the VLAN source session and port source session, two copies are needed at two destination ports. Due to the hardware limitation, only the VLAN source SPAN and the specific destination port receive the SPAN packets. This limitation applies only to the following Cisco devices:

Table 1: Cisco Nexus 9000 Series Switches

Cisco Nexus 93120TX	Cisco Nexus 93128TX	Cisco Nexus 9332PQ
Cisco Nexus 9372PX	Cisco Nexus 9372PX-E	Cisco Nexus 9372TX
Cisco Nexus 9396PX	Cisco Nexus 9372TX-E	Cisco Nexus 9396TX

Table 2: Cisco Nexus 9000 Series Line Cards, Fabric Modules, and GEM Modules

N9K-X9408PC-CFP2	N9K-X9536PQ	N9K-C9504-FM
N9K-X9432PQ	N9K-X9464TX	—

- When you filter a monitor session, make sure that the access-group specified must be a VACL, or VLAN access-map and not a regular ACL for filtering purpose. This guidelines is not applicable for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.
- An access-group filter in a SPAN session must be configured as vlan-accessmap. This guideline does not apply for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.
- Supervisor-generated stream of bytes module header (SOBMH) packets have all the information to go out on an interface and can bypass all forwarding lookups in the hardware, including SPAN and ERSPAN. CPU-generated frames for Layer 3 interfaces and the Bridge Protocol Data Unit (BPDU) class of packets are sent using SOBMH. This guideline does not apply for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards. The Cisco Nexus 9636C-R and 9636Q-R both support inband SPAN and local SPAN.
- Cisco NX-OS does not span Link Layer Discovery Protocol (LLDP) or Link Aggregation Control Protocol (LACP) packets when the source interface is not a host interface port channel.
- SPAN copies for multicast packets are made before rewrite. Therefore, the TTL, VLAN ID, any remarking due to an egress policy, and so on, are not captured in the SPAN copy.
- If SPAN is mirroring the traffic which ingresses on an interface in an ASIC instance and egresses on a Layer 3 interface (SPAN Source) on a different ASIC instance, then a Tx mirrored packet has a VLAN

ID of 4095 on Cisco Nexus 9300 platform switches (except EX, FX, or FX2) and Cisco Nexus 9500 platform modular switches.

- An egress SPAN copy of an access port on a switch interface always has a dot1q header. This guideline does not apply for Cisco Nexus 9508 platform switches with 9636C-R and 9636Q-R line cards.
- The flows for post-routed unknown unicast flooded packets are in the SPAN session, even if the SPAN session is configured not to monitor the ports on which this flow is forwarded. This limitation applies to Network Forwarding Engine (NFE) and NFE2-enabled EOR switches and SPAN sessions that have Tx port sources.
- VLAN sources are spanned only in the Rx direction. This limitation does not apply to the following switch platforms which support VLAN spanning in both directions:
  - Cisco Nexus 9300-EX platform switches
  - Cisco Nexus 9300-FX platform switches
  - Cisco Nexus 9300-FX2 platform switches
  - Cisco Nexus 9300-FX3 platform switches
  - Cisco Nexus 9300-GX platform switches
  - Cisco Nexus 9504, 9508, and 9516 switches with the 97160YC-EX line card.
  - Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.
- If a VLAN source is configured as both directions in one session and the physical interface source is configured in two other sessions, Rx SPAN is not supported for the physical interface source session. This limitation applies to the Cisco Nexus 97160YC-EX line card.
- With regard to session filtering functionality, ACL filter is supported only in Rx source, and VLAN filter is supported in both Tx and Rx sources. This guideline does not apply for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.
- Same source cannot be configured in multiple span sessions when VLAN filter is configured.
- The FEX NIF interfaces or port-channels cannot be used as a SPAN source or SPAN destination. If the FEX NIF interfaces or port-channels are specified as a SPAN source or SPAN destination, the software displays an unsupported error.
- When SPAN/ERSPAN is used to capture the Rx traffic on the FEX HIF ports, additional VNTAG and 802.1Q tags are present in the captured traffic.
- VLAN and ACL filters are not supported for FEX ports.
- If the sources used in bidirectional SPAN sessions are from the same FEX, the hardware resources are limited to two SPAN sessions.
- Truncation is supported only for local and ERSPAN source sessions. It is not supported for ERSPAN destination sessions.
- When sFlow is configured on N9K-C9508-FM-G with the N9K-X9716D-GX line card, disable sFlow before configuring SPAN sessions.
- Configuring MTU on a SPAN session truncates all packets egressing on the SPAN destination (for that session) to the MTU value specified.

- The cyclic redundancy check (CRC) is recalculated for the truncated packet.
- The bytes specified are retained starting from the header of the packets. The rest are truncated if the packet is longer than the MTU.

- Beginning with Cisco NX-OS Release 10.1(2), SPAN is supported on the Cisco Nexus N9K-X9624D-R2 line card.
- Beginning with Cisco NX-OS Release 10.2(1q)F, SPAN is supported on the N9K-C9332D-GX2B platform switches.
- MTU truncation is not supported on Cisco Nexus 9504/9508 modular chassis with the N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX, and N9K-X96136YC-R line cards.
- Beginning with Cisco NX-OS Release 10.2(2)F, Multicast SPAN Tx is supported on Cisco Nexus 9300-GX, 9300-GX2, and 9300-FX3 platform switches.
- Beginning with Cisco NX-OS Release 10.3(1)F, SPAN is supported on Cisco Nexus 9808 platform switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, SPAN is supported on the following switches and line cards:
  - Cisco Nexus 9804 switch
  - Cisco Nexus 9332D-H2R switch
  - Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 and 9804 switches
- Beginning with Cisco NX-OS Release 10.4(2)F, Layer 3 port-channel interface as SPAN source is supported on 9232E-B1, 9808 and 9804 platform switches, and port-channel interface as SPAN destination is supported only on 9232E-B1 platform switch. Note that load balancing of mirrored traffic on port channel is not supported.
- Beginning with Cisco NX-OS Release 10.4(2)F, SPAN is supported on Cisco Nexus 93400LD-H1 platform switch.
- Beginning with Cisco NX-OS Release 10.4(3)F, SPAN is supported on Cisco Nexus 9364C-H1 platform switch.
- Beginning with Cisco NX-OS Release 10.5(3)F, SPAN is supported on Cisco Nexus 9364E-SG2 ToR switches.

SPAN guidelines and limitations for Cisco Nexus 9324C-SE1U switches

Beginning with Cisco NX-OS Release 10.5(3s), SPAN is supported on Cisco N9324C-SE1U ToR switches. This section lists the guidelines and limitations that you need to follow while configuring SPAN on this switch.

- **Sessions**—The switch supports a maximum of 10 active monitor sessions at a time, irrespective of the sessions being local SPAN or ERSPAN.
- **MTU truncation**—MTU truncation for SPAN Rx mirroring supports 144 bytes excluding FCS.
- **Multicast traffic**—When multicast traffic on front panel is mirrored by local SPAN, it is accounted as multicast under monitor port.

- **Port-channel interface**—When port-channel interface with more than one member port is used as SPAN destination, only one member interface is used to send mirrored traffic. Member selection is done in software, which can lead to packet loss when membership changes.
- **Packet mirroring**—N9324C-SE1U mirrors packets on sub-interface when parent service-ethernet interface is configured as source. SPAN mirrored packets do not have separate SPAN egress queue, they take the default queue (Q0) on SPAN destination interface. SPAN can be used to mirror traffic ingress or egress out of service-ethernet interface.
- **SPAN to CPU**—Only Rx mirroring is supported on SPAN to CPU.
- **Unsupported features**—The features that are not supported include:
  - mirroring packets on Layer 3 sub interfaces or Layer 3 port-channel sub interfaces when the respective parent interface is configured as source,
  - sharing of the same source port or interface across sessions,
  - tunnel ports, VLAN, SUP Ethernet, and management interface as a source, and
  - UDF and SPAN ACL filter.

Default Settings for SPAN

The following table lists the default settings for SPAN parameters.

Parameters	Default
SPAN sessions	Created in the shut state

Configuring SPAN

Note Cisco NX-OS commands for this feature may differ from those in Cisco IOS.

Configuring a SPAN Session

You can configure a SPAN session on the local device only. By default, SPAN sessions are created in the shut state.

Note For bidirectional traditional sessions, you can configure the sessions without specifying the direction of the traffic.

Before you begin

You must configure the destination ports in access or trunk mode. For more information, see the Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *ethernet slot/port*
3. **switchport**
4. **switchport monitor**
5. (Optional) Repeat Steps 2 through 4 to configure monitoring on additional SPAN destinations.
6. **no monitor session** *session-number*
7. **monitor session** *session-number*[**rx | tx**] [**shut**]
8. **description** *description*
9. **source** {**interface** *type* [**rx** | **tx** | **both**] | [**vlan** {*number* | *range*}[**rx**]} | [**vsan** {*number* | *range*}[**rx**]}
10. (Optional) Repeat Step 9 to configure all SPAN sources.
11. **filter vlan** {*number* | *range*}
12. (Optional) Repeat Step 11 to configure all source VLANs to filter.
13. (Optional) **filter access-group** *acl-filter*
14. **destination interface** *type slot/port*
15. **no shut**
16. (Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**]
17. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**interface** *ethernet slot/port* **Example:** `switch(config)# interface ethernet 2/5` `switch(config-if)#`	Enters interface configuration mode on the selected slot and port.
Step 3	**switchport** **Example:** `switch(config-if)# switchport`	Configures switchport parameters for the selected slot and port or range of ports.
Step 4	**switchport monitor** **Example:** `switch(config-if)# switchport monitor`	Configures the switchport interface as a SPAN destination.

	Command or Action	Purpose
Step 5	(Optional) Repeat Steps 2 through 4 to configure monitoring on additional SPAN destinations.	—
Step 6	**no monitor session** *session-number* **Example:** `switch(config)# no monitor session 3`	Clears the configuration of the specified SPAN session. The new session configuration is added to the existing session configuration.
Step 7	**monitor session** *session-number*[**rx** | **tx**] [**shut**] **Example:** `switch(config)# monitor session 3 rx` `switch(config-monitor)#` **Example:** `switch(config)# monitor session 3 tx` `switch(config-monitor)#` **Example:** `switch(config)# monitor session 3 shut` `switch(config-monitor)#`	Enters the monitor configuration mode. The new session configuration is added to the existing session configuration. By default, the session is created in the shut state, and the session is a local SPAN session. The optional keyword shut specifies a shut state for the selected session.
Step 8	**description** *description* **Example:** `switch(config-monitor)# description my_span_session_3`	Configures a description for the session. By default, no description is defined. The description can be up to 32 alphanumeric characters.
Step 9	**source** {**interface** *type* [**rx** | **tx** | **both**] | [**vlan** {*number* | *range*}[**rx**]} | [**vsan** {*number* | *range*}[**rx**]} **Example:** `switch(config-monitor)# source interface ethernet 2/1-3, ethernet 3/1 rx` **Example:** `switch(config-monitor)# source interface fc1/1 both` **Example:** `switch(config-monitor)# source interface port-channel 2` **Example:** `switch(config-monitor)# source interface san-port-channel201 both` **Example:** `switch(config-monitor)# source interface sup-eth 0 rx` **Example:** `switch(config-monitor)# source vlan 3, 6-8 rx` **Example:** `switch(config-monitor)# source vsan 500 rx`	Configures sources and the traffic direction in which to copy packets. You can enter a range of Ethernet ports, FC ports, a port channel, SAN port channels, an inband interface, a range of VLANs, a range of VSANs, or a satellite port or host interface port channel on the Cisco Nexus 2000 Series Fabric Extender (FEX). You can configure one or more sources, as either a series of comma-separated entries or a range of numbers. You can specify the traffic direction to copy as ingress (rx), egress (tx), or both. **Note** Source VLANs are supported only in the ingress direction. Source FEX ports are supported in the ingress direction for all traffic and in the egress direction only for known Layer 2 unicast traffic. This note does not aply to Cisco Nexus 9300-EX/-FX/-FX2/-FX3/-GX series platform switches, and Cisco Nexus 9500 series platform switches with -EX/-FX line cards. Supervisor as a source is only supported in the Rx direction.

	Command or Action	Purpose
	Example: `switch(config-monitor)# source interface ethernet 101/1/1-3`	For a unidirectional session, the direction of the source must match the direction specified in the session. **Note** Source VSANs are also supported only in the ingress direction.
Step 10	(Optional) Repeat Step 9 to configure all SPAN sources.	
Step 11	**filter vlan** {*number* | *range*} **Example:** `switch(config-monitor)# filter vlan 3-5, 7`	Configures which VLANs to select from the configured sources. You can configure one or more VLANs, as either a series of comma-separated entries or a range of numbers **Note** A FEX port that is configured as a SPAN source does not support VLAN filters. **Note** Filters are not supported when the source is either FC interface or VSAN.
Step 12	(Optional) Repeat Step 11 to configure all source VLANs to filter.	
Step 13	(Optional) **filter access-group** *acl-filter* **Example:** `switch(config-monitor)# filter access-group ACL1`	Associates an ACL with the SPAN session. **Note** Filters are not supported when the source is either FC interface or VSAN.
Step 14	Required: **destination interface** *type slot*/*port* **Example:** `switch(config-monitor)# destination interface ethernet 2/5` **Example:** `switch(config-monitor)# destination interface sup-eth 0`	Configures a destination for copied source packets. **Note** FC ports are not supported as a destination interface. **Note** The SPAN destination port must be either an access port or a trunk port. **Note** You must enable monitor mode on the destination port. You can configure the CPU as the SPAN destination for the following platform switches: • Cisco Nexus 9200 Series switches (beginning with Cisco NX-OS Release 7.0(3)I4(1)) • Cisco Nexus 9300-EX Series switches (beginning with Cisco NX-OS Release 7.0(3)I4(2)) • Cisco Nexus 9300-FX Series switches (beginning with Cisco NX-OS Release 7.0(3)I7(1))

	Command or Action	Purpose
		• Cisco Nexus 9300-FX2 Series switches (beginning with Cisco NX-OS Release 7.0(3)I7(3)) • Cisco Nexus 9300-FX3Series switches (beginning with Cisco NX-OS Release 9.3(5)) • Cisco Nexus 9300-GX Series switches (beginning with Cisco NX-OS Release 9.3(3)) • Cisco Nexus 9500-EX Series switches with -EX/-FX line cards To do so, enter **sup-eth 0** for the interface type.
Step 15	Required: **no shut** **Example:** `switch(config-monitor)# no shut`	Enables the SPAN session. By default, the session is created in the shut state.
Step 16	(Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**] **Example:** `switch(config-monitor)# show monitor session 3`	Displays the SPAN configuration.
Step 17	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring UDF-Based SPAN

You can configure the device to match on user-defined fields (UDFs) of the outer or inner packet fields (header or payload) and to send the matching packets to the SPAN destination. Doing so can help you to analyze and isolate packet drops in the network.

Before you begin

Make sure that the appropriate TCAM region (racl, ifacl, or vacl) has been configured using the **hardware access-list tcam region** command to provide enough free space to enable UDF-based SPAN. For more information, see the "Configuring ACL TCAM Region Sizes" section in the *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

SUMMARY STEPS

1. **configure terminal**
2. **udf** *udf-name offset-base offset length*
3. **hardware access-list tcam region** {**racl | ifacl | vacl** } **qualify** *qualifier-name*
4. **copy running-config startup-config**
5. **reload**

6. **ip access-list** *span-acl*
7. Enter one of the following commands:
   - **permit udf** *udf-name value mask*
   - **permit ip** *source destination* **udf** *udf-name value mask*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**udf** *udf-name offset-base offset length* **Example:** `switch(config)# udf udf-x packet-start 12 1` `switch(config)# udf udf-y header outer l3 20 2`	Defines the UDF as follows: • *udf-name*—Specifies the name of the UDF. You can enter up to 16 alphanumeric characters for the name. • *offset-base*—Specifies the UDF offset base as follows, where **header** is the packet header to consider for the offset: **packet-start** | **header** {**outer** | **inner** {**l3** | **l4**}}. • *offset*—Specifies the number of bytes offset from the offset base. To match the first byte from the offset base (Layer 3/Layer 4 header), configure the offset as 0. • *length*—Specifies the number of bytes from the offset. Only 1 or 2 bytes are supported. To match additional bytes, you must define multiple UDFs. You can define multiple UDFs, but Cisco recommends defining only required UDFs.
Step 3	**hardware access-list tcam region** {**racl | ifacl | vacl** } **qualify** *qualifier-name* **Example:** `switch(config)# hardware access-list tcam region racl qualify ing-l3-span-filter`	Attaches the UDFs to one of the following TCAM regions: • racl—Applies to Layer 3 ports. • ifacl—Applies to Layer 2 ports • vacl—Applies to source VLANs. You can attach up to 8 UDFs to a TCAM region. **Note** When the UDF qualifier is added, the TCAM region goes from single wide to double wide. Make sure enough free space is available; otherwise, this command will be rejected. If necessary, you can reduce the TCAM space from unused regions and then re-enter this command. For

	Command or Action	Purpose
		more information,see the "Configuring ACL TCAM Region Sizes" section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide. **Note** The **no** form of this command detaches the UDFs from the TCAM region and returns the region to single wide.
Step 4	Required: **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 5	Required: **reload** **Example:** `switch(config)# reload`	Reloads the device. **Note** Your UDF configuration is effective only after you enter **copy running-config startup-config** + **reload**.
Step 6	**ip access-list** *span-acl* **Example:** `switch(config)# ip access-list span-acl-udf-only` `switch(config-acl)#`	Creates an IPv4 access control list (ACL) and enters IP access list configuration mode.
Step 7	Enter one of the following commands: • **permit udf** *udf-name value mask* • **permit ip** *source destination* **udf** *udf-name value mask* **Example:** `switch(config-acl)# permit udf udf-x 0x40 0xF0 udf-y 0x1001 0xF00F` **Example:** `switch(config-acl)# permit ip 10.0.0./24 any udf udf-x 0x02 0x0F udf-y 0x1001 0xF00F`	Configures the ACL to match only on UDFs (example 1) or to match on UDFs along with the current access control entries (ACEs) for the outer packet fields (example 2). A single ACL can have ACEs with and without UDFs together. Each ACE can have different UDF fields to match, or all ACEs can match for the same list of UDFs.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring SPAN Truncation

You can configure truncation for local and SPAN source sessions only.

SUMMARY STEPS

1. **configure terminal**
2. **monitor session** *session number*
3. **source interface** *type slot/port* [**rx** | **tx** | **both**]

4. **mtu** *size*
5. **destination interface** *type slot/port*
6. **no shut**
7. (Optional) **show monitor session** *session*
8. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**monitor session** *session number* **Example:** `switch(config)# monitor session 5` `switch(config-monitor)#`	Enters monitor configuration mode for the specified SPAN session.
Step 3	**source interface** *type slot/port* [**rx** | **tx** | **both**] **Example:** `switch(config-monitor)# source interface ethernet 1/5 both`	Configures the source interface.
Step 4	**mtu** *size* **Example:** `switch(config-monitor)# mtu 320` **Example:** `switch(config-monitor)# mtu ?` `<320-1518> Enter the value of MTU truncation size for SPAN packets`	Configures the MTU size for truncation. Any SPAN packet that is larger than the configured MTU size is truncated to the configured size. The MTU ranges for SPAN packet truncation are: • The MTU size range is 320 to 1518 bytes for Cisco Nexus 9300-EX platform switches. • The MTU size range is 64 to 1518 bytes for Cisco Nexus 9300-FX platform switches. • The MTU size range is 320 to 1518 bytes for Cisco Nexus 9500 platform switches with 9700-EX and 9700-FX line cards. • The MTU size is 343 bytes (excluding FCS) for Cisco Nexus 9808 and 9804 platform switches.
Step 5	**destination interface** *type slot/port* **Example:** `switch(config-monitor)# destination interface Ethernet 1/39`	Configures the Ethernet SPAN destination port.

	Command or Action	Purpose
Step 6	**no shut** **Example:** `switch(config-monitor)# no shut`	Enables the SPAN session. By default, the session is created in the shut state.
Step 7	(Optional) **show monitor session** *session* **Example:** `switch(config-monitor)# show monitor session 5`	Displays the SPAN configuration.
Step 8	**copy running-config startup-config** **Example:** `switch(config-monitor)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring SPAN for Multicast Tx Traffic Across Different LSE Slices

Beginning with Cisco NX-OS Release 7.0(3)I7(1), you can configure SPAN for multicast Tx traffic across different leaf spine engine (LSE) slices on Cisco Nexus 9300-EX platform switches.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **hardware multicast global-tx-span**
3. **copy running-config startup-config**
4. **reload**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	[**no**] **hardware multicast global-tx-span** **Example:** `switch(config)# hardware multicast global-tx-span`	Configures SPAN for multicast Tx traffic across different leaf spine engine (LSE) slices. **Note** Beginning from Cisco NX-OS Release 10.2(2)F, if source and destination are on different slices, use this command for multicast SPAN Tx.
Step 3	**copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch(config)# copy running-config startup-config`	
Step 4	**reload** **Example:** `switch(config)# reload`	Reloads the device.

Configuring SPAN to CPU

Introduction

A SPAN-to-CPU is for troubleshooting packet flow through Cisco Nexus 9000 Series switches. Similarly, to a normal SPAN or Encapsulated Remote SPAN (ERSPAN) session, a SPAN-to-CPU monitor session involves the definition of one or more source interfaces and traffic directions. Any traffic that matches the direction (TX, RX, or both) defined on a source interface is replicated to the supervisor CPU. This traffic is filtered and analyzed with the use of ethanalyzer or saved to a local storage device for reviewing the results.

To verify whether packets generated by the CPU of a Cisco Nexus 9000 Series Switches are transmitted out of a specific interface, Cisco recommends using a packet capture utility on the remote device connected to the interface.

1. **Configuring SPAN as CPU destination**

   You must be able to configure CPU as monitor session destination and same must be configured on hardware. On Tahoe platforms, this configuration is supported for local span only as there is no customer requirement to support it for ERSPAN termination session. The same will be supported for N9K-C9508-FM-R2.

2. **Analyzing SPAN Traffic**

   When SPAN traffic reaches mentioned supervisor CPU. The modules identify as SPAN packets and takes necessary actions and ethanalyzer displays these packets. The Ethanalyzer control plane packet capture utility can be used to view traffic replicated to the CPU. The mirror keyword in the Ethanalyzer command filters traffic such that only traffic replicated by a SPAN-to-CPU monitor session is shown. Ethanalyzer capture and display filters can be used to further limit the traffic displayed.

3. **Limiting SPAN traffic rate**

   Spanned traffic for CPU must be rate limited to avoid control plane disruption. Ethanalyzer uses libpcap module for processing, stripping, and decoding packet headers. Ethanalyzer uses mirror option to display the span traffic reaching supervisor CPU. To match SPAN to CPU a separate span class is created. All the traffic will be created as SPAN class and separate rate is created for this class as Control Plane Policing (COPP). The COPP traffic rate limit will be 50 kbps.

4. **Filtering ACL**

   This will give customers the ability to choose the traffic which they want to monitor. This feature will be supported on all kind of monitor session. For span to cpu this particularly important as traffic will be rate limited and so, it becomes important to categorize the traffic which is intended to be spanned.

Guidelines and Limitations

SPAN-to-CPU has the following configuration guidelines and limitations:

- No ACL Filtering is supported on inband sources.
- Sources such as Physical Interfaces (L2 and L3), port channels, and L3 subinterface are supported with ACL filter.
- ACL Filter is supported for Rx sources only.
- No ACL filtering supported on VLAN sources.
- Configuring multiple span sessions for the same source is not supported.
- MTU truncation is not supported on N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX, N9K-X96136YC-R, N9K-X9624D-R2, N9K-C9508-FM-R, N9K-C9504-FM-R, N9K-C9508-FM-R2, N9K-C9504-FM-R2, N3K-C36180YC-R, N3K-C3636C-R, and N3K-C36480LD-R2.
- ACL filters are not supported on N9K-X9624D-R2 Line card until Cisco NX-OS release 10.2(2)F.
- Beginning with Cisco NX-OS Release 10.2(3)F, ACL filters is supported on N9K-X9624D-R2 Line card.

Configuring SPAN to CPU

You can configure SPAN to CPU.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**configure CPU as SPAN** **Example:** `switch(config-monitor)# destination interface sup-eth0`	Configures the CPU as the SPAN destination.
Step 3	**configure ACL Filter** **Example:** `switch(config-monitor)# filter access-group <acl_filter_name>`	Configures the access list which will be honored for filtering.
Step 4	**configure ethanalyzer** **Example:** `switch# ethanalyzer local interface inband mirror`	Displays spanned packets.

Example

This example shows the output of monitor session.

```
show monitor session 1 session 1
type : local
state : up
acl-name : acl-name not specified
source intf :
rx : Eth3/44
tx : Eth3/44
both : Eth3/44
source VLANs :
rx :
tx :
both :
filter VLANs : filter not specified
source fwd drops :
destination ports : sup-eth0
PFC On Interfaces :
source VSANs :
rx :
```

This example shows the output of copp.

```
# show policy-map interface control-plane | begin span
class-map copp-system-p-class-span (match-any)
match exception span
set cos 0
police cir 50 pps , bc 256 packets
module 1 : <Designated Module>
conformed 910228778 bytes;
7217965 packets;
violated 7217965 bytes;
0 packets;
module 3 :
conformed 0 bytes;
0 packets;
violated 0 bytes;
0 packets;
0 packets;
```

Shutting Down or Resuming a SPAN Session

You can shut down SPAN sessions to discontinue the copying of packets from sources to destinations. You can shut down one session in order to free hardware resources to enable another session. By default, SPAN sessions are created in the shut state.

You can resume (enable) SPAN sessions to resume the copying of packets from sources to destinations. In order to enable a SPAN session that is already enabled but operationally down, you must first shut it down and then enable it.

You can configure the shut and enabled SPAN session states with either a global or monitor configuration mode command.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **monitor session** {*session-range* | **all**} **shut**
3. **monitor session** *session-number*
4. [**no**] **shut**
5. (Optional) **show monitor**

6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	[**no**] **monitor session** {*session-range* | **all**} **shut** **Example:** `switch(config)# monitor session 3 shut`	Shuts down the specified SPAN sessions. By default, sessions are created in the shut state. The **no** form of the command resumes (enables) the specified SPAN sessions. By default, sessions are created in the shut state. **Note** If a monitor session is enabled but its operational status is down, to enable the session, you must first specify the **monitor session shut** command followed by the **no monitor session shut** command.
Step 3	**monitor session** *session-number* **Example:** `switch(config)# monitor session 3` `switch(config-monitor)#`	Enters the monitor configuration mode. The new session configuration is added to the existing session configuration.
Step 4	[**no**] **shut** **Example:** `switch(config-monitor)# shut`	Shuts down the SPAN session. By default, the session is created in the shut state. The **no** form of the command enables the SPAN session. By default, the session is created in the shut state.
Step 5	(Optional) **show monitor** **Example:** `switch(config-monitor)# show monitor`	Displays the status of SPAN sessions.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the SPAN Configuration

To display the SPAN configuration, perform one of the following tasks:

Command	Purpose
show monitor session {**all** | *session-number* | **range** *session-range*} [**brief**]	Displays the SPAN session configuration.

Configuration Examples for SPAN

Configuration Example for a SPAN Session

To configure a SPAN session, follow these steps:

SUMMARY STEPS

1. Configure destination ports in access mode and enable SPAN monitoring.
2. Configure a SPAN session.

DETAILED STEPS

Procedure

Step 1 Configure destination ports in access mode and enable SPAN monitoring.

Example:

```
switch# configure terminal
switch(config)# interface ethernet 2/5
switch(config-if)# switchport
switch(config-if)# switchport monitor
switch(config-if)# no shut
switch(config-if)# exit
switch(config)#
```

Step 2 Configure a SPAN session.

Example:

```
switch(config)# no monitor session 3
switch(config)# monitor session 3
switch(config-monitor)# source interface ethernet 2/1-3, ethernet 3/1 rx
switch(config-monitor)# source interface port-channel 2
switch(config-monitor)# source interface sup-eth 0 both
switch(config-monitor)# source vlan 3, 6-8 rx
switch(config-monitor)# source interface ethernet 101/1/1-3
switch(config-monitor)# filter vlan 3-5, 7
switch(config-monitor)# destination interface ethernet 2/5
switch(config-monitor)# no shut
switch(config-monitor)# exit
switch(config)# show monitor session 3
switch(config)# copy running-config startup-config
```

Example:

```
switch(config)# monitor session 1
switch(config-monitor)# source interface fc 1/9/1
switch(config-monitor)# source interface san-port-channel 171
switch(config-monitor)# source vsan 3701
switch(config-monitor)# destination interface ethernet 1/8
switch(config-monitor)# no shutdown
switch(config-monitor)# exit
switch(config)# show monitor session 1
switch(config)# copy running-config startup-config
```

Configuration Example for a Unidirectional SPAN Session

To configure a unidirectional SPAN session, follow these steps:

SUMMARY STEPS

1. Configure destination ports in access mode and enable SPAN monitoring.
2. Configure a SPAN session.

DETAILED STEPS

Procedure

Step 1 Configure destination ports in access mode and enable SPAN monitoring.

Example:

```
switch# configure terminal
switch(config)# interface ethernet 2/5
switch(config-if)# switchport
switch(config-if)# switchport monitor
switch(config-if)# no shut
switch(config-if)# exit
switch(config)#
```

Step 2 Configure a SPAN session.

Example:

```
switch(config)# no monitor session 3
switch(config)# monitor session 3 rx
switch(config-monitor)# source interface ethernet 2/1-3, ethernet 3/1 rx
switch(config-monitor)# filter vlan 3-5, 7
switch(config-monitor)# destination interface ethernet 2/5
switch(config-monitor)# no shut
switch(config-monitor)# exit
switch(config)# show monitor session 3
switch(config)# copy running-config startup-config
```

Configuration Example for a SPAN ACL

This example shows how to configure a SPAN ACL:

```
switch# configure terminal
switch(config)# ip access-list match_11_pkts
switch(config-acl)# permit ip 11.0.0.0 0.255.255.255 any
switch(config-acl)# exit
switch(config)# ip access-list match_12_pkts
switch(config-acl)# permit ip 12.0.0.0 0.255.255.255 any
switch(config-acl)# exit
switch(config)# vlan access-map span_filter 5
switch(config-access-map)# match ip address match_11_pkts
switch(config-access-map)# action forward
switch(config-access-map)# exit
switch(config)# vlan access-map span_filter 10
switch(config-access-map)# match ip address match_12_pkts
switch(config-access-map)# action forward
switch(config-access-map)# exit
switch(config)# monitor session 1
switch(config-erspan-src)# filter access_group span_filter
```

Configuration Examples for UDF-Based SPAN

This example shows how to configure UDF-based SPAN to match on the inner TCP flags of an encapsulated IP-in-IP packet using the following match criteria:

- Outer source IP address: 10.0.0.2
- Inner TCP flags: Urgent TCP flag is set
- Bytes: Eth Hdr (14) + Outer IP (20) + Inner IP (20) + Inner TCP (20, but TCP flags at 13th byte)
- Offset from packet-start: 14 + 20 + 20 + 13 = 67
- UDF match value: 0x20
- UDF mask: 0xFF

```
udf udf_tcpflags packet-start 67 1
hardware access-list tcam region racl qualify ing-l3-span-filter
copy running-config startup-config
reload
ip access-list acl-udf
permit ip 10.0.0.2/32 any udf udf_tcpflags 0x20 0xff
monitor session 1
source interface Ethernet 1/1
filter access-group acl-udf
```

This example shows how to configure UDF-based SPAN to match regular IP packets with a packet signature (DEADBEEF) at 6 bytes after a Layer 4 header start using the following match criteria:

- Outer source IP address: 10.0.0.2
- Inner TCP flags: Urgent TCP flag is set
- Bytes: Eth Hdr (14) + IP (20) + TCP (20) + Payload: 112233445566DEADBEEF7788
- Offset from Layer 4 header start: 20 + 6 = 26
- UDF match value: 0xDEADBEEF (split into two-byte chunks and two UDFs)

- UDF mask: 0xFFFFFFFF

```
udf udf_pktsig_msb header outer l4 26 2
udf udf_pktsig_lsb header outer l4 28 2
hardware access-list tcam region racl qualify ing-l3-span-filter
copy running-config startup-config
reload
ip access-list acl-udf-pktsig
permit udf udf_pktsig_msb 0xDEAD 0xFFFF udf udf_pktsig_lsb 0xBEEF 0xFFFF
monitor session 1
source interface Ethernet 1/1
filter access-group acl-udf-pktsig
```

Configuration Example for SPAN Truncation

This example shows how to configure SPAN truncation for use with MPLS stripping:

```
mpls strip
ip access-list mpls
statistics per-entry
20 permit ip any any redirect Ethernet1/5
interface Ethernet1/5
switchport
switchport mode trunk
mtu 9216
no shutdown
monitor session 1
source interface Ethernet1/5 tx
mtu 64
destination interface Ethernet1/6
  no shut
```

Configuration Examples for Multicast Tx SPAN Across LSE Slices

This example shows how to configure multicast Tx SPAN across LSE slices for Cisco Nexus 9300-EX platform switches. It also shows sample output before and after multicast Tx SPAN is configured.

Before Multicast Tx SPAN Is Configured

```
switch# show interface eth1/15-16, ethernet 1/27 counters

--------------------------------------
Port           InOctets    InUcastPkts
--------------------------------------
Eth1/15          580928              0
Eth1/16             239              0
Eth1/27               0              0

--------------------------------------
Port        InMcastPkts    InBcastPkts
--------------------------------------
Eth1/15            9077              0
Eth1/16               1              0
Eth1/27               0              0

--------------------------------------
Port          OutOctets   OutUcastPkts
--------------------------------------
Eth1/15             453              0
```

```
Eth1/16          581317              0
Eth1/27               0              0

----------------------------------------
Port       OutMcastPkts   OutBcastPkts
----------------------------------------
Eth1/15               4              0
Eth1/16            9080              0
Eth1/27               0              0
```

Configuring Multicast Tx SPAN

```
switch(config)# hardware multicast global-tx-span
Warning: Global Tx SPAN setting changed, please save config and reload
switch(config)# copy running-config start-up config
[########################################] 100%
Copy complete.
switch(config)# reload
This command will reboot the system. (y/n)?  [n] y
```

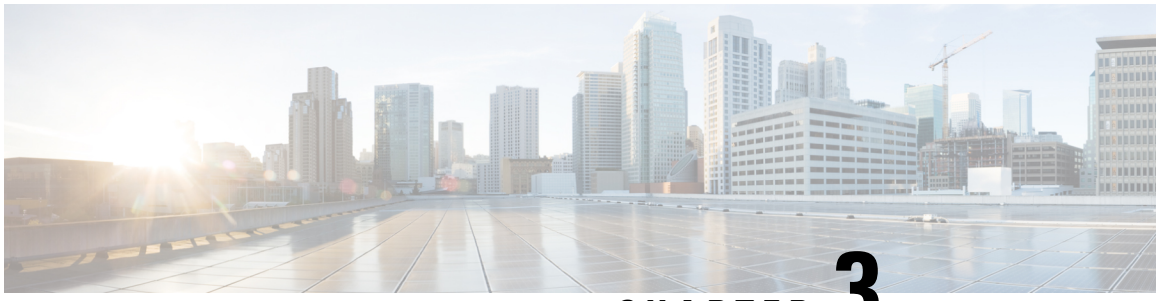
After Multicast Tx SPAN Is Configured

```
switch# show interface eth1/15-16, eth1/27 counters

----------------------------------------
Port           InOctets    InUcastPkts
----------------------------------------
Eth1/15          392576              0
Eth1/16               0              0
Eth1/27               0              0
----------------------------------------
Port        InMcastPkts    InBcastPkts
----------------------------------------
Eth1/15            6134              0
Eth1/16               0              0
Eth1/27               0              0

----------------------------------------
Port          OutOctets   OutUcastPkts
----------------------------------------
Eth1/15               0              0
Eth1/16          392644              0
Eth1/27          417112              0

----------------------------------------
Port       OutMcastPkts   OutBcastPkts
----------------------------------------
Eth1/15               0              0
Eth1/16            6135              0
Eth1/27            6134              0
```

Additional References

Related Documents

Related Topic	Document Title
ACL TCAM regions	*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*
FEX	*Cisco Nexus 2000 Series NX-OS Fabric Extender Software Configuration Guide for Cisco Nexus 9000 Series Switches*



CHAPTER 3

Configure ERSPAN

This chapter describes how to configure an encapsulated remote switched port analyzer (ERSPAN) to transport mirrored traffic in an IP network on Cisco NX-OS devices.

About ERSPAN

ERSPAN transports mirrored traffic over an IPv4 or IPv6 network, which provides remote monitoring of multiple switches across your network. The traffic is encapsulated at the source router and is transferred across the network. The packet is decapsulated at the destination router and then sent to the destination interface. Another method is that the destination can be the analyzer itself, which needs to understand the ERSPAN encapsulation format to parse the packet and access the inner (SPAN copy) frame.

ERSPAN Sources

The interfaces from which traffic can be monitored are called ERSPAN sources. Sources designate the traffic to monitor and whether to copy ingress, egress, or both directions of traffic. ERSPAN sources include the following:

- Ethernet ports (but not subinterfaces)
- Port channels
- The inband interface to the control plane CPU

Note When you specify the supervisor inband interface as a SPAN source, the device monitors all packets that are sent by the Supervisor CPU.

Note If you use the supervisor inband interface as a SPAN source, all packets generated by the supervisor hardware (egress) are monitored.

Rx is from the perspective of the ASIC (traffic egresses from the supervisor over the inband and is received by the ASIC/SPAN).

- VLANs
  - When a VLAN is specified as an ERSPAN source, all supported interfaces in the VLAN are ERSPAN sources.
  - VLANs can be ERSPAN sources only in the ingress direction, except for Cisco Nexus 9300-EX/-FX/-FX2/-FX3/-GX series platform switches, and Cisco Nexus 9500 series platform switches with -EX/-FX line cards.

Note A single ERSPAN session can include mixed sources in any combination of the above.

ERSPAN Destination

Destination ports receive the copied traffic from ERSPAN sources. The destination port is a port that is connected to the device such as a Remote Monitoring (RMON) probe or security device that can receive and analyze the copied packets from single or multiple source port. Destination ports do not participate in any spanning tree instance or any Layer 3 protocols

Cisco Nexus 9200, 9300-EX, 9300-FX, and 9300-FX2 platform switches support an ERSPAN destination session configured on physical or port-channel interfaces in switchport mode through the use of GRE header traffic flow. The source IP address should be configured on the default VRF. Multiple ERSPAN destination sessions should be configured with the same source IP address.

ERSPAN Sessions

You can create ERSPAN sessions that designate sources to monitor.

Localized ERSPAN Sessions

An ERSPAN session is localized when all of the source interfaces are on the same line card.

Note An ERSPAN session with a VLAN source is not localized

ERSPAN Truncation

Beginning with Cisco NX-OS Release 7.0(3)I7(1), you can configure the truncation of source packets for each ERSPAN session based on the size of the MTU. Truncation helps to decrease ERSPAN bandwidth by

reducing the size of monitored packets. Any ERSPAN packet that is larger than the configured MTU size is truncated to the given size. For ERSPAN, an additional ERSPAN header is added to the truncated packet from 54 to 166 bytes depending on the ERSPAN header type. For example, if you configure the MTU as 300 bytes, the packets are replicated with an ERSPAN header size from 354 to 466 bytes depending on the ERSPAN header type configuration.

ERSPAN truncation is disabled by default. To use truncation, you must enable it for each ERSPAN session.

Prerequisites for ERSPAN

ERSPAN has the following prerequisites:

- You must first configure the ports on each device to support the desired ERSPAN configuration. For more information, see the Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide.

Guidelines and Limitations for ERSPAN

Note For scale information, see the release-specific *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.

ERSPAN has the following configuration guidelines and limitations:

- A maximum of 48 source interfaces are supported per ERSPAN session (Rx and Tx, Rx, or Tx).
- ERSPAN destination handles jumbo frames for MTU differently based on the platform. For the following Cisco Nexus 9300 platform switches and Cisco Nexus 9500 platform switches with supporting line cards, ERSPAN destination drops the jumbo frames:
  - Cisco Nexus 9332PQ
  - Cisco Nexus 9372PX
  - Cisco Nexus 9372PX-E
  - Cisco Nexus 9372TX
  - Cisco Nexus 9372TX-E
  - Cisco Nexus 93120TX
  - Cisco Nexus 9500 platform switches with the following line cards:
    - Cisco Nexus 9564PX
    - Cisco Nexus 9464TX
    - Cisco Nexus 9464TX2
    - Cisco Nexus 9564TX
    - Cisco Nexus 9464PX
    - Cisco Nexus 9536PQ

- Cisco Nexus 9636PQ
- Cisco Nexus 9432PQ

For the following Cisco Nexus 9200 platform switches and Cisco Nexus 9500 platform switches with supporting line cards, ERSPAN truncates the packets at port MTU, and issues a TX Output error:

- Cisco Nexus 92160YC-X
- Cisco Nexus 92304QC
- Cisco Nexus 9272Q
- Cisco Nexus 9232C
- Cisco Nexus 9236C
- Cisco Nexus 92300YC
- Cisco Nexus 93108TC-EX
- Cisco Nexus 93180LC-EX
- Cisco Nexus 93180YC-EX
- Cisco Nexus 9500 platform switches with the following line cards:
  - Cisco Nexus 9736C-EX
  - Cisco Nexus 97160YC-EX
  - Cisco Nexus 9732C-EX
  - Cisco Nexus 9732C-EXM

- Using the ACL filter to ERSPAN subinterface traffic on the parent interface is not supported on the Cisco Nexus 9200 platform switches.
- Using the ACL filter to ERSPAN subinterface traffic on the parent interface is not supported on the Cisco Nexus 9300-EX/FX/FX2/FX3/GX platform switches.
- ERSPAN mirroring is not supported for PBR traffic.
- ERSPAN with a Type three header is not supported in Cisco NX-OS Release 9.3(3).
- For ERSPAN session limits, see the *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.
- The number of ERSPAN sessions per line card reduces to two if the same interface is configured as a bidirectional source in more than one session.
- Configuring two SPAN or ERSPAN sessions on the same source interface with only one filter is not supported. If the same source is used in multiple SPAN or ERSPAN sessions either all the sessions must have different filters or no sessions should have filters.
- Beginning with Cisco NX-OS Release 9.3(5), the following ERSPAN features are supported on Cisco Nexus 9300-GX platform switch:
  - ERSPAN Type III Header
  - ERSPAN Destination Support

- Packets with FCS errors are not mirrored in an ERSPAN session.
- TCAM carving is not required for SPAN/ERSPAN on the following line cards:
  - Cisco Nexus 9636C-R
  - Cisco Nexus 9636Q-R
  - Cisco Nexus 9636C-RX
  - Cisco Nexus 96136YC-R
  - Cisco Nexus 9624D-R2

Note All other switches supporting SPAN/ERSPAN must use TCAM carving.

- Statistics are not supported for the filter access group.
- An access-group filter in an ERSPAN session must be configured as vlan-accessmap.
- Control plane packets that are generated by the Supervisor cannot be ERSPAN encapsulated or filtered by an ERSPAN access control list (ACL).
- ERSPAN is not supported for management ports.
- ERSPAN does not support destinations on Layer 3 port-channel subinterfaces.
- VLAN as a source is not supported with ERSPAN configuration on R-series linecards and N3K-C36180YC-R, N3KC36480LD-R2, and N3K-C3636C-R platform switches.
- A VLAN can be part of only one session when it is used as an ERSPAN source or filter.
- VLAN ERSPAN monitors only the traffic that leaves or enters Layer 2 ports in the VLAN.
- If you enable ERSPAN on a vPC and ERSPAN packets must be routed to the destination through the vPC, packets that come through the vPC peer link cannot be captured.
- ERSPAN is not supported over a VXLAN overlay.
- ERSPAN copies for multicast packets are made before rewrite. Therefore, the TTL, VLAN ID, any remarking due to egress policy, and so on, are not captured in the ERSPAN copy.
- The timestamp granularity of ERSPAN Type III sessions is not configurable through the CLI. It is 100 picoseconds and driven through PTP.
- ERSPAN works on default and nondefault VRFs, but ERSPAN marker packets work only on the default VRF.
- The same source can be part of multiple sessions.

The following guidelines and limitations apply to egress (Tx) ERSPAN:

- The flows for post-routed unknown unicast flooded packets are in the ERSPAN session, even if the ERSPAN session is configured to not monitor the ports on which this flow is forwarded. This limitation applies to Network Forwarding Engine (NFE) and NFE2-enabled EOR switches and ERSPAN sessions that have TX port sources.

- For ERSPAN Tx multicast for Layer 2, ERSPAN copies are created independent of multicast replication. Due to this, multicast and SPAN packet have different values for VLAN tag, which is the ingress interface VLAN ID.
- The following guidelines and limitations apply to ingress (Rx) ERSPAN:
  - VLAN sources are spanned only in the Rx direction.
  - Session filtering functionality (VLAN or ACL filters) is supported only for Rx sources.
  - VLANs are supported as ERSPAN sources only in the ingress direction.
- Priority flow control (PFC) ERSPAN has the following guidelines and limitations:
  - It cannot coexist with filters.
  - It is supported only in the Rx direction on physical or port-channel interfaces. It is not supported in the Rx direction on VLAN interfaces or in the Tx direction.
- The following guidelines and limitations apply to FEX ports:
  - If the sources used in bidirectional ERSPAN sessions are from the same FEX, the hardware resources are limited to two ERSPAN sessions.
  - FEX ports are supported as ERSPAN sources in the ingress direction for all traffic and in the egress direction only for known Layer 2 unicast traffic.
  - Cisco Nexus 9300 platform switches do not support ERSPAN destination being connected on a FEX interface. The ERSPAN destination must be connected to a front panel port.
  - VLAN and ACL filters are not supported for FEX ports. It cannot coexist with filters.
- The following guidelines and limitations apply to ERSPAN destination:
  - Cisco Nexus 9200, 9300-EX, 9300-FX, and 9300-FX2 platform switches support an ERSPAN destination session that is configured on physical or port-channel interfaces in switchport mode by using GRE header traffic flow.
  - ERSPAN destination cannot coexist with other tunnel features such as MPLS and VXLAN for Cisco Nexus 9200, 9300, 9300-EX, 9300-FX, and 9300-FX2 platform switches.
  - On Cisco Nexus 9300-GX switches, dot1q-tagged broadcast or multicast packets passing through a device where the ERSPAN destination session is active, get tagged with the native VLAN instead of the correct VLAN due to a hardware limitation.
  - ERSPAN destination supports only default VRF.
  - Cisco Nexus 9300-EX/FX switches cannot serve as an ERSPAN destination for Cisco Nexus 3000 and non-EX/FX Cisco Nexus 9000 switches.
- Beginning with Cisco NX-OS Release 10.1(2), ERSPAN is supported on the Cisco Nexus N9K-X9624D-R2 Line Card.
- The following guidelines and limitations apply to ERSPAN over IPv6:
  - Beginning with Cisco NX-OS Release 10.2(1)F, the ERSPAN over IPv6 feature is supported on Cisco Nexus 9300-GX2, 9300-GX, 9300-FXP, 9300-FX2, 9300-EX, 9300-FX3, 9300-FX3S, and

9300-FX3P platform switches and N9K-X9716D-GX, N9K-X9736C-EX, N9K-X9732C-EX(X86_64 Atom), N9K-X9732C-EXM, N9K-X97160YC-EX, and N9K-X9736C-FX line cards.

- This feature is not supported for Load balancing across egress port-channel members and egress ECMP path.
- This feature is not supported for header-type 3, udf in filter ACL, and marker-packets.
- This feature is not supported for FEX host interface as ERSPAN source with IPv6.

- Beginning with Cisco NX-OS Release 10.2(3)F, IPv6 is supported on ERSPAN destination/termination on Cisco Nexus 9300-GX2, 9300-GX, 9300-FXP, 9300-FX2, 9300-EX, 9300-FX3, 9300-FX3S, and 9300-FX3P platform switches and N9K-X9716D-GX, N9K-X9736C-EX, N9K-X9732C-EX(X86_64 Atom), N9K-X9732C-EXM, N9K-X97160YC-EX, and N9K-X9736C-FX line cards.
- The following guidelines and limitations are applicable:
  - Only VRF default is supported.
  - You can only have one IPv6 address per switch.
  - This feature is not supported with other tunnel features.
  - You can bring up four ERSPAN destination sessions at a time.
  - ERSPAN ID is unique per session and the range is 1–32.
- Beginning with Cisco NX-OS Release 10.3(1)F, ERSPAN is supported on Cisco Nexus 9808 platform switches.
  - Only RX is supported on ERSPAN.
  - Type 3 header is not supported.
  - ERSPAN destination/termination is not supported.
- Beginning with Cisco NX-OS Release 10.4(1)F, ERSPAN is supported on the following switches and line cards:
  - Cisco Nexus 9332D-H2R switch
  - Cisco Nexus 9804 switch
    - Type 3 header is not supported
    - ERSPAN destination/termination is not supported
  - Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 and 9804 switches
- Beginning with Cisco NX-OS Release 10.4(2)F, Cisco Nexus 9300-H2R platform switches supports SPAN on ACL drop in ingress direction for the ERSPAN source session.
- Beginning with Cisco NX-OS Release 10.4(2)F, Layer 3 Port-channel interface as ERSPAN source and destination is supported on 9804 and 9808, and 9232E-B1 platform switches. However, the following guidelines and limitations are applicable:
  - Load balancing of mirrored traffic on port channel is not supported.

- Sharing of the same source port or interface across sessions is not supported.
- A maximum of 10 monitor sessions are supported at a time.
- 10 active ERSPAN sessions are supported at a time.
- ERSPAN MTU truncation is only supported for 343 bytes on 9804 and 9808 switches and for 152 bytes on 9232E-B1 switch excluding FCS.
- ERSPAN Type 3 header is not supported.
- ERSPAN destination/termination is not supported.
- ERSPAN Layer 2 interface (switch port) and VLAN as source is not supported.
- UDF-based ERSPAN is not supported.
- ERSPAN mirrored packets do not have separate SPAN egress queue, they take the default queue.
- When port-channel interface (with more than one member port) is configured as ERSPAN destination, only one member interface is used for sending out mirrored traffic.
- The member selection is done in software, so there will be packet loss when membership changes.

- Beginning with Cisco NX-OS Release 10.4(2)F, ERSPAN is supported on Cisco Nexus 93400LD-H1 platform switch.
- Beginning with Cisco NX-OS Release 10.4(3)F, ERSPAN is supported on Cisco Nexus 9364C-H1 platform switch.

ERSPAN guidelines and limitations for Cisco Nexus 9324C-SE1U switches

Beginning with Cisco NX-OS Release 10.5(3s), ERSPAN is supported on Cisco N9324C-SE1U ToR switches. This section lists the guidelines and limitations that you need to follow while configuring ERSPAN on this switch.

- **Sessions**—The switch supports a maximum of 10 active monitor sessions at a time, irrespective of the sessions being local SPAN or ERSPAN.
- **MTU truncation**—MTU truncation for ERSPAN Rx mirroring supports 144 bytes excluding FCS.
- **Port-channel interface**—When port-channel interface with more than one member port is used as ERSPAN destination, only one member interface is used to send mirrored traffic. Member selection is done in software, which can lead to packet loss when membership changes.
- **Packet mirroring**—N9324C-SE1U mirrors packets on sub-interface when parent service-ethernet interface is configured as source. ERSPAN mirrored packets do not have separate SPAN egress queue, they take the default queue (Q0) on ERSPAN destination interface. ERSPAN can be used to mirror traffic ingress or egress out of service-ethernet interface.
- **Unsupported features**—The features that are not supported include:
  - mirroring packets on Layer 3 sub interfaces or Layer 3 port-channel sub interfaces when the respective parent interface is configured as source,
  - sharing of the same source port or interface across sessions,
  - tunnel ports, VLAN, SUP Ethernet, and management interface as a source,

- ERSPAN Type 3 header,
- ERSPAN destination/termination, and
- UDF and ERSPAN ACL filter.

Default Settings

The following table lists the default settings for ERSPAN parameters.

Table 3: Default ERSPAN Parameters

Parameters	Default
ERSPAN sessions	Created in the shut state
ERSPAN marker packet interval	100 milliseconds
Timestamp granularity of ERSPAN Type III sessions	100 picoseconds

Configuring ERSPAN

Note Be aware that the Cisco NX-OS commands for this feature may differ from those commands used in Cisco IOS.

Configuring an ERSPAN Source Session

You can configure an ERSPAN session on the local device only. By default, ERSPAN sessions are created in the shut state.

Note ERSPAN does not monitor any packets that are generated by the supervisor, regardless of their source.

SUMMARY STEPS

1. **configure terminal**
2. **monitor erspan origin ip-address** *ip-address* **global** or **monitor erspan origin ipv6-address** *ipv6-address* **global**
3. **no monitor session** {*session-number* | **all**}
4. **monitor session** {*session-number* | **all**} **type erspan-source** [**shut**]
5. **description** *description*
6. **source** {**interface** *type* [**tx** | **rx** |**both**] **vlan** {**number** | **range**} [**rx**]}
7. (Optional) Repeat Step 7 to configure all ERSPAN sources.

8. **filter vlan** {*number* | *range*}
9. (Optional) Repeat Step 9 to configure all source VLANs — to filter.
10. (Optional) **filter access-group** *acl-filter*
11. **destination ip** *ip-address*
12. **erspan-id** *erspan-id*
13. **vrf** *vrf-name*
14. (Optional) **ip ttl** *ttl-number*
15. (Optional) **ip dscp** *dscp-number*
16. (Optional) [**no**] **marker-packet** *milliseconds*
17. **no shut**
18. **exit**
19. (Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**]
20. (Optional) **show running-config monitor**
21. (Optional) **show startup-config monitor**
22. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**monitor erspan origin ip-address** *ip-address* **global** or **monitor erspan origin ipv6-address** *ipv6-address* **global** **Example:** `switch(config)# monitor erspan origin ip-address 10.0.0.1 global` `switch(config)# monitor erspan origin ipv6-address 2001:DB8:1::1 global`	Configures the ERSPAN global origin IPv4 or IPv6 address.
Step 3	**no monitor session** {*session-number*	**all**} **Example:** `switch(config)# no monitor session 3`	Clears the configuration of the specified ERSPAN session. The new session configuration is added to the existing session configuration.
Step 4	**monitor session** {*session-number*	**all**} **type erspan-source** [**shut**] **Example:** `switch(config)# monitor session 3 type erspan-source` `switch(config-erspan-src)#`	Configures an ERSPAN Type II source session. By default the session is bidirectional. The optional keyword shut specifies a shut state for the selected session.

	Command or Action	Purpose
Step 5	**description** *description* **Example:** `switch(config-erspan-src)# description erspan_src_session_3`	Configures a description for the session. By default, no description is defined. The description can be up to 32 alphanumeric characters.
Step 6	**source** {**interface** *type* [**tx** | **rx** |**both**] **vlan** {**number** | **range**} [**rx**]} **Example:** `switch(config-erspan-src)# source interface ethernet 2/1-3, ethernet 3/1 rx` **Example:** `switch(config-erspan-src)# source interface port-channel 2` **Example:** `switch(config-erspan-src)# source interface sup-eth 0 rx` **Example:** `switch(config-erspan-src)# source vlan 3, 6-8 rx` **Example:** `switch(config-erspan-src)# source interface ethernet 101/1/1-3`	Configures the sources and traffic direction in which to copy packets. You can enter a range of Ethernet ports, a port channel, an inband interface, a range of VLANs, or a satellite port or host interface port channel on the Cisco Nexus 2000 Series Fabric Extender (FEX). You can configure one or more sources, as either a series of comma-separated entries or a range of numbers. You can specify the traffic direction to copy as ingress, egress, or both. For a unidirectional session, the direction of the source must match the direction specified in the session. **Note** Source VLANs are supported only in the ingress direction. Source FEX ports are supported in the ingress direction for all traffic and in the egress direction only for known Layer 2 unicast traffic. Supervisor as a source is only supported in the Rx direction.
Step 7	(Optional) Repeat Step 7 to configure all ERSPAN sources.	—
Step 8	**filter vlan** {*number* | *range*} **Example:** `switch(config-erspan-src)# filter vlan 3-5, 7`	Configures which VLANs to select from the configured sources. You can configure one or more VLANs, as either a series of comma-separated entries or a range of numbers. For information on the VLAN range, see the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide. **Note** A FEX port that is configured as an ERSPAN source does not support VLAN filters.
Step 9	(Optional) Repeat Step 9 to configure all source VLANs — to filter.	—
Step 10	(Optional) **filter access-group** *acl-filter* **Example:** `switch(config-erspan-src)# filter access-group ACL1`	Associates an ACL with the ERSPAN session. (You can create an ACL using the standard ACL configuration process. For more information, see the *Cisco Nexus 9000 Series NX-OS Security Configuration Guide.*) **Note**

	Command or Action	Purpose
		Before executing this command, configure ip access list and associated vlan access map. See Configuring ERSPAN ACL.
Step 11	**destination ip** *ip-address* **Example:** `switch(config-erspan-src)# destination ip 10.1.1.1` `switch(config-erspan-src)# destination ipv6 2001:DB8:1::1`	**destination ipv6** *ipv6-address* Configures the destination IPv4 or IPv6 address in the ERSPAN session. **Note** Only one destination IPv4 or IPv6 address is supported per ERSPAN source session.
Step 12	**erspan-id** *erspan-id* **Example:** `switch(config-erspan-src)# erspan-id 5`	Configures the ERSPAN ID for the ERSPAN source session. The ERSPAN range is from 1 to 1023.
Step 13	**vrf** *vrf-name* **Example:** `switch(config-erspan-src)# vrf default`	Configures the virtual routing and forwarding (VRF) instance that the ERSPAN source session uses for traffic forwarding. The VRF name can be any case-sensitive, alphanumeric string up to 32 characters.
Step 14	(Optional) **ip ttl** *ttl-number* **Example:** `switch(config-erspan-src)# ip ttl 25`	Configures the IP time-to-live (TTL) value for the ERSPAN traffic. The range is from 1 to 255.
Step 15	(Optional) **ip dscp** *dscp-number* **Example:** `switch(config-erspan-src)# ip dscp 42`	Configures the differentiated services code point (DSCP) value of the packets in the ERSPAN traffic. The range is from 0 to 63.
Step 16	(Optional) [**no**] **marker-packet** *milliseconds* **Example:** `switch(config-erspan-src)# marker-packet 100`	Enables the ERSPAN marker packet for a session in order to recover the real value of the ERSPAN timestamp. The interval can range from 100 to 1000 milliseconds. The **no** form of this command disables the marker packet for the session. **Note** ERSPAN marker packets only apply to Type 3 sessions.
Step 17	**no shut** **Example:** `switch(config-erspan-src)# no shut`	Enables the ERSPAN source session. By default, the session is created in the shut state.
Step 18	**exit** **Example:** `switch(config-erspan-src)# exit` `switch(config)#`	Exits the monitor configuration mode.

	Command or Action	Purpose
Step 19	(Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**] **Example:** `switch(config)# show monitor session 3`	Displays the ERSPAN session configuration.
Step 20	(Optional) **show running-config monitor** **Example:** `switch(config)# show running-config monitor`	Displays the running ERSPAN configuration.
Step 21	(Optional) **show startup-config monitor** **Example:** `switch(config)# show startup-config monitor`	Displays the ERSPAN startup configuration.
Step 22	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Shutting Down or Activating an ERSPAN Session

You can shut down ERSPAN sessions to discontinue the copying of packets from sources to destinations. You can shut down one session in order to free hardware resources to enable another session. By default, ERSPAN sessions are created in the shut state.

You can enable ERSPAN sessions to activate the copying of packets from sources to destinations. To enable an ERSPAN session that is already enabled but operationally down, you must first shut it down and then enable it. You can shut down and enable the ERSPAN session states with either a global or monitor configuration mode command.

SUMMARY STEPS

1. **configure terminal**
2. **monitor session** {*session-range* | **all**} **shut**
3. **no monitor session** {*session-range* | **all**} **shut**
4. **monitor session** *session-number* **type erspan-source**
5. **shut**
6. **no shut**
7. **exit**
8. (Optional) **show monitor session all**
9. (Optional) **show running-config monitor**
10. (Optional) **show startup-config monitor**
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**monitor session** {*session-range* | **all**} **shut** **Example:** `switch(config)# monitor session 3 shut`	Shuts down the specified ERSPAN sessions. By default, sessions are created in the shut state.
Step 3	**no monitor session** {*session-range* | **all**} **shut** **Example:** `switch(config)# no monitor session 3 shut`	Resumes (enables) the specified ERSPAN sessions. By default, sessions are created in the shut state. If a monitor session is enabled but its operational status is down, then to enable the session, you must first specify the **monitor session shut** command followed by the **no monitor session shut** command.
Step 4	**monitor session** *session-number* **type erspan-source** **Example:** `switch(config)# monitor session 3 type erspan-source` `switch(config-erspan-src)#`	Enters the monitor configuration mode for the ERSPAN source type. The new session configuration is added to the existing session configuration.
Step 5	**shut** **Example:** `switch(config-erspan-src)# shut`	Shuts down the ERSPAN session. By default, the session is created in the shut state.
Step 6	**no shut** **Example:** `switch(config-erspan-src)# no shut`	Enables the ERSPAN session. By default, the session is created in the shut state.
Step 7	**exit** **Example:** `switch(config-erspan-src)# exit` `switch(config)#`	Exits the monitor configuration mode.
Step 8	(Optional) **show monitor session all** **Example:** `switch(config)# show monitor session all`	Displays the status of ERSPAN sessions.
Step 9	(Optional) **show running-config monitor** **Example:** `switch(config)# show running-config monitor`	Displays the ERSPAN running configuration.

	Command or Action	Purpose
Step 10	(Optional) **show startup-config monitor** **Example:** `switch(config)# show startup-config monitor`	Displays the ERSPAN startup configuration.
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring an ERSPAN ACL

You can create an IPv4 ERSPAN ACL on the device and add rules to it.

SUMMARY STEPS

1. **configure terminal**
2. **ip access-list** *acl-name*
3. [*sequence-number*] {**permit** | **deny**} *protocol source destination*
4. **vlan access-map erpsan-acl** *map name* [*sequence-number*]
5. **match ip address** *acl-name*
6. **action forward**
7. **exit**
8. **monitor session** [*session-number*| **all**] **type erspan-source** [**shut**]
9. **filter access_group** *name*
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**ip access-list** *acl-name* **Example:** `switch(config)# ip access-list erspan-acl` `switch(config-acl)#`	Creates the ERSPAN ACL and enters IP ACL configuration mode. The *acl-name* argument can be up to 64 characters.
Step 3	[*sequence-number*] {**permit**	**deny**} *protocol source destination* **Example:**	Creates a rule in the ERSPAN ACL. You can create many rules. The *sequence-number* argument can be a whole number between 1 and 4294967295.

	Command or Action	Purpose
	`switch(config-acl)# permit ip 192.168.2.0/24 any` **Example:** `switch(config)# ip access-list match_11_pkts` `switch(config-acl)# permit ip 10.0.0.0/24 any` `switch(config-acl)# exit`	The **permit** and **deny** commands support many ways of identifying traffic.
Step 4	**vlan access-map erpsan-acl** *map name* [*sequence-number*] **Example:** `switch(config)# vlan access-map erspan_filter`	Enters VLAN access-map configuration mode for the VLAN access map specified. If the VLAN access map does not exist, the device creates it. If you do not specify a sequence number, the device creates a new entry whose sequence number is 10 greater than the last sequence number in the access map.
Step 5	**match ip address** *acl-name* **Example:** `switch(config-access-map)# match ip address erspan-acl`	Specifies an ACL for the access-map entry.
Step 6	**action forward** **Example:** `switch(config-access-map)# action forward`	Specifies the action that the device applies to traffic that matches the ACL.
Step 7	**exit** **Example:** `switch(config-access-map)# exit`	Exits VLAN access-map configuration mode.
Step 8	**monitor session** [*session-number*| **all**] **type erspan-source** [**shut**] **Example:** `switch(config)# monitor session 1 type erspan-source`	Configures an ERSPAN Type II source session. By default, the session is bidirectional. The optional keyword **shut** specifies a shut state for the selected session.
Step 9	**filter access_group** *name* **Example:** `switch(config-erspan-src)# filter access_group erspan_filter`	Associates an ACL with the ERSPAN session. (You can create an ACL using the standard ACL configuration process. For more information, see *Cisco Nexus 9000 Series NX-OS Security Configuration Guide.*)
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config-acl)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying ERSPAN ACL Configuration

To display the ERSPAN ACL configuration, execute the appropriate show commands from the following table.

Command	Purpose
show ip access-lists *name* **Example:** `switch(config-acl)# show ip access-lists erpsan-acl`	Displays the ERSPAN ACL configuration.
show vlan access-map *name* **Example:** `switch(config-acl)# show vlan access-map erspan_filter`	Displays information about VLAN access maps.
show monitor session {**all** | *session-number* | **range** *session-range*} [**brief**] **Example:** `switch(config-acl)# show monitor session 1`	Displays the ERSPAN session configuration.

Configuring UDF-Based ERSPAN

You can configure the device to match on user-defined fields (UDFs) of the outer or inner packet fields (header or payload) and to send the matching packets to the ERSPAN destination. Doing so can help you to analyze and isolate packet drops in the network.

Before you begin

Make sure that the appropriate TCAM region (racl, ifacl, or vacl) has been configured using the **hardware access-list tcam region** command to provide enough free space to enable UDF-based ERSPAN. For information, see the "Configuring ACL TCAM Region Sizes" section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

SUMMARY STEPS

1. **configure terminal**
2. **udf** *udf-name offset-base offset length*
3. **hardware access-list tcam region** {**racl | ifacl | vacl** }**qualify udf** *udf-names*
4. **copy running-config startup-config**
5. **reload**
6. **ip access-list** *erspan-acl*
7. Enter one of the following commands:
   - **permit udf** *udf-name value mask*
   - **permit ip** *source destination* **udf** *udf-name value mask*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**udf** *udf-name offset-base offset length* **Example:** `switch(config)# udf udf-x packet-start 12 1` `switch(config)# udf udf-y header outer l3 20 2`	Defines the UDF as follows: • *udf-name*—Specifies the name of the UDF. You can enter up to 16 alphanumeric characters for the name. • *offset-base*—Specifies the UDF offset base as follows, where **header** is the packet header to consider for the offset: **packet-start** | **header** {**outer** | **inner** {**l3** | **l4**}}. • *offset*—Specifies the number of bytes offset from the offset base. To match the first byte from the offset base (Layer 3/Layer 4 header), configure the offset as 0. • *length*—Specifies the number of bytes from the offset. Only 1 or 2 bytes are supported. To match additional bytes, you must define multiple UDFs. You can define multiple UDFs, but Cisco recommends defining only required UDFs.
Step 3	**hardware access-list tcam region** {**racl | ifacl | vacl** }**qualify udf** *udf-names* **Example:** `switch(config)# hardware access-list tcam region racl qualify udf udf-x udf-y`	Attaches the UDFs to one of the following TCAM regions: • racl—Applies to Layer 3 ports.—Applies to layer 2 and Layer 3 ports. • ifacl—Applies to Layer 2 ports. • vacl—Applies to source VLANs. You can attach up to 8 UDFs to a TCAM region. **Note** When the UDF qualifier is added, the TCAM region goes from single wide to double wide. Make sure enough free space is available; otherwise, this command will be rejected. If necessary, you can reduce the TCAM space from unused regions and then re-enter this command. For more information, see the "Configuring ACL TCAM Region Sizes" section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide. **Note** The **no** form of this command detaches the UDFs from the TCAM region and returns the region to single wide.

	Command or Action	Purpose
Step 4	Required: **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 5	Required: **reload** **Example:** `switch(config)# reload`	Reloads the device. **Note** Your UDF configuration is effective only after you enter **copy running-config startup-config** + **reload**.
Step 6	**ip access-list** *erspan-acl* **Example:** `switch(config)# ip access-list erspan-acl-udf-only` `switch(config-acl)#`	Creates an IPv4 access control list (ACL) and enters IP access list configuration mode.
Step 7	Enter one of the following commands: • **permit udf** *udf-name value mask* • **permit ip** *source destination* **udf** *udf-name value mask* **Example:** `switch(config-acl)# permit udf udf-x 0x40 0xF0 udf-y 0x1001 0xF00F` **Example:** `switch(config-acl)# permit ip 10.0.0.0/24 any udf udf-x 0x02 0x0F udf-y 0x1001 0xF00F`	Configures the ACL to match only on UDFs (example 1) or to match on UDFs along with the current access control entries (ACEs) for the outer packet fields (example 2). A single ACL can have ACEs with and without UDFs together. Each ACE can have different UDF fields to match, or all ACEs can match for the same list of UDFs.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring ERSPAN Truncation

You can configure truncation for local and ERSPAN source sessions only.

SUMMARY STEPS

1. **configure terminal**
2. **monitor session** *session-number* **type erspan-source**
3. **source interface** *type slot/port* [**rx** | **tx** | **both**]
4. **mtu** *size*
5. **destination interface** *type slot/port*
6. **no shut**
7. (Optional) **show monitor session** *session*
8. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**monitor session** *session-number* **type erspan-source** **Example:** `switch(config)# monitor session 10 type erspan-source` `switch(config-erspan-src)#`	Enters monitor configuration mode for the specified ERSPAN session.
Step 3	**source interface** *type slot/port* [**rx** | **tx** | **both**] **Example:** `switch(config-erspan-src)# source interface ethernet 1/5 both`	Configures the source interface.
Step 4	**mtu** *size* **Example:** `switch(config-erspan-src)# mtu 512` **Example:** `switch(config-erspan-src)# mtu ?` `<512-1518> Enter the value of MTU truncation size for ERSPAN packets (erspan header + truncated original packet)`	Configures the MTU size for truncation. Any ERSPAN packet that is larger than the configured MTU size is truncated to the configured size. The MTU ranges for ERSPAN packet truncation are: • The MTU size range is 512 to 1518 bytes for Cisco Nexus 9300-EX Series switches. • The MTU size range is 64 to 1518 bytes for Cisco Nexus 9300-FX Series switches. • The MTU size range is 512 to 1518 bytes for Cisco Nexus 9500 platform switches with 9700-EX and 9700-FX line cards. • The MTU size is 343 bytes (excluding FCS) for Cisco Nexus 9808 and 9804 platform switches.
Step 5	**destination interface** *type slot/port* **Example:** `switch(config-erspan-src)# destination interface Ethernet 1/39`	Configures the Ethernet ERSPAN destination port.
Step 6	**no shut** **Example:** `switch(config-erspan-src)# no shut`	Enables the ERSPAN session. By default, the session is created in the shut state.
Step 7	(Optional) **show monitor session** *session* **Example:**	Displays the ERSPAN configuration.

	Command or Action	Purpose
	`switch(config-erspan-src)# show monitor session 5`	
Step 8	**copy running-config startup-config** **Example:** `switch(config-erspan-src)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring an ERSPAN Destination Session

You can configure a ERSPAN destination session to copy packets from a source IP address to destination ports on the local device. By default, ERSPAN destination sessions are created in the shut state.

Before you begin

Ensure that you have already configured the destination ports in switchport monitor mode.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*[*-port*]
3. **switchport**
4. **switchport mode** [**access** | **trunk**]
5. **switchport monitor**
6. Repeat Steps 2 to 5 to configure monitoring on additional ERSPAN destinations.
7. **no monitor session** {*session-number* | **all**}
8. **monitor session** {*session-number* | **all**} **type erspan-destination**
9. **description** *description*
10. **source ip** *ip-address*
11. **destination** {[**interface** [*type slot/port*[*-port*]]] [**port-channel** *channel-number*]]}
12. (Optional) Repeat Step 11 to configure all ERSPAN destinations.
13. **erspan-id** *erspan-id*
14. **no shut**
15. **exit**
16. **exit**
17. (Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*}
18. (Optional) **show running-config monitor**
19. (Optional) **show startup-config monitor**
20. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**interface ethernet** *slot/port*[*-port*] **Example:** `switch(config)# interface ethernet 2/5` `switch(config-if)#`	Enters interface configuration mode on the selected slot and port or range of ports.
Step 3	**switchport** **Example:** `switch(config-if)# switchport`	Configures switchport parameters for the selected slot and port or range of ports.
Step 4	**switchport mode** [**access** | **trunk**] **Example:** `switch(config-if)# switchport mode trunk`	Configures the following switchport modes for the selected slot and port or range of ports: • access • trunk
Step 5	**switchport monitor** **Example:** `switch(config-if)# switchport monitor`	Configures the switchport interface as an ERSPAN destination.
Step 6	Repeat Steps 2 to 5 to configure monitoring on additional ERSPAN destinations.	—
Step 7	**no monitor session** {*session-number* | **all**} **Example:** `switch(config-if)# no monitor session 3`	Clears the configuration of the specified ERSPAN session. The new session configuration is added to the existing session configuration.
Step 8	**monitor session** {*session-number* | **all**} **type erspan-destination** **Example:** `switch(config-if)# monitor session 3 type erspan-destination` `switch(config-erspan-dst)#`	Configures an ERSPAN destination session.
Step 9	**description** *description* **Example:** `switch(config-erspan-dst)# description erspan_dst_session_3`	Configures a description for the session. By default, no description is defined. The description can be up to 32 alphanumeric characters.

	Command or Action	Purpose
Step 10	**source ip** *ip-address* **Example:** `switch(config-erspan-dst)# source ip 10.1.1.1` `switch(config-erspan-dst)# source ipv6 2001:DB8:1::1`	**source ipv6** *ipv6-address* Configures the source IPv4 or IPv6 address in the ERSPAN session. The source IPv4 or IPv6 address is a locally configured IPv4 or IPv6 address. The source IPv4 or IPv6 address in an ERSPAN destination session must match the destination IPv4 or IPv6 address configured in the ERSPAN source session from which the encapsulated data is received. Only one source IPv4 or IPv6 address is supported per ERSPAN destination session. **Note** IPv6 is supported from Cisco NX-OS Release 10.2(3)F.
Step 11	**destination** {[**interface** [*type slot/port*[*-port*]]] [**port-channel** *channel-number*]]} **Example:** `switch(config-erspan-dst)# destination interface ethernet 2/5`	Configures a destination for copied source packets. You can configure a destination interface. **Note** You can configure destination ports as trunk ports.
Step 12	(Optional) Repeat Step 11 to configure all ERSPAN destinations.	—
Step 13	**erspan-id** *erspan-id* **Example:** `switch(config-erspan-dst)# erspan-id 5`	Configures the ERSPAN ID for the ERSPAN session. The range is from 1 to 1023.
Step 14	**no shut** **Example:** `switch(config-erspan-dst)# no shut`	Enables the ERSPAN destination session. By default, the session is created in the shut state.
Step 15	**exit** **Example:** `switch(config-erspan-dst)# exit`	Exits monitor configuration mode.
Step 16	**exit** **Example:** `switch(config)# exit`	Exits global configuration mode.
Step 17	(Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} **Example:** `switch(config)# show monitor session 3`	Displays the ERSPAN session configuration.
Step 18	(Optional) **show running-config monitor** **Example:** `switch(config-erspan-src)# show running-config monitor`	Displays the running ERSPAN configuration.

	Command or Action	Purpose
Step 19	(Optional) **show startup-config monitor** **Example:** `switch(config-erspan-src)# show startup-config monitor`	Displays the ERSPAN startup configuration.
Step 20	(Optional) **copy running-config startup-config** **Example:** `switch(config-erspan-src)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the ERSPAN Configuration

To display the ERSPAN configuration, perform one of the following tasks:

Command	Purpose
show ip access-lists *name*	Displays the ERSPAN ACL configuration.
show monitor session {**all** | *session-number* | **range** *session-range*} [**brief**]	Displays the ERSPAN session configuration. The output includes the egress interface that is used to send the ERSPAN packets. The output varies depending on the type of egress interface used: • Physical Layer 3 interface—Displays the interface name. • SVI interface—Displays the member interface through which the route was learned. • Layer 3 port channel—Displays the port-channel interface name. • Layer 3 subinterface—Displays the parent interface name. • ECMP path—Displays the name of one of the equal-cost multipath (ECMP) member interfaces. Only the interface that is displayed will be used for mirroring the traffic even though the route is ECMP. • PFC on interfaces—Displays the priority flow control (PFC) status on the interface.
show running-config monitor	Displays the running ERSPAN configuration.
show startup-config monitor	Displays the ERSPAN startup configuration.

Configuration Examples for ERSPAN

Configuration Example for an ERSPAN Source Session Over IPv6

This example shows how to configure an ERSPAN source session over IPv6:

```
switch# configure terminal
switch(config)# monitor erspan origin ipv6-address 2001::10:0:0:9 global
switch(config)# moni session 10 type erspan-source
switch(config-erspan-src)# erspan-id 10
switch(config-erspan-src)# vrf default
switch(config-erspan-src)# source interface ethernet 1/64
switch(config-erspan-src)# destination ip 10.1.1.2
```

Configuration Example for a Unidirectional ERSPAN Session

This example shows how to configure a unidirectional ERSPAN session:

```
switch# configure terminal
switch(config)# interface ethernet 14/30
switch(config-if)# no shut
switch(config-if)# exit
switch(config)# no monitor session 3
switch(config)# monitor session 3 rxswitch(config-erspan-src)# source interface ethernet
2/1-3 rx
switch(config-erspan-src)# erspan-id 1
switch(config-erspan-src)# ip ttl 16
switch(config-erspan-src)# ip dscp 5
switch(config-erspan-src)# vrf default
switch(config-erspan-src)# destination ip 10.1.1.2
switch(config-erspan-src)# no shut
switch(config-erspan-src)# exit
switch(config)# show monitor session 1
```

Configuration Example for an ERSPAN ACL

This example shows how to configure an ERSPAN ACL:

```
switch# configure terminal
switch(config)# ip access-list match_10_pkts
switch(config-acl)# permit ip 10.0.0.0/24 any
switch(config-acl)# exit
switch(config)# ip access-list match_172_pkts
switch(config-acl)# permit ip 172.16.0.0/24 any
switch(config-acl)# exit
```

In the case of different ERSPAN destinations where the interesting traffic is chosen based on the defined ACL filters, the last configured session would always have the higher priority.

For example, if Monitor Session 1 is configured; then Monitor Session 2 is configured; then ERSPAN traffic filter works as intended. But, if the user goes back to Monitor Session 1 and re-applies one of the existing configuration line (no new changes in the config); then the spanned traffic switches back to Monitor Session 1.

Configuration Example for a Marker Packet

This example shows how to enable the ERSPAN marker packet with an interval of 2 seconds:

```
switch# configure terminal
switch(config)# monitor erspan origin ip-address 172.28.15.250 global
switch(config)# monitor session 1 type erspan-source
switch(config-erspan-src)# header-type 3
switch(config-erspan-src)# erspan-id 1
switch(config-erspan-src)# ip ttl 16
switch(config-erspan-src)# ip dscp 5
switch(config-erspan-src)# vrf default
switch(config-erspan-src)# destination ip 10.1.1.2
switch(config-erspan-src)# source interface ethernet 1/15 both
switch(config-erspan-src)# marker-packet 100
switch(config-erspan-src)# no shut
switch(config-erspan-src)# show monitor session 1
session 1
---------------
type              : erspan-source
state             : up
granularity       : nanoseconds
erspan-id         : 1
vrf-name          : default
destination-ip    : 10.1.1.2
ip-ttl            : 16
ip-dscp           : 5
header-type       : 3
origin-ip         : 172.28.15.250 (global)
source intf       :
    rx            : Eth1/15
    tx            : Eth1/15
    both          : Eth1/15
    rx            :
marker-packet     : enabled
packet interval   : 100
packet sent       : 25
packet failed     : 0
egress-intf       :
```

Configuration Examples for UDF-Based ERSPAN

This example shows how to configure UDF-based ERSPAN to match on the inner TCP flags of an encapsulated IP-in-IP packet using the following match criteria:

- Outer source IP address: 10.0.0.2
- Inner TCP flags: Urgent TCP flag is set
- Bytes: Eth Hdr (14) + Outer IP (20) + Inner IP (20) + Inner TCP (20, but TCP flags at 13th byte)
- Offset from packet-start: 14 + 20 + 20 + 13 = 67
- UDF match value: 0x20
- UDF mask: 0xFF

```
udf udf_tcpflags packet-start 67 1
hardware access-list tcam region racl qualify udf udf_tcpflags
copy running-config startup-config
```

```
reload
ip access-list acl-udf
permit ip 10.0.0.2/32 any udf udf_tcpflags 0x20 0xff
monitor session 1 type erspan-source
source interface Ethernet 1/1
filter access-group acl-udf
```

This example shows how to configure UDF-based ERSPAN to match regular IP packets with a packet signature (DEADBEEF) at 6 bytes after a Layer 4 header start using the following match criteria:

- Outer source IP address: 10.0.0.2
- Inner TCP flags: Urgent TCP flag is set
- Bytes: Eth Hdr (14) + IP (20) + TCP (20) + Payload: 112233445566DEADBEEF7788
- Offset from Layer 4 header start: 20 + 6 = 26
- UDF match value: 0xDEADBEEF (split into two-byte chunks and two UDFs)
- UDF mask: 0xFFFFFFFF

```
udf udf_pktsig_msb header outer l3 26 2
udf udf_pktsig_lsb header outer l3 28 2
hardware access-list tcam region racl qualify udf udf_pktsig_msb udf_pktsig_lsb
copy running-config startup-config
reload
ip access-list acl-udf-pktsig
permit udf udf_pktsig_msb 0xDEAD 0xFFFF udf udf_pktsig_lsb 0xBEEF 0xFFFF
monitor session 1 type erspan-source
source interface Ethernet 1/1
filter access-group acl-udf-pktsig
```

Configuration Example for ERSPAN Truncation

This example shows how to configure ERSPAN truncation for use with MPLS stripping:

```
mpls strip
ip access-list mpls
  statistics per-entry
  20 permit ip any any redirect Ethernet1/5

interface Ethernet1/5
  switchport
  switchport mode trunk
  mtu 9216
  no shutdown

monitor session 1
  source interface Ethernet1/5 tx
  mtu 64
  destination interface Ethernet1/6
  no shut
monitor session 21 type erspan-source
  description "ERSPAN Session 21"
  header-type 3
  erspan-id 21
  vrf default
  destination ip 10.1.1.2
  source interface Ethernet1/5 tx
  mtu 64
```

```
  no shut
monitor session 22 type erspan-source
  description "ERSPAN Session 22"
  erspan-id 22
  vrf default
  destination ip 10.2.1.2
  source interface Ethernet1/5 tx
  mtu 750
  no shut
monitor session 23 type erspan-source
  description "ERSPAN Session 23"
  header-type 3
  marker-packet 1000
  erspan-id 23
  vrf default
  destination ip 10.3.1.2
  source interface Ethernet1/5 tx
  mtu 1000
  no shut
```

Configuration Example for an ERSPAN Destination Session Over IPv4

This example shows how to configure an ERSPAN destination session over IPv4:

The **destination interface eth1/1** is in switchport monitor mode. This interface can not co-exist with mpls strip, tunnel, nv overlay, vn-segment-vlan-based, mpls segment-routing, mpls evpn, mpls static, mpls oam, mpls l3vpn , mpls ldp, and nv overlay evpn features.

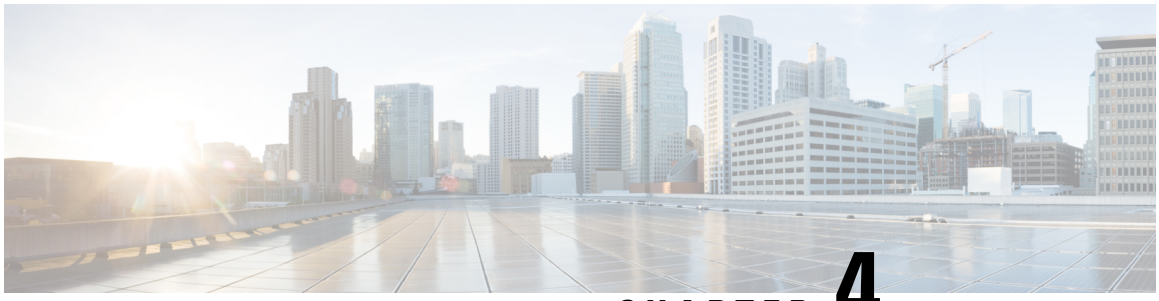
```
switch# monitor session 1 type erspan-destination
switch(config)# erspan-id 1
switch(config-erspan-dst)# source ip 10.1.1.1
switch(config-erspan-dst)# destination interface eth1/1
switch(config-erspan-dst)# no shut
switch(config-erspan-dst)# exit
```

Configuration Example for an ERSPAN Destination Session Over IPv6

This example shows how to configure an ERSPAN destination session over IPv6:

The **destination interface eth1/1** is in switchport monitor mode. This interface can not co-exist with mpls strip, tunnel, nv overlay, vn-segment-vlan-based, mpls segment-routing, mpls evpn, mpls static, mpls oam, mpls l3vpn , mpls ldp, and nv overlay evpn features.

```
switch# monitor session 1 type erspan-destination
switch(config)# erspan-id 1
switch(config-erspan-dst)# source ipv6 2001:DB8:1::1
switch(config-erspan-dst)# destination interface eth1/1
switch(config-erspan-dst)# no shut
switch(config-erspan-dst)# exit
```



CHAPTER 4

Configure MACsec

This chapter describes how to configure MACsec on Cisco NX-OS devices.

About MACsec

Media Access Control Security (MACsec) an IEEE 802.1AE along with MACsec Key Agreement (MKA) protocol provide secure communications on Ethernet links. It offers the following :

- Provides line rate encryption capabilities.
- Helps to ensure data confidentiality by providing strong encryption at Layer 2.
- Provides integrity checking to help ensure that data cannot be modified in transit.
- Can be selectively enabled using a centralized policy to help ensure that it is enforced where required while allowing non-MACsec-capable components to access the network.

- Encrypts packets on a hop-by-hop basis at Layer 2, allowing the network to inspect, monitor, mark, and forward traffic according to your existing policies (unlike end-to-end Layer 3 encryption techniques that hide the contents of packets from the network devices they cross).

Key Lifetime and Hitless Key Rollover

A MACsec keychain can have multiple pre-shared keys (PSKs), each configured with a key ID and an optional lifetime. A key lifetime specifies at which time the key activates and expires. In the absence of a lifetime configuration, the default lifetime is unlimited. When a lifetime is configured, MKA rolls over to the next configured pre-shared key in the keychain after the lifetime is expired. The time zone of the key can be local or UTC. The default time zone is UTC.

To configure a MACsec keychain, see Configuring a MACsec Keychain and Keys, on page 68.

A key can roll over to a second key within the same keychain by configuring the second key (in the keychain) and configuring a lifetime for the first key. When the lifetime of the first key expires, it automatically rolls over to the next key in the list. If the same key is configured on both sides of the link at the same time, then the key rollover is hitless (that is, the key rolls over without traffic interruption).

Fallback Key

A MACsec session can fail due to a key/key name (CKN) mismatch or a finite key duration between the switch and a peer. If a MACsec session does fail, a fallback session can take over if a fallback key is configured. A fallback session prevents downtime due to primary session failure and allows a user time to fix the key issue causing the failure. A fallback key also provides a backup session if the primary session fails to start. This feature is optional.

To configure a MACsec fallback key, see Configuring MACsec Fallback Key, on page 70.

Licensing Requirements for MACsec

Product	License Requirement
Cisco NX-OS	MACsec requires a Security license. For a complete explanation of the Cisco NX-OS licensing scheme to obtain and apply licenses, see the Cisco NX-OS Licensing Guide.

Guidelines and Limitations for MACsec

MACsec has the following guidelines and limitations:

- MACsec is supported on the following interface types:
  - Layer 2 switch ports (access and trunk)
  - Layer 3 routed interfaces (no subinterfaces)

Note Enabling MACsec on the Layer 3 routed interface also enables encryption on all the subinterfaces that are defined under that interface. However, selectively enabling MACsec on a subset of subinterfaces of the same Layer 3 routed interface is not supported.

  - Layer 2 and Layer 3-port channels (no subinterfaces)

- Beginning with Cisco Nexus Release 10.2(1)F, Secure Channel Identifier (SCI) can be disabled from MACSec security tag (SecTAG) on Cisco Nexus 9000 ToR switches.
  - It is supported in FX2 and FX3 platforms.
  - It is supported in FX platforms with XPN cipher suites only
- When the Cisco Nexus ToR switches are downgraded from Cisco NX-OS Release 9.3.7 to Cisco NX-OS Release 9.3.6 and below releases, MACsec is not supported.
- MKA is the only supported key exchange protocol for MACsec. The Security Association Protocol (SAP) is not supported.
- Link-level flow control (LLFC) and priority flow control (PFC) are not supported with MACsec.
- Multiple MACsec peers (different SCI values) for the same interface are not supported.
- You can retain the MACsec configuration when you disable MACsec using the **macsec shutdown** command.
- MACsec sessions are liberal in accepting packets from a key server whose latest Rx and latest Tx flags have been retired after Tx SA installation for the first time. The MACsec session then converges into a secure state.
- Beginning with Cisco NX-OS Release 9.2(1), the following configurations are allowed:
  - Allowing MACSec policy to be modified while the policy is referenced by an interface.
  - Allowing different MACsec policies across different lanes of a breakout port.
- Beginning with Cisco Nexus Release 9.2(1), MACsec is supported on Cisco Nexus 93180YC-FX and Cisco Nexus 3264C-E switches.
- Beginning with Cisco Nexus Release 9.3(1), MACsec is supported on the Cisco Nexus 9364C, 9332C, and 9348GC-FXP switches. The following limitations are applicable when you use MACsec with these switches:
  - Cisco Nexus C9364C—MACsec is supported on 16 ports (Ports 49–64).
  - Cisco Nexus C9332C—MACsec is supported on 8 ports (Ports 25–32).
  - Cisco Nexus 9348GC-FXP—MACsec is supported on 6 ports (Ports 49–54).

Note On the Cisco Nexus 9364C, and 9332C platform switches, when MACsec is either configured or unconfigured on a port, there will be a port-flap occurrence irrespective of MACsec security-policy type.

- Beginning with Cisco Nexus Release 9.3(1), you cannot apply MACsec configuration directly on port-channel interface. However, you can apply MACsec configurations directly on port-channel member ports. This applies to both NX-OS and vPC port-channels.
- Beginning with Cisco Nexus Release 9.3(3), MACsec is supported on Cisco Nexus 93216TC-FX2, Cisco Nexus 93360YC-FX2.
- Beginning with Cisco NX-OS Release 9.3(5), MACsec is supported on the following switches and line cards:
  - Cisco Nexus 93180YC-FX3S switches - MACsec is supported on all ports.
  - Cisco Nexus X9732C-FX, and X9788TC-FX line cards
- The N9K-X9736C-FX, N9K-X9732C-FX, N9K-C9348GC-FXP, N9K-C93180YC-FX, N9K-C93108TC-FX, N9K-X9788TC-FX, N9K-C9336C-FX2, N9K-C93240YC-FX2, N9K-C93216TC-FX2, N9K-C93360YC-FX2, N9K-C9364C, and N9K-C9332C cards and switches do not support MACsec on 1G ports. MACsec is not supported on any port on a mac block that has 1G ports on it.
- Beginning with Cisco NX-OS Release 10.1(1), the Cisco Nexus 93180YC-FX3, and 93108TC-FX3P switches support MACsec on all port speeds including 1G and 10G port speeds.
- MACsec is supported on Cisco Nexus 93240YC-FX2, 9336C-FX2, 93108TC-FX, 93180YC-FX switches and the X9736C-FX, and X9732C-EXM line cards.
- 1G is not supported on BV ports or retimer ports. For the retimer port details, see *Supported Retimer Ports* section of **Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide.**
- Cisco Nexus 9000 Series switches do not support MACsec on any of the MACsec capable ports when QSA is being used.
  - Beginning with Cisco NX-OS Release 9.3(7), MACsec is supported by Cisco Nexus 9364C and 9336C-FX2 switches when QSA is being used.
  - Beginning with Cisco NX-OS Release 10.1(1), MACsec is supported by Cisco Nexus 9336C-FX2, 9336C-FX2-E, and 9364C switches when QSA is being used.
  - Beginning with Cisco NX-OS Release 10.1(2), MACsec is supported by Cisco Nexus 9300-FX3 platform switches when QSA is being used.
- Beginning with Cisco Nexus Release 10.1(1), MACsec is supported on Cisco Nexus 9336C-FX2-E.
- Beginning with Cisco Nexus Release 10.2(1)F, MACsec is supported on Cisco Nexus X9716D-GX.
- Beginning with Cisco NX-OS Release 10.2(1q)F, MACsec is supported on ports 25-32 of Cisco Nexus 9332D-GX2B switches.
- Beginning with Cisco NX-OS Release 10.2(2)F, MACsec is supported on the Cisco Nexus N9K-C9348D-GX2A switches on 1-48 ports.

- Beginning with Cisco NX-OS Release 10.2(2)F, MACsec supports Cisco Nexus X9736C-FX, and X9736Q-FX line cards with 10G QSA links.
- Beginning with Cisco NX-OS Release 10.2(2)F, MACsec is supported on ports 1-16 of Cisco Nexus 9364D-GX2A switches.
- On the Cisco Nexus 9332D-GX2B, 9364D-GX2A and 9348D-GX2A switches and Cisco Nexus X9836DM-A line card, when MACsec is either configured or unconfigured on a port, a port-flap occurs irrespective of MACsec security-policy type.
- Beginning with Cisco NX-OS Release 10.3(1)F, MACsec is supported on Cisco Nexus X9836DM-A line card of Cisco Nexus 9800 platform switches.
- Beginning with Cisco NX-OS Release 10.3(2)F, MACsec is supported on Cisco Nexus 9408 switches with LEM modules X9400-16W and X9400-8D on all supported links.
- Beginning with Cisco Nexus Release 10.3(3)F, the cipher key enforcement feature provides the option to define the supported cipher suites from the most preferred to the least preferred on the Cisco Nexus 9332D-GX2B, 9336C-FX2, 93180YC-FX, and 93180YC-FX3 switches with following limitations:
  - Cipher Key Enforcement feature will work effectively only if it is prioritized as key server, else it will be in **init** or **pending** state of the session.
  - Cipher Key Enforcement feature is only supported for direct connections between 2 peers. If MKA session is with multiple peers, this feature is not expected to work properly.
  - During a peer cipher suite allowance change, session may not secure on the most preferred supported cipher suite.
  - When changing cipher from any to an enforced-peer cipher on a policy that is used on any secured MACsec session, it is recommended to flap the port after changing the cipher to reach expected behavior. If flapping is not done, the session shows secured on the switch while peer session shows pending on unsupported ciphers. It could also cause session to not get secured immediately even if supported cipher is present in enforce-peer cipher suites.
  - The Allowed Peer Cipher Suites (APSC) can not be empty or cannot have duplicates.
  - Cipher-suite and cipher-suite enforce-peer commands cannot coexist under the same policy.
  - While waiting for SAK Cipher-Enforcing Timer to timeout to try the next cipher suite in line, the data and control traffic might face one way traffic interruptions even with should secure mode. The interruption will only recover when session is secured.
- Beginning with Cisco Nexus Release 10.4(1)F, MACsec is supported on ports 49 to 54 of Cisco Nexus 9348GC-FX3 and 9348GC-FX3PH switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, MACsec is supported on all front panel ports (port 1 to 32) of Cisco Nexus 9332D-H2R platform switches. However, MACsec is not supported on Ethernet1/33 and Ethernet1/34.
- Beginning with Cisco Nexus Release 10.4(2)F, MACsec is supported on the below switches:
  - Cisco Nexus 93400LD-H1 on all ports.
  - Cisco Nexus 93108TC-FX3 on ports 49 to 54.

- Beginning with Cisco Nexus Release 10.4(3)F, MACsec is supported on the Cisco Nexus 9364C-H1 switches on ports 49 to 64.
- If the MACsec feature is configured, non-disruptive ISSU is not supported.

Keychain restrictions:

- You cannot overwrite the octet string for a MACsec key. Instead, you must create a new key or a new keychain.
- A new key in the keychain is configured when you enter **end** or **exit**. The default timeout for editor mode is 6 seconds. If the key is not configured with the key octet string or/and the send lifetime within the 6-second window, incomplete information may be used to bring up the MACsec session and could result in the session being stuck in an Authorization Pending state. If the MACsec sessions are not converged after the configuration is complete, you might be advised to shut/no shut the ports.
- For a given keychain, key activation times should overlap to avoid any period of time when no key is activated. If a time period occurs during which no key is activated, session negotiation fails and traffic drops can occur. The key with the latest start time among the currently active keys takes precedence for a MACsec key rollover.
- In addition to enabling the MACsec feature, a MACsec keychain must be configured on at least one interface to consume the security add-on license.

Fallback restrictions:

- If a MACsec session is secured on an old primary key, it does not go to a fallback session in case of mismatched latest active primary key. So the session remains secured on the old primary key and will show as rekeying on the old CA under status. And the MACsec session on the new key on primary PSK will be in init state.
- Use only one key with infinite lifetime in the fallback key chain. Multiple keys are not supported.
- The key ID (CKN) used in the fallback key chain must not match any of the key IDs (CKNs) used in the primary key chain.
- Once configured, fallback configuration on an interface cannot be removed, unless the complete MACsec configuration on the interface is removed.

MACsec policy restrictions:

- BPDU packets can be transmitted before a MACsec session becomes secure.

Layer 2 Tunneling Protocol (L2TP) restrictions:

- MACsec is not supported on ports configured for dot1q tunneling or L2TP.
- L2TP does not work if STP is enabled on trunk ports for non-native VLANs.

Statistics restrictions:

- Few CRC errors should occur during the transition between MACsec and non-MACsec mode (regular port shut/no shut).
- Secy statistics are cumulative and polled every 30 seconds.

- The IEEE8021-SECY-MIB OIDs secyRxSAStatsOKPkts, secyTxSAStatsProtectedPkts, and secyTxSAStatsEncryptedPkts can carry only up to 32 bits of counter values, but the traffic may exceed 32 bits.
- On Cisco Nexus 9300-FX3 platform switches, the **show macsec secy statistics** command supports rate statistics and the following rate related "CISCO-SECY-EXT-MIB" OIDs beginning with Cisco NX-OS Release 10.4(2)F.
  - cseSecyIfRxUncontrolledPktRate,
  - cseSecyIfRxControlledPktRate,
  - cseSecyIfTxUncontrolledPktRate,
  - cseSecyIfTxControlledPktRate
  - cseSecyIfRxControlledOctetRate
  - cseSecyIfTxControlledOctetRate
  - cseSecyIfRxUnControlledOctetRate
  - cseSecyIfTxUnControlledOctetRate

Interoperability restrictions:

- Interoperability of N9K-X9732C-EXM and other peer switches (other Cisco and non-Cisco switches) is supported only with the XPN cipher suite.
- MACsec peers must run the same Cisco NX-OS release in order to use the AES_128_CMAC cryptographic algorithm. For interoperability between previous releases and Cisco NX-OS Release 9.2(1), you must use keys with the AES_256_CMAC cryptographic algorithm.
- For interoperability between previous releases and Cisco NX-OS Release 9.2(1), pad the MACsec key with zeros if it is less than 32 octets.
- On any Cisco NX-OS switch, you can configure only one unique combination of an alternate MAC address and Ethernet type on all interfaces.
- When using 1G optics on MACSEC capable module, it is recommended to change diagnostics mode to 'minimal'.
- When you attempt to downgrade from Cisco NX-OS Release 9.3(1) to a Cisco NX-OS release without per port channel member MACsec configuration support, when the switch has MACsec configurations on members of the same port channel interface that are different from each other, you may see the following error message:

```
Asymmetric macsec config is present on port-channel members. Please use
symmetric macsec config across members to perform Non-disruptive ISSU.
```

- Software support for MACsec and 50G is not available on the Cisco Nexus X9400-22L LEM card.

EAPOL has the following guidelines and limitations:

- In Cisco NX-OS Release 9.3(1), EAPOL configuration is not supported on Cisco Nexus 9332C and 9364C Series switches.

- Within the same slice of the forwarding engine, EAPOL ethertype and dot1q ethertype cannot have the same value.
- For enabling EAPOL configuration, the range of ethernet type between 0 to 0x599 is invalid.
- For enabling EAPOL configuration, on N9K-X9836DM-A line card, the only supported EAPOL mac addresses are range 0x0180c2000000 to 0x0180c20000ff.
- While configuring EAPOL packets, the following combinations must not be used:
  - Mac address 0100.0ccd.cdd0 with any ethertype
  - Any mac address with Ether types: 0xfff0, 0x800, 0x86dd
  - The default destination MAC address, 0180.c200.0003 with the default Ethernet type, 0x888e
  - Different EAPOL DMAC addresses on both MACsec peers. The MACsec session works only if the MACsec peer is sending MKAPDUs with the DMAC configured locally.
- Beginning with Cisco NX-OS Release 10.2(1)F, EAPOL is supported on Cisco Nexus 9300-FX3 Series switches.

MACsec guidelines and limitations for Cisco Nexus N9324C-SE1U switches

- Beginning with Cisco Nexus Release 10.5(3s), Cisco Nexus N9324C-SE1U switches support MACsec features on all 24 front-panel ports.
- Silicon one switches do not support MACsec egress SA counters.
- On Cisco Nexus N9324C-SE1U, when the operational cipher-suite changes, there will be port flap occurance.
- The **no protocol lldp encrypted** command is not supported on N9323C-SE1U switches.

Enabling MACsec

Before you can access the MACsec and MKA commands, you must enable the MACsec feature.

SUMMARY STEPS

1. **configure terminal**
2. **feature macsec**
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	
Step 2	**feature macsec** **Example:** `switch(config)# feature macsec`	Enables MACsec and MKA on the device.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Disabling MACsec

Beginning with Cisco NX-OS Release 9.2(1), disabling the MACsec feature only deactivates this feature and does not remove the associated MACsec configurations.

Disabling MACsec has the following conditions:

- MACsec shutdown is global command and is not available at the interface level.
- The macsec shutdown, show macsec mka session/summary, show macsec mka session detail, and show macsec mka/secy statistics commands will display the 'Macsec is shutdown' message. However, the show macsec policy and show key chain commands will display the output.
- Consecutive MACsec status changes from macsec shutdown to no macsec shutdown and vice versa needs a 30 seconds time interval in between the status change.

SUMMARY STEPS

1. **configure terminal**
2. **macsec shutdown**
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**macsec shutdown** **Example:** `switch(config)# macsec shutdown`	Disables the MACsec configuration on the device. The **no** option restores the MACsec feature.

	Command or Action	Purpose
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration. This step is required only if you want to retain the MACsec in the shutdown state after the switch reload.

Configuring a MACsec Keychain and Keys

You can create a MACsec keychain and keys on the device.

Note Only MACsec keychains will result in converged MKA sessions.

Before you begin

Make sure that MACsec is enabled.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) [**no**] **key-chain macsec-psk no-show**
3. **key chain** *name* **macsec**
4. **key** *key-id*
5. **key-octet-string** *octet-string* **cryptographic-algorithm** {**AES_128_CMAC** | **AES_256_CMAC**}
6. **send-lifetime** *start-time* **duration** *duration*
7. (Optional) **show key chain** *name*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	(Optional) [**no**] **key-chain macsec-psk no-show** **Example:** `switch(config)# key-chain macsec-psk no-show`	Hides the encrypted key octet string in the output of the **show running-config** and **show startup-config** commands by replacing the string with a wildcard character. By default, PSK keys are displayed in encrypted format and can be easily decrypted. This command applies only to MACsec keychains.

	Command or Action	Purpose
		Note The octet string is also hidden when you save the configuration to a file.
Step 3	**key chain** *name* **macsec** **Example:** `switch(config)# key chain 1 macsec` `switch(config-macseckeychain)#`	Creates a MACsec keychain to hold a set of MACsec keys and enters MACsec keychain configuration mode.
Step 4	**key** *key-id* **Example:** `switch(config-macseckeychain)# key 1000` `switch(config-macseckeychain-macseckey)#`	Creates a MACsec key and enters MACsec key configuration mode. The range is from 1 to 32 octets, and the maximum size is 64. **Note** The key must consist of an even number of characters.
Step 5	**key-octet-string** *octet-string* **cryptographic-algorithm** {**AES_128_CMAC** | **AES_256_CMAC**} **Example:** `switch(config-macseckeychain-macseckey)# key-octet-string abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789 cryptographic-algorithm AES_256_CMAC`	Configures the octet string for the key. The *octet-string* argument can contain up to 64 hexadecimal characters. The octet key is encoded internally, so the key in clear text does not appear in the output of the **show running-config macsec** command. The key octet string includes the following: • 0 Encryption Type - No encryption (default) • 6 Encryption Type - Proprietary (Type-6 encrypted). For more information, see Enabling Type-6 Encryption on MACsec Keys. • 7 Encryption Type - Proprietary WORD key octet string with maximum 64 characters
Step 6	**send-lifetime** *start-time* **duration** *duration* **Example:** `switch(config-macseckeychain-macseckey)# send-lifetime 00:00:00 Oct 04 2016 duration 100000`	Configures a send lifetime for the key. By default, the device treats the start time as UTC. The *start-time* argument is the time of day and date that the key becomes active. The *duration* argument is the length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years).
Step 7	(Optional) **show key chain** *name* **Example:** `switch(config-macseckeychain-macseckey)# show key chain 1`	Displays the keychain configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config-macseckeychain-macseckey)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

MACsec Packet-Number Exhaustion

Every MACsec frame contains a 32-bit packet number (PN), and it is unique for a given Security Association Key (SAK). Upon PN exhaustion (after reaching 75% of 2^{32}- 1), SAK rekey takes place automatically to refresh the data plane keys and the PN will wrap around.

For example, on 10G full line rate @ 64 bytes, the SAK rekey will occur every 216 seconds due to PN exhaustion.

This is applicable when using GCM-AES-PN-128 or GCM-AES-PN-256 cipher-suites.

When GCM-AES-XPN-128 or GCM-AES-XPN-256 cipher-suite is used , the SAK rekey happens automatically when reaching 75% of 2^{64}- 1 , which will take several years to exhaust the packet numbering. The cipher-suite is configurable under the macsec policy and the operational cipher-suite is determined by the key-server device.

It is recommended to use XPN ciphersuite on N9K-X9732C-EXM line card

Configuring MACsec Fallback Key

Beginning with Cisco NX-OS Release 9.2(1), you can configure a fallback key on the device to initiate a backup session if the primary session fails as a result of a key/key name (CKN) mismatch or a finite key duration between the switch and peer.

Before you begin

Make sure that MACsec is enabled and a primary and fallback keychain and key ID are configured. See Configuring a MACsec Keychain and Keys.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *name*
3. **macsec keychain** *keychain-name* **policy** *policy-name* **fallback-keychain** *keychain-name*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters the global configuration mode.

	Command or Action	Purpose
Step 2	**interface** *name* **Example:** `switch(config)# interface ethernet 1/1` `switch(config-if)#`	Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use ethernet slot/port.
Step 3	**macsec keychain** *keychain-name* **policy** *policy-name* **fallback-keychain** *keychain-name* **Example:** `switch(config-if)# macsec keychain kc2 policy abc fallback-keychain fb_kc2`	Specifies the fallback keychain to use after a MACsec session failure due to a key/key ID mismatch or a key expiration. The fallback key ID should not match any key ID from a primary keychain. Fallback keychain configuration for each interface can be changed on the corresponding interface, without removing the MACsec configuration, by reissuing the same command with the fallback keychain name changed. **Note** The command must be entered exactly the same as the existing configuration command for the interface, except for the fallback keychain name. See Configuring a MACsec Keychain and Keys.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring a MACsec Policy

You can create multiple MACsec policies with different parameters. However, only one policy can be active on an interface.

Before you begin

Make sure that MACsec is enabled.

SUMMARY STEPS

1. **configure terminal**
2. **macsec policy** *name*
3. (Optional) **[no] cipher-suite** { { **enforce-peer** <*allowed-peer-cipher-suite1*> [*allowed-peer-cipher-suite2*> [*allowed-peer-cipher-suite3*> [*allowed-peer-cipher-suite4*>]]] } | **<suite>**}
4. (Optional) **[no] include-sci**
5. (Optional) **no protocol lldp encrypted**
6. (Optional) **key-server-priority** *number*
7. (Optional) **security-policy** *name*
8. (Optional) **window-size** *number*

9. (Optional) **sak-expiry-time** *time*
10. (Optional) **conf-offset** *name*
11. (Optional) **show macsec policy**
12. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**macsec policy** *name* **Example:** `switch(config)# macsec policy abc` `switch(config-macsec-policy)#`	Creates a MACsec policy.
Step 3	(Optional) **[no] cipher-suite** { { **enforce-peer** <*allowed-peer-cipher-suite1*> [*allowed-peer-cipher-suite2*> [*allowed-peer-cipher-suite3*> [*allowed-peer-cipher-suite4*>]]] } | **<suite>**} **Example:** `switch(config-macsec-policy)# cipher-suite enforce-peer GCM-AES-XPN-256 GCM-AES-XPN-128`	Configures the sequence for the following cipher suites from the most preferred to the least preferred. The session gets secured on the most preferred cipher suite that is supported by the peer.: GCM-AES-128, GCM-AES-256, GCM-AES-XPN-128, or GCM-AES-XPN-256. To unconfigure, you can either use the **no** form or overwrite the existing sequence with the required sequence preference. **Note** • For this feature to work, ensure that the Cisco NX-OS switch is set as key server. • If the peer supports cipher suites that are not included in the set of cipher suites defined in the **cipher-suite enforce-peer** command, the MKA session state will not be secured instead it will be pending.
Step 4	(Optional) **[no] include-sci** **Example:** `switch(config-macsec-policy)# no include-sci`	Disables SCI in SecTAG. By default, SCI is always enabled. **Note** To prevent packet drops, ensure that SCI tagging settings are consistent at both ingress and egress points.
Step 5	(Optional) **no protocol lldp encrypted** **Example:**	Permits LLDP packets, even when the MACsec configuration on a port with a must-secure policy is unsecured. Before Cisco NX-OS Release 10.5(3)F, LLDP

	Command or Action	Purpose
	`switch(config-macsec-policy)# no protocol lldp encrypted`	packets were dropped if the MACsec configuration on a port with a must-secure policy was not in a secured state. **Note** This command is supported only on Cisco Nexus 9300-GX2, H2R, H1 Series switches.
Step 6	(Optional) **key-server-priority** *number* **Example:** `switch(config-macsec-policy)# key-server-priority 0`	Configures the key server priority to break the tie between peers during a key exchange. The range is from 0 (highest) and 255 (lowest), and the default value is 16.
Step 7	(Optional) **security-policy** *name* **Example:** `switch(config-macsec-policy)# security-policy should-secure`	Configures one of the following security policies to define the handling of data and control packets: • **must-secure**—Packets not carrying MACsec headers will be dropped. • **should-secure**—Packets not carrying MACsec headers will be permitted. This is the default value.
Step 8	(Optional) **window-size** *number* **Example:** `switch(config-macsec-policy)# window-size 512`	Configures the replay protection window such that the secured interface will not accept any packet that is less than the configured window size. The range is from 0 to 596000000.
Step 9	(Optional) **sak-expiry-time** *time* **Example:** `switch(config-macsec-policy)# sak-expiry-time 100`	Configures the time in seconds to force an SAK rekey. This command can be used to change the session key to a predictable time interval. The default is 0. **Note** Prior to the 10.5(3)F release, the minimum time for SAK expiry was 60 seconds. Starting with the 10.5(3)F release, a minimum time of 30 seconds is supported.
Step 10	(Optional) **conf-offset** *name* **Example:** `switch(config-macsec-policy)# conf-offset CONF-OFFSET-0`	Configures one of the following confidentiality offsets in the Layer 2 frame, where encryption begins: CONF-OFFSET-0, CONF-OFFSET-30, or CONF-OFFSET-50. This command might be necessary for intermediate switches to use packet headers {dmac, smac, etype} like MPLS tags.
Step 11	(Optional) **show macsec policy** **Example:** `switch(config-macsec-policy)# show macsec policy`	Displays the MACsec policy configuration.
Step 12	(Optional) **copy running-config startup-config** **Example:** `switch(config-macsec-policy)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring MACsec EAP

Beginning with Cisco NX-OS Release 10.4(1)F, you can use MACsec EAP profile for 802.1X authentication.

Note MACsec EAP is not supported on Cisco 9800 Series switches and N9324C-SE1U switches.

Before you begin

- Enable the 802.1X feature on the Cisco NX-OS device.
- Configure MACsec command which specifies should-secure (default) or must-secure macsec policy

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot / port*
3. **[no] macsec eap policy** *policy name*
4. **[no] dot1x supplicant eap profile** *eap profile name* }

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**interface ethernet** *slot / port* **Example:** `switch(config)# interface ethernet 1/30` `switch(config-if)#`	Selects the interface to configure and enters interface configuration mode.
Step 3	**[no] macsec eap policy** *policy name* **Example:** `switch(config-if)# macsec eap policy P1` `switch(config-eap-profile)#`	Creates the MACsec eap profile. The **no** form of the command is used to disable the MACsec eap profile.
Step 4	**[no] dot1x supplicant eap profile** *eap profile name* } **Example:** `switch(config-if)# dot1x supplicant eap profile`	Enters global configuration mode. Configures the eap profile to be used by the supplicant.

QKD integration with SKIP on MACsec

About Quantum-Safe Encryption

Recent advancements in quantum computing have exposed vulnerabilities in various cryptographic algorithms, making them unsecured for future applications. The RSA (integer factorization) and DHE (discrete logarithms) public key algorithms, which rely on computational complexity, are now at risk of being solved by quantum computers using Shor's or Grover's algorithm.

As a result, establishing a shared secret key between communicating parties has become a significant challenge. To avoid this issue, you can configure quantum-safe algorithms or implement a Quantum Key Distribution (QKD).

About QKD Integration with Secure Key Integration Protocol

Integrating Secure Key Integration Protocol (SKIP) protocol to the switches empowers to establish communication with external quantum devices. This advancement allows for the utilization of Quantum Key Distribution (QKD) devices in the exchange of MACsec encryption keys between switches.

QKD operates on the principles of quantum physics, utilizing the quantum state of photons to encode and share information through an optical link. Additionally, an authenticated classical channel is used for sharing measurements. The change in quantum states helps the two end parties of the communication channel to identify any interception of their key.

QKD is a secured key exchange mechanism against quantum attacks even in the future advancements in cryptanalysis or quantum computing. QKD doesn’t require continual updates based on discovered vulnerabilities.

Postquantum Preshared Keys (PPK)

Session keys that are based on preshared keys are not vulnerable to quantum attacks if the preshared keys have sufficient entropy and the pseudorandom function (PRF), encryption, and authentication transformations are quantum secure. The resulting system is then believed to be secure against classical attackers of today or future attackers with a quantum computer.

Guidelines and Limitations

The integration of QKD with SKIP for MACsec communication has the following guidelines and limitations:

- Beginning with Cisco NX-OS Release 10.4(3)F, Secure Key Integration Protocol is supported on the following Cisco Nexus switches:
  - N9K-C9348GC-FXP
  - N9K-C93216TC-FX2
  - N9K-C93360YC-FX2
  - N9K-C9336C-FX2
  - N9K-C9348GC-FX3

  - N9K-C9348D-GX2A
  - N9K-C9332D-H2R
- You can use the SKIP protocol only in a point-to-point MACsec link encryption scenario.
- The SKIP protocol is available only on the interfaces that support MACsec encryption.
- Ensure that the QKD server is accessible through the management interface if switches have an HTTPS connection that is established with it.
- If MACsec peers are connected to two different QKD servers, the QKD servers synchronize the keys to establish an MKA session. This synchronization ensures that the MACsec key (CKN) and key-string (CAK) are the same at both ends.
- To establish a secure Transport Layer Security (TLS) connection and enable mutual authentication, you must install trustpoint certificates on the switch. These certificates allow the switch to obtain keys from the server. For more information, see chapter Configuring PKI in *Cisco Nexus 9000 Series NX-OS Security Configuration Guide.*
- MACsec PPK session and EAP-TLS sessions are not supported on the same interface.
- A switch can connect to only one QKD server and one QKD profile per switch.
- For SKIP protocol, only a single remoteSystemID is supported.
- For QKD connectivity, IPv6 is not supported.
- MACsec peers exchanging QKD Keys must be Cisco NX-OS switches.
- Once the MACsec session is established, any modifications to the QKD profile will cause traffic loss in **MUST SECURE MACSEC** mode.
- Once the MACsec session is up with QKD keys derived from KME Server, trustpoint modifications that are part of the QKD profile will not have any effect on the current sessions.
- The remoteSystemID attribute is mandatory for the capability response.
- Beginning with Cisco NX-OS Release 10.5(2)F, QKD MACsec fallback to PSK support is provided to establish a secured MKA session when the primary PPK fails with the following guidelines and limitations:
  - When PPK is configured ensure that the PSK is configured first.
  - PPK mode does not support the keychain fallback method for PSK.
  - While MACsec sessions are secured in PPK mode, if the QKD server goes down or is disconnected, or if the cryptopqc feature is removed, the current PPK sessions and keys will still be maintained and used until the next re-key event is triggered, such as SAK-expiry timeout or PN-exhaustion. After that, PPK sessions will fall back to secured PSK mode. However, if the PPK crypto-QKD profile is removed from the MACsec policy, PPK sessions will immediately fall back to PSK mode.

QKD guidelines and limitations for Cisco Nexus N9324C-SE1U switches

- Beginning with Cisco NX-OS Release 10.5(3s), Secure Key Integration Protocol is supported on the Cisco Nexus N9324C-SE1U switches.

Configuring point-to-point MACsec Link Encryption Using SKIP

In point-to-point MACsec Link Encryption, secure encryption is established by using SKIP in switches. This encryption is set up between two interfaces in peer switches and requires the assistance of a QKD device network. Instead of the switches network, the QKD network shares the MACsec encryption key. Therefore, when a switch is required to create a MACsec link between peer switch interfaces, it contacts the external QKD device and requests the key. The external QKD device then generates a key pair consisting of the key ID and the key.

The Key ID acts as the unique ID string for the key (Shared Secret). The QKD device shares both the key ID and Key with the switch, while the switch only shares the key ID with its peer. The Peer switch uses this Key ID to retrieve encryption keys from its QKD device. Hence, Quantum networks always securely communicate encryption keys.

Enabling Postquantum Cryptography

Before you begin

- Configure MACsec Pre-Shared Key (PSK).
- Configure MACsec in the PPK mode.
- An external QKD device network.
- Add the QKD server CA to the trustpoint in the switch and import the QKD server root CA certificate to the switch.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature cryptopqc**
3. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	switch(config)# **feature cryptopqc** **Example:** `switch(config)# feature cryptopqc`	Enables post quantum cryptography (cryptopqc) on the switch.

	Command or Action	Purpose
Step 3	(Optional) switch(config)# **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Enabling MACsec and MKA features

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature macsec**
3. (Optional) switch(config)# copy running-config startup-config

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	switch(config)# **feature macsec** **Example:** `switch(config)# feature macsec`	Enables MACsec and MKA on the switch.
Step 3	(Optional) switch(config)# copy running-config startup-config **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Configuring Quantum Key Distribution Profile

SUMMARY STEPS

1. switch (config)# **crypto qkd profile***name*
2. switch (config)# **kme server**<*hostname*/*IP*> **port** *portnumber*
3. switch(config)#**transport tls authentication-type trustpoint**<*trustpoint name*>
4. (Optional) switch(config)#**copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch (config)# **crypto qkd profile***name* **Example:** `switch(config)# crypto qkd profile ppk1`	Creates a QKD profile with name ppk1.
Step 2	switch (config)# **kme server**<*hostname*/*IP*> **port** *portnumber* **Example:** `switch(config-crypto-qkd-profile)# kme server 172.0.0.2 port 6000`	Configures Key Management Engine (KME)/QKD server IP and TCP port number. **Note** Port number is optional. By default, port number will be 443.
Step 3	switch(config)#**transport tls authentication-type trustpoint**<*trustpoint name*> **Example:** `switch(config-crypto-qkd-profile)# transport tls authentication-type trustpoint tp1`	Configures the CA (Certificate Authority) trustpoint. To create a trustpoint, refer to Configuring PKI section.
Step 4	(Optional) switch(config)#**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Enabling MACsec and MKA features

SUMMARY STEPS

1. switch(config)# **macsec policy**<*name*>
2. switch(config)# **ppk crypto-qkd-profile**<*name*>
3. (Optional) switch(config)# copy running-config startup-config

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch(config)# **macsec policy**<*name*> **Example:** `switch(config)# [no] macsec policy test-policy`	Creates a MACsec policy.

	Command or Action	Purpose
Step 2	switch(config)# **ppk crypto-qkd-profile**<*name*> **Example:** `switch(config-macsec-policy)# [no] ppk crypto-qkd-profile ppk1`	Configures the PPK profile name.
Step 3	(Optional) switch(config)# copy running-config startup-config **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Configuration Examples

The following examples shows configuration of QKD profile and display of the configured details:

- Configuring QKD profile

```
switch(config)# feature cryptopqc
switch(config)#
switch(config)# crypto qkd profile ppk1
switch(config-crypto-qkd-profile)# kme server 168.20.1.2 port 5000
switch(config-crypto-qkd-profile)# transport tls authentication-type trustpoint tp1
switch(config-crypto-qkd-profile)# end
switch#
```

- Displaying QKD configuration

```
switch# show running-config cryptopqc
!Command: show running-config cryptopqc
!Running configuration last done at: Mon Jan 29 22:19:16 2024
!Time: Mon Jan 29 22:19:35 2024
version 10.4(3) Bios:version 05.51
feature cryptopqc
crypto qkd profile ppk1
kme server 168.20.1.2 port 5000
transport tls authentication-type trustpoint tp1
switch#
```

The following examples shows configuration of PPK profile on MACsec policy and display of the configured details:

- Configuring PPK profile on MACsec policy

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# macsec policy test
switch(config-macsec-policy) # ppk crypto-qkd-profile ppk1
switch(config-macsec-policy)# sak-expiry-time 1800
switch(config-macsec-policy)# exit
switch(config)# end
```

- Display of configured MACsec policy

```
switch# show macsec policy test
MACSec Cipher     Pri Window Offset Security     SAKRekey   timeICV     Policy Indicator
     Include-SCI
--------------------------------- ---------------- ---- ------------ --------
-------------- -------------- ------------- ------------
```

```
test   GCM-AES-XPN-256 16     148809600 0         should-secure 1800         FALSE
            TRUE
MACSec Policy     PPK Crypto-QKD-Profile Name
-------------------------------- --------------------------------
test                ppk1
switch#
```

The following example shows configuration of key chain, MACsec policy on an interface, and display of configured details:

- Configuring key chain

```
switch(config)# key chain KC1 macsec
switch(config-macseckeychain)#key 10100000
switch(config-macseckeychain-macseckey)#key-octet-string
F123456789ABCDEF0123456789ABCDEFF123456789ABCDEF0123456789ABCDEF cryptographic-algorithm
 AES_256_CMAC
switch(config-macseckeychain-macseckey)#exit
```

- Configuring MACsec policy to an interface

```
switch(config)# interface Ethernet 1/21
switch(config-if)# macsec keychain KC1 policy test
```

- Displaying MACsec session

```
switch(config)# show macsec mka session
Interface     Local-TxSCI #     Peers     Status     Key-Server Auth Mode
----------------- -------------------------------- ----------------- -----------------
 ------------------ ------------------
Ethernet1/21  6cb2.ae9f.e766/0001 1     Secured         No     PRIMARY-PPK
```

The following examples show configuring point-to-point MACsec QKD profile, binding QKD profile to MACsec policy and binding the MACsec policy to the interface:

Note Make sure that the KME1 and KME2 servers must be active for connection through management port.

Configuring Swtich 1

```
switch1# configure terminal
switch1(config)# crypto ca trustpoint tp1
switch1config-trustpoint)# end
switch1#

switch1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch1(config)# feature cryptopqc
switch1(config)#
switch1(config)# crypto qkd profile PPK1
switch1(config-crypto-qkd-profile)# kme server KME1 port 7010
switch1(config-crypto-qkd-profile)# transport tls authentication-type trustpoint tp1
switch1(config-crypto-qkd-profile)# end
switch1#

switch1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch1(config)# feature macsec
switch1(config)#

switch1(config)# macsec policy MP1
```

```
switch1(config-macsec-policy)# ppk crypto-qkd-profile PPK1
switch1(config-macsec-policy)#exit
switch1(config-if)# interface Ethernet1/21
switch1(config-if)# macsec keychain KC1 policy MP1
switch1(config-if)#

switch1(config-if)# interface Ethernet1/22
switch1(config-if)# macsec keychain KC1 policy MP1
switch1(config-if)#
switch1(config-if)# end
switch1#
```

Configuring Swtich 2

```
switch2# configure terminal
switch2(config)# crypto ca trustpoint tp1
switch2config-trustpoint)# end
switch2#

switch2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch2(config)# feature cryptopqc
switch2(config)#
switch2(config)# crypto qkd profile PPK1
switch2(config-crypto-qkd-profile)# kme server KME2 port 7010
switch2(config-crypto-qkd-profile)# transport tls authentication-type trustpoint tp1
switch2(config-crypto-qkd-profile)#
switch2(config-crypto-qkd-profile)# end
switch2#

switch2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch2(config)# feature macsec
switch2(config)#

switch2(config)# macsec policy MP1
switch2(config-macsec-policy)# ppk crypto-qkd-profile PPK1
switch2(config-macsec-policy)# exit

switch2(config-if)# interface Ethernet1/53
switch2(config-if)# macsec keychain KC1 policy MP1
switch2(config-if)#
switch2(config-if)# interface Ethernet1/54
switch2(config-if)# macsec keychain KC1 policy MP1
switch2(config-if)# end
switch2#
```

The following shows the output of configuration on Switch 1 and Switch 2:

Switch 1:

```
switch1#
switch1# show macsec mka session
Interface     Local-TxSCI #     Peers     Status     Key-Server     Auth Mode
----------------- -------------------------------- ------------------ ------------------
------------------ ------------------
Ethernet1/22   3c8b.7ffe.0244/0001 1     Secured         Yes        PRIMARY-PPK
Ethernet1/21   3c8b.7ffe.0240/0001 1     Secured         Yes        PRIMARY-PPK
N9K3K STANDARD TEMPLATE FOR FEATURE REVIEWS
----------------- -------------------------------- ------------------ ------------------
------------------ ------------------
Total Number of Sessions : 2
Secured Sessions : 2
```

```
Pending Sessions : 0
switch1#
```

Switch 2:

```
switch2#
switch2# show macsec mka session
Interface     Local-TxSCI #     Peers     Status     Key-Server     Auth Mode
------------------ -------------------------------- ------------------ ------------------
------------------ ------------------
Ethernet1/53  5451.deb8.62b4/0001 1         Secured     No          PRIMARY-PPK
Ethernet1/54  5451.deb8.62b8/0001 1         Secured     No          PRIMARY-PPK
------------------ -------------------------------- ------------------ ------------------
------------------ ------------------
Total Number of Sessions : 2
Secured Sessions : 2
Pending Sessions : 0
switch2#
```

About Configurable EAPOL Destination and Ethernet Type

Beginning Cisco NX-OS Release 9.2(2), Cisco enables networks with WAN MACsec to change the Extensible Authentication Protocol (EAP) over LAN (EAPOL) protocol destination address, and the Ethernet type values to nonstandard values.

Configurable EAPOL MAC and Ethernet type provides you the ability to change the MAC address and the Ethernet type of the MKA packet, inorder to allow CE device to form MKA sessions over the ethernet networks that consume the standard MKA packets.

The EAPOL destination Ethernet type can be changed from the default Ethernet type of 0x888E to an alternate value or, the EAPOL destination MAC address can be changed from the default DMAC of 01:80:C2:00:00:03 to an alternate value, to avoid being consumed by a provider bridge.

This feature is available at the interface level and the alternate EAPOL configuration can be changed on any interface at any given time as follows:

- If the MACsec is already configured on an interface, the sessions will come up with a new alternate EAPOL configuration.
- When MACsec is not configured on an interface, the EAPOL configuration is applied to the interface and is effective when MACsec is configured on that inferface.

Enabling EAPOL Configuration

You can enable the EAPOL configuration on any available interface.

Before you begin

Make sure that MACsec is enabled.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *name*
3. **eapol mac-address** *mac_address* [**ethertype** *eth_type*]

4. **eapol mac-address broadcast-address** [**ethertype** *eth_type*]
5. (Optional) **copy running-config startup-config**
6. **show macsec mka session detail**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**interface** *name* **Example:** `switch(config)# interface ethernet 1/1` `switch(config-if)#`	Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use ethernet slot/port.
Step 3	**eapol mac-address** *mac_address* [**ethertype** *eth_type*]	Enables the EAPOL configuration on the specified interface type and identity. **Note** If the ethernet type is not specified, the default ethernet type of MKA packets, which is 0x888e, is considered.
Step 4	**eapol mac-address broadcast-address** [**ethertype** *eth_type*]	Enables the broadcast address as the alternate mac address.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-macseckeychain-macseckey)# copy running-config startup-config`	Copies the running configuration to the startup configuration.
Step 6	**show macsec mka session detail**	Displays the EAPOL settings.

Disabling EAPOL Configuration

You can disable the EAPOL configuration on any available interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *name*
3. [**no**] **eapol mac-address** *mac_address* [**ethertype** *eth_type*]
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**interface** *name* **Example:** `switch(config)# interface ethernet 1/1` `switch(config-if)#`	Specifies the interface that you are configuring. You can specify the interface type and identity. For an Ethernet port, use ethernet slot/port.
Step 3	[**no**] **eapol mac-address** *mac_address* [**ethertype** *eth_type*]	Disables the EAPOL configuration on the specified interface type and identity.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-macseckeychain-macseckey)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the MACsec Configuration

To display MACsec configuration information, perform one of the following tasks:

Command	Purpose
show key chain *name*	Displays the keychain configuration.
show macsec mka session [**interface** *type slot*/*port*] [**detail**]	Displays information about the MACsec MKA session for a specific interface or for all interfaces.
show macsec mka session details	Displays information about the MAC address and the ethernet type that is currently used by the interfaces for all EAPOL packets.
show macsec mka summary	Displays the MACsec MKA configuration.
show macsec policy [*policy-name*]	Displays the configuration for a specific MACsec policy or for all MACsec policies.
show running-config macsec	Displays the running configuration information for MACsec.

The following example displays information about the MACsec MKA session for all interfaces. .

```
switch# show macsec mka session
Interface          Local-TxSCI                     #Peers            Status
Key-Server         Auth Mode
------------------ ------------------------------- ------------------ -----------------
```

```
------------------ ------------------
Ethernet2/2        2c33.11b8.7d14/0001             1                  Secured
Yes                PRIMARY-PSK
Ethernet2/3        2c33.11b8.7d18/0001             1                  Secured
Yes                PRIMARY-PSK
------------------ ------------------------------- ------------------ ----------------
------------------ ------------------
Total Number of Sessions : 2
        Secured Sessions : 2
        Pending Sessions : 0
```

The following example displays information about the MACsec MKA session for a specific interface. In addition to the common elements of the table as described in the previous example, the following also identifies the authentication mode which defines the current MACsec session type.

```
switch# show macsec mka session interface ethernet 1/1

Interface      Local-TxSCI          # Peers   Status        Key-Server     Auth Mode
-------------- -------------------- --------- ------------- -------------- -------------
Ethernet1/1    70df.2fdc.baf4/0001      0     Pending       Yes            PRIMARY-PSK
Ethernet1/1    70df.2fdc.baf4/0001      1     Secured       No             FALLBACK-PSK
```

The following example displays detailed information about the MACsec MKA session for a specific Ethernet interface:

```
Interface Name          : Ethernet2/2
    Session Status                    : SECURED - Secured MKA Session with MACsec
    Local Tx-SCI                      : 2c33.11b8.7d14/0001
    Local Tx-SSCI                     : 2
    MKA Port Identifier               : 2
    CAK Name (CKN)                    : 12
    CA Authentication Mode            : PRIMARY-PSK
    Member Identifier (MI)            : B54263EF7949A561E25CE617
    Message Number (MN)               : 523
    MKA Policy Name                   : tests2
    Key Server Priority               : 16
    Key Server                        : Yes
    Include ICV                       : No
    SAK Cipher Suite                  : GCM-AES-XPN-256
    SAK Cipher Suite (Operational)    : GCM-AES-XPN-256
    Replay Window Size                : 148809600
    Confidentiality Offset            : CONF-OFFSET-0
    Confidentiality Offset (Operational): CONF-OFFSET-0
    Latest SAK Status                 : Rx & TX
    Latest SAK AN                     : 0
    Latest SAK KI                     : B54263EF7949A561E25CE61700000001
    Latest SAK KN                     : 1
    Last SAK key time                 : 12:59:38 PST Tue Mar 19 2019
    CA Peer Count                     : 1
    Eapol dest mac                    : 0180.c200.0003
    Ether-type                        : 0x888e
Peer Status:
    Peer MI                           : 2C2C090E62A96F4D6E018210
    RxSCI                             : 2c33.11b8.8b88/0001
    Peer CAK                          : Match
    Latest Rx MKPDU                   : 13:16:54 PST Tue Mar 19 2019
```

The following example displays the MACsec MKA configuration:

```
switch# show macsec mka summary
Interface          MACSEC-policy                   Keychain
------------------ ------------------------------- ------------------------
Ethernet2/13       1                               1/10000000000000000
```

```
Ethernet2/14       1                                1/10000000000000000
```

The following example displays the configuration for all MACsec policies:

```
switch# show macsec policy
MACSec Policy        Cipher          Pri  Window     Offset   Security   SAK Rekey time ICV
 Indicator  Include-SCI
------------------ -------------     --- --------   --------  --------  --------------
------------ -------------
KC256-Po117b        GCM-AES-256      16   148809600    0    should-secure  pn-rollover
FALSE        True
pol1                GCM-AES-XPN-256  100  148809600    30   must-secure    60
FALSE        True
pol256-FanO         GCM-AES-XPN-256  16   148809600    0    must-secure    60
FALSE        True
pol256-MCT          GCM-AES-XPN-256  16   148809600    0    should-secure  60
FALSE        FALSE
system-default-
macsec-policy       GCM-AES-XPN-256  16   148809600    0    should-secure  pn-rollover
FALSE        FALSE
test1               GCM-AES-XPN-256  16   148809600    0    should-secure  pn-rollover
FALSE        True
```

The following example displays the key octet string in the output of the **show running-config** and **show startup-config** commands when the **key-chain macsec-psk no-show** command is not configured:

```
key chain KC256-1 macsec
  key 2000
    key-octet-string 7 075e701e1c5a4a5143475e5a527d7c7c706a6c724306170103555a5c57510b051e47080
a05000101005e0e50510f005c4b5f5d0b5b070e234e4d0a1d0112175b5e cryptographic-algorithm
AES_256_CMAC
```

The following example displays the key octet string in the output of the **show running-config** and **show startup-config** commands when the **key-chain macsec-psk no-show** command is configured:

```
key chain KC256-1 macsec
  key 2000
    key-octet-string 7 ****** cryptographic-algorithm AES_256_CMAC
```

Displaying MACsec Statistics

You can display MACsec statistics using the following commands.

Command	Description
show macsec mka statistics [**interface** *type slot*/*port*]	Displays MACsec MKA statistics.
show macsec secy statistics [**interface** *type slot*/*port*]	Displays MACsec security statistics.

The following example shows the MACsec MKA statistics for a specific Ethernet interface:

```
switch# show macsec mka statistics interface ethernet 2/2

Per-CA MKA Statistics for Session on interface (Ethernet2/2) with CKN 0x10
==============================================================================
CA Statistics
   Pairwise CAK Rekeys..... 0
```

```
SA Statistics
   SAKs Generated..... 0
   SAKs Rekeyed..... 0
   SAKs Received..... 0
   SAK Responses Received.. 0

MKPDU Statistics
   MKPDUs Transmitted...... 1096
      "Distributed SAK".. 0

   MKPDUs Validated & Rx... 0
      "Distributed SAK".. 0

MKA Statistics for Session on interface (Ethernet2/2)
=====================================================
CA Statistics
   Pairwise CAK Rekeys..... 0

SA Statistics
   SAKs Generated..... 0
   SAKs Rekeyed..... 0
   SAKs Received..... 0
   SAK Responses Received.. 0

MKPDU Statistics
   MKPDUs Transmitted...... 1096
      "Distributed SAK".. 0
   MKPDUs Validated & Rx... 0
      "Distributed SAK".. 0
   MKPDUs Tx Success..... 1096
   MKPDUs Tx Fail..... 0
   MKPDUS Tx Pkt build fail... 0
   MKPDUS No Tx on intf down.. 0
   MKPDUS No Rx on intf down.. 0
   MKPDUs Rx CA Not found..... 0
   MKPDUs Rx Error..... 0
   MKPDUs Rx Success..... 0

MKPDU Failures
   MKPDU Rx Validation ..... 0
   MKPDU Rx Bad Peer MN..... 0
   MKPDU Rx Non-recent Peerlist MN..... 0
   MKPDU Rx Drop SAKUSE, KN mismatch..... 0
   MKPDU Rx Drop SAKUSE, Rx Not Set..... 0
   MKPDU Rx Drop SAKUSE, Key MI mismatch.... 0
   MKPDU Rx Drop SAKUSE, AN Not in Use..... 0
   MKPDU Rx Drop SAKUSE, KS Rx/Tx Not Set... 0
   MKPDU Rx Drop Packet, Ethertype Mismatch. 0

SAK Failures
   SAK Generation..... 0
   Hash Key Generation..... 0
   SAK Encryption/Wrap..... 0
   SAK Decryption/Unwrap..... 0

CA Failures
   ICK Derivation..... 0
   KEK Derivation..... 0
   Invalid Peer MACsec Capability... 0

MACsec Failures
   Rx SA Installation..... 0
   Tx SA Installation..... 0
```

The following example shows the MACsec security statistics for a specific Ethernet interface.

Note The following differences exist for uncontrolled and controlled packets in Rx and Tx statistics:

- Rx statistics:
  - Uncontrolled = Encrypted and unencrypted
  - Controlled = Decrypted
- Tx statistics:
  - Uncontrolled = Unencrypted
  - Controlled = Encrypted
  - Common = Encrypted and unencrypted

```
switch(config)# show macsec secy statistics interface e2/28/1

Interface Ethernet2/28/1 MACSEC SecY Statistics:
-----------------------------------------
Interface Rx Statistics:
   Unicast Uncontrolled Pkts: 14987
   Multicast Uncontrolled Pkts: 1190444
   Broadcast Uncontrolled Pkts: 4
   Uncontrolled Pkts - Rx Drop: 0
   Uncontrolled Pkts - Rx Error: 0
   Unicast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Multicast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Broadcast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Controlled Pkts: 247583
   Controlled Pkts - Rx Drop: N/A (N9K-X9736C-FX not supported)
   Controlled Pkts - Rx Error: N/A (N9K-X9736C-FX not supported)
   In-Octets Uncontrolled: 169853963 bytes
   In-Octets Controlled: 55027017 bytes
   Input rate for Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Input rate for Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Input rate for Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Input rate for Controlled Pkts: N/A (N9K-X9736C-FX not supported)

Interface Tx Statistics:
   Unicast Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Multicast Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Broadcast Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Uncontrolled Pkts - Rx Drop: N/A (N9K-X9736C-FX not supported)
   Uncontrolled Pkts - Rx Error: N/A (N9K-X9736C-FX not supported)
   Unicast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Multicast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Broadcast Controlled Pkts: N/A (N9K-X9736C-FX not supported)
   Controlled Pkts: 205429
   Controlled Pkts - Rx Drop: N/A (N9K-X9736C-FX not supported)
   Controlled Pkts - Rx Error: N/A (N9K-X9736C-FX not supported)
   Out-Octets Uncontrolled: N/A (N9K-X9736C-FX not supported)
   Out-Octets Controlled: 20612648 bytes
   Out-Octets Common: 151787484 bytes
   Output rate for Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Output rate for Uncontrolled Pkts: N/A (N9K-X9736C-FX not supported)
   Output rate for Controlled Pkts: N/A (N9K-X9736C-FX not supported)
```

```
   Output rate for Controlled Pkts: N/A (N9K-X9736C-FX not supported)

SECY Rx Statistics:
  Transform Error Pkts: N/A (N9K-X9736C-FX not supported)
   Control Pkts: 952284
   Untagged Pkts: N/A (N9K-X9736C-FX not supported)
   No Tag Pkts: 0
   Bad Tag Pkts: 0
   No SCI Pkts: 0
   Unknown SCI Pkts: 0
   Tagged Control Pkts: N/A (N9K-X9736C-FX not supported)

SECY Tx Statistics:
   Transform Error Pkts: N/A (N9K-X9736C-FX not supported)
   Control Pkts: 967904
   Untagged Pkts: N/A (N9K-X9736C-FX not supported)

SAK Rx Statistics for AN [3]:
   Unchecked Pkts: 0
   Delayed Pkts: 0
   Late Pkts: 0
   OK Pkts: 1
   Invalid Pkts: 0
   Not Valid Pkts: 0
   Not-Using-SA Pkts: 0
   Unused-SA Pkts: 0
   Decrypted In-Octets: 235 bytes
   Validated In-Octets: 0 bytes

SAK Tx Statistics for AN [3]:
   Encrypted Protected Pkts: 2
   Too Long Pkts: N/A (N9K-X9736C-FX not supported)
   SA-not-in-use Pkts: N/A (N9K-X9736C-FX not supported)
   Encrypted Protected Out-Octets: 334 bytes
switch(config)#
```

Configuration Example for MACsec

The following example shows how to configure a user-defined MACsec policy and then apply the policy to interfaces:

```
switch(config)# macsec policy 1
switch(config-macsec-policy)# cipher-suite GCM-AES-256
switch(config-macsec-policy)# window-size 512
switch(config-macsec-policy)# key-server-priority 0
switch(config-macsec-policy)# conf-offset CONF-OFFSET-0
switch(config-macsec-policy)# security-policy should-secure
switch(config-macsec-policy)# exit

switch(config)# int e2/13-14
switch(config-if-range)# macsec keychain 1 policy 1
switch(config-if-range)# exit
switch(config)# show macsec mka summary
Interface          MACSEC-policy                    Keychain
------------------ -------------------------------- ------------------------
Ethernet2/13       1                                1/10000000000000000
Ethernet2/14       1                                1/10000000000000000

switch(config)# show macsec mka session
Interface      Local-TxSCI          # Peers    Status     Key-Server
-------------- -------------------- ---------- ---------- -------------
Ethernet2/13   006b.f1be.d31c/0001  1          Secured    Yes
```

```
Ethernet2/14   006b.f1be.d320/0001  1          Secured    No

switch(config)# show running-config macsec
!Command: show running-config macsec
!Time: Mon Dec  5 04:53:40 2016

version 9.2(1)feature macsec
macsec policy 1
  cipher-suite GCM-AES-256
  key-server-priority 0
  window-size 512
  conf-offset CONF-OFFSET-0
  security-policy should-secure

interface Ethernet2/13
  macsec keychain 1 policy 1

interface Ethernet2/14
  macsec keychain 1 policy 1
```

The following example shows how to configure a MACsec keychain and then add the system default MACsec policy to the interfaces:

```
switch(config)# key chain 1 macsec
switch(config-macseckeychain)# key 1000
switch(config-macseckeychain-macseckey)# key-octet-string
abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789 cryptographic-algorithm
aes_256_CMAC
switch(config-macseckeychain-macseckey)# exit

switch(config)# int e2/13-14
switch(config-if-range)# macsec keychain 1
switch(config-if-range)# exit
switch(config)#

switch(config)# show running-config macsec
!Command: show running-config macsec
!Time: Mon Dec  5 04:50:16 2016
version 7.0(3)I4(5)
feature macsec
interface Ethernet2/13
  macsec keychain 1 policy system-default-macsec-policy
interface Ethernet2/14
  macsec keychain 1 policy system-default-macsec-policy

switch(config)# show macsec mka session
Interface          Local-TxSCI                     # Peers            Status
Key-Server         Auth Mode
------------------ ------------------------------- ------------------ ------------------
------------------ ------------------
Ethernet2/2        2c33.11b8.7d14/0001             1                  Secured
Yes                PRIMARY-PSK
Ethernet2/3        2c33.11b8.7d18/0001             1                  Secured
Yes                PRIMARY-PSK
------------------ ------------------------------- ------------------ ------------------
------------------ ------------------
Total Number of Sessions : 2
        Secured Sessions : 2
        Pending Sessions : 0

switch(config)# show macsec mka summary
Interface          Status   Cipher (Operational)   Key-Server   MACSEC-policy   Keychain
Fallback-keychain
------------------ -------- ---------------------- ------------ -------------- ----------
```

```
---------------
Ethernet2/1        down       -                      -          tests1           keych1
 no keychain
Ethernet2/2        Secured  GCM-AES-XPN-256        Yes          tests2           keych2
 no keychain
Ethernet2/3        Secured  GCM-AES-256            Yes          tests3           keyc3
 no keychain
```

The following example shows the configuration and output of Peer Enforce Cipher configuration feature MACsec:

```
switch# show key chain
Key-Chain KC1 Macsec
Key 10000000 -- text 7
"0729701e1d5d4c53404a522d26090f010e63647040534355560e007971772a263e30080a040707030353022725 7b73213556550958525a771b165038273
4362e2a"
cryptographic-algorithm AES_256_CMAC
send lifetime (always valid) [active]

Key-Chain KC2 Macsec
Key 10100000 -- text 7
"0729701e1d5d4c53404a522d26090f010e63647040534355560e007971772a263e30080a040707030353022725 7b73213556550958525a771b165038273
4362e2a"
cryptographic-algorithm AES_256_CMAC
send lifetime (always valid) [active]

switch#
switch# show run macsec

!Command: show running-config macsec
!Running configuration last done at: Mon Apr 17 16:49:57 2023
!Time: Mon Apr 17 16:50:09 2023

version 10.3(3) Bios:version 05.47
feature macsec

macsec policy MP1
no protocol lldp encrypted
cipher-suite enforce-peer GCM-AES-XPN-256 GCM-AES-XPN-128
macsec policy MP2
no protocol lldp encrypted
cipher-suite enforce-peer GCM-AES-256
interface Ethernet1/97/1
macsec keychain KC1 policy MP1

interface Ethernet1/97/2
macsec keychain KC2 policy MP2

switch#

switch# show macsec policy
MACSec Policy Cipher Pri Window Offset Security SAK Rekey time ICV Indicator Include-SCI
------------------------------- ---------------- ---- ------------ -------- --------------
 -------------- ------------- -------------
MP1 Enforce-Peer 16 148809600 0 should-secure pn-rollover FALSE TRUE
MP2 Enforce-Peer 16 148809600 0 should-secure pn-rollover FALSE TRUE
system-default-macsec-policy GCM-AES-XPN-256 16 148809600 0 should-secure pn-rollover FALSE
 TRUE

MACSec Policy                 Lldp-bypass
------------------------------- -------------
MP1                               True
MP2                               True
system-default-macsec-policy      FALSE
```

```
MACSec Policy PPK Crypto-QKD-Profile Name
-------------------------------- --------------------------------

MACSec Policy Cipher-Suite Enforce-Peer
-------------------------------- ------------------------------------------------------
MP1 GCM-AES-XPN-256 GCM-AES-XPN-128
MP2 GCM-AES-256
switch#
```

The following example shows the sample output of the **show macsec mka session detail** command:

```
switch# show macsec mka session details
Detailed Status for MKA Session
---------------------------------
Interface Name : Ethernet1/97/1
Session Status : SECURED - Secured MKA Session with MACsec
Local Tx-SCI : c4f7.d530.1484/0001
Local Tx-SSCI : 1
MKA Port Identifier : 1
CAK Name (CKN) : 10000000
CA Authentication Mode : PRIMARY-PSK
Member Identifier (MI) : D94B90E3FDB111CE583E7158
Message Number (MN) : 111
MKA Policy Name : MP1
Key Server Priority : 16
Key Server : Yes
Include ICV : No
SAK Cipher Suite : GCM-AES-XPN-128
SAK Cipher Suite (Operational) : GCM-AES-XPN-128
Replay Window Size : 148809600
Confidentiality Offset : CONF-OFFSET-0
Confidentiality Offset (Operational): CONF-OFFSET-0
Latest SAK Status : Rx & TX
Latest SAK AN : 1
Latest SAK KI : D94B90E3FDB111CE583E715800000001
Latest SAK KN : 1
Last SAK key time : 16:48:41 PST Mon Apr 17 2023
CA Peer Count : 1
Eapol dest mac : 0180.c200.0003
Ether-type : 0x888e
Peer Status:
Peer MI : 00110000000010001000000001
RxSCI : 0011.0000.0001/0001
Peer CAK : Match
Latest Rx MKPDU : 16:52:07 PST Mon Apr 17 2023

Interface Name : Ethernet1/97/2
Session Status : SECURED - Secured MKA Session with MACsec
Local Tx-SCI : c4f7.d530.1485/0001
Local Tx-SSCI : 1
MKA Port Identifier : 1
CAK Name (CKN) : 10100000
CA Authentication Mode : PRIMARY-PSK
Member Identifier (MI) : 43AE54C19982238C298E0241
Message Number (MN) : 107
MKA Policy Name : MP2
Key Server Priority : 16
Key Server : Yes
Include ICV : No
SAK Cipher Suite : GCM-AES-256
SAK Cipher Suite (Operational) : GCM-AES-256
Replay Window Size : 148809600
Confidentiality Offset : CONF-OFFSET-0
Confidentiality Offset (Operational): CONF-OFFSET-0
```

```
Latest SAK Status : Rx & TX
Latest SAK AN : 0
Latest SAK KI : 43AE54C19982238C298E024100000001
Latest SAK KN : 1
Last SAK key time : 16:48:42 PST Mon Apr 17 2023
CA Peer Count : 1
Eapol dest mac : 0180.c200.0003
Ether-type : 0x888e
Peer Status:
Peer MI : 00270000000100010000001
RxSCI : 0027.0000.0001/0001
Peer CAK : Match
Latest Rx MKPDU : 16:52:06 PST Mon Apr 17 2023
switch#
```

XML Examples

MACsec supports XML output for the following **show** commands for scripting purposes using **| xml**:

- **show key chain** *name* **| xml**
- **show macsec mka session interface** *interface slot/port* **details | xml**
- **show macsec mka statistics interface** *interface slot/port* **| xml**
- **show macsec mka summary | xml**
- **show macsec policy** *name* **| xml**
- **show macsec secy statistics interface** *interface slot/port* **| xml**
- **show running-config macsec | xml**

The following are example outputs for each of the preceding **show** commands:

Example 1: Displays the keychain configuration.

```
switch# show key chain "Kc2" | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0:rpm">
 <nf:data>
  <show>
   <key>
    <chain>
     <__XML__OPT_Cmd_rpm_show_keychain_cmd_keychain>
      <keychain>Kc2</keychain>
     </__XML__OPT_Cmd_rpm_show_keychain_cmd_keychain>
    </chain>
   </key>
  </show>
 </nf:data>
</nf:rpc-reply>
]]>]]>
```

Example 2: Displays information about the MACsec MKA session for a specific interface.

```
switch# show macsec mka session interface ethernet 4/31 details | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0">
```

```
 <nf:data>
  <show>
   <macsec>
    <mka>
     <session>
      <__XML__OPT_Cmd_show_macsec_mka_session_interface>
       <interface>
        <__XML__INTF_ifname>
         <__XML__PARAM_value>
          <__XML__INTF_output>Ethernet4/31</__XML__INTF_output>
         </__XML__PARAM_value>
        </__XML__INTF_ifname>
       </interface>
       <__XML__OPT_Cmd_show_macsec_mka_session_details>
        <details/>
        <__XML__OPT_Cmd_show_macsec_mka_session___readonly__>
         <__readonly__>
          <TABLE_mka_session_details>
           <ROW_mka_session_details>
            <ifname>Ethernet4/31</ifname>
            <status>Secured</status>
            <sci>0c75.bd03.5360/0001</sci>
            <ssci>1</ssci>
            <port_id>1</port_id>
            <mi>02000000000000000000000000000000000000000000000000000000000000000000
</mi>
            <mi>F511280A765CE41C79458753</mi>
            <mn>2770</mn>
            <policy>am2</policy>
            <ks_prio>0</ks_prio>
            <keyserver>No</keyserver>
            <cipher>GCM-AES-XPN-256</cipher>
            <window>512</window>
            <conf_offset>CONF-OFFSET-0</conf_offset>
            <sak_status>Rx &amp;amp; TX</sak_status>
            <sak_an>1</sak_an>
            <sak_ki>516486241</sak_ki>
            <sak_kn>90</sak_kn>
            <last_sak_rekey_time>07:12:02 UTC Fri Jan 20 2017</last_sak_rekey_ti
me>
           </ROW_mka_session_details>
          </TABLE_mka_session_details>
         </__readonly__>
        </__XML__OPT_Cmd_show_macsec_mka_session___readonly__>
       </__XML__OPT_Cmd_show_macsec_mka_session_details>
      </__XML__OPT_Cmd_show_macsec_mka_session_interface>
     </session>
    </mka>
   </macsec>
  </show>
 </nf:data>
</nf:rpc-reply>
]]>]]>
```

Example 3: Displays MACsec MKA statistics.

```
switch# show macsec mka statistics interface ethernet 4/31 | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0">
 <nf:data>
  <show>
<macsec>
    <mka>
```

```
<statistics>
 <__XML__OPT_Cmd_some_macsec_mka_statistics_interface>
  <interface>
   <__XML__INTF_ifname>
    <__XML__PARAM_value>
     <__XML__INTF_output>Ethernet4/31</__XML__INTF_output>
     <__XML__INTF_output>Ethernet4/31</__XML__INTF_output>
    </__XML__PARAM_value>
   </__XML__INTF_ifname>
  </interface>
  <__XML__OPT_Cmd_some_macsec_mka_statistics___readonly__>
   <__readonly__>
    <TABLE_mka_intf_stats>
     <ROW_mka_intf_stats>
      <TABLE_ca_stats>
       <ROW_ca_stats>
        <ca_stat_ckn>0x2</ca_stat_ckn>
        <ca_stat_pairwise_cak_rekey>0</ca_stat_pairwise_cak_rekey>
        <sa_stat_sak_generated>0</sa_stat_sak_generated>
        <sa_stat_sak_rekey>0</sa_stat_sak_rekey>
        <sa_stat_sak_received>91</sa_stat_sak_received>
        <sa_stat_sak_response_rx>0</sa_stat_sak_response_rx>
        <mkpdu_stat_mkpdu_tx>2808</mkpdu_stat_mkpdu_tx>
        <mkpdu_stat_mkpdu_tx_distsak>0</mkpdu_stat_mkpdu_tx_distsak>
        <mkpdu_stat_mkpdu_rx>2714</mkpdu_stat_mkpdu_rx>
        <mkpdu_stat_mkpdu_rx_distsak>91</mkpdu_stat_mkpdu_rx_distsak>
       </ROW_ca_stats>
      </TABLE_ca_stats>
     </ROW_mka_intf_stats>
    </TABLE_mka_intf_stats>
   </__readonly__>
  </__XML__OPT_Cmd_some_macsec_mka_statistics___readonly__>
  <interface>
   <__XML__INTF_ifname>
    <__XML__PARAM_value>
     <__XML__INTF_output>Ethernet4/31</__XML__INTF_output>
    </__XML__PARAM_value>
   </__XML__INTF_ifname>
  </interface>
  <__XML__OPT_Cmd_some_macsec_mka_statistics___readonly__>
   <__readonly__>
    <TABLE_mka_intf_stats>
     <ROW_mka_intf_stats>
      <TABLE_idb_stats>
       <ROW_idb_stats>
        <ca_stat_pairwise_cak_rekey>0</ca_stat_pairwise_cak_rekey>
        <sa_stat_sak_generated>0</sa_stat_sak_generated>
        <sa_stat_sak_rekey>0</sa_stat_sak_rekey>
        <sa_stat_sak_received>91</sa_stat_sak_received>
        <sa_stat_sak_response_rx>0</sa_stat_sak_response_rx>
        <mkpdu_stat_mkpdu_tx>2808</mkpdu_stat_mkpdu_tx>
        <mkpdu_stat_mkpdu_tx_distsak>0</mkpdu_stat_mkpdu_tx_distsak>
        <mkpdu_stat_mkpdu_rx>2714</mkpdu_stat_mkpdu_rx>
        <mkpdu_stat_mkpdu_rx_distsak>91</mkpdu_stat_mkpdu_rx_distsak>
        <idb_stat_mkpdu_tx_success>2808</idb_stat_mkpdu_tx_success>
        <idb_stat_mkpdu_tx_fail>0</idb_stat_mkpdu_tx_fail>
        <idb_stat_mkpdu_tx_pkt_build_fail>0</idb_stat_mkpdu_tx_pkt_build_fail>
        <idb_stat_mkpdu_no_tx_on_intf_down>0</idb_stat_mkpdu_no_tx_on_intf_down>
        <idb_stat_mkpdu_no_rx_on_intf_down>0</idb_stat_mkpdu_no_rx_on_intf_down>
        <idb_stat_mkpdu_rx_ca_notfound>0</idb_stat_mkpdu_rx_ca_notfound>
        <idb_stat_mkpdu_rx_error>0</idb_stat_mkpdu_rx_error>
        <idb_stat_mkpdu_rx_success>2714</idb_stat_mkpdu_rx_success>
        <idb_stat_mkpdu_failure_rx_integrity_check_error>0</idb_stat_mkpdu_
failure_rx_integrity_check_error>
```

```
            <idb_stat_mkpdu_failure_invalid_peer_mn_error>0</idb_stat_mkpdu_fai
lure_invalid_peer_mn_error>
            <idb_stat_mkpdu_failure_nonrecent_peerlist_mn_error>1</idb_stat_mkp
du_failure_nonrecent_peerlist_mn_error>
            <idb_stat_mkpdu_failure_sakuse_kn_mismatch_error>0</idb_stat_mkpdu_
failure_sakuse_kn_mismatch_error>
<idb_stat_mkpdu_failure_sakuse_rx_not_set_error>0</idb_stat_mkpdu_f
ailure_sakuse_rx_not_set_error>
            <idb_stat_mkpdu_failure_sakuse_key_mi_mismatch_error>0</idb_stat_mk
pdu_failure_sakuse_key_mi_mismatch_error>
            <idb_stat_mkpdu_failure_sakuse_an_not_in_use_error>0</idb_stat_mkpd
u_failure_sakuse_an_not_in_use_error>
            <idb_stat_mkpdu_failure_sakuse_ks_rx_tx_not_set_error>0</idb_stat_m
kpdu_failure_sakuse_ks_rx_tx_not_set_error>
            <idb_stat_mkpdu_failure_sakuse_eapol_ethertype_mismatch_error>0</id
b_stat_mkpdu_failure_sakuse_eapol_ethertype_mismatch_error>
            <idb_stat_sak_failure_sak_generate_error>0</idb_stat_sak_failure_sa
k_generate_error>
            <idb_stat_sak_failure_hash_generate_error>0</idb_stat_sak_failure_h
ash_generate_error>
            <idb_stat_sak_failure_sak_encryption_error>0</idb_stat_sak_failure_
sak_encryption_error>
            <idb_stat_sak_failure_sak_decryption_error>0</idb_stat_sak_failure_
sak_decryption_error>
            <idb_stat_sak_failure_ick_derivation_error>0</idb_stat_sak_failure_
ick_derivation_error>
            <idb_stat_sak_failure_kek_derivation_error>0</idb_stat_sak_failure_
kek_derivation_error>
            <idb_stat_sak_failure_invalid_macsec_capability_error>0</idb_stat_s
ak_failure_invalid_macsec_capability_error>
            <idb_stat_macsec_failure_rx_sa_create_error>0</idb_stat_macsec_fail
ure_rx_sa_create_error>
            <idb_stat_macsec_failure_tx_sa_create_error>0</idb_stat_macsec_fail
ure_tx_sa_create_error>
           </ROW_idb_stats>
          </TABLE_idb_stats>
         </ROW_mka_intf_stats>
        </TABLE_mka_intf_stats>
       </__readonly__>
      </__XML__OPT_Cmd_some_macsec_mka_statistics___readonly__>
     </__XML__OPT_Cmd_some_macsec_mka_statistics_interface>
    </statistics>
   </mka>
  </macsec>
 </show>
</nf:data>
</nf:rpc-reply>
]]>]]>
```

Example 4: Displays the MACsec MKA configuration.

```
switch# show macsec mka summary | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0">
 <nf:data>
  <show>
   <macsec>
    <mka>
     <__XML__OPT_Cmd_some_macsec_summary>
      <__XML__OPT_Cmd_some_macsec___readonly__>
       <__readonly__>
        <TABLE_mka_summary>
         <ROW_mka_summary>
```

```
          <ifname>Ethernet2/1</ifname>
          <policy>am2</policy>
<keychain>kc2/020000000000000000000000000000000000000000000000000000000000
00000000</keychain>
         </ROW_mka_summary>
         <ROW_mka_summary>
          <ifname>Ethernet3/1</ifname>
          <policy>am2</policy>
          <keychain>kc2/020000000000000000000000000000000000000000000000000000000000
00000000</keychain>
         </ROW_mka_summary>

[TRUNCATED FOR READABILITY]

<ROW_mka_summary>
          <ifname>Ethernet3/32</ifname>
          <policy>am2</policy>
          <keychain>kc2/020000000000000000000000000000000000000000000000000000000000
00000000</keychain>
         </ROW_mka_summary>
        </TABLE_mka_summary>
       </__readonly__>
      </__XML__OPT_Cmd_some_macsec___readonly__>
     </__XML__OPT_Cmd_some_macsec_summary>
    </mka>
   </macsec>
  </show>
 </nf:data>
</nf:rpc-reply>
]]>]]>
```

Example 5: Displays the configuration for a specific MACsec policy.

```
switch# show macsec policy am2 | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0">
 <nf:data>
  <show>
   <macsec>
    <policy>
     <__XML__OPT_Cmd_some_macsec_policy_name>
      <policy_name>am2</policy_name>
      <__XML__OPT_Cmd_some_macsec___readonly__>
       <__readonly__>
        <TABLE_macsec_policy>
         <ROW_macsec_policy>
          <name>am2</name>
          <cipher_suite>GCM-AES-XPN-256</cipher_suite>
          <keyserver_priority>0</keyserver_priority>
          <window_size>512</window_size>
          <conf_offset>0</conf_offset>
          <security_policy>must-secure</security_policy>
          <sak-expiry-time>60</sak-expiry-time>
         </ROW_macsec_policy>
        </TABLE_macsec_policy>
       </__readonly__>
      </__XML__OPT_Cmd_some_macsec___readonly__>
     </__XML__OPT_Cmd_some_macsec_policy_name>
    </policy>
   </macsec>
  </show>
 </nf:data>
</nf:rpc-reply>
```

```
]]>]]>
```

Example 6: Displays MACsec security statistics.

```
switch# show macsec secy statistics interface ethernet 4/31 | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0">
 <nf:data>
  <show>
   <macsec>
    <secy>
     <statistics>
      <interface>
       <__XML__INTF_ifname>
        <__XML__PARAM_value>
         <__XML__INTF_output>Ethernet4/31</__XML__INTF_output>
        </__XML__PARAM_value>
        <__XML__OPT_Cmd_some_macsec_secy_statistics___readonly__>
         <__readonly__>
          <TABLE_statistics>
           <ROW_statistics>
            <in_pkts_unicast_uncontrolled>0</in_pkts_unicast_uncontrolled>
            <in_pkts_multicast_uncontrolled>42</in_pkts_multicast_uncontrolled>
            <in_pkts_broadcast_uncontrolled>0</in_pkts_broadcast_uncontrolled>
            <in_rx_drop_pkts_uncontrolled>0</in_rx_drop_pkts_uncontrolled>
            <in_rx_err_pkts_uncontrolled>0</in_rx_err_pkts_uncontrolled>
            <in_pkts_unicast_controlled>0</in_pkts_unicast_controlled>
            <in_pkts_multicast_controlled>2</in_pkts_multicast_controlled>
            <in_pkts_broadcast_controlled>0</in_pkts_broadcast_controlled>
            <in_rx_drop_pkts_controlled>0</in_rx_drop_pkts_controlled>
            <in_rx_err_pkts_controlled>0</in_rx_err_pkts_controlled>
            <in_octets_uncontrolled>7230</in_octets_uncontrolled>
            <in_octets_controlled>470</in_octets_controlled>
            <input_rate_uncontrolled_pps>0</input_rate_uncontrolled_pps>
            <input_rate_uncontrolled_bps>9</input_rate_uncontrolled_bps>
            <input_rate_controlled_pps>0</input_rate_controlled_pps>
            <input_rate_controlled_bps>23</input_rate_controlled_bps>
            <out_pkts_unicast_uncontrolled>0</out_pkts_unicast_uncontrolled>
            <out_pkts_multicast_uncontrolled>41</out_pkts_multicast_uncontrolled>
            <out_pkts_broadcast_uncontrolled>0</out_pkts_broadcast_uncontrolled>
            <out_rx_drop_pkts_uncontrolled>0</out_rx_drop_pkts_uncontrolled>
            <out_rx_err_pkts_uncontrolled>0</out_rx_err_pkts_uncontrolled>
            <out_pkts_unicast_controlled>0</out_pkts_unicast_controlled>
            <out_pkts_multicast_controlled>2</out_pkts_multicast_controlled>
            <out_pkts_broadcast_controlled>0</out_pkts_broadcast_controlled>
            <out_rx_drop_pkts_controlled>0</out_rx_drop_pkts_controlled>
            <out_rx_err_pkts_controlled>0</out_rx_err_pkts_controlled>
            <out_octets_uncontrolled>6806</out_octets_uncontrolled>
            <out_octets_controlled>470</out_octets_controlled>
            <out_octets_common>7340</out_octets_common>
            <output_rate_uncontrolled_pps>2598190092</output_rate_uncontrolled_pps>
            <output_rate_uncontrolled_bps>2598190076</output_rate_uncontrolled_bps>
            <output_rate_controlled_pps>0</output_rate_controlled_pps>
            <output_rate_controlled_bps>23</output_rate_controlled_bps>
            <in_pkts_transform_error>0</in_pkts_transform_error>
            <in_pkts_control>40</in_pkts_control>
            <in_pkts_untagged>0</in_pkts_untagged>
            <in_pkts_no_tag>0</in_pkts_no_tag>
            <in_pkts_badtag>0</in_pkts_badtag>
            <in_pkts_no_sci>0</in_pkts_no_sci>
            <in_pkts_unknown_sci>0</in_pkts_unknown_sci>
            <in_pkts_tagged_ctrl>0</in_pkts_tagged_ctrl>
            <out_pkts_transform_error>0</out_pkts_transform_error>
```

```
            <out_pkts_control>41</out_pkts_control>
            <out_pkts_untagged>0</out_pkts_untagged>
            <rx_sa_an>1</rx_sa_an>
            <in_pkts_unchecked>0</in_pkts_unchecked>
            <in_pkts_delayed>0</in_pkts_delayed>
            <in_pkts_late>0</in_pkts_late>
            <in_pkts_ok>1</in_pkts_ok>
            <in_pkts_invalid>0</in_pkts_invalid>
            <in_pkts_not_valid>0</in_pkts_not_valid>
            <in_pkts_not_using_sa>0</in_pkts_not_using_sa>
            <in_pkts_unused_sa>0</in_pkts_unused_sa>
            <in_octets_decrypted>223</in_octets_decrypted>
            <in_octets_validated>0</in_octets_validated>
            <tx_sa_an>1</tx_sa_an>
            <out_pkts_encrypted_protected>1</out_pkts_encrypted_protected>
            <out_pkts_too_long>0</out_pkts_too_long>
            <out_pkts_sa_not_inuse>0</out_pkts_sa_not_inuse>
            <out_octets_encrypted_protected>223</out_octets_encrypted_protected>
           </ROW_statistics>
          </TABLE_statistics>
         </__readonly__>
        </__XML__OPT_Cmd_some_macsec_secy_statistics___readonly__>
       </__XML__INTF_ifname>
      </interface>
     </statistics>
    </secy>
   </macsec>
  </show>
 </nf:data>
</nf:rpc-reply>
]]>]]>
```

Example 7: Displays the running configuration information for MACsec.

```
switch# show running-config macsec | xml

!Command: show running-config macsec
!Time: Fri Jan 20 07:12:34 2017

version 7.0(3)I4(6)
****************************************
This may take time. Please be patient.
****************************************
<?xml version="1.0"?>
<nf:rpc xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://www.cis
co.com/nxos:7.0.3.I4.6.:configure_" xmlns:m="http://www.cisco.com/nxos:7.0.3.I4.
6.:_exec" xmlns:m1="http://www.cisco.com/nxos:7.0.3.I4.6.:configure__macsec-poli
cy" xmlns:m2="http://www.cisco.com/nxos:7.0.3.I4.6.:configure__if-eth-non-member
" message-id="1">
  <nf:get-config>
    <nf:source>
      <nf:running/>
    </nf:source>
    <nf:filter>
      <m:configure>
        <m:terminal>
          <feature>
            <macsec/>
          </feature>
          <macsec>
            <policy>
              <__XML__PARAM__policy_name>
                <__XML__value>am2</__XML__value>
                <m1:cipher-suite>
```

```
                    <m1:__XML__PARAM__suite>
                      <m1:__XML__value>GCM-AES-XPN-256</m1:__XML__value>
                    </m1:__XML__PARAM__suite>
                  </m1:cipher-suite>
                  <m1:key-server-priority>
                    <m1:__XML__PARAM__pri>
                      <m1:__XML__value>0</m1:__XML__value>
                    </m1:__XML__PARAM__pri>
                  </m1:key-server-priority>
<m1:window-size>
<m1:__XML__PARAM__size>
                      <m1:__XML__value>512</m1:__XML__value>
                    </m1:__XML__PARAM__size>
                  </m1:window-size>
                  <m1:conf-offset>
                    <m1:__XML__PARAM__offset>
                      <m1:__XML__value>CONF-OFFSET-0</m1:__XML__value>
                    </m1:__XML__PARAM__offset>
                  </m1:conf-offset>
                  <m1:security-policy>
                    <m1:__XML__PARAM__policy>
                      <m1:__XML__value>must-secure</m1:__XML__value>
                    </m1:__XML__PARAM__policy>
                  </m1:security-policy>
                  <m1:sak-expiry-time>
                    <m1:__XML__PARAM__ts>
                      <m1:__XML__value>60</m1:__XML__value>
                    </m1:__XML__PARAM__ts>
                  </m1:sak-expiry-time>
                </__XML__PARAM__policy_name>
              </policy>
            </macsec>
            <interface>
              <__XML__PARAM__interface>
                <__XML__value>Ethernet2/1</__XML__value>
                <m2:macsec>
                  <m2:keychain>
                    <m2:__XML__PARAM__keychain_name>
                      <m2:__XML__value>kc2</m2:__XML__value>
                      <m2:policy>
                        <m2:__XML__PARAM__policy_name>
                          <m2:__XML__value>am2</m2:__XML__value>
                        </m2:__XML__PARAM__policy_name>
                      </m2:policy>
                    </m2:__XML__PARAM__keychain_name>
                  </m2:keychain>
                </m2:macsec>
              </__XML__PARAM__interface>
            </interface>

[TRUNCATED FOR READABILITY]

<interface>
              <__XML__PARAM__interface>
                <__XML__value>Ethernet4/31</__XML__value>
                <m2:macsec>
                  <m2:keychain>
                    <m2:__XML__PARAM__keychain_name>
                      <m2:__XML__value>kc2</m2:__XML__value>
                      <m2:policy>
                        <m2:__XML__PARAM__policy_name>
                          <m2:__XML__value>am2</m2:__XML__value>
                        </m2:__XML__PARAM__policy_name>
                      </m2:policy>
```

```
                    </m2:__XML__PARAM__keychain_name>
                  </m2:keychain>
                </m2:macsec>
              </__XML__PARAM__interface>
            </interface>
          </m:terminal>
        </m:configure>
      </nf:filter>
    </nf:get-config>
</nf:rpc>
]]>]]>
```

MIBs

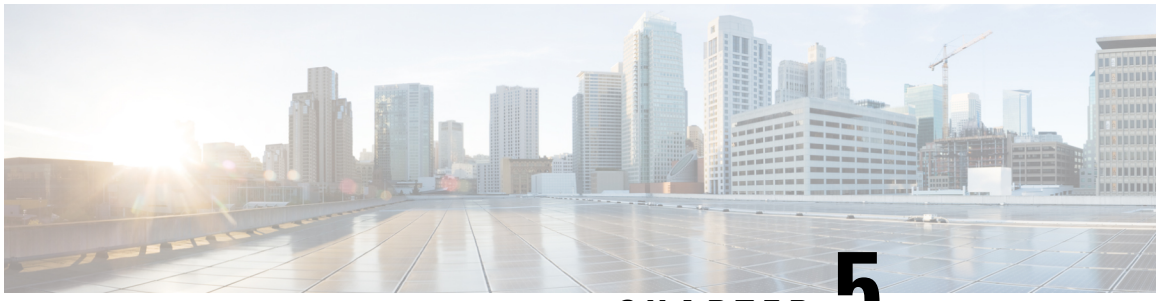
MACsec supports the following MIBs:

- IEEE8021-SECY-MIB
- CISCO-SECY-EXT-MIB

To locate and download supported MIBs, go to the following URL: ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html.

Related Documentation

Related Topic	Document Title
Keychain management	Cisco Nexus 9000 Series NX-OS Security Configuration Guide
System messages	Cisco Nexus 9000 Series NX-OS System Messages References



CHAPTER 5

Configure Control Plane Policing

This chapter describes how to configure Control Plane Policing on Cisco NX-OS devices.

About CoPP

Control Plane Policing (CoPP) protects the control plane and separates it from the data plane, which ensures network stability, reachability, and packet delivery.

This feature allows a policy map to be applied to the control plane. This policy map looks like a normal QoS policy and is applied to all traffic entering the switch from a non-management port. A common attack vector for network devices is the denial-of-service (DoS) attack, where excessive traffic is directed at the device interfaces.

The Cisco NX-OS device provides CoPP to prevent DoS attacks from impacting performance. Such attacks, which can be perpetrated either inadvertently or maliciously, typically involve high rates of traffic destined to the supervisor module or CPU itself.

The supervisor module divides the traffic that it manages into three functional components or planes:

Data plane
: Handles all the data traffic. The basic functionality of a Cisco NX-OS device is to forward packets from one interface to another. The packets that are not meant for the switch itself are called the transit packets. These packets are handled by the data plane.

Control plane
Handles all routing protocol control traffic. These protocols, such as the Border Gateway Protocol (BGP) and the Open Shortest Path First (OSPF) Protocol, send control packets between devices. These packets are destined to router addresses and are called control plane packets.

Management plane
Runs the components meant for Cisco NX-OS device management purposes such as the command-line interface (CLI) and Simple Network Management Protocol (SNMP).

The supervisor module has both the management plane and control plane and is critical to the operation of the network. Any disruption or attacks to the supervisor module will result in serious network outages. For example, excessive traffic to the supervisor module could overload and slow down the performance of the entire Cisco NX-OS device. For example, a DoS attack on the supervisor module could generate IP traffic streams to the control plane at a very high rate, forcing the control plane to spend a large amount of time in handling these packets and preventing the control plane from processing genuine traffic.

Examples of DoS attacks include:

- Internet Control Message Protocol (ICMP) echo requests
- IP fragments
- TCP SYN flooding

These attacks can impact the device performance and have the following negative effects:

- Reduced service quality (such as poor voice, video, or critical applications traffic)
- High route processor or switch processor CPU utilization
- Route flaps due to loss of routing protocol updates or keepalives
- Unstable Layer 2 topology
- Slow or unresponsive interactive sessions with the CLI
- Processor resource exhaustion, such as the memory and buffers
- Indiscriminate drops of incoming packets

Caution It is important to ensure that you protect the supervisor module from accidental or malicious attacks by configuring control plane protection.

Control Plane Protection

To protect the control plane, the Cisco NX-OS device segregates different packets destined for the control plane into different classes. Once these classes are identified, the Cisco NX-OS device polices the packets, which ensures that the supervisor module is not overwhelmed.

Control Plane Packet Types

Different types of packets can reach the control plane:

Receive packets
Packets that have the destination address of a router. The destination address can be a Layer 2 address (such as a router MAC address) or a Layer 3 address (such as the IP address of a router interface). These packets include router updates and keepalive messages. Multicast packets can also be in this category where packets are sent to multicast addresses that are used by a router.

Exception packets
Packets that need special handling by the supervisor module. For example, if a destination address is not present in the Forwarding Information Base (FIB) and results in a miss, the supervisor module sends an ICMP unreachable packet back to the sender. Another example is a packet with IP options set.

The following exceptions are possible from line cards only:

- match exception ip option
- match exception ipv6 option
- match exception ttl-failure

The following exceptions are possible from fabric modules only:

- match exception ipv6 icmp unreachable
- match exception ip icmp unreachable

The following exceptions are possible from line cards and fabric modules:

- match exception mtu-failure

Redirected packets
Packets that are redirected to the supervisor module.

Glean packets
If a Layer 2 MAC address for a destination IP address is not present in the FIB, the supervisor module receives the packet and sends an ARP request to the host.

All of these different packets could be maliciously used to attack the control plane and overwhelm the Cisco NX-OS device. CoPP classifies these packets to different classes and provides a mechanism to individually control the rate at which the supervisor module receives these packets.

Classification for CoPP

For effective protection, the Cisco NX-OS device classifies the packets that reach the supervisor modules to allow you to apply different rate controlling policies based on the type of the packet. For example, you might want to be less strict with a protocol packet such as Hello messages but more strict with a packet that is sent to the supervisor module because the IP option is set. You configure packet classifications and rate controlling policies using class maps and policy maps.

Egress CoPP

Beginning with Cisco NX-OS Release 10.2(3)F, egress CoPP is supported on the Nexus 93180YC-EX, Nexus 93180YC-FX, Nexus 93240YC-FX2, Nexus 93360YC-FX2, Nexus 9336C-FX2, Nexus 9336C-FX2-E, Nexus 93180YC-FX3, N9K-C9316D-GX, N9K-C93600CD-GX, Nexus 9364C-GX, N9K-C9332D-GX2B, Nexus 9364C and Nexus 9332C cloudscale switches.

Egress CoPP can be applied on top of custom/default CoPP policy.

Rate Controlling Mechanisms

Once the packets are classified, the Cisco NX-OS device has different mechanisms to control the rate at which packets arrive at the supervisor module. Two mechanisms control the rate of traffic to the supervisor module. One is called policing and the other is called rate limiting.

Using hardware policers, you can define separate actions for traffic that conforms to or violates certain conditions. The actions can transmit the packet, mark down the packet, or drop the packet.

You can configure the following parameters for policing:

Committed information rate (CIR)
Desired bandwidth, specified as a bit rate or a percentage of the link rate.

Committed burst (BC)
Size of a traffic burst that can exceed the CIR within a given unit of time and not impact scheduling

In addition, you can set separate actions such as transmit or drop for conform and violate traffic.

For more information on policing parameters, see the *Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide*.

Dynamic and Static CoPP ACLs

CoPP access control lists (ACLs) are classified as either dynamic or static. Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches use only dynamic CoPP ACLs. Cisco Nexus 9200 Series switches use both dynamic and static CoPP ACLs.

Dynamic CoPP ACLs work only for Forwarding Information Base (FIB)-based supervisor redirected packets, and static CoPP ACLs work for ACL-based supervisor redirected packets. Dynamic CoPP ACLs are supported for myIP and link-local multicast traffic, and static CoPP ACLs are supported for all other types of traffic.

Static CoPP ACLs are identified by a substring. Any ACL that has one of these substrings is categorized as a static CoPP ACL.

- MAC-based static CoPP ACL substrings:
  - acl-mac-cdp-udld-vtp
  - acl-mac-cfsoe
  - acl-mac-dot1x
  - acl-mac-l2-tunnel
  - acl-mac-l3-isis
  - acl-mac-lacp
  - acl-mac-lldp
  - acl-mac-sdp-srp
  - acl-mac-stp
  - acl-mac-undesirable
- Protocol-based static CoPP ACL substrings:
  - acl-dhcp

- acl-dhcp-relay-response
- acl-dhcp6
- acl-dhcp6-relay-response
- acl-ptp

- Multicast-based static CoPP ACL substrings:
  - acl-igmp

For more information on static CoPP ACLs, see Guidelines and Limitations for CoPP, on page 120.

Default Policing Policies

When you bring up your Cisco NX-OS device for the first time, the Cisco NX-OS software installs the default copp-system-p-policy-strict policy to protect the supervisor module from DoS attacks. You can set the level of protection by choosing one of the following CoPP policy options from the initial setup utility:

- Strict—This policy is 1 rate and 2 color.
- Moderate—This policy is 1 rate and 2 color. The important class burst size is greater than the strict policy but less than the lenient policy.
- Lenient—This policy is 1 rate and 2 color. The important class burst size is greater than the moderate policy but less than the dense policy.
- Dense—This policy is 1 rate and 2 color. The policer CIR values are less than the strict policy.
- Skip—No control plane policy is applied. (Cisco does not recommend using the Skip option because it will impact the control plane of the network.)

If you do not select an option or choose not to execute the setup utility, the software applies strict policing. We recommend that you start with the strict policy and later modify the CoPP policies as required.

Note Strict policing is not applied by default when using POAP, so you must configure a CoPP policy.

The copp-system-p-policy policy has optimized values suitable for basic device operations. You must add specific class and access-control list (ACL) rules that meet your DoS protection requirements. The default CoPP policy does not change when you upgrade the software.

Caution Selecting the skip option and not subsequently configuring CoPP protection can leave your Cisco NX-OS device vulnerable to DoS attacks.

You can reassign the CoPP default policy by entering the setup utility again using the **setup** command from the CLI prompt or by using the **copp profile** command.

Related Topics

Changing or Reapplying the Default CoPP Policy, on page 132

Default Class Maps

The copp-system-class-critical class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-critical
      match access-group name copp-system-p-acl-bgp
      match access-group name copp-system-p-acl-rip
      match access-group name copp-system-p-acl-vpc
      match access-group name copp-system-p-acl-bgp6
      match access-group name copp-system-p-acl-ospf
      match access-group name copp-system-p-acl-rip6
      match access-group name copp-system-p-acl-eigrp
      match access-group name copp-system-p-acl-ospf6
      match access-group name copp-system-p-acl-eigrp6
      match access-group name copp-system-p-acl-auto-rp
      match access-group name copp-system-p-acl-mac-l3-isis
```

The copp-system-class-exception class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-exception
      match exception ip option
      match exception ip icmp unreachable
      match exception ipv6 option
      match exception ipv6 icmp unreachable
```

The copp-system-class-exception-diag class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-exception-diag
      match exception ttl-failure
      match exception mtu-failure
```

The copp-system-class-important class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-important
      match access-group name copp-system-p-acl-hsrp
      match access-group name copp-system-p-acl-vrrp
      match access-group name copp-system-p-acl-hsrp6
      match access-group name copp-system-p-acl-vrrp6
      match access-group name copp-system-p-acl-mac-lldp
```

The copp-system-class-l2-default class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-l2-default
      match access-group name copp-system-p-acl-mac-undesirable
```

The copp-system-class-l2-unpoliced class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-l2-unpoliced
      match access-group name copp-system-p-acl-mac-stp
      match access-group name copp-system-p-acl-mac-lacp
      match access-group name copp-system-p-acl-mac-cfsoe
      match access-group name copp-system-p-acl-mac-sdp-srp
      match access-group name copp-system-p-acl-mac-l2-tunnel
      match access-group name copp-system-p-acl-mac-cdp-udld-vtp
```

The copp-system-class-l3mc-data class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-l3mc-data
      match exception multicast rpf-failure
      match exception multicast dest-miss
```

The copp-system-class-l3uc-data class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-l3uc-data
      match exception glean
```

The copp-system-class-management class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-management
      match access-group name copp-system-p-acl-ftp
      match access-group name copp-system-p-acl-ntp
      match access-group name copp-system-p-acl-ssh
      match access-group name copp-system-p-acl-http
      match access-group name copp-system-p-acl-ntp6
      match access-group name copp-system-p-acl-sftp
      match access-group name copp-system-p-acl-snmp
      match access-group name copp-system-p-acl-ssh6
      match access-group name copp-system-p-acl-tftp
      match access-group name copp-system-p-acl-https
      match access-group name copp-system-p-acl-snmp6
      match access-group name copp-system-p-acl-tftp6
      match access-group name copp-system-p-acl-radius
      match access-group name copp-system-p-acl-tacacs
      match access-group name copp-system-p-acl-telnet
      match access-group name copp-system-p-acl-radius6
      match access-group name copp-system-p-acl-tacacs6
      match access-group name copp-system-p-acl-telnet6
```

The copp-system-class-monitoring class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-monitoring
      match access-group name copp-system-p-acl-icmp
      match access-group name copp-system-p-acl-icmp6
      match access-group name copp-system-p-acl-traceroute
```

The copp-system-class-multicast-host class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-multicast-host
      match access-group name copp-system-p-acl-mld
```

The copp-system-class-multicast-router class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-multicast-router
      match access-group name copp-system-p-acl-pim
      match access-group name copp-system-p-acl-msdp
      match access-group name copp-system-p-acl-pim6
      match access-group name copp-system-p-acl-pim-reg
      match access-group name copp-system-p-acl-pim6-reg
      match access-group name copp-system-p-acl-pim-mdt-join
```

The copp-system-class-nat-flow class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-nat-flow
      match exception nat-flow
```

The copp-system-class-ndp class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-ndp
      match access-group name copp-system-p-acl-ndp
```

The copp-system-class-normal class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-normal
      match access-group name copp-system-p-acl-mac-dot1x
      match protocol arp
```

The copp-system-class-normal-dhcp class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-normal-dhcp
      match access-group name copp-system-p-acl-dhcp
      match access-group name copp-system-p-acl-dhcp6
```

The copp-system-class-normal-dhcp-relay-response class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-normal-dhcp-relay-response
      match access-group name copp-system-p-acl-dhcp-relay-response
      match access-group name copp-system-p-acl-dhcp6-relay-response
```

The copp-system-class-normal-igmp class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-normal-igmp
      match access-group name copp-system-p-acl-igmp
```

The copp-system-class-redirect class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-redirect
      match access-group name copp-system-p-acl-ptp
```

The copp-system-class-undesirable class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-undesirable
      match access-group name copp-system-p-acl-undesirable
      match exception multicast sg-rpf-failure
```

The copp-system-class-fcoe class has the following configuration:

```
class-map type control-plane match-any copp-system-p-class-fcoe
      match access-group name copp-system-p-acl-mac-fcoe
```

Note The copp-system-class-fcoe class is not supported for Cisco Nexus 9200 Series switches.

Strict Default CoPP Policy

On Cisco Nexus 9200 Series switches, the strict CoPP policy has the following configuration:

Note The CIR value for copied or custom CoPP profiles which are pre-existing before upgrade to a new image will have the same CIR values as before. The new CoPP profiles which are copied from default profiles in the new image will have a new CIR value.

```
policy-map type control-plane copp-system-p-policy-strict
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 800 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
```

```
    police cir 36000 kbps bc 1280000 bytes conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 2500 kbps bc 1280000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 2600 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 10000 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 1000 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 2400 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 2200 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1400 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 1300 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 1500 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 3000 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 280 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 150 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 150 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 150 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 50 mbps bc 8192000 bytes conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
    police cir 800 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 400 kbps bc 32000 bytes conform transmit violate drop
  class class-default
    set cos 0
    police cir 400 kbps bc 32000 bytes conform transmit violate drop
```

On Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches, the strict CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-strict
  class copp-system-p-class-l3uc-data
```

```
  set cos 1
  police cir 250 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-critical
  set cos 7
  police cir 19000 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-important
  set cos 6
  police cir 3000 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-multicast-router
  set cos 6
  police cir 3000 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-management
  set cos 2
  police cir 3000 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-multicast-host
  set cos 1
  police cir 2000 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-l3mc-data
  set cos 1
  police cir 3000 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-normal
  set cos 1
  police cir 1500 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-ndp
  set cos 6
  police cir 1500 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-normal-dhcp
  set cos 1
  police cir 300 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-normal-dhcp-relay-response
  set cos 1
  police cir 400 pps bc 64 packets conform transmit violate drop
class copp-system-p-class-normal-igmp
  set cos 3
  police cir 6000 pps bc 64 packets conform transmit violate drop
class copp-system-p-class-redirect
  set cos 1
  police cir 1500 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-exception
  set cos 1
  police cir 50 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-exception-diag
  set cos 1
  police cir 50 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-monitoring
  set cos 1
  police cir 300 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-l2-unpoliced
  set cos 7
  police cir 20000 pps bc 8192 packets conform transmit violate drop
class copp-system-p-class-undesirable
  set cos 0
  police cir 15 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-fcoe
  set cos 6
  police cir 1500 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-nat-flow
  set cos 7
  police cir 100 pps bc 64 packets conform transmit violate drop
class copp-system-p-class-l2-default
  set cos 0
  police cir 50 pps bc 32 packets conform transmit violate drop
class class-default
  set cos 0
```

```
    police cir 50 pps bc 32 packets conform transmit violate drop
```

Moderate Default CoPP Policy

On Cisco Nexus 9200 Series switches, the moderate CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-moderate
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 800 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 36000 kbps bc 1920000 bytes conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 2500 kbps bc 1920000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 2600 kbps bc 192000 bytes conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 10000 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 1000 kbps bc 192000 bytes conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 2400 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1400 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1400 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 1300 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 1500 kbps bc 96000 bytes conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 3000 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 280 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 150 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 150 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 150 kbps bc 192000 bytes conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 50 mbps bc 8192000 bytes conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 200 kbps bc 48000 bytes conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
```

```
    police cir 800 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 400 kbps bc 48000 bytes conform transmit violate drop
  class class-default
    set cos 0
    police cir 400 kbps bc 48000 bytes conform transmit violate drop
```

On Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches, the moderate CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-moderate
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 250 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 19000 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 3000 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 3000 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 3000 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 2000 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 3000 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1500 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1500 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 300 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 400 pps bc 96 packets conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 6000 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 1500 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 50 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 50 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 300 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 20000 pps bc 8192 packets conform transmit violate drop
  class copp-system-p-class-undesirable
```

```
    set cos 0
    police cir 15 pps bc 48 packets conform transmit violate drop
  class copp-system-p-class-fcoe
    set cos 6
    police cir 1500 pps bc 192 packets conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
    police cir 100 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 50 pps bc 48 packets conform transmit violate drop
  class class-default
    set cos 0
    police cir 50 pps bc 48 packets conform transmit violate drop
```

Lenient Default CoPP Policy

On Cisco Nexus 9200 Series switches, the lenient CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-lenient
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 800 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 36000 kbps bc 2560000 bytes conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 2500 kbps bc 2560000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 2600 kbps bc 256000 bytes conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 10000 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 1000 kbps bc 256000 bytes conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 2400 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1400 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1400 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 1300 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 1500 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 3000 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 280 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 150 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-exception-diag
```

```
    set cos 1
    police cir 150 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 150 kbps bc 256000 bytes conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 50 mbps bc 8192000 bytes conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 200 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
    police cir 800 kbps bc 64000 bytes conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 400 kbps bc 64000 bytes conform transmit violate drop
  class class-default
    set cos 0
    police cir 400 kbps bc 64000 bytes conform transmit violate drop
```

On Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches, the lenient CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-lenient
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 250 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 19000 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 3000 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 3000 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 3000 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 2000 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 3000 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1500 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 1500 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 300 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 400 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 6000 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 1500 pps bc 64 packets conform transmit violate drop
```

```
  class copp-system-p-class-exception
    set cos 1
    police cir 50 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 50 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 300 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 20000 pps bc 8192 packets conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 15 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-fcoe
    set cos 6
    police cir 1500 pps bc 256 packets conform transmit violate drop
  class copp-system-p-class-nat-flow
    set cos 7
    police cir 100 pps bc 64 packets conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 50 pps bc 64 packets conform transmit violate drop
  class class-default
    set cos 0
    police cir 50 pps bc 64 packets conform transmit violate drop
```

Dense Default CoPP Policy

On Cisco Nexus 9200 Series switches, the dense CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-dense
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 800 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 4500 kbps bc 1280000 bytes conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 2500 kbps bc 1280000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 370 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 2500 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 2
    police cir 300 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 600 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 1400 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 1
    police cir 350 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
    police cir 750 kbps bc 128000 bytes conform transmit violate drop
```

```
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 750 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-normal-igmp
    set cos 3
    police cir 1400 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-exception-diag
    set cos 1
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 150 kbps bc 128000 bytes conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    set cos 7
    police cir 50 mbps bc 8192000 bytes conform transmit violate drop
  class copp-system-p-class-undesirable
    set cos 0
    police cir 100 kbps bc 32000 bytes conform transmit violate drop
  class copp-system-p-class-l2-default
    set cos 0
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
  class class-default
    set cos 0
    police cir 200 kbps bc 32000 bytes conform transmit violate drop
```

On Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches, the dense CoPP policy has the following configuration:

```
policy-map type control-plane copp-system-p-policy-dense
  class copp-system-p-class-l3uc-data
    set cos 1
    police cir 250 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-critical
    set cos 7
    police cir 2500 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 1200 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 1200 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 1200 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 2
    police cir 1000 pps bc 128 packets conform transmit violate drop
  class copp-system-p-class-l3mc-data
    set cos 1
    police cir 1200 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 750 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 1
    police cir 750 pps bc 32 packets conform transmit violate drop
  class copp-system-p-class-normal-dhcp
    set cos 1
```

```
  police cir 150 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-normal-dhcp-relay-response
  set cos 1
  police cir 200 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-normal-igmp
  set cos 3
  police cir 2500 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-redirect
  set cos 1
  police cir 1500 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-exception
  set cos 1
  police cir 50 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-exception-diag
  set cos 1
  police cir 50 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-monitoring
  set cos 1
  police cir 50 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-l2-unpoliced
  set cos 7
  police cir 20000 pps bc 8192 packets conform transmit violate drop
class copp-system-p-class-undesirable
  set cos 0
  police cir 15 pps bc 32 packets conform transmit violate drop
class copp-system-p-class-fcoe
  set cos 6
  police cir 750 pps bc 128 packets conform transmit violate drop
class copp-system-p-class-l2-default
  set cos 0
  police cir 25 pps bc 32 packets conform transmit violate drop
class class-default
  set cos 0
  police cir 25 pps bc 32 packets conform transmit violate drop
```

Packets Per Second Credit Limit

The aggregate packets per second (PPS) for a given policy (sum of PPS of each class part of the policy) is capped by an upper PPS Credit Limit (PCL). If an increase in PPS of a given class causes a PCL exceed, the configuration is rejected. To increase the desired PPS, the additional PPS beyond PCL should be decreased from other class(es).

Modular QoS Command-Line Interface

CoPP uses the Modular Quality of Service Command-Line Interface (MQC). MQC is a CLI structure that allows you to define a traffic class, create a traffic policy (policy map), and attach the traffic policy to an interface. The traffic policy contains the CoPP feature that will be applied to the traffic class.

SUMMARY STEPS

1. Define a traffic class using the **class-map** command. A traffic class is used to classify traffic.
2. Create a traffic policy using the **policy-map** command. A traffic policy (policy map) contains a traffic class and one or more CoPP features that will be applied to the traffic class. The CoPP features in the traffic policy determine how to treat the classified traffic.
3. Attach the traffic policy (policy map) to the control plane using the **control-plane** and **service-policy** commands.

DETAILED STEPS

Procedure

Step 1 Define a traffic class using the **class-map** command. A traffic class is used to classify traffic.

This example shows how to create a new class-map called copp-sample-class:

```
class-map type control-plane copp-sample-class
```

Step 2 Create a traffic policy using the **policy-map** command. A traffic policy (policy map) contains a traffic class and one or more CoPP features that will be applied to the traffic class. The CoPP features in the traffic policy determine how to treat the classified traffic.

Step 3 Attach the traffic policy (policy map) to the control plane using the **control-plane** and **service-policy** commands.

This example shows how to attach the policy map to the control plane:

```
control-plane
service-policy input copp-system-policy
```

Note
The copp-system-policy is always configured and applied. There is no need to use this command explicitly.

CoPP and the Management Interface

The Cisco NX-OS device supports only hardware-based CoPP, which does not support the management interface (mgmt0). The out-of-band mgmt0 interface connects directly to the CPU and does not pass through the in-band traffic hardware where CoPP is implemented.

On the mgmt0 interface, ACLs can be configured to give or deny access to a particular type of traffic.

Related Topics

Configure IP ACLs, on page 147
Configuring MAC ACLs

Guidelines and Limitations for CoPP

CoPP has the following configuration guidelines and limitations:

- We recommend that you use the strict default CoPP policy initially and then later modify the CoPP policies that are based on the data center and application requirements.
- First-generation Cisco Nexus 9000 Series switches (non -EX/FX/FX2), do not support source-based CoPP. This limitation does not exist for cloud scale ASIC-based Cisco Nexus switches.
- The **match-all** option is not supported in CoPP class-map and it always defaults to the **match-any** option.
- Customizing CoPP is an ongoing process. CoPP must be configured according to the protocols and features that are used in your specific environment and the supervisor features that are required by the server environment. As these protocols and features change, CoPP must be modified.

- We recommend that you continuously monitor CoPP. If drops occur, determine if CoPP dropped traffic unintentionally or in response to a malfunction or attack. In either event, analyze the situation and evaluate the need to modify the CoPP policies.
- All the traffic that you do not specify in the other class maps is put into the last class, the default class. Monitor the drops in this class and investigate if these drops are based on traffic that you do not want or the result of a feature that was not configured and you need to add.
- All broadcast traffic is sent through CoPP logic in order to determine which packets (for example, ARP and DHCP) must be redirected through an access control list (ACL) to the router processor. Broadcast traffic that does not need to be redirected is matched against the CoPP logic, and both conforming and violated packets are counted in the hardware but not sent to the CPU. Broadcast traffic that must be sent to the CPU and broadcast traffic that does not need to be sent to the CPU must be separated into different classes.
- After you have configured CoPP, delete anything that is not being used, such as old class maps and unused routing protocols.
- You must ensure that the CoPP policy does not filter critical traffic such as routing protocols or interactive access to the device. Filtering this traffic could prevent remote access to the Cisco NX-OS device and require a console connection.
- The Cisco NX-OS software does not support egress CoPP or silent mode. CoPP is supported only on ingress (you cannot use the **service-policy output copp** command to the control plane interface).
- You can use the access control entry (ACE) hit counters in the hardware only for ACL logic. Use the software ACE hit counters and the **show access-lists** and **show policy-map type control-plane** commands to evaluate CPU traffic.
- The Cisco NX-OS device hardware performs CoPP on a per-forwarding-engine basis. CoPP does not support distributed policing. Therefore, you should choose rates so that the aggregate traffic does not overwhelm the supervisor module.
- If multiple flows map to the same class, individual flow statistics will not be available.
- If you upgrade from a Cisco NX-OS release that supports the CoPP feature to a Cisco NX-OS release that supports the CoPP feature with other classes for new protocols, you must either run the setup utility using the **setup** command or use the **copp profile** command for the new CoPP classes to be available.
- Before you downgrade from a Cisco NX-OS release that supports the CoPP feature to an earlier Cisco NX-OS release that supports the CoPP feature, you should verify compatibility using the **show incompatibility nxos bootflash:***filename* command. If an incompatibility exists, disable any features that are incompatible with the downgrade image before downgrading the software.
- You cannot disable CoPP. If you attempt to disable it, packets are rate limited at 50 packets per seconds.
- Skip CoPP policy option has been removed from the Cisco NX-OS initial setup utility because using it can impact the control plane of the network.
- Cisco Nexus 9200 Series switches support CoPP policer rates only in multiples of 10 kbps. If a rate is configured that is not a multiple of 10 kbps, the rate is rounded down. For example, the switch uses 50 kbps if a rate of 55 kbps is configured. (The **show policy-map type control-plane** command shows the user configured rate. See Verifying the CoPP Configuration, on page 139 for more information.)
- For Cisco Nexus 9200 Series switches, ip icmp redirect, IPv6 icmp redirect, ip ICMP unreachable, ipv6 icmp unreachable, and mtu-failure use the same TCAM entry, and they will all be classified to the class map where the first exception is present in the policy. In the CoPP strict profile, they are classified to

the class-exception class map. In a different CoPP policy, if the first exception is in a different class map (for example, class-exception-diag), the rest of the exceptions will be classified to the same class map.

- The copp-system-class-fcoe class is not supported for Cisco Nexus 9200 Series switches.
- The following guidelines and limitations apply to static CoPP ACLs:
  - Only Cisco Nexus 9200 Series switches use static CoPP ACLs.
  - Static CoPP ACLs can be remapped to a different CoPP class.
  - Access control entries (ACEs) cannot be modified or removed for static CoPP ACLs.
  - If a CoPP ACL has a static ACL substring, it maps to that type of traffic. For example, if the ACL includes the acl-mac-stp substring, STP traffic classifies to the class map for that ACL.
  - Static CoPP ACLs take priority over dynamic CoPP ACLs, regardless of their position in the CoPP policy, the order in which they are configured, and how they appear in the output of the **show policy-map type control-plane** command.
  - You must have static CoPP ACLs in the CoPP policy. Otherwise, the CoPP policy is rejected.
- Beginning with Cisco Nexus Release 9.2(2), Cisco Nexus 9300-EX, Cisco Nexus 9300-FX Series switches and Cisco Nexus 9500 platform switches support protocol ACL filtering. In this release, IPv6 ACL is not supported.
- Beginning with Cisco NX-OS Release 9.2(3), IPv6 ACL is supported for dynamic CoPP on the Cisco Nexus 9300-EX, Cisco Nexus 9300-FX Series switches, and Cisco Nexus 9500 platform switches.
- The protocol ACL filtering for egress CoPP has the following limitations:
  - Once the egress CoPP ACL is defined, you cannot add or remove an existing rule. This is applicable for all class-maps and policy-maps attached to the egress CoPP ACLs.
  - You cannot override the existing egress CoPP with a new policy. You must remove the existing egress CoPP before you add a new policy.
  - The deny action is not applicable.
  - Every entry is programmed in TCAM and uses a different TCAM space if two MAC or IP ACLs with the same entries are created and bound to either the same or a different class-map.
  - The maximum TCAM carving supported for the egress CoPP is 128 entries (24 entries are reserved and the remaining 104 entries are for egress CoPP, which are all double wide), which can be any of 52 (Ipv4, mac, Ipv6) entries.
  - Policer can be used to drop the traffic completely, with cir and burst as 0.
  - SNMP MIB is not supported.
- When a packet meets multiple exception conditions, CoPP matches the packet based on the order in which the CoPP ACLs are configured and matches it only against a single class. This is an expected CoPP behavior.

  Beginning with Cisco NX-OS Release 9.3(4), the UC FIB MISS exception is counted against the CoPP class (copp-system-p-class-exception). Therefore, if a packet has both, the TTL (accounted user class copp-system-p-class-exception-diag) and the UC FIM MISS exceptions, it is accounted against the UC FIB MISS exception. This behavior occurs because the order of the CoPP classes where the

copp-system-p-class-exception class has an order higher than the copp-system-p-class-exception-diag class. For NX-OS releases earlier to NX-OS Release 9.3(4), the UC FIB MISS exception was not explicitly handled by the CoPP rules.

- CoPP processing comprises of 2 stages: In the first stage, the actual packet size is reused in each class policy, however when the packet enters the second stage, an internal header of 44 bytes is added. This causes an alteration in the conform or violation policies of all the CoPP classes. This limitation is applicable to Cisco Nexus 9300-FX, Nexus 9300-FX2, Nexus 9364C, Nexus 9332C, and 9300-GX platform switches.
- Beginning with Cisco NX-OS Release 10.1(2), CoPP is supported on the Cisco Nexus X9624D-R2 line cards and 9508-FM-R2 switches.
- Beginning with Cisco NX-OS Release 10.1(2), CoPP is supported on the Cisco Nexus 9364D-GX2A and 9332D-GX2B switches.
- Cloudscale IPv6 link-local BGP support requires carving > 512 ing-sup TCAM region (this requires a reload to take effect).
- Beginning with Cisco NX-OS Release 10.3(1)F, CoPP ACL is supported on Cisco Nexus 9808 switches.
  - Beginning with Cisco NX-OS Release 10.4(1)F, CoPP ACL is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, CoPP ACL is supported on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.
- Cisco Nexus 9808/9804 switches have the following limitations for SUP CoPP ACL support:
  - Policer rate is in multiples of 161 PPS at Stage-1.
  - There is no shaper in Stage 0.
  - Stage-2 output is at LC/Module level, and Stage-3 output is at SUP/CPU level.
  - Fabrics/FMs are not involved in in-band path.
  - CoPP policy for Stage-1, Stage-2, and Stage-3 are in PPS.
  - CoPP Stage3 stats gets reset to zero after system switchover.
  - Only policer rate changes are supported in Custom CoPP.
- To avoid traffic loss during traffic impact, configure the CoPP class normal CIR value to 2200 kbps on Cisco Nexus 9300 GX/FX/FX2/FX3, 9504-FM-G, and 9508-FM-G switches and X9716D-GX line cards.
- Beginning with Cisco NX-OS Release 10.3(2)F, source IP based filtering in CoPP is supported on Cisco Nexus 9504 and 9508 modular chassis with R/RX line cards.

Note For IPv6, source IP based filtering is supported up to 24b MSB.

- Beginning with Cisco NX-OS Release 10.4(1)F, CoPP ACL is supported on the Cisco Nexus 9332D-H2R switches.
- Beginning with Cisco NX-OS Release 10.4(2)F, CoPP ACL is supported on the Cisco Nexus 93400LD-H1, and 93108TC-FX3 switches.

- Beginning with Cisco Nexus Release 10.4(3)F, CoPP ACL is supported on the Cisco Nexus 9364C-H1 switches.

CoPP guidelines and limitations for Cisco Nexus N9364E-SG2 switches

- Beginning with Cisco NX-OS Release 10.5(3)F, on the Cisco Nexus N9364E-SG2-Q and N9364E-SG2-O switches, Custom CoPP is supported. Below are the functionalities and limitations:
  - You can modify the policer rates for the copy of inbuilt classes.
  - You can create fully user-defined CoPP classes with user-defined match criteria and policer rates.
  - You can configure policer rates in packets per second (PPS) only.
  - User-defined MAC access lists and source MAC addresses (SMACs) are not supported within Custom CoPP. However, MAC ACLs with well-known destination MAC addresses (DMACs) from the default profile are supported.
  - At least one IPv4 ACL must be present in the Custom CoPP configuration.
  - All Bridge Protocol Data Units (BPDUs) are mapped to a single policer; therefore, distinct policing configurations for protocols such as Cisco Discovery Protocol (CDP), Link Layer Discovery Protocol (LLDP), and Spanning Tree Protocol (STP) are not supported.
  - You can create up to 28 class maps in a CoPP policy.
  - Custom CoPP scale limit (unidimensional) - Maximum number of TCAM entries :
    - IPv4 - 360
    - IPv6 – 180
  - CoPP Consistency checker is not supported for Custom CoPP.
  - Only a single burst value of 126 packets is supported.
  - Destination IP match does not support IPv4 or IPv6 link-local multicast addresses.
  - IPv4 and IPv6 ACLs of link-local multicast entries like OSPF, RIP, EIGRP, and AUTO-RP should be added within the same CoPP class. Similarly, IPv4 and IPv6 ACLs for HSRP and VRRP must also be in the same CoPP class as in the default CoPP profile. Additionally, the ACL name for link-local multicast entries should include the string from the default CoPP policy, such as "acl-ospf" or "acl-eigrp".

CoPP guidelines and limitations for Cisco Nexus N9324C-SE1U switches

- Beginning with Cisco NX-OS Release 10.5(3s), Cisco Nexus N9324C-SE1U switches support only policer rate changes in Custom CoPP. However, Zero CIR is not supported with custom CoPP or hardware rate-limiter. When a user configures zero CIR, it is set to the minimum possible value in the hardware.

Default Settings for CoPP

This table lists the default settings for CoPP parameters.

Table 4: Default CoPP Parameters Settings

Parameters	Default
Default policy	Strict
Default policy	9 policy entries **Note** The maximum number of supported policies with associated class maps is 128.
Scale factor value	1.00

Configuring CoPP

This section describes how to configure CoPP.

Configuring a Control Plane Class Map

You must configure control plane class maps for control plane policies.

You can classify traffic by matching packets based on existing ACLs. The permit and deny ACL keywords are ignored in the matching.

You can configure policies for IP version 4 (IPv4) and IP version 6 (IPv6) packets.

Before you begin

Ensure that you have configured the IP ACLs if you want to use ACE hit counters in the class maps.

SUMMARY STEPS

1. **configure terminal**
2. **class-map type control-plane** [**match-all** | **match-any**] *class-map-name*
3. (Optional) **match access-group name** *access-list-name*
4. (Optional) **match exception** {**ip** | **ipv6**} **icmp redirect**
5. (Optional) **match exception** {**ip** | **ipv6**} **icmp unreachable**
6. (Optional) **match exception** {**ip** | **ipv6**} **option**
7. **match protocol arp**
8. **exit**
9. (Optional) **show class-map type control-plane** [*class-map-name*]
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**class-map type control-plane** [**match-all** | **match-any**] *class-map-name* **Example:** `switch(config)# class-map type control-plane ClassMapA` `switch(config-cmap)#`	Specifies a control plane class map and enters class map configuration mode. The default class matching is match-any. The name can be a maximum of 64 characters long and is case sensitive. **Note** You cannot use class-default, match-all, or match-any as class map names.
Step 3	(Optional) **match access-group name** *access-list-name* **Example:** `switch(config-cmap)# match access-group name MyAccessList`	Specifies matching for an IP ACL. **Note** The permit and deny ACL keywords are ignored in the CoPP matching.
Step 4	(Optional) **match exception** {**ip** | **ipv6**} **icmp redirect** **Example:** `switch(config-cmap)# match exception ip icmp redirect`	Specifies matching for IPv4 or IPv6 ICMP redirect exception packets.
Step 5	(Optional) **match exception** {**ip** | **ipv6**} **icmp unreachable** **Example:** `switch(config-cmap)# match exception ip icmp unreachable`	Specifies matching for IPv4 or IPv6 ICMP unreachable exception packets.
Step 6	(Optional) **match exception** {**ip** | **ipv6**} **option** **Example:** `switch(config-cmap)# match exception ip option`	Specifies matching for IPv4 or IPv6 option exception packets.
Step 7	**match protocol arp** **Example:** `switch(config-cmap)# match protocol arp`	Specifies matching for IP Address Resolution Protocol (ARP) and Reverse Address Resolution Protocol (RARP) packets.
Step 8	**exit** **Example:** `switch(config-cmap)# exit` `switch(config)#`	Exits class map configuration mode.

	Command or Action	Purpose
Step 9	(Optional) **show class-map type control-plane** [*class-map-name*] **Example:** `switch(config)# show class-map type control-plane`	Displays the control plane class map configuration.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring a Control Plane Policy Map

You must configure a policy map for CoPP, which includes policing parameters. If you do not configure a policer for a class, the following default is configured:

- 50 packets per second (pps) with a burst of 32 packets (for Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches)
- 150 kilobits per second (kbps) with a burst of 32,000 bytes (for Cisco Nexus 9200 Series switches)

Before you begin

Ensure that you have configured a control plane class map.

SUMMARY STEPS

1. **configure terminal**
2. **policy-map type control-plane** *policy-map-name*
3. **class** {*class-map-name* [**insert-before** *class-map-name2*] | **class-default**}
4. Enter one of the following commands:
   - **police** [**cir**] {*cir-rate* [*rate-type*]}
   - **police** [**cir**] {*cir-rate* [*rate-type*]} [**bc**] *burst-size* [*burst-size-type*]
   - **police** [**cir**] {*cir-rate* [*rate-type*]]} **conform transmit** [**violate drop**]
5. (Optional) **logging drop threshold** [*drop-count* [**level** *syslog-level*]]
6. (Optional) **set cos** *cos-value*
7. **exit**
8. **exit**
9. (Optional) **show policy-map type control-plane** [**expand**] [**name** *class-map-name*]
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**policy-map type control-plane** *policy-map-name* **Example:** `switch(config)# policy-map type control-plane ClassMapA` `switch(config-pmap)#`	Specifies a control plane policy map and enters policy map configuration mode. The policy map name can have a maximum of 64 characters and is case sensitive.
Step 3	**class** {*class-map-name* [**insert-before** *class-map-name2*] | **class-default**} **Example:** `switch(config-pmap)# class ClassMapA` `switch(config-pmap-c)#`	Specifies a control plane class map name or the class default and enters control plane class configuration mode. The class-default class map is always at the end of the class map list for a policy map.
Step 4	Enter one of the following commands: • **police** [**cir**] {*cir-rate* [*rate-type*]} • **police** [**cir**] {*cir-rate* [*rate-type*]} [**bc**] *burst-size* [*burst-size-type*] • **police** [**cir**] {*cir-rate* [*rate-type*]]} **conform transmit** [**violate drop**] **Example:** `switch(config-pmap-c)# police cir 52000 bc 1000 packets` **Example:** `switch(config-pmap-c)# police cir 3400 kbps bc 200 kbytes`	Specifies the committed information rate (CIR). The rate range is as follows: • 0 to 268435456 pps (for Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches) • 0 to 80000000000 bps/gbps/kbps/mbps (for Cisco Nexus 9200 Series switches) **Note** The CIR rate range starts with 0. In previous releases, the CIR rate range starts with 1. A value of 0 drops the packet. The committed burst (BC) range is as follows: • 1 to 1073741 packets (for Cisco Nexus 9300 and 9500 Series and 3164Q, 31128PQ, 3232C, and 3264Q switches) • 1 to 512000000 bytes/kbytes/mbytes (for Cisco Nexus 9200 Series switches) The **conform transmit** action transmits the packet. **Note** You can specify the BC and conform action for the same CIR.
Step 5	(Optional) **logging drop threshold** [*drop-count* [**level** *syslog-level*]]	Specifies the threshold value for dropped packets and generates a syslog if the drop count exceeds the configured

	Command or Action	Purpose
	Example: `switch(config-pmap-c)# logging drop threshold 100`	threshold. The range for the *drop-count* argument is from 1 to 8000000000 bytes. The range for the *syslog-level* argument is from 1 to 7, and the default level is 4.
Step 6	(Optional) **set cos** *cos-value* **Example:** `switch(config-pmap-c)# set cos 1`	Specifies the 802.1Q class of service (CoS) value. The range is from 0 to 7. The default value is 0.
Step 7	**exit** **Example:** `switch(config-pmap-c)# exit` `switch(config-pmap)#`	Exits policy map class configuration mode.
Step 8	**exit** **Example:** `switch(config-pmap)# exit` `switch(config)#`	Exits policy map configuration mode.
Step 9	(Optional) **show policy-map type control-plane** [**expand**] [**name** *class-map-name*] **Example:** `switch(config)# show policy-map type control-plane`	Displays the control plane policy map configuration.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring a Control Plane Class Map, on page 125

Configuring the Control Plane Service Policy

You can configure one or more policy maps for the CoPP service policy.

Note When you try to change the CoPP policy and apply a custom CoPP policy, it is configured in the hardware as non-atomic, and the following system message appears:

```
This operation can cause disruption of control traffic. Proceed (y/n)?  [no] y
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT24-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT23-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT21-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT25-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT26-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT22-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
2013 Nov 13 23:16:46 switch %ACLQOS-SLOT4-5-ACLQOS_NON_ATOMIC: Non atomic ACL/QoS policy
update done for CoPP
```

Before you begin

Ensure that you have configured a control plane policy map.

SUMMARY STEPS

1. **configure terminal**
2. **control-plane**
3. [**no**] **service-policy input** *policy-map-name*
4. **exit**
5. (Optional) **show running-config copp** [**all**]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**control-plane** **Example:** `switch(config)# control-plane` `switch(config-cp)#`	Enters control plane configuration mode.

	Command or Action	Purpose
Step 3	[**no**] **service-policy input** *policy-map-name* **Example:** `switch(config-cp)# service-policy input PolicyMapA`	Specifies a policy map for the input traffic. Repeat this step if you have more than one policy map. You cannot disable CoPP. If you enter the **no** form of this command, packets are rate limited at 50 packets per seconds.
Step 4	**exit** **Example:** `switch(config-cp)# exit` `switch(config)#`	Exits control plane configuration mode.
Step 5	(Optional) **show running-config copp** [**all**] **Example:** `switch(config)# show running-config copp`	Displays the CoPP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring a Control Plane Policy Map, on page 127

Configuring the CoPP Scale Factor Per Line Card

You can configure the CoPP scale factor per line card.

The scale factor configuration is used to scale the policer rate of the applied CoPP policy for a particular line card. The accepted value is from 0.10 to 2.00. You can increase or reduce the policer rate for a particular line card without changing the current CoPP policy. The changes are effective immediately, so you do not need to reapply the CoPP policy.

SUMMARY STEPS

1. **configure terminal**
2. **control-plane**
3. **scale-factor** *value* **module** *multiple-module-range*
4. (Optional) **show policy-map interface control-plane**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	
Step 2	**control-plane** **Example:** `switch(config)# control-plane` `switch(config-cp)#`	Enters control plane configuration mode.
Step 3	**scale-factor** *value* **module** *multiple-module-range* **Example:** `switch(config-cp)# scale-factor 1.10 module 1-2`	Configures the policer rate per line card. The allowed scale factor value is from 0.10 to 2.00. When the scale factor value is configured, the policing values are multiplied by the corresponding scale factor value of the module, and it is programmed in the particular module. To revert to the default scale factor value of 1.00, use the **no scale-factor** *value* **module** *multiple-module-range* command, or explicitly set the default scale factor value to 1.00 using the **scale-factor 1 module** *multiple-module-range* command.
Step 4	(Optional) **show policy-map interface control-plane** **Example:** `switch(config-cp)# show policy-map interface control-plane`	Displays the applied scale factor values when a CoPP policy is applied.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Changing or Reapplying the Default CoPP Policy

You can change to a different default CoPP policy, or you can reapply the same default CoPP policy.

SUMMARY STEPS

1. [**no**] **copp profile** [**strict** | **moderate** | **lenient** | **dense**]
2. (Optional) **show copp status**
3. (Optional) **show running-config copp**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	[**no**] **copp profile** [**strict**	**moderate**	**lenient**	**dense**] **Example:** `switch(config)# copp profile moderate`	Applies the CoPP best practice policy. You cannot disable CoPP. If you enter the **no** form of this command, packets are rate limited at 50 packets per seconds.

	Command or Action	Purpose
Step 2	(Optional) **show copp status** **Example:** `switch(config)# show copp status`	Displays the CoPP status, including the last configuration operation and its status. This command also enables you to verify that the CoPP best practice policy is attached to the control plane.
Step 3	(Optional) **show running-config copp** **Example:** `switch(config)# show running-config copp`	Displays the CoPP configuration in the running configuration.

Related Topics

Changing or Reapplying the Default CoPP Policy Using the Setup Utility, on page 145

Copying the CoPP Best Practice Policy

The CoPP best practice policy is read-only. If you want to modify its configuration, you must copy it.

SUMMARY STEPS

1. **copp copy profile** {**strict** | **moderate** | **lenient** | **dense**} {**prefix** | **suffix**} *string*
2. (Optional) **show copp status**
3. (Optional) **show running-config copp**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**copp copy profile** {**strict**	**moderate**	**lenient**	**dense**} {**prefix**	**suffix**} *string* **Example:** `switch# copp copy profile strict prefix abc`	Creates a copy of the CoPP best practice policy. CoPP renames all class maps and policy maps with the specified prefix or suffix.
Step 2	(Optional) **show copp status** **Example:** `switch# show copp status`	Displays the CoPP status, including the last configuration operation and its status. This command also enables you to verify that the copied policy is not attached to the control plane.
Step 3	(Optional) **show running-config copp** **Example:** `switch# show running-config copp`	Displays the CoPP configuration in the running configuration, including the copied policy configuration.

Protocol ACL Filtering for Egress CoPP

The protocol ACL filtering for egress CoPP enables the NX-OS swtich to filter all traffic to control plane based on the host MAC, IPv4, and IPv6 address.

Configuring ARP ACL Filtering for Egress CoPP

You can configure MAC ACL filtering at egress CoPP.

Before you begin

Ensure that you have configured a control plane policy map.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **hardware access-list tcam region erg-copp** *size*
3. **copy running-config startup-config**
4. **reload**
5. **configure terminal**
6. **mac access-list mac-foo-1**
7. **class-map type control-plane** [**match-all** | **match-any**] *class-map-name*
8. (Optional) **match access-group name** *access-list-name*
9. **policy-map type control-plane** *policy-map-name*
10. **class** {*class-map-name* [**insert-before** *class-map-name2*] | **class-default**}
11. Enter one of the following commands:
    - **police** [**cir**] {*cir-rate* [*rate-type*]}
    - **police** [**cir**] {*cir-rate* [*rate-type*]} [**bc**] *burst-size* [*burst-size-type*]
    - **police** [**cir**] {*cir-rate* [*rate-type*]]} **conform transmit** [**violate drop**]
12. **control-plane Dynamic mode**
13. **service-policy-dynamic input** *policy-map-name*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	[**no**] **hardware access-list tcam region erg-copp** *size* **Example:**	Configures the size of the CoPP TCAM region.

	Command or Action	Purpose
	`switch(config)# hardware access-list tcam region erg-copp 128`	
Step 3	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.
Step 4	**reload** **Example:** `switch(config)# reload`	Reloads the device. **Note** The new size values are effective only after you enter **copy running-config startup-config** + **reload** or reload all line card modules.
Step 5	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 6	**mac access-list mac-foo-1** **Example:** `switch# mac access-list mac-foo-1` `switch(config-mac-acl)#`	
Step 7	**class-map type control-plane** [**match-all** | **match-any**] *class-map-name* **Example:** `switch(config)# class-map type control-plane match-any c-map2` `switch(config-cmap)#`	Specifies a control plane class map and enters class map configuration mode. The default class matching is match-any. The name can be a maximum of 64 characters long and is case-sensitive.
Step 8	(Optional) **match access-group name** *access-list-name* **Example:** `switch(config-cmap)# match access-group name IP-foo-1`	
Step 9	**policy-map type control-plane** *policy-map-name* **Example:** `switch(config)# policy-map type control-plane ClassMapA` `switch(config-pmap)#`	Specifies a control plane policy map and enters policy map configuration mode. The policy map name can have a maximum of 64 characters and is case-sensitive.
Step 10	**class** {*class-map-name* [**insert-before** *class-map-name2*] | **class-default**} **Example:** `switch(config-pmap)# class ClassMap2` `switch(config-pmap-c)#`	Specifies a control plane class map name or the class default and enters control plane class configuration mode. The class-default class map is always at the end of the class map list for a policy map.

	Command or Action	Purpose
Step 11	Enter one of the following commands: • **police** [**cir**] {*cir-rate* [*rate-type*]} • **police** [**cir**] {*cir-rate* [*rate-type*]} [**bc**] *burst-size* [*burst-size-type*] • **police** [**cir**] {*cir-rate* [*rate-type*]]} **conform transmit** [**violate drop**] **Example:** `switch(config-pmap-c)# police cir 52000 bc 1000 packets`	Specifies the committed information rate (CIR). The rate range is as follows: The committed burst (BC) range is as follows:
Step 12	**control-plane Dynamic mode** **Example:** `switch(config)# control-plane dynamic` `switch(config-cp-dyn)#`	Enters the control plane dynamic configuration mode.
Step 13	**service-policy-dynamic input** *policy-map-name* **Example:** `switch(config-cp-dyn)# service-policy-dynamic input PolicyMap1`	Specifies a policy map for the input traffic.

Configuring IP ACL Filtering for Egress CoPP

You can configure IP ACL filtering at egress CoPP.

Before you begin

Ensure that you have configured a control plane policy map.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **hardware access-list tcam region erg-copp** *size*
3. **copy running-config startup-config**
4. **reload**
5. **configure terminal**
6. **ip access-list IP-foo-1**
7. **permit tcp access-list IP-foo-1** *eq bgp*
8. **class-map type control-plane** [**match-all** | **match-any**] *class-map-name*
9. **match access-group name** *access-list-name*
10. **policy-map type control-plane** *policy-map-name*
11. **class** {*class-map-name* [**insert-before** *class-map-name2*] | **class-default**}
12. Enter one of the following commands:
    - **police** [**cir**] {*cir-rate* [*rate-type*]}
    - **police** [**cir**] {*cir-rate* [*rate-type*]} [**bc**] *burst-size* [*burst-size-type*]
    - **police** [**cir**] {*cir-rate* [*rate-type*]]} **conform transmit** [**violate drop**]

13. **control-plane Dynamic mode**
14. **service-policy-dynamic input** *policy-map-name*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	[**no**] **hardware access-list tcam region erg-copp** *size* **Example:** `switch(config)# hardware access-list tcam region erg-copp 128`	Configures the size of the egress CoPP TCAM region.
Step 3	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.
Step 4	**reload** **Example:** `switch(config)# reload`	Reloads the device. **Note** The new size values are effective only after you enter **copy running-config startup-config** + **reload** or reload all line card modules.
Step 5	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 6	**ip access-list IP-foo-1** **Example:** `switch# ip access-list mac-foo-1` `switch(config-acl)#`	
Step 7	**permit tcp access-list IP-foo-1** *eq bgp* **Example:** `switch(config-acl)# 10 permit tcp 10.1.1.1/32 10.1.1.2/32 eq bgp`	
Step 8	**class-map type control-plane** [**match-all** | **match-any**] *class-map-name* **Example:**	Specifies a control plane class map and enters class map configuration mode. The default class matching is match-any. The name can be a maximum of 64 characters long and is case sensitive.

	Command or Action	Purpose
	`switch(config)# class-map type control-plane match-any c-map2` `switch(config-cmap)#`	
Step 9	**match access-group name** *access-list-name* **Example:** `switch(config-cmap)# match access-group name IP-foo-1`	
Step 10	**policy-map type control-plane** *policy-map-name* **Example:** `switch(config)# policy-map type control-plane ClassMapA` `switch(config-pmap)#`	Specifies a control plane policy map and enters policy map configuration mode. The policy map name can have a maximum of 64 characters and is case sensitive.
Step 11	**class** {*class-map-name* [**insert-before** *class-map-name2*] | **class-default**} **Example:** `switch(config-pmap)# class ClassMap2` `switch(config-pmap-c)#`	Specifies a control plane class map name or the class default and enters control plane class configuration mode. The class-default class map is always at the end of the class map list for a policy map.
Step 12	Enter one of the following commands: • **police** [**cir**] {*cir-rate* [*rate-type*]} • **police** [**cir**] {*cir-rate* [*rate-type*]} [**bc**] *burst-size* [*burst-size-type*] • **police** [**cir**] {*cir-rate* [*rate-type*]]} **conform transmit** [**violate drop**] **Example:** `switch(config-pmap-c)# police cir 52000 bc 1000 packets` **Example:** `switch(config-pmap-c)# police cir 3400 kbps bc 200 kbytes`	Specifies the committed information rate (CIR). The rate range is as follows: The committed burst (BC) range is as follows:
Step 13	**control-plane Dynamic mode** **Example:** `switch(config)# control-plane dynamic` `switch(config-cp-dyn)#`	Enters the control plane dynamic configuration mode.
Step 14	**service-policy-dynamic input** *policy-map-name* **Example:** `switch(config-cp-dyn)# service-policy-dynamic input PolicyMap1`	Specifies a policy map for the input traffic. ENd

Verifying the CoPP Configuration

To display CoPP configuration information, perform one of the following tasks:

Command	Purpose
show policy-map type control-plane [**expand**] [**name** *policy-map-name*]	Displays the control plane policy map with associated class maps and CIR and BC values.
show policy-map interface control-plane	Displays the policy values with associated class maps and drops per policy or class map. It also displays the scale factor values when a CoPP policy is applied. When the scale factor value is the default (1.00), it is not displayed. **Note** The scale factor changes the CIR and BC values internally on each module, but the display shows the configured CIR and BC values only. The actual applied value on a module is the scale factor multiplied by the configured value.
show class-map type control-plane [*class-map-name*]	Displays the control plane class map configuration, including the ACLs that are bound to this class map.

Command	Purpose
show copp diff profile {**strict** | **moderate** | **lenient** | **dense**} [**prior-ver**] **profile** {**strict** | **moderate** | **lenient** | **dense**} **show copp diff profile**	Displays the difference between two CoPP best practice policies. When you do not include the prior-ver option, this command displays the difference between two currently applied default CoPP best practice policies (such as the currently applied strict and currently applied moderate policies). When you include the prior-ver option, this command displays the difference between a currently applied default CoPP best practice policy and a previously applied default CoPP best practice policy (such as the currently applied strict and the previously applied lenient policies).
show copp profile {**strict** | **moderate** | **lenient** | **dense**}	Displays the details of the CoPP best practice policy, along with the classes and policer values.
show running-config aclmgr [**all**]	Displays the user-configured access control lists (ACLs) in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration.
show running-config copp [**all**]	Displays the CoPP configuration in the running configuration.
show startup-config aclmgr [**all**]	Displays the user-configured access control lists (ACLs) in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration.

Displaying the CoPP Configuration Status

SUMMARY STEPS

1. switch# **show copp status**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch# **show copp status**	Displays the configuration status for the CoPP feature.

Example

This example shows how to display the CoPP configuration status:

```
switch# show copp status
```

CoPP Consistency Checker

Beginning with Cisco NX-OS Release 10.5(2)F, CoPP consistency checker is introduced to ensure the consistency of all SUP related ACLs across the Software, HAL, and Hardware layers. This feature is supported on Cisco Nexus 9300-FX3/GX/GX2/HX, Nexus 9808, and Nexus 9804 series switches.

Note Beginning with Cisco NX-OS Release 10.5(3)F, the **show consistency-checker copp extended module** command is deprecated. For more information on the CoPP consistency checker, see the **Consistency Checker Commands** section of *Cisco Nexus 9000 Series NX-OS Troubleshooting Guide, Release 10.5(x)*.

SUMMARY STEPS

1. switch# **show consistency-checker copp extended module** <*module_no*> **[brief | detail]**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch# **show consistency-checker copp extended module** <*module_no*> **[brief | detail]**	Performs CoPP consistency check. **brief** – shows consistency checker structured output in brief. **detail** - shows consistency checker structured output in detail.

Monitoring CoPP

SUMMARY STEPS

1. switch# **show policy-map interface control-plane**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch# **show policy-map interface control-plane**	Displays packet-level statistics for all classes that are part of the applied CoPP policy. Statistics are specified in terms of OutPackets (packets admitted to the control plane) and DropPackets (packets dropped because of rate limiting).

Example

This example shows how to monitor CoPP:

```
switch# show policy-map interface control-plane
Control Plane

  Service-policy  input: copp-system-p-policy-strict

    class-map copp-system-p-class-critical (match-any)
      set cos 7
      police cir 19000 pps , bc 128 packets
      module 4 :
        transmitted 373977 packets;
        dropped 0 packets;
```

Monitoring CoPP with SNMP

Beginning with Cisco Nexus Release 9.2(3), CoPP supports the Cisco class-based QoS MIB (cbQoSMIB). All CoPP elements can now be monitored (but not modified) using SNMP. This feature applies only to policies and their subelements (such as classes, match rules, and set actions) that are attached to the control plane. Elements of policies that are not in service on the control plane are not visible through SNMP.

The following cbQoSMIB tables are supported:

- ccbQosServicePolicy
- cbQosInterfacePolicy
- cbQosObjects

- cbQosPolicyMapCfg
- cbQosClassMapCfg
- cbQosMatchStmtCfg
- cbQosPoliceCfg
- cbQosSetCfg

Note SNMP MIB is not supported for Dynamic CoPP.

Clearing the CoPP Statistics

SUMMARY STEPS

1. (Optional) switch# **show policy-map interface control-plane**
2. switch# **clear copp statistics**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	(Optional) switch# **show policy-map interface control-plane**	Displays the currently applied CoPP policy and per-class statistics.
Step 2	switch# **clear copp statistics**	Clears the CoPP statistics. **Note** On Cisco Nexus 9800 Series switches with N9K-X9836DM-A and N9K-X98900CD-A line cards, N9K-C9232E-B1, N9364E-SG2, and N9324C-SE1U switches, the **clear copp statistics** command clears the CoPP statistics for all the policers except the **hardware rate-limiter** policers.

Example

This example shows how to clear the CoPP statistics for your installation:

```
switch# show policy-map interface control-plane
switch# clear copp statistics
```

Configuration Examples for CoPP

This section includes example CoPP configurations.

CoPP Configuration Example

The following example shows how to configure CoPP using IP ACLs and MAC ACLs:

```
configure terminal
ip access-list copp-system-p-acl-igmp
permit igmp any 10.0.0.0/24

ip access-list copp-system-p-acl-msdp
permit tcp any any eq 639

mac access-list copp-system-p-acl-arp
permit any any 0x0806

ip access-list copp-system-p-acl-tacas
permit udp any any eq 49

ip access-list copp-system-p-acl-ntp
permit udp any 10.0.1.1/23 eq 123

ip access-list copp-system-p-acl-icmp
permit icmp any any

class-map type control-plane match-any copp-system-p-class-critical
match access-group name copp-system-p-acl-igmp
match access-group name copp-system-p-acl-msdp

class-map type control-plane match-any copp-system-p-class-normal
match access-group name copp-system-p-acl-icmp
match exception ip icmp redirect
match exception ip icmp unreachable
match exception ip option

policy-map type control-plane copp-system-p-policy

class copp-system-p-class-critical
police cir 19000 pps bc 128 packets conform transmit violate drop

class copp-system-p-class-important
police cir 500 pps bc 128 packets conform transmit violate drop

class copp-system-p-class-normal
police cir 300 pps bc 32 packets conform transmit violate drop

class class-default
police cir 50 pps bc 32 packets conform transmit violate drop

control-plane
service-policy input copp-system-p-policy
```

Create CoPP class and associate ACL:

```
class-map type control-plane copp-arp-class
match access-group name copp-arp-acl
```

Add the class to the CoPP policy:

```
policy-map type control-plane copp-system-policy
class copp-arp-class
police pps 500
```

The following example shows to customize COPP limit:

```
copp copy profile strict suffix CUSTOMIZED-COPP
policy-map type control-plane copp-policy-strict-CUSTOMIZED-COPP
class copp-class-redirect-CUSTOMIZED-COPP
police cir 1500 mbps bc 125 mbytes conform transmit violate drop
control-plane
service-policy input copp-policy-strict-CUSTOMIZED-COPP
```

Changing or Reapplying the Default CoPP Policy Using the Setup Utility

The following example shows how to change or reapply the default CoPP policy using the setup utility.

```
switch# setup

         ---- Basic System Configuration Dialog ----

This setup utility will guide you through the basic configuration of
the system. Setup configures only enough connectivity for management
of the system.


*Note: setup is mainly used for configuring the system initially,
when no configuration is present. So setup always assumes system
defaults and not the current system configuration values.


Press Enter at anytime to skip a dialog. Use ctrl-c at anytime
to skip the remaining dialogs.

Would you like to enter the basic configuration dialog (yes/no): yes

Do you want to enforce secure password standard (yes/no)[y]: <CR>

  Create another login account (yes/no) [n]: n

  Configure read-only SNMP community string (yes/no) [n]: n

  Configure read-write SNMP community string (yes/no) [n]: n

  Enter the switch name : <CR>

  Enable license grace period? (yes/no) [n]: n

  Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]: n

  Configure the default gateway? (yes/no) [y]: n

  Configure advanced IP options? (yes/no) [n]: <CR>

  Enable the telnet service? (yes/no) [n]: y

  Enable the ssh service? (yes/no) [y]: <CR>

    Type of ssh key you would like to generate (dsa/rsa) : <CR>
```

```
  Configure the ntp server? (yes/no) [n]: n

  Configure default interface layer (L3/L2) [L3]: <CR>

  Configure default switchport interface state (shut/noshut) [shut]: <CR>

  Configure best practices CoPP profile (strict/moderate/lenient/dense/skip) [strict]:
strict


The following configuration will be applied:
  password strength-check
  no license grace-period
  no telnet server enable
  no system default switchport
  system default switchport shutdown
  policy-map type control-plane copp-system-p-policy

Would you like to edit the configuration? (yes/no) [n]: <CR>

Use this configuration and save it? (yes/no) [y]: y

switch#
```

Changing CoPP Policy limit

The following exampls shows to change CoPP limit to set PTP state stable across PTP interfaces.

```
copp copy profile strict suffix CUSTOMIZED-COPP
policy-map type control-plane copp-policy-strict-CUSTOMIZED-COPP
class copp-class-redirect-CUSTOMIZED-COPP
police cir 1500 mbps bc 125 mbytes conform transmit violate drop
control-plane
service-policy input copp-policy-strict-CUSTOMIZED-COPP
```

Additional References for CoPP

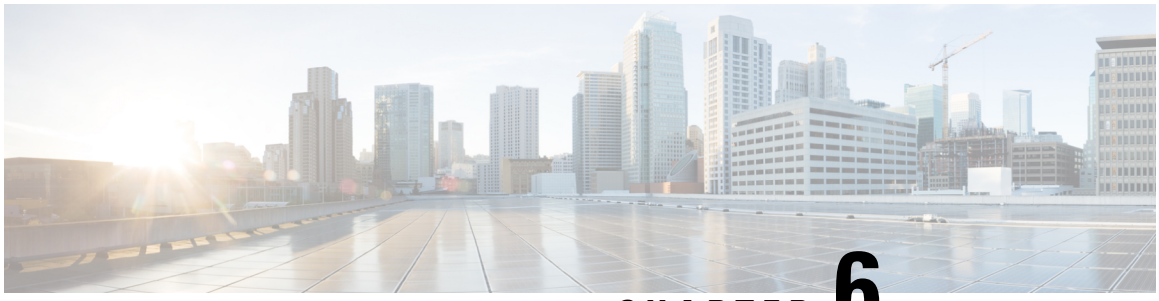
This section provides additional information related to implementing CoPP.

Related Documents

Related Topic	Document Title
Licensing	*Cisco NX-OS Licensing Guide*

Standards

Standards	Title
RFC 2698	A Two Rate Three Color Marker



CHAPTER 6

Configure IP ACLs

This chapter describes how to configure IP access control lists (ACLs) on Cisco NX-OS devices.

Unless otherwise specified, the term IP ACL refers to IPv4 and IPv6 ACLs.

About ACLs

An ACL is an ordered set of rules that you can use to filter traffic. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When the device determines that an ACL applies to a packet, it tests the packet against the conditions of all rules. The first matching rule determines whether the packet is permitted or denied. If there is no match, the device applies the applicable implicit rule. The device continues processing packets that are permitted and drops packets that are denied.

You can use ACLs to protect networks and specific hosts from unnecessary or unwanted traffic. For example, you could use ACLs to disallow HTTP traffic from a high-security network to the Internet. You could also use ACLs to allow HTTP traffic but only to specific sites, using the IP address of the site to identify it in an IP ACL.

ACL Types and Applications

The device supports the following types of ACLs for security traffic filtering:

IPv4 ACLs
The device applies IPv4 ACLs only to IPv4 traffic.
IPv6 ACLs
The device applies IPv6 ACLs only to IPv6 traffic.
MAC ACLs
The device applies MAC ACLs only to non-IP traffic.

IP and MAC ACLs have the following types of applications:

Port ACL
Filters Layer 2 traffic
MAC ACL with UDF-based match
Filters MAC ACLs with UDF-based match
Router ACL
Filters Layer 3 traffic
VLAN ACL
Filters VLAN traffic
VTY ACL
Filters virtual teletype (VTY) traffic

This table summarizes the applications for security ACLs.

Table 5: Security ACL Applications

Application	Supported Interfaces	Types of ACLs Supported
Port ACL	• Layer 2 interfaces • Layer 2 Ethernet port-channel interfaces When a port ACL is applied to a trunk port, the ACL filters traffic on all VLANs on the trunk port.	• IPv4 ACLs • IPv4 ACLs with UDF-based match • IPv6 ACLs • IPv6 ACLs with UDF-based match • MAC ACLs • MAC ACLs with UDF-based match
Router ACL	• VLAN interfaces • Physical Layer 3 interfaces • Layer 3 Ethernet subinterfaces • Layer 3 Ethernet port-channel interfaces • Management interfaces **Note** You must enable VLAN interfaces globally before you can configure a VLAN interface.	• IPv4 ACLs • IPv6 ACLs **Note** MAC ACLs are supported on Layer 3 interfaces only if you enable MAC packet classification.
VLAN ACL	• VLANs	• IPv4 ACLs • IPv6 ACLs • MAC ACLs

Application	Supported Interfaces	Types of ACLs Supported
VTY ACL	• VTYs	• IPv4 ACLs • IPv6 ACLs

Related Topics

About VLAN ACLs
About MAC ACLs

Order of ACL Application

When the device processes a packet, it determines the forwarding path of the packet. The path determines which ACLs that the device applies to the traffic. The device applies the ACLs in the following order:

1. Port ACL
2. Ingress VACL
3. Ingress router ACL
4. Ingress VTY ACL
5. Egress VTY ACL
6. Egress router ACL
7. Egress VACL

If the packet is bridged within the ingress VLAN, the device does not apply router ACLs.

Figure 2: Order of ACL Application

The following figure shows the order in which the device applies ACLs.

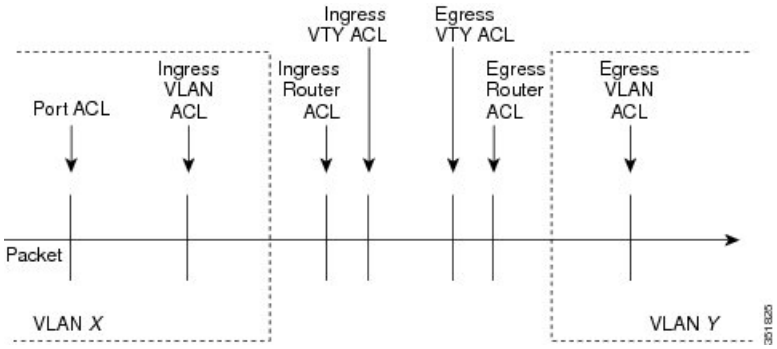
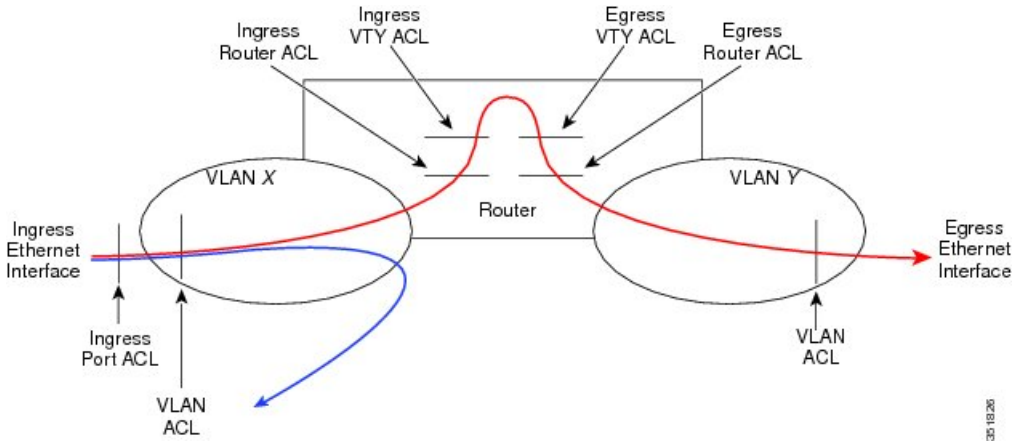


Figure 3: ACLs and Packet Flow

The following figure shows where the device applies ACLs, depending upon the type of ACL. The red path indicates a packet sent to a destination on a different interface than its source. The blue path indicates a packet that is bridged within its VLAN.

The device applies only the applicable ACLs. For example, if the ingress port is a Layer 2 port and the traffic is on a VLAN that is a VLAN interface, a port ACL and a router ACL both can apply. In addition, if a VACL is applied to the VLAN, the device applies that ACL too.



About Rules

Rules are what you create, modify, and remove when you configure how an ACL filters network traffic. Rules appear in the running configuration. When you apply an ACL to an interface or change a rule within an ACL that is already applied to an interface, the supervisor module creates ACL entries from the rules in the running configuration and sends those ACL entries to the applicable I/O module. Depending upon how you configure the ACL, there may be more ACL entries than rules, especially if you implement policy-based ACLs by using object groups when you configure rules.

You can create rules in access-list configuration mode by using the **permit** or **deny** command. The device allows traffic that matches the criteria in a permit rule and blocks traffic that matches the criteria in a deny rule. You have many options for configuring the criteria that traffic must meet in order to match the rule.

This section describes some of the options that you can use when you configure a rule.

Protocols for IP ACLs and MAC ACLs

IPv4, IPv6, and MAC ACLs allow you to identify traffic by protocol. For your convenience, you can specify some protocols by name. For example, in an IPv4 or IPv6 ACL, you can specify ICMP by name.

You can specify any protocol by number. In MAC ACLs, you can specify protocols by the EtherType number of the protocol, which is a hexadecimal number. For example, you can use 0x0800 to specify IP traffic in a MAC ACL rule.

In IPv4 and IPv6 ACLs, you can specify protocols by the integer that represents the Internet protocol number.

Source and Destination

In each rule, you specify the source and the destination of the traffic that matches the rule. You can specify both the source and destination as a specific host, a network or group of hosts, or any host. How you specify the source and destination depends on whether you are configuring IPv4 ACLs, IPv6 ACLs, or MAC ACLs.

Implicit Rules for IP and MAC ACLs

IP and MAC ACLs have implicit rules, which means that although these rules do not appear in the running configuration, the device applies them to traffic when no other rules in an ACL match. When you configure the device to maintain per-rule statistics for an ACL, the device does not maintain statistics for implicit rules.

All IPv4 ACLs include the following implicit rule:

```
deny ip any any
```

This implicit rule ensures that the device denies unmatched IP traffic.

All IPv6 ACLs include the following implicit rule:

```
deny ipv6 any any
```

This implicit rule ensures that the device denies unmatched IPv6 traffic.

Note

- IPv6 Neighbor Discovery packets (Router Solicitation, and Router Advertisement) will not be permitted due to the implicit **deny ipv6 any any** rule of an IPv6 ACL.
- You must add the following rules explicitly to allow IPv6 Neighbor Discovery packets in the Cisco Nexus 93180YC-EX, Nexus 93180YC-FX, Nexus 93240YC-FX2, Nexus 93360YC-FX2, Nexus 9336C-FX2, Nexus 9336C-FX2-E, Nexus 93180YC-FX3, N9K-C9316D-GX, N9K-C93600CD-GX, Nexus 9364C-GX, N9K-C9332D-GX2B, Nexus 9364C and Nexus 9332C platform switches:
  - **permit icmp any any router-advertisement**
  - **permit icmp any any router-solicitation**
- Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages do not match under the implicit rule. The following commands are required to match the NS or NA IPv6 traffic.
  - **permit/deny icmp any any nd-na**
  - **permit/deny icmp any any nd-ns**

All MAC ACLs include the following implicit rule:

```
deny any any protocol
```

This implicit rule ensures that the device denies the unmatched traffic, regardless of the protocol specified in the Layer 2 header of the traffic.

Additional Filtering Options

You can identify traffic by using additional options. These options differ by ACL type. The following list includes most but not all additional filtering options:

- IPv4 ACLs support the following additional filtering options:
  - Layer 4 protocol
  - TCP and UDP ports
  - ICMP types and codes
  - IGMP types
  - Precedence level
  - Differentiated Services Code Point (DSCP) value
  - TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set
  - Established TCP connections

- Packet length

- IPv6 ACLs support the following additional filtering options:
  - Layer 4 protocol
  - Encapsulating Security Payload
  - Payload Compression Protocol
  - Stream Control Transmission Protocol (SCTP)
  - SCTP, TCP, and UDP ports
  - ICMP types and codes
  - DSCP value
  - TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set
  - Established TCP connections
  - Packet length

- MAC ACLs support the following additional filtering options:
  - Layer 3 protocol (Ethertype)
  - VLAN ID
  - Class of Service (CoS)

- Beginning Cisco NX-OS Release 9.2(4), IPv4 ACLs and IPv6 in Cisco Nexus 9500 platform switches with N9K-X96136YC-R, N9K-X9636C-R, and N9K-X9636C-RX line cards and N9K-C9504-FM-R fabric module support the following additional filtering options:
  - TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set
  - Established TCP connections

Note

- TCP flag options are correctly processed by Netstack rather than the kernel (KStack), due to the kernel's lack of support for TCP flags. Additionally, the following syslog message is generated:

  ```
  <HOSTNAME> %NPACL-2-IPT_WARNING: npacl [<#>] WARNING: Mgmt ACL: <ACL>
  Seq:<Seq#> has ACL option: tcp-flags that is not supported in kernel
  stack. Hence that option is not added in its filter rule.
  ```

- The **tcp-flags-mask** option is not supported.

Sequence Numbers

The device supports sequence numbers for rules. Every rule that you enter receives a sequence number, either assigned by you or assigned automatically by the device. Sequence numbers simplify the following ACL tasks:

Adding new rules between existing rules

By specifying the sequence number, you specify where in the ACL a new rule should be positioned. For example, if you need to insert a rule between rules numbered 100 and 110, you could assign a sequence number of 105 to the new rule.

Removing a rule

Without using a sequence number, removing a rule requires that you enter the whole rule, as follows:

```
switch(config-acl)# no permit tcp 10.0.0.0/8 any
```

However, if the same rule had a sequence number of 101, removing the rule requires only the following command:

```
switch(config-acl)# no 101
```

Moving a rule

With sequence numbers, if you need to move a rule to a different position within an ACL, you can add a second instance of the rule using the sequence number that positions it correctly, and then you can remove the original instance of the rule. This action allows you to move the rule without disrupting traffic.

If you enter a rule without a sequence number, the device adds the rule to the end of the ACL and assigns a sequence number that is 10 greater than the sequence number of the preceding rule to the rule. For example, if the last rule in an ACL has a sequence number of 225 and you add a rule without a sequence number, the device assigns the sequence number 235 to the new rule.

In addition, Cisco NX-OS allows you to reassign sequence numbers to rules in an ACL. Resequencing is useful when an ACL has rules numbered contiguously, such as 100 and 101, and you need to insert one or more rules between those rules.

Logical Operators and Logical Operation Units

IP ACL rules for TCP and UDP traffic can use logical operators to filter traffic based on port numbers. Cisco NX-OS supports logical operators in only the ingress direction.

The device stores operator-operand couples in registers called logical operator units (LOUs). The LOU usage for each type of operator is as follows:

eq
Is never stored in an LOU

gt
Uses 1 LOU

lt
Uses 1 LOU

neq
Uses 1 LOU

range
Uses 1 LOU

Note For range operators, LOU threshold configuration is used to control how the port range is expanded when configuring an ACL entry. If you want to use the LOU operator when the number of the ACL rules exceed the configured threshold value, run the following command: **hardware access-list lou resource threshold** <*x*>, wherein <*x*> denotes the number of ACL rules to be used before the LOU threshold is reached. The range value for <*x*> is 1 to 50, and the default value for LOU threshold is 5.

ACL Logging

The ACL logging feature monitors ACL flows and logs statistics.

A flow is defined by the source interface, protocol, source IP address, source port, destination IP address, and destination port values. The statistics maintained for a flow include the number of forwarded packets (for each flow that matches the permit conditions of the ACL entry) and dropped packets (for each flow that matches the deny conditions of the ACL entry).

Beginning with Cisco NX-OS Release 10.4(3)F, ACL logging for Security Group ACL (SGACL) is provided on Cisco Nexus 9300-FX3/GX/GX2/H2R platform switches.

For SGACL, a flow is defined by the security group tag (SGT), destination group tag (DGT), source MAC (SMAC), destination MAC (DMAC), SGACL permit/deny information, physical interface on which packet arrived, and hit counts for that particular SGACL flow apart from the basic 5 tuples. To enable the SGACL logging, see Configuring ACL Logging, on page 209.

Time Ranges

You can use time ranges to control when an ACL rule is in effect. For example, if the device determines that a particular ACL applies to traffic arriving on an interface, and a rule in the ACL uses a time range that is not in effect, the device does not compare the traffic to that rule. The device evaluates time ranges based on its clock.

When you apply an ACL that uses time ranges, the device updates the affected I/O module whenever a time range referenced in the ACL starts or ends. Updates that are initiated by time ranges occur on a best-effort priority. If the device is especially busy when a time range causes an update, the device may delay the update by up to a few seconds.

IPv4, IPv6, and MAC ACLs support time ranges. When the device applies an ACL to traffic, the rules in effect are as follows:

- All rules without a time range specified
- Rules with a time range that includes the second when the device applies the ACL to traffic

The device supports named, reusable time ranges, which allows you to configure a time range once and specify it by name when you configure many ACL rules. Time range names have a maximum length of 64 alphanumeric characters.

A time range contains one or more rules. The two types of rules are as follows:

Absolute

A rule with a specific start date and time, specific end date and time, both, or neither. The following items describe how the presence or absence of a start or end date and time affect whether an absolute time range rule is active:

- Start and end date and time both specified—The time range rule is active when the current time is later than the start date and time and earlier than the end date and time.
- Start date and time specified with no end date and time—The time range rule is active when the current time is later than the start date and time.
- No start date and time with end date and time specified—The time range rule is active when the current time is earlier than the end date and time.
- No start or end date and time specified—The time range rule is always active.

For example, you could prepare your network to allow access to a new subnet by specifying a time range that allows access beginning at midnight of the day that you plan to place the subnet online. You can use that time range in ACL rules that apply to the subnet. After the start time and date have passed, the device automatically begins applying the rules that use this time range when it applies the ACLs that contain the rules.

Periodic

A rule that is active one or more times per week. For example, you could use a periodic time range to allow access to a lab subnet only during work hours on weekdays. The device automatically applies ACL rules that use this time range only when the range is active and when it applies the ACLs that contain the rules.

Note The order of rules in a time range does not affect how a device evaluates whether a time range is active. Cisco NX-OS includes sequence numbers in time ranges to make editing the time range easier.

Time ranges also allow you to include remarks, which you can use to insert comments into a time range. Remarks have a maximum length of 100 alphanumeric characters.

The device determines whether a time range is active as follows:

- The time range contains one or more absolute rules—The time range is active if the current time is within one or more absolute rules.
- The time range contains one or more periodic rules—The time range is active if the current time is within one or more periodic rules.
- The time range contains both absolute and periodic rules—The time range is active if the current time is within one or more absolute rules and within one or more periodic rules.

When a time range contains both absolute and periodic rules, the periodic rules can only be active when at least one absolute rule is active.

Policy-Based ACLs

The device supports policy-based ACLs (PBACLs), which allow you to apply access control policies across object groups. An object group is a group of IP addresses or a group of TCP or UDP ports. When you create a rule, you specify the object groups rather than specifying IP addresses or ports.

Using object groups when you configure IPv4 or IPv6 ACLs can help reduce the complexity of updating ACLs when you need to add or remove addresses or ports from the source or destination of rules. For example,

if three rules reference the same IP address group object, you can add an IP address to the object instead of changing all three rules.

PBACLs do not reduce the resources required by an ACL when you apply it to an interface. When you apply a PBACL or update a PBACL that is already applied, the device expands each rule that refers to object groups into one ACL entry per object within the group. If a rule specifies the source and destination both with object groups, the number of ACL entries created on the I/O module when you apply the PBACL is equal to the number of objects in the source group multiplied by the number of objects in the destination group.

The following object group types apply to port, router, policy-based routing (PBR), and VLAN ACLs:

IPv4 Address Object Groups

Can be used with IPv4 ACL rules to specify source or destination addresses. When you use the **permit** or **deny** command to configure a rule, the **addrgroup** keyword allows you to specify an object group for the source or destination.

IPv6 Address Object Groups

Can be used with IPv6 ACL rules to specify source or destination addresses. When you use the **permit** or **deny** command to configure a rule, the **addrgroup** keyword allows you to specify an object group for the source or destination.

Protocol Port Object Groups

Can be used with IPv4 and IPv6 TCP and UDP rules to specify source or destination ports. When you use the **permit** or **deny** command to configure a rule, the **portgroup** keyword allows you to specify an object group for the source or destination.

Note Policy-based routing (PBR) ACLs do not support deny access control entries (ACEs) or **deny** commands to configure a rule.

Kernel Stack ACL

The Kernel Stack ACL is a common CLI infrastructure to configure ACLs for management of inband and outband components.

The Kernel Stack ACL uses NX-OS ACL CLI to secure management applications on management and front panel ports. Configuring a single ACL must be able to secure all management applications on NX-OS.

Kernel Stack ACL is the component that fixes the manual intervention of the user and automatically programs iptable entries when the ACL is applied to mgmt0 interface.

The following is an example for configuring Kernel Stack ACL:

```
swtich# conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# ip access-list kacl1
switch(config-acl)# statistics per-entry
switch(config-acl)# 10 deny tcp any any eq 443
switch(config-acl)# 20 permit ip any any
switch(config-acl)# end
switch#

switch(config-if)# interface mgmt0
switch(config-if)#   ip access-group acl1 in
```

```
switch(config-if)#   ipv6 traffic-filter acl6 in
switch(config-if)#
```

```
switch# sh ip access-lists kacl1
IP access list kacl1
statistics per-entry
10 deny tcp any any eq 443 [match=136]
20 permit ip any any [match=44952]
switch(config)#
```

The following is the Kernel Stack filtering for iptables entries based on the configuration:

```
bash-4.4# ip netns exec management iptables -L -n -v --line-numbers
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
num pkts bytes target prot opt in out source destination
1 9 576 DROP tcp -- * * 0.0.0.0/0 0.0.0.0/0 tcp dpt:443
2 0 0 ACCEPT all -- * * 0.0.0.0/0 0.0.0.0/0
3 0 0 DROP all -- * * 0.0.0.0/0 0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
num pkts bytes target prot opt in out source destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num pkts bytes target prot opt in out source destination
bash-4.4#
```

The following are the limitations for the Kernel Stack ACL support:

- This feature is supported only on mgmt0 interface and not on other inband interfaces.
- Five tuples (protocol, source-ip, destination-ip, source-port, and destination-port) of the ACL entry are programmed in the iptables. Rest of the options provided in the ACL entry are not programmed in the iptables and throws a warning syslog in such instances.

  For example, "WARNING: Some ACL options are not supported in kstack. Only partial rule will be installed".
- If the device user has host bash access, then the user can manually update the iptables. This update could potentially corrupt the iptable rules for which they are programmed.
- The verified maximum number of ACEs is 100 for IPv4 traffic and an additional 100 for IPv6 traffic. Throughput may be impacted if more than this scale is applied.

Statistics and ACLs

The device can maintain global statistics for each rule that you configure in IPv4, IPv6, and MAC ACLs. If an ACL is applied to multiple interfaces, the maintained rule statistics are the sum of packet matches (hits) on all the interfaces on which that ACL is applied.

Note The device does not support interface-level ACL statistics.

For each ACL that you configure, you can specify whether the device maintains statistics for that ACL, which allows you to turn ACL statistics on or off as needed to monitor traffic filtered by an ACL or to help troubleshoot the configuration of an ACL.

The device does not maintain statistics for implicit rules in an ACL. For example, the device does not maintain a count of packets that match the implicit **deny ip any any** rule at the end of all IPv4 ACLs. If you want to

maintain statistics for implicit rules, you must explicitly configure the ACL with rules that are identical to the implicit rules.

Related Topics

Monitoring and Clearing IP ACL Statistics, on page 217
Implicit Rules for IP and MAC ACLs, on page 150

Atomic ACL Updates

By default, when a supervisor module of a Cisco Nexus 9000 Series device updates an I/O module with changes to an ACL, it performs an atomic ACL update. An atomic update does not disrupt traffic that the updated ACL applies to; however, an atomic update requires that an I/O module that receives an ACL update has enough available resources to store each updated ACL entry in addition to all pre-existing entries in the affected ACL. After the update occurs, the additional resources used for the update are freed. If the I/O module lacks the required resources, the device generates an error message and the ACL update to the I/O module fails.

If an I/O module lacks the resources required for an atomic update, you can disable atomic updates by using the **no hardware access-list update atomic** command; however, during the brief time required for the device to remove the preexisting ACL and implement the updated ACL, traffic that the ACL applies to is dropped by default.

If you want to permit all traffic that an ACL applies to while it receives a nonatomic update, use the **hardware access-list update default-result permit** command.

This example shows how to disable atomic updates to ACLs:

```
switch# config t
switch(config)# no hardware access-list update atomic
```

This example shows how to permit affected traffic during a nonatomic ACL update:

```
switch# config t
switch(config)# hardware access-list update default-result permit
```

This example shows how to revert to the atomic update method:

```
switch# config t
switch(config)# no hardware access-list update default-result permit
switch(config)# hardware access-list update atomic
```

Session Manager Support for IP ACLs

Session Manager supports the configuration of IP and MAC ACLs. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration.

ACL TCAM Regions

You can change the size of the ACL ternary content addressable memory (TCAM) regions in the hardware.

On Cisco Nexus 9300 and 9500 Series switches and Cisco Nexus 3164Q, 31128PQ, 3232C, and 3264Q switches, the egress TCAM size is 1K, divided into four 256 entries. On Cisco Nexus NFE2-enabled devices

(such as the Cisco Nexus 3232C and 3264Q switches), the ingress TCAM size is 6K, divided into twelve 512 slices. Three slices are in one group. On other Cisco Nexus 9300 and 9500 Series switches and the 3164Q and 31128PQ switches, the ingress TCAM size is 4K, divided into eight 256 slices and four 512 slices. A slice is the unit of allocation. A slice can be allocated to one region only. For example, a 512-size slice cannot be used to configure two features of size 256 each. Similarly, a 256-size slice cannot be used to configure two features of size 128 each. The IPv4 TCAM regions are single wide. The IPv6, QoS, MAC, CoPP, and system TCAM regions are double wide and consume double the physical TCAM entries. For example, a logical region size of 256 entries actually consumes 512 physical TCAM entries.

You can create IPv6, port ACLs, VLAN ACLs, and router ACLs, and you can match IPv6 and MAC addresses for QoS. However, Cisco NX-OS cannot support all of them simultaneously. You must remove or reduce the size of the existing TCAM regions (TCAM carving) to enable the IPv6, MAC, or other desired TCAM regions. For every TCAM region configuration command, the system evaluates if the new change can be fit in the TCAM. If not, it reports an error, and the command is rejected. You must remove or reduce the size of existing TCAM regions to make room for new requirements.

On Cisco Nexus 9200 Series switches, the egress TCAM size is 2K, and the ingress TCAM size is 4K. The concepts of TCAM slices and single- and double-wide regions do not apply to these switches. For example, the ing-ifacl region can host IPv4, IPv6, or MAC type entries. IPv4 and MAC types occupy one TCAM entry whereas IPv6 types occupy two TCAM entries.

For N9K-X9636C-RX, when PACL uses external TCAM region, the internal TCAM needs to take 2K for ifacl and the ingress RACL-IPv4 can take upto 2044. Additional four entries are required when egress PACL external TCAM region is used.

ACL TCAM region sizes have the following guidelines and limitations:

- To enable RACL or PACL on existing TCAM regions, you must carve the TCAM region beyond 12, 288.
- On Cisco Nexus 9300 Series switches, the X9536PQ, X9564PX, and X9564TX line cards are used to enforce the QoS classification policies applied on 40G ports. It has 768 TCAM entries available for carving in 256-entry granularity. These region names are prefixed with "ns-".
- For the X9536PQ, X9564PX, and X9564TX line cards, only the IPv6 TCAM regions consume double-wide entries. The rest of the TCAM regions consume single-wide entries.
- When a VACL region is configured, it is configured with the same size in both the ingress and egress directions. If the region size cannot fit in either direction, the configuration is rejected.
- On Cisco Nexus 9200 Series switches, the ing-sup region occupies a minimum size of 512 entries, and the egr-sup region occupies a minimum size of 256 entries. These regions cannot be configured to lesser values. Any region size can be carved with a value only in multiples of 256 entries (with the exception of the span region, which can be carved only in multiples of 512 entries).
- RACL v6, CoPP, and multicast have default TCAM sizes and these TCAM sizes must be non-zero on the following Cisco Nexus 9504 and Cisco Nexus 9508 line cards to avoid line card failure during reload:
  - N9K-X96136YC-R
  - N9K-X9636C-RX
  - N9K-X9636Q-R
  - N9K-X9636C-R

- When the egress RACL is beyond 4K, the TCAM carving configuration has to be ingress RACL (RACL) + egress RACL (e-racl) summing to 20480. See the following TCAM carving example:

```
hardware access-list tcam region ifacl 0
hardware access-list tcam region ipv6-ifacl 0
hardware access-list tcam region mac-ifacl 0
hardware access-list tcam region racl 0
hardware access-list tcam region ipv6-racl 0
hardware access-list tcam region span 0
hardware access-list tcam region redirect_v4 0
hardware access-list tcam region redirect_v6 0
hardware access-list tcam region e-racl 20480
```

- You can partially use IPv6 RACL with IPv6 IFCAL. This is applicable to Cisco Nexus N9K-C9508 and N9K-C9504 with N9K-X96136YC-R, N9K-X9636C-R, N9K-X9636Q-R, and N9K-X9636C-RX line cards.
- The N9K-X9636C-R and N9K-X9636Q-R line cards support a maximum TCAM region size of 12K. If you configure a greater number, the TCAM region is set to 12K.
- The N9K-X96136YC-R and N9K-X9636C-R line cards support egress RACL of 2K.
- The N9K-X9636C-RX line card supports a TCAM region size beyond 12K. If you configure the RACL IPv4 TCAM region to 100K, the TCAM region is set to 12K for the N9K-X9636C-R and N9K-X9636Q-R line cards and to 100K for the N9K-X9636C-RX line card, provided you have set all of the other TCAM regions and made space for the N9K-X9636C-R and N9K-X9636Q-R line cards to accommodate 12K.
- In addition to the internal TCAM, an external TCAM of 128K is available on the N9K-X9636C-RX line card.

The following table summarizes the regions that need to be configured for a given feature to work. The region sizes should be selected based on the scale requirements of a given feature.

Table 6: Features per ACL TCAM Region

Feature Name	Region Name
Port ACL	ifacl: For IPv4 port ACLs ifacl-udf: For UDFs on IPv4 port ACLs ing-ifacl: For ingress IPv4, IPv6, and MAC port ACLs ing-ifacl: For ingress IPv4, IPv6, MAC port ACLs, and MAC port ACLs with UDF ipv6-ifacl: For IPv6 port ACLs mac-ifacl: For MAC port ACLs

Feature Name	Region Name
Port QoS (QoS classification policy applied on Layer 2 ports or port channels)	qos, qos-lite, rp-qos, rp-qos-lite, ns-qos, e-qos, or e-qos-lite: For classifying IPv4 packets ing-l2-qos: For classifying ingress Layer 2 packets ipv6-qos, rp-ipv6-qos, ns-ipv6-qos, or e-ipv6-qos: For classifying IPv6 packets mac-qos, rp-mac-qos, ns-mac-qos, or e-mac-qos: For classifying non-IP packets **Note** For traffic that needs to be classified on 40G ports on Cisco Nexus 9300 Series switches, you must carve the qos regions and the corresponding ns-*qos regions.
VACL	vacl: For IPv4 packets ipv6-vacl: For IPv6 packets mac-vacl: For non-IP packets
VLAN QoS (QoS classification policy applied on a VLAN)	vqos or ns-vqos: For classifying IPv4 packets ipv6-vqos or ns-ipv6-vqos: For classifying IPv6 packets ing-l3-vlan-qos: For classifying ingress Layer 3, VLAN, and SVI QoS packets mac-vqos or ns-mac-vqos: For classifying non-IP packets **Note** For traffic that needs to be classified on 40G ports on Cisco Nexus 9300 Series switches, you must carve the qos regions and the corresponding ns-*qos regions.
RACL	egr-racl: For egress IPv4 and IPv6 RACLs e-racl: For egress IPv4 RACLs e-ipv6-racl: For egress IPv6 RACLs ing-racl: For ingress IPv4 and IPv6 RACLs racl: For IPv4 RACLs racl-lite: For IPv4 RACLs racl-udf: For UDFs on IPv4 RACLs ipv6-racl: For IPv6 RACLs

Feature Name	Region Name
Layer 3 QoS (QoS classification policy applied on Layer 3 ports or port channels)	l3qos, l3qos-lite, or ns-l3qos: For classifying IPv4 packets ipv6-l3qos or ns-ipv6-l3qos: For classifying IPv6 packets **Note** For traffic that needs to be classified on 40G ports on Cisco Nexus 9300 Series switches, you must carve qos regions and the corresponding ns-*qos regions.
VLAN source or VLAN filter SPAN (for Cisco Nexus 9500 or 9300 Series switches) Rx SPAN on 40G ports (for Cisco Nexus 9300 Series switches only)	span
SPAN filters	ifacl: For filtering IPv4 traffic on Layer 2 (switch port) source interfaces. ifacl-udf: For UDFs on IPv4 port ACLs ipv6-ifacl: For filtering IPv6 traffic on Layer 2 (switch port) source interfaces. mac-ifacl: For filtering Layer 2 traffic on Layer 2 (switch port) source interfaces. racl-udf: For UDFs on IPv4 RACLs vacl: For filtering IPv4 traffic on VLAN sources. ipv6-vacl: For filtering IPv6 traffic on VLAN sources. mac-vacl: For filtering Layer 2 traffic on VLAN sources. racl: For filtering IPv4 traffic on Layer 3 interfaces. ipv6-racl: For filtering IPv6 traffic on Layer 3 interfaces. ing-l2-span-filter: For filtering ingress Layer 2 SPAN traffic ing-l3-span-filter: For filtering ingress Layer 3 and VLAN SPAN traffic
SVI counters **Note** This region enables the packet counters for Layer 3 SVI interfaces.	svi

Feature Name	Region Name
BFD, DHCP relay, or DHCPv6 relay	redirect **Note** BFD uses the ing-sup region while DHCPv4 relay, DHCPv4 snooping, and DHCPv4 client use the ing-redirect region.
CoPP	copp **Note** The region size cannot be 0.
System-managed ACLs	system **Note** The region size cannot be changed.
vPC convergence **Note** This region boosts the convergence times when a vPC link goes down and traffic needs to be redirected to the peer link.	vpc-convergence **Note** Setting this region size to 0 might affect the convergence times of vPC link failures.
Fabric extender (FEX)	fex-ifacl, fex-ipv6-ifacl, fex-ipv6-qos, fex-mac-ifacl, fex-mac-qos, fex-qos, fex-qos-lite
Dynamic ARP inspection (DAI)	arp-ether
IP source guard (IPSG)	ipsg
Multicast PIM Bidir	mcast_bidir
Static MPLS	mpls
Network address translation (NAT)	nat
NetFlow	ing-netflow
OpenFlow	openflow
sFlow	sflow
Supervisor modules	egr-sup: Egress supervisor ing-sup: Ingress supervisor

Related Topics

Configuring ACL TCAM Region Sizes, on page 184
Configuring TCAM Carving, on page 194

Maximum Label Sizes Supported for ACL Types

Cisco NX-OS switches support the following label sizes for the corresponding ACL types:

Table 7: ACL Types and Maximum Label Sizes

ACL Types	Direction	Label	Label Type
RACL/PBR/VACL/ L3-VLAN QoS/L3-VLAN SPAN ACL	Ingress	62	BD
PACL/L2 QoS/L2 SPAN ACL	Ingress	62[1]	IF
RACL/VACL/L3-VLAN QoS	Egress	254	BD
L2 QoS	Egress	31	IF
RACL	Ingress	510	L3

[1] The label size can be increased to 62 when you enter the **hardware access-list tcam label ing-ifacl 6** command and reload the switch.

Beginning with Cisco NX-OS Release 9.3(6), the **hardware access-list tcam label ing-ifacl 6** command is introduced and is applicable only for Cisco Nexus 9300-FX platform switches.

Beginning with Cisco NX-OS Release 10.1(2), the **hardware access-list tcam label ing-ifacl 6** command is also supported on Cisco Nexus 9300-FX2 platform switches.

Beginning with Cisco NX-OS Release 10.4(3)F, the **hardware access-list tcam label ing-ifacl 6** command is also supported on Cisco Nexus 9300-FX3, GX, GX2, H2R, H1 platform switches.

Prerequisites for IP ACLs

IP ACLs have the following prerequisites:

- You must be familiar with IP addressing and protocols to configure IP ACLs.
- You must be familiar with the interface types that you want to configure with ACLs.

Guidelines and Limitations for IP ACLs

IP ACLs have the following configuration guidelines and limitations:

Note For more information about the Cisco Nexus 9000 series platform switches that support various features spanning from release 7.0(3)I7(1) to the current release, refer to Nexus Switch Platform Support Matrix.

- Beginning with Cisco NX-OS Release 10.2(1)F, Egress PACL is supported on the Cisco Nexus 9364D-GX2A, and 9332D-GX2B switches.
- If you configure egress PACL and egress VACL on the same interface, only egress VACL is enabled.

- We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources that are required by the configuration are available before committing them to the running configuration. This recommendation is especially useful for ACLs that include more than 1000 rules. For more information about Session Manager, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.
- Configuring a IPv4 PACL in the range of 12K to 64K is supported on Cisco Nexus 9500 Series switches with -RX line cards.
- Duplicate ACL entries with different sequence numbers are allowed in the configuration. However, these duplicate entries are not programmed in the hardware access-list.
- Only 62 unique ACLs can be configured. Each ACL takes one label. If the same ACL is configured on multiple interfaces, the same label is shared. If each ACL has unique entries, the ACL labels are not shared, and the label limit is 62. This is not applicable to Cisco Nexus 9500 Series switches and Cisco Nexus 3636C-R switches.
- Usually, ACL processing for IP packets occurs on the I/O modules, which use hardware that accelerates ACL processing. In some circumstances, processing occurs on the supervisor module, which can result in slower ACL processing, especially during processing that involves an ACL with many rules. Management interface traffic is always processed on the supervisor module. If IP packets in any of the following categories are exiting a Layer 3 interface, they are sent to the supervisor module for processing:
  - Packets that fail the Layer 3 maximum transmission unit check and therefore require fragmenting.
  - IPv4 packets that have IP options (other IP packet header fields following the destination address field).
  - IPv6 packets that have extended IPv6 header fields.

  Rate limiters prevent redirected packets from overwhelming the supervisor module.
- When you apply an ACL that uses time ranges, the device updates the ACL entries whenever a time range that is referenced in an ACL entry starts or ends. Updates that are initiated by time ranges occur on a best-effort priority. If the device is especially busy when a time range causes an update, the device may delay the update by up to a few seconds.
- To apply an IP ACL to a VLAN interface, you must have enabled VLAN interfaces globally. For more information about VLAN interfaces, see the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*.
- The VTY ACL feature restricts all traffic for all VTY lines. You cannot specify different traffic restrictions for different VTY lines. Any router ACL can be configured as a VTY ACL.
- An egress VTY ACL (an IP ACL applied to the VTY line in the outbound direction) prevents the switch from copying files using a file transfer protocol (TFTP, FTP, SCP, SFTP, etc.) unless the file transfer protocol is explicitly permitted within the egress VTY ACL.
- When you apply an undefined ACL to an interface, the system treats the ACL as empty and permits all traffic.
- IP tunnels do not support ACLs or QoS policies.
- The following guidelines apply to ACLs for VXLANs:
  - Ingress port ACLs applied on a Layer 2 port for traffic in the access to a network direction (Layer 2 to Layer 3 encapsulation path) are supported on the inner payload.

- We recommend using port ACLs on the access side to filter out traffic entering the overlay network.
- Ingress router ACLs applied on an uplink Layer 3 interface matching on the inner or outer payload in the network to access direction (Layer 3 to Layer 2 decapsulation path) are not supported.
- Egress router ACLs applied on an uplink Layer 3 interface matching on the inner or outer payload in the access to a network direction (encapsulation path) are not supported.

- Cisco Nexus 9300 and 9500 Series switches, and Cisco Nexus 9200 and 9300-EX Series switches have the following limitations for ACL options that can be used on VXLAN traffic:
  - Does not support egress port ACLs applied on a Layer 2 port for traffic in the network to access direction (decapsulation path).
  - Supports ingress VACLs applied on a VLAN for traffic in the access to a network direction (encapsulation path).
  - Supports egress VACLs applied on a VLAN for traffic in the network to access direction (decapsulation path).
  - Supports ingress RACLs applied on a tenant or server facing SVI for traffic in the access to network direction (encapsulation path).
  - Supports egress RACLs applied on a tenant or server facing SVI for traffic in the network to access direction (decapsulation path).
- IPv6 ACL logging is not supported for egress PACL.
- IPv4 ACL logging in the egress direction is not supported.
- ACL logging for VACLs is not supported.
- ACL logging applies to port ACLs configured by the **ip port access-group** command and to router ACLs configured by the **ip access-group** command only.
- The total number of IPv4 ACL flows is limited to a user-defined maximum value to prevent DoS attacks. If this limit is reached, no new logs are created until an existing flow finishes.
- The number of syslog entries that are generated by IPv4 ACL logging is limited by the configured logging level of the ACL logging process. If the number of syslog entries exceeds this limit, the logging facility might drop some logging messages. Therefore, IPv4 ACL logging should not be used as a billing tool or as an accurate source of the number of matches to an ACL.
- Egress router ACLs are not supported on Cisco Nexus 9300 Series switch uplink ports.
- For Network Forwarding Engine (NFE)-enabled switches, ingress RACLs matching the outer header of the tunnel interface are not supported.
- If the same QoS policy and ACL are applied to multiple interfaces, the label is shared only when the QoS policy is applied with the no-stats option.
- The switch hardware does not support range checks (Layer 4 operations) in the egress TCAM. Therefore, ACL and QoS policies with a Layer 4 operations-based classification must be expanded to multiple entries in the egress TCAM.

  The switch hardware supports only up to 16 Layer 4 operands. Make sure to consider this limitation for egress TCAM space planning. For more information see the Logical Operators and Logical Operation Units, on page 153 section.

- For Cisco Nexus X96136YC-R, X9636C-RX, X9636C-RX, and X9636Q-R line cards, run the **hardware profile acl-eg-ext module all** command before applying **eg-racl-v6** configuration on a SVI or port object on an EoR switch.
- TCAM resources are shared in the following scenarios:
  - When a routed ACL is applied to multiple switched virtual interfaces (SVIs) in the ingress direction.
  - When a routed ACL is applied to multiple layer 2 interfaces in the ingress or egress direction.
- TCAM resources are not shared in the following scenarios:
  - VACL (VLAN ACL) is applied to multiple VLANs.
  - Routed ACL is applied to multiple SVIs in the egress direction.
- Access-lists based on HTTP methods are not supported on the Cisco Nexus 9200, 9300-EX, 9300-FX, 9300-FX2, 9300-FXP, and 9300-GX platform switches and the 9500 switches with the X9700-EX, and X9700-FX line cards. For all these switches, you must use UDF-based ACLs.
- HTTP methods are not supported on FEX ports.
- The following guidelines and limitations apply to Cisco Nexus 9200 and 9300-EX Series switches:
  - Egress MAC ACLs are not supported.
  - Egress RACLs are not supported on an interface if the packet matches the outer header of the tunnel interface on the device where the tunnel is originating the traffic.
  - Ingress RACLs matching the outer header of the tunnel interface are not supported.
  - IP length-based matches are not supported.
  - All ACL-based features cannot be enabled at the same time.
  - 16 Layer 4 operations are supported.
  - Layer 4 operations are not supported on egress TCAM regions.
  - The MAC compression table size is 4096 + 512 overflow TCAM.
  - An overlap of MAC addresses and MAC masks is rejected.
  - The ACL log rate limiter does not have any hardware counters for transmitted or dropped packets.
  - The ACL log rate limiter is implemented at the per-TCAM entry level (instead of using aggregated rate limiting), and the default is 1 pps.
  - The Network Address Translation (NAT) exception counters are zero.
  - Only PACL redirects are supported for TAP aggregation. VACL redirects are not supported.
  - Only three of the following four features can be supported at a time: DHCPv4 snooping or relay, DHCPv6 relay, ARP snooping, VXLAN. The first three configured features take effect, but the fourth one will fail because all three bridge domain label bits are already in use.
- The following guidelines and limitations apply to Cisco Nexus 9364C-GX, Cisco Nexus 9316D-GX, and Cisco Nexus 93600CD-GX series switches:
  - The MAC compression table size is 4096 + 512 overflow TCAM.

- An overlap of MAC addresses and MAC masks is rejected.

- Cisco Nexus 9504 and Cisco Nexus 9508 switches with -R line cards do not support the following TCAM:
  - All FEX related TCAM
  - All xxx-lite related TCAM region
  - Ranger related TCAM
  - All FCoE related TCAM

- TCAM carving configuration of the ing-netflow region can be performed on -FX line cards. -EX line cards have a default ing-netflow region TCAM carving of 1024 and cannot be configured otherwise. For ports on the -EX and -FX line cards, the suggested maximum for the ing-netflow region is 1024.
- On the Cisco Nexus 9200 and 9300-EX platform switches, router ACL with the ACL log option will not take into effect as the sup-redirect ACLs have higher priority for the traffic that is destined to SUP.
- On the Cisco Nexus 9300-GX platform switches, dot1q VLAN with ACL redirect supports only the VLAN IDs from 1 to 511.

  If PACL redirect or TapAgg is configured, the **switchport access vlan** *vlan-id* command supports only the vlan IDs from 1 to 511.
- For traffic destined to the FHRP VIP and ingressing on FHRP standby which matches an ACL log enabled ACE designed to permit the traffic, the Cisco Nexus 9000 Series switch drops this packet.
- For Cisco Nexus 3172TQ, 3172TQ-XL, 36180YC-R, and 3636C-R switches, when there is a SVI and subinterface matching the same VLAN tag, the traffic that gets routed out through a subinterface gets dropped if the access-list is configured on that SVI. This is due to an ASIC limitation and egress router ACL on L3 subinterfaces is not supported due to this limitation.
- Cisco Nexus 9364D-GX2A, and 9332D-GX2B switches do not support the following on egress router ACL:
  - UDF to support ICMP Type Match.
  - ACL log-on egress
  - Egress IPv4 router ACL with additional filter option tcp/udp ports with lt/gt
  - Egress IPv4 router ACL with additional filter option tcp/udp ports with neq
  - Egress IPv4 router ACL with extra filter option tcp/udp ports with range
  - Egress IPv4 router ACL with a flag
  - Egress router ACL on an external TCAM
  - Egress PACL support
  - Statistics support
  - Label sharing

- Cisco Nexus 9500 platform switches with -R and -RX line cards have the following guidelines:

- Atomic ACL update is supported for all the ingress ACL features except for the Multihop BFD and CoPP features.
- Atomic ACL update is not supported for the egress ACL features.
- Label sharing is supported only for the same policy on different interfaces within the same ASIC.
- In Cisco NX-OS Release 9.2(3), ACL statistics are supported for the following:
  - PACL - IPv4 (including system ACL for both, internal, and external TCAM)
  - Router ACL - IPv4 (internal TCAM for both, ingress RACL-IPv4 and egress RACL-IPv4)
  - Only 2K counters are supported in the egress.
- ACL statistics are not supported for the following:
  - BFD
  - DHCP - IPv4 and IPv6
  - PACL-MAC
  - PACL- IPv6
  - PBR - IPv4 and IPv6
  - RACL-IPv6
  - RACL-IPv4 when using an external TCAM

- ACL label sharing works on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2 and 9300C platform switches with following limitations:
  - ACL statistics are disabled by default. However, statistics are enabled by default only for QoS policies.
  - ACL target (port / VLAN / SVI etc.,) must be on the same slice, and port.
  - Additionally, label space is shared with following features:
    - Ingress RACL, PBR and Ingress L3 QoS
    - Ingress PACL, Ingress L2 QoS
    - Egress RACL, Egress QoS

Note For label sharing to work, ensure that the same set of features are supported on interfaces.

- When you enable the counters for the ACL TCAM entries using the hardware profile acl-stats module *xx* command, the input discard field in the show interface is always zero. This limitation is applicable only to the Cisco Nexus 9500 platform switches with -R and -RX line cards.
- Cisco Nexus 9500 platform switches with -R and -RX line cards do not support the following:

- Egress atomic updates
- Egress router ACL on external TCAM
- Egress router ACL with UDF
- Router ACL v6 counters for both egress and ingress
- Egress and ingress router ACL IPv6 with l4 ops
- Egress router ACL on subinterface
- Egress and ingress router ACL with IPv6 ICMP Type and Code
- IPv6 ingress router ACL with tcp-flag
- IPv4 router ACL with extra option

- In Cisco NX-OS Release 9.3(3), egress IPv4 RACLs support the following on Cisco Nexus 9504 and 9508 switches with -R and -RX line cards:
  - TCP flags
  - ICMP Type and Code
  - ACL logs

- IPv6 Egress ACL support the following on Cisco Nexus 9504 and 9508 switches with -R and -RX line cards:
  - Layer 4 Protocol
  - TCP flags
  - Fragment
  - ACL logs for IPv4
  - IPv6 header fields

  The following limitations are applicable for the IPv6 egress ACL:
  - Port groups and Layer 4 Operations are not supported. The ranges expand to multiple ACE entries for eg-racl-ipv6.
  - Address group defined host is not supported.
  - Counters are not supported.
  - Egress IPv6 RACL is not supported on sub-interfaces and external TCAM.
  - Atomic updates are not supported.
  - VXLAN is not supported when acl-eg-ext is enabled.

- PACL redirects are supported on Cisco Nexus 9300-GX switches. The following limitations are applicable:
  - To support PACL redirects, you must run the **mode tap-agg** command on the ingress tap interface.

  - To support the MPLS strip feature, the **mpls strip** and the **hardware acl tap-agg** commands must be configured and the switch reloaded.
  - For double tag VLAN, the range of the second VLAN is 2-510.
  - MPLS strip with dot1q VLAN is not supported.
  - The redirect port carries the tag if the incoming packet is tagged, even when the redirect port is configured as an access port.
  - TapAgg redirect is not supported for deny ACE.

- In Cisco NX-OS Release 10.1(2), PACL redirect feature is not supported in mixed mode on Cisco Nexus X9736C-FX, X9788TC-FX, and X97160YC-EX line cards.
- Egress ACL does not support traffic that is destined to the IP address of the second VLAN in inter-VLAN routing flow.
- In Cisco Nexus 9300-EX/FX/FX2/FX3/GX platform switches and 93180YC-FX switches, RACLs cannot match on packets with multicast MAC destination addresses on Layer-3 interfaces. Use the **ignore routable** command when you configure the ACL to remove the routable qualifier. However, when you add ignore-routable to a RACL and apply on SVI, RACL will match with the bridged packets too.
- The Get operation provides incomplete data/no sequence number when wildcard bits are in A.B.C.D format. This is a known behavior. The Open Config model does not have srcPrefixMask/dstPrefixMask. Also, no meaningful value can be returned for prefix length because it is not possible to convert the mask to prefix length for non-contiguous mask.
- The ing-sup region occupies a minimum size of 512 entries, and the egr-sup region occupies a minimum size of 256 entries. These regions cannot be configured to lesser values. Any region size can be carved with a value only in multiples of 256 entries (with the exception of the span region, which can be carved only in multiples of 512 entries).
- Beginning with Cisco NX-OS Release 9.3(9), the Layer 3 subinterface egress router ACL feature is supported on Cisco Nexus 9300-EX, 9300-FX, and 9300-FX2 platform switches.
- Beginning with Cisco NX-OS Release 10.2(3)F, the Layer 3 subinterface egress router ACL feature is supported on Cisco Nexus 9300 Series platform switches.
- For egress RACL V6 region, you need to configure **hw profile mdb-balanced-exem**.
- From Cisco NX-OS Release 10.2(2)F, the egress PACL feature is supported on egress router ACL on Cisco Nexus 9300-GX platform switches and 93108TC-FX3P, and 93180YC-FX3 switches.
- Beginning with Cisco NX-OS Release 10.2(3)F, the egress filtering on subinterfaces feature supports Layer 3 subinterface egress router ACL on Cisco Nexus 9300-FX/FX2/FX3/GX/GX2 platform switches.
- Beginning with Cisco NX-OS Release 10.2(3)F, the increase ACL LOU threshold feature supports configurable LOU threshold limit for ACL configuration on Cisco Nexus 9500-R platform switches.
- Beginning with Cisco NX-OS Release 10.3(1)F, ITD NAT VRF configuration is provided on Cisco Nexus 9300-GX platform switches.
- Beginning with Cisco NX-OS Release 10.3(1)F, ACL Consistency Checker support is provided on Cisco Nexus 9808 switches.
  - Beginning with Cisco NX-OS Release 10.4(1)F, ACL Consistency Checker is supported on Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 switches.

- Beginning with Cisco NX-OS Release 10.4(1)F, ACL Consistency Checker support is provided on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards.
- Cisco Nexus 9808/9804 switches have the following limitations for ACL SUP support:
  - In ACE, match COS and match VLAN are not supported.
  - Ensure nd-na and nd-ns packets matches with IPv6 ACE.
  - TCAM carving is not supported. However, you can view the currently allocated TCAM for each individual feature. To view the currently allocated TCAM, use the **show hardware access-list resource utilization** command.
  - Central TCAM is supported. However, it is shared among both ingress and egress ACLs.
  - UDF is not supported.
  - LOUs is not supported
  - IPv6 fragments are not matched in egress RACL.
  - L2 ACL features are not supported.
  - ODM merge is not supported.
  - IPv6 next header match will match the innermost next header, the pipeline is able to parse.
  - Only 16 unique burst values are supported. Due to this, user configured burst values are mapped to nearest 2 power value (min 64 to max 65536).
  - Each IPv6 ACL is limited to 1,000 ACEs. This applies to all IPv6 ACLs (RACL, QoS or SPAN filter). No such limitation applies for IPv4 ACL.
- Beginning with Cisco NX-OS Release 10.3(1)F, RACL (Ingress-IPv4/IPv6 and Egress-IPv4/IPv6) with statistics are supported on Cisco Nexus 9808 switches. However, UDF is not supported.
  - Beginning with Cisco NX-OS Release 10.4(1)F, RACL (Ingress-IPv4/IPv6 and Egress-IPv4/IPv6) with statistics are supported on Cisco Nexus X98900CD-A, and X9836DM-A line cards with Cisco Nexus 9808 switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, RACL (Ingress-IPv4/IPv6 and Egress-IPv4/IPv6) with statistics are supported on Cisco Nexus 9804 switches, and Cisco Nexus X98900CD-A and X9836DM-A line cards. However, UDF is not supported.
- To display ACL statistics on Cisco Nexus 9808/9804 switches, the **hardware access-list tcam per-entry-stats template racl** command has to be enabled and reload of switch is required after configuring the **hardware access-list tcam per-entry-stats template racl** command.
- Cisco Nexus 9808/9804 switches have the following limitations for CoPP support:
  - CoPP policer stats for Stage-1, Stage-2, and Stage-3 are in PPS.

Note CoPP Stage3 stats gets reset to zero after system switchover.

  - Stage-2 output is at LC/Module level, and Stage-3 output is at SUP/CPU level.

  - Policer rates and CoS changes are supported in Custom CoPP.
  - Fabrics/FMs are not involved in in-band path.
  - CoPP Consistency checker is not supported.
  - Supported CIR minimum value is 125 PPS.
  - CIR 0 is supported.
  - There are no per entry statistics for CoPP ACL entries.
  - MACsec packets are mapped to BPDU queue.
  - Only 16 unique burst values are supported. Due to this, user configured burst values are mapped to nearest 2 power value (min 64 to max 65536).

- Beginning with Cisco NX-OS Release 10.4(3)F, Cisco Nexus 9364C-H1 switches have the following limitations for CoPP support:
  - There is only one stage of policing and CoPP policer stats for Stage-1 are in PPS
  - Policer rate and CoS changes are supported in Custom CoPP
  - Policer rates are in multiples of 572
  - CoPP Consistency checker is not supported

- Deny ACE in MAC ACL or PACL (Port ACL) with redirect option is not supported on Cisco Nexus 9000 Series switches.
- Beginning with Cisco NX-OS Release 10.3(3)F, ACL auto name completion feature is supported on Cisco Nexus 9000 Series platform switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, a new ACE keyword (all) is provided for applying the IP or IPv6 ACL rule priority over SUP rule on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2, C9364C, C9332C, and Cisco Nexus 9500 with 9700-EX/FX/GX line cards.
- Beginning with Cisco NX-OS Release 10.4(1)F, Security ACL is supported on the Cisco Nexus 9332D-H2R switches.
- Beginning with Cisco NX-OS Release 10.4(2)F, Security ACL is supported on the Cisco Nexus 93400LD-H1 switches.
- Beginning with Cisco Nexus Release 10.4(3)F, Security ACL is supported on the Cisco Nexus 9364C-H1 switches.
- The flexible TCAM configuration is supported on the Cisco Nexus 9332D-H2R, 9364C-H1, and 93400LD-H1 switches. Ingress and Egress regions uses 14K shared TCAM on this platform switch.

Note

- In Cisco Nexus 9332D-H2R, 9364C-H1, and 93400LD-H1 switches, there is no per-direction total TCAM size limit for the shared TCAM model as in Cisco Nexus 9300-FX3, GX2 switches. Limit 14K applies to the sum of both direction TCAM sizes.
- 14K is split into 28 blocks of 512 entries each. Allocation to ingress or egress direction TCAM happens in block size granularity. If the sum of ingress region sizes configured is multiple of 256, additionally 256 is allocated (and unused) for block level allocation granularity. Same applies for the sum of egress region sizes configured.

- Beginning with Cisco NX-OS Release 10.4(2)F, the new ACE configuration filter route-tag default-route is supported. This configuration enables the filtering of traffic for QoS classification when traffic matches the default route. The following guidelines and limitations are applicable for this enhancement:
  - ACL using **route-tag default-route** command is supported for class-map, QoS type only. This feature is supported for following switches:
    - Cisco Nexus 9300-FX3
    - Cisco Nexus 9300-GX
    - Cisco Nexus 9300-GX2
    - Cisco Nexus 93400LD-H1
    - Cisco Nexus 9332D-H2R
- The **route-tag default-route** configuration is supported for IPv4 and IPv6 access-list ACEs only.
- You cannot configure both PBR ACLs and default-route on a switch. Both configurations cannot co-exist.
- If you have configured the **hardware access-list tcam pbr match-default-route** command on a switch, you cannot configure the **route-tag default-route** command in PBR policy configuration.
- Ensure that you do not have **FabricPath to VXLAN** feature configured to use the **hardware access-list tcam label ing-ifacl 6** command.
- Beginning with Cisco NX-OS Release 10.4(3)F, Class E IP address is supported for Security Group ACL with Endpoint Security Group (ESG).
- Beginning with Cisco NX-OS Release 10.5(1)F, Security Group ACL (SGACL) feature support is provided for the following features:
  - Ethernet Segment Identifier (ESI)
  - Virtual Extensible LAN Traffic Engineering (VXLAN-TE)
  - Virtual Extensible LAN Policy-Based Routing (VXLAN-PBR)
  - Cloud Security (CloudSec) for Data Center Interconnect (DCI)
  - Traffic Route Management (TRM)

- Beginning with Cisco NX-OS Release 10.5(3)F, on the Cisco Nexus N9364E-SG2-Q and N9364E-SG2-O switches, ACL is supported.
  - When an Access Control List (ACL) entry with Layer 4 (TCP/UDP) port numbers matches, the "Fragment" option is not added for IPv6 Access Control Entries (ACEs) on egress.
  - ACL logging functionality is not supported.
  - Modifying the hardware rate limiter value for ACLs is not supported.
  - PACL is not supported.
- Beginning with Cisco NX-OS Release 10.5(3)F, Security Group ACL (SGACL) now support Layer 3 subinterfaces, routed interfaces, and port-channel subinterfaces. This expands the types of interfaces usable with SGACL policies.
- Beginning with Cisco NX-OS Release 10.5(3)F, on the Cisco N93-C64E-SG2-Q switch, RACL is supported for both ingress and egress directions.
  - Supported interfaces: L3 physical, L3 Port-Channel, L3 sub-interfaces, and L3 Port-Channel sub-interfaces.
  - Supported ACL types: IPv4 and IPv6.
  - Ingress ACL: Up to 1,450 IPv4 or 725 IPv6 entries per slice.
  - Egress ACL: Up to 1,022 IPv4 or 511 IPv6 entries per slice.
  - Ingress label scale: 14 unique ACLs per TOR per feature.
  - Egress label scale: 6 unique ACLs per TOR.
  - RACL is not supported on SVI interfaces due to dense mode dependency.

ACL guidelines and limitations for Cisco Nexus N9324C-SE1U switches

- Beginning with Cisco NX-OS Release 10.5(3s), IP RACL (Ingress/Egress) is supported on Cisco Nexus N9324C-SE1U switches.

Default Settings for IP ACLs

This table lists the default settings for IP ACL parameters.

Table 8: Default IP ACL Parameters

Parameters	Default
IP ACLs	No IP ACLs exist by default
IP ACL entries	1024
ACL rules	Implicit rules apply to all ACLs

Parameters	Default
Object groups	No object groups exist by default
Time ranges	No time ranges exist by default

Related Topics

Implicit Rules for IP and MAC ACLs, on page 150

Configuring IP ACLs

Creating an IP ACL

You can create an IPv4 ACL or IPv6 ACL on the device and add rules to it.

Before you begin

We recommend that you perform the ACL configuration using the Session Manager. This feature allows you to verify the ACL configuration and confirm that the resources that are required by the configuration are available before committing them to the running configuration. This feature is especially useful for ACLs that include more than about 1000 rules.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **ip access-list** *name*
   - **ipv6 access-list** *name*
3. (Optional) **fragments** {**permit-all** | **deny-all**}
4. [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ip-prefix* | *source-ip-mask*} {*destination-ip-prefix* | *destination-ip-mask*}
5. (Optional) **statistics per-entry**
6. **hardware profile acl-stats module** *xx*
7. **reload module** *xx*
8. **ignore routeable**
9. (Optional) Enter one of the following commands:
   - **show ip access-lists** *name*
   - **show ipv6 access-lists** *name*
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	Enter one of the following commands: • **ip access-list** *name* • **ipv6 access-list** *name* **Example:** `switch(config)# ip access-list acl-01` `switch(config-acl)#`	Creates the IP ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.
Step 3	(Optional) **fragments** {**permit-all** | **deny-all**} **Example:** `switch(config-acl)# fragments permit-all`	Optimizes fragment handling for noninitial fragments. When a device applies to traffic an ACL that contains the **fragments** command, the **fragments** command only matches noninitial fragments that do not match any explicit **permit** or **deny** commands in the ACL.
Step 4	[*sequence-number*] {**permit** | **deny**} *protocol* {*source-ip-prefix* | *source-ip-mask*} {*destination-ip-prefix* | *destination-ip-mask*} **Example:** `switch(config-acl)# 10 permit ipv6 1::1 2::2 3::3 4::4`	Creates a rule in the IP ACL. You can create many rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic. For IPv4 and IPv6 access lists, you can specify a source and destination IPv4 or IPv6 prefix, which matches only on the first contiguous bits, or you can specify a source and destination IPv4 or IPv6 wildcard mask, which matches on any bit in the address. IPv6 wildcard masks are supported for Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2/FXP switches and the Cisco Nexus 9364C switch.
Step 5	(Optional) **statistics per-entry** **Example:** `switch(config-acl)# statistics per-entry`	Specifies that the device maintains global statistics for packets that match the rules in the ACL. **Note** Beginning Cisco NX-OS Release 9.2(3), ACL statistics is supported on Cisco Nexus 9500 platform switches with -R line cards. This is a mandatory step if you are using the Cisco Nexus 9500 platform switches.
Step 6	**hardware profile acl-stats module** *xx* **Example:**	Enables counters for the ACL TCAM entries on both, the internal and external TCAM. **Note**

	Command or Action	Purpose
	`switch(config-acl)# hardware profile acl-stats module 10`	This command is applicable only for Cisco Nexus 9500 platform switches with -R and -RX line cards and Cisco Nexus 3636C-R and 36180YC-R switches. VLAN and SVI statistics are lost when you enable the counters.
Step 7	**reload module** *xx* **Example:** `switch(config)# reload module 10`	Reloads the switch. **Note** For the Cisco Nexus 9500 platform switches, this command is optional and only those module (s) where the**hardware profile ac-stats** is applied must be reloaded.
Step 8	**ignore routeable** **Example:** `switch(config)# ignore routeable`	Enables the filtering of multicast traffic on Cisco Nexus 9300-EX and 9300-FX platform switches.
Step 9	(Optional) Enter one of the following commands: • **show ip access-lists** *name* • **show ipv6 access-lists** *name* **Example:** `switch(config-acl)# show ip access-lists acl-01`	Displays the IP ACL configuration.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config-acl)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Changing an IP ACL

You can add and remove rules in an existing IPv4 or IPv6 ACL, but you cannot change existing rules. Instead, to change a rule, you can remove it and recreate it with the desired changes.

If you need to add more rules between existing rules than the current sequence numbering allows, you can use the **resequence** command to reassign sequence numbers.

Before you begin

We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. This feature is especially useful for ACLs that include more than about 1000 rules.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **ip access-list** *name*
   - **ipv6 access-list** *name*

3. (Optional) [*sequence-number*] {**permit** | **deny**} *protocol source destination*
4. (Optional) [**no**] **fragments** {**permit-all** | **deny-all**}
5. (Optional) **no** {*sequence-number* | {**permit** | **deny**} *protocol source destination*}
6. (Optional) [**no**] **statistics per-entry**
7. (Optional) Enter one of the following commands:
   - **show ip access-lists** *name*
   - **show ipv6 access-lists** *name*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	Enter one of the following commands: • **ip access-list** *name* • **ipv6 access-list** *name* **Example:** `switch(config)# ip access-list acl-01` `switch(config-acl)#`	Enters IP ACL configuration mode for the ACL that you specify by name.
Step 3	(Optional) [*sequence-number*] {**permit**	**deny**} *protocol source destination* **Example:** `switch(config-acl)# 100 permit ip 192.168.2.0/24 any`	Creates a rule in the IP ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic.
Step 4	(Optional) [**no**] **fragments** {**permit-all**	**deny-all**} **Example:** `switch(config-acl)# fragments permit-all`	Optimizes fragment handling for noninitial fragments. When a device applies to traffic an ACL that contains the **fragments** command, the **fragments** command only matches noninitial fragments that do not match any explicit **permit** or **deny** commands in the ACL. The **no** option removes fragment-handling optimization.
Step 5	(Optional) **no** {*sequence-number*	{**permit**	**deny**} *protocol source destination*} **Example:** `switch(config-acl)# no 80`	Removes the rule that you specified from the IP ACL. The **permit** and **deny** commands support many ways of identifying traffic.

	Command or Action	Purpose
Step 6	(Optional) [**no**] **statistics per-entry** **Example:** `switch(config-acl)# statistics per-entry`	Specifies that the device maintains global statistics for packets that match the rules in the ACL. The **no** option stops the device from maintaining global statistics for the ACL.
Step 7	(Optional) Enter one of the following commands: • **show ip access-lists** *name* • **show ipv6 access-lists** *name* **Example:** `switch(config-acl)# show ip access-lists acl-01`	Displays the IP ACL configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config-acl)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Changing Sequence Numbers in an IP ACL, on page 181

Creating a VTY ACL

You can configure a VTY ACL to control access to all IPv4 or IPv6 traffic over all VTY lines in the ingress or egress direction.

Before you begin

Set identical restrictions on all the virtual terminal lines because a user can connect to any of them.

We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration, which is especially useful for ACLs that include more than about 1000 rules.

SUMMARY STEPS

1. **configure terminal**
2. {**ip** | **ipv6**} **access-list** *name*
3. {**permit** | **deny**} *protocol source destination* [**log**] [**time-range** *time*]
4. **exit**
5. **line vty**
6. {**ip** | **ipv6**} **access-class** *name* {**in** | **out**}
7. (Optional) **show** {**ip** | **ipv6**} **access-lists**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	{**ip** | **ipv6**} **access-list** *name* **Example:** `switch(config)# ip access-list vtyacl`	Creates an ACL and enters IP access list configuration mode for that ACL. The maximum length for the *name* argument is 64 characters.
Step 3	{**permit** | **deny**} *protocol source destination* [**log**] [**time-range** *time*] **Example:** `switch(config-ip-acl)# permit tcp any any`	Creates an ACL rule that permits TCP traffic from and to the specified sources.
Step 4	**exit** **Example:** `switch(config-ip-acl)# exit` `switch(config)#`	Exits IP access list configuration mode.
Step 5	**line vty** **Example:** `switch(config)# line vty` `switch(config-line)#`	Specifies the virtual terminal and enters line configuration mode.
Step 6	{**ip** | **ipv6**} **access-class** *name* {**in** | **out**} **Example:** `switch(config-line)# ip access-class vtyacl out`	Restricts incoming or outgoing connections to and from all VTY lines using the specified ACL. The maximum length for the *name* argument is 64 characters.
Step 7	(Optional) **show** {**ip** | **ipv6**} **access-lists** **Example:** `switch# show ip access-lists`	Displays the configured ACLs, including any VTY ACLs.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Changing Sequence Numbers in an IP ACL

You can change all the sequence numbers assigned to the rules in an IP ACL.

Before you begin

We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. This feature is especially useful for ACLs that include more than about 1000 rules.

SUMMARY STEPS

1. **configure terminal**
2. **resequence** {**ip** | **ipv6**} **access-list** *name starting-sequence-number increment*
3. (Optional) **show ip access-lists** *name*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**resequence** {**ip** | **ipv6**} **access-list** *name starting-sequence-number increment* **Example:** `switch(config)# resequence access-list ip acl-01 100 10`	Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify. The *starting-sequence-number* argument and the *increment* argument can be a whole number between 1 and 4294967295.
Step 3	(Optional) **show ip access-lists** *name* **Example:** `switch(config)# show ip access-lists acl-01`	Displays the IP ACL configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Removing an IP ACL

You can remove an IP ACL from the device.

Before you begin

Ensure that you know whether the ACL is applied to an interface. The device allows you to remove ACLs that are currently applied. Removing an ACL does not affect the configuration of interfaces where you have applied the ACL. Instead, the device considers the removed ACL to be empty. Use the **show ip access-lists** command or the **show ipv6 access-lists** command with the summary keyword to find the interfaces that an IP ACL is configured on.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **no ip access-list** *name*
   - **no ipv6 access-list** *name*
3. (Optional) Enter one of the following commands:
   - **show ip access-lists** *name* **summary**
   - **show ipv6 access-lists** *name* **summary**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	Enter one of the following commands: • **no ip access-list** *name* • **no ipv6 access-list** *name* **Example:** `switch(config)# no ip access-list acl-01`	Removes the IP ACL that you specified by name from the running configuration.
Step 3	(Optional) Enter one of the following commands: • **show ip access-lists** *name* **summary** • **show ipv6 access-lists** *name* **summary** **Example:** `switch(config)# show ip access-lists acl-01 summary`	Displays the IP ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.
Step 4	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch(config)# copy running-config startup-config`	

Configuring ACL TCAM Region Sizes

You can change the size of the ACL ternary content addressable memory (TCAM) regions in the hardware. After TCAM carving, for the TCAM to qualify, you must save the configuration and reload the switch. If the switch has a faulty module, saving the configuration will take a longer time.

You can use this procedure for all Cisco Nexus 9200, 9300, and 9500 Series switches and the Cisco Nexus 3164Q, 31128PQ, 3232C, and 3264Q switches, except for NFE2-enabled devices (such as the X9432C-S 100G line card and the C9508-FM-S fabric module), which must use TCAM templates to configure ACL TCAM region sizes. For more information on using TCAM templates, see "Using Templates to Configure ACL TCAM Region Sizes."

Note

- Once you apply a template (using Using Templates to Configure ACL TCAM Region Sizes, on page 192), the **hardware access-list tcam region** command in this section will not work. You must uncommit the template in order to use the command.
- The **hardware access-list tcam region** command for the Multicast PIM Bidir feature is applicable only to the Broadcom-based Cisco Nexus 9000 Series switches.
- For information on configuring QoS TCAM carving, see the *Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide*.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **hardware access-list tcam region** *region tcam-size*
3. **copy running-config startup-config**
4. (Optional) **show hardware access-list tcam region**
5. **hardware access-list tcam label vrf-nat**
6. **reload**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	[**no**] **hardware access-list tcam region** *region tcam-size* **Example:** `switch(config)# hardware access-list tcam region mpls 256`	Changes the ACL TCAM region size. These are the available regions: • **n9k-arp-acl**—Configures the rate limit for arp packets entering an interface on their way to the CPU. You will have to set this rate limit per interface to ensure that arp packets conform to the configured rate. • **arp-ether**—Configures the size of the ARP/Layer 2 Ethertype TCAM region. • **copp**—Configures the size of the CoPP TCAM region. • **e-flow**—Configures the size of the egress flow counters TCAM region. • **egr-copp**—Configures the size of the egress CoPP TCAM region. • **egr-racl**—Configures the size of the egress IPv4 or IPv6 router ACL (RACL) TCAM region. • **egr-sup**—Configures the size of the egress supervisor TCAM region. • **e-ipv6-qos**—Configures the size of the IPv6 egress QoS TCAM region. • **e-ipv6-racl**—Configures the size of the IPv6 egress router ACL (ERACL) TCAM region. • **e-mac-qos**—Configures the size of the egress MAC QoS TCAM region. • **e-qos**—Configures the size of the IPv4 egress QoS TCAM region. • **e-qos-lite**—Configures the size of the IPv4 egress QoS lite TCAM region. • **e-racl**—Configures the size of the IPv4 egress router ACL (ERACL) TCAM region. • **fex-ifacl**—Configures the size of the FEX IPv4 port ACL TCAM region. • **fex-ipv6-ifacl**—Configures the size of the FEX IPv6 port ACL TCAM region. • **fex-ipv6-qos**—Configures the size of the FEX IPv6 port QoS TCAM region. • **fex-mac-ifacl**—Configures the size of the FEX MAC port ACL TCAM region.

	Command or Action	Purpose
		• **fex-mac-qos**—Configures the size of the FEX MAC port QoS TCAM region. • **fex-qos**—Configures the size of the FEX IPv4 port QoS TCAM region. • **fex-qos-lite**—Configures the size of the FEX IPv4 port QoS lite TCAM region. • **fhs**—Configures the size of the fhs TCAM region. You can configure TCAM for the fhs region on the Cisco Nexus 9300 and 9500 Series switches. • **flow**—Configures the size of the ingress flow counters TCAM region. • **ifacl**—Configures the size of the IPv4 port ACL TCAM region. • **ifacl-udf**—Configures the size of the IPv4 port ACL user-defined field (UDF) TCAM region. • **ing-ifacl**—Configures the size of the ingress IPv4, IPv6, or MAC port ACL TCAM region. **Note** You can attach user-defined fields (UDFs) to the ing-ifacl TCAM region to configure UDF-based IPv4 or IPv6 port ACLs. UDF-based IPv6 port ACLs. For more information and configuration instructions, see Configuring UDF-Based Port ACLs, on page 201. • **ing-l2qos**—Configures the size of the ingress Layer 2 QoS TCAM region. • **ing-l2-span-filter**—Configures the size of the ingress Layer 2 SPAN filter TCAM region. • **ing-l3-span-filter**—Configures the size of the ingress Layer 3 and VLAN SPAN filter TCAM region. • **ing-l3-vlan-qos**—Configures the size of the ingress Layer 3, VLAN, and SVI QoS TCAM region. • **ing-racl**—Configures the size of the IPv4 or IPv6 ingress router ACL (RACL) TCAM region. • **ing-redirect**—Configures the size of the redirect TCAM region for DHCPv4 relay, DHCPv4 snooping, and DHCPv4 client. • **ing-sup**—Configures the size of the ingress supervisor TCAM region.

	Command or Action	Purpose
		• **ipsg**—Configures the size of the IP source guard SMAC-IP binding TCAM region. • **ipv6-ifacl**—Configures the size of the IPv6 port ACL TCAM region. • **ipv6-l3qos**—Configures the size of the IPv6 Layer 3 QoS TCAM region. • **ipv6-qos**—Configures the size of the IPv6 port QoS TCAM region. • **ipv6-racl**—Configures the size of the IPv6 RACL TCAM region. • **ipv6-vacl**—Configures the size of the IPv6 VACL TCAM region. • **ipv6-vqos**—Configures the size of the IPv6 VLAN QoS TCAM region. • **l3qos**—Configures the size of the IPv4 Layer 3 QoS TCAM region. • **l3qos-lite**—Configures the size of the IPv4 Layer 3 QoS lite TCAM region. • **mac-ifacl**—Configures the size of the MAC port ACL TCAM region. • **mac-l3qos**—Configures the size of the MAC Layer 3 QoS TCAM region. • **mac-qos**—Configures the size of the MAC port QoS TCAM region. • **mac-vacl**—Configures the size of the MAC VACL TCAM region. • **mac-vqos**—Configures the size of the MAC VLAN QoS TCAM region. • **mcast_bidir**—Configures the size of the multicast PIM Bidir TCAM region. • **mpls**—Configures the size of the static MPLS TCAM region. • **nat**—Configures the size of the network address translation (NAT) TCAM region. • **ns-ipv6-l3qos**—Configures the size of the IPv6 Layer 3 QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM).

	Command or Action	Purpose
		• **ns-ipv6-qos**—Configures the size of the IPv6 port QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM). • **ns-ipv6-vqos**—Configures the size of the IPv6 VLAN QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM). • **ns-l3qos**—Configures the size of the IPv4 Layer 3 QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM). • **ns-mac-l3qos**—Configures the size of the MAC Layer 3 QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM). • **ns-mac-qos**—Configures the size of the MAC port QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM). • **ns-mac-vqos**—Configures the size of the MAC VLAN QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM). • **ns-qos**—Configures the size of the IPv4 port QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM). • **ns-vqos**—Configures the size of the IPv4 VLAN QoS TCAM region for the X9536PQ, X9564PX, and X9564TX line cards and the M12PQ generic expansion module (GEM). • **openflow**—Configures the size of the OpenFlow TCAM region. • **qos**—Configures the size of the IPv4 port QoS TCAM region. • **qos-lite**—Configures the size of the IPv4 port QoS lite TCAM region. • **racl**—Configures the size of the IPv4 router ACL (RACL) TCAM region. • **racl-lite**—Configures the size of the IPv4 router ACL (RACL) lite TCAM region.

	Command or Action	Purpose
		• **racl-udf**—Configures the size of the IPv4 router ACL (RACL) user-defined field (UDF) TCAM region. • **redirect**—Configures the size of the redirect TCAM region. • **redirect-tunnel**—Configures the size of the redirect-tunnel TCAM region, which is used for BFD over VXLAN. **Note** This command is supported only if the TP_SERVICES_PKG license is installed. • **rp-ipv6-qos**—Configures the size of the IPv6 port QoS TCAM region for the 100G 9408PC line card and the 100G M4PC generic expansion module (GEM). • **rp-mac-qos**—Configures the size of the MAC port QoS TCAM region for the 100G 9408PC line card and the 100G M4PC generic expansion module (GEM). • **rp-qos**—Configures the size of the IPv4 port QoS TCAM region for the 100G 9408PC line card and the 100G M4PC generic expansion module (GEM). • **rp-qos-lite**—Configures the size of the IPv4 port QoS lite TCAM region for the 100G 9408PC line card and the 100G M4PC generic expansion module (GEM). • **sflow**—Configures the size of the sFlow TCAM region. • **span**—Configures the size of the SPAN TCAM region. • **svi**—Configures the size of the ingress SVI counters TCAM region. • **vacl**—Configures the size of the IPv4 VACL TCAM region. • **vpc-convergence**—Configures the size of the vPC convergence TCAM region. • **vqos**—Configures the size of the IPv4 VLAN QoS TCAM region. • **vqos-lite**—Configures the size of the IPv4 VLAN QoS lite TCAM region. • *tcam-size*—TCAM size. The size has to a multiple of 256. If the size is more than 256, it has to be multiple of 512. For FHS, the range is from 0-4096.

	Command or Action	Purpose
		You can use the **no** form of this command to revert to the default TCAM region size. **Note** You can attach IPv4 user-defined fields (UDFs) to the racl, ifacl, and vacl TCAM regions using the **hardware access-list tcam region** {**racl** | **ifacl** | **vacl**} **qualify udf** *udf-names* command to configure IPv4 UDF-based SPAN or ERSPAN.You can attach IPv6 UDFs to the ing-l2-span-filter and ing-l3-span-filter TCAM regions using the **hardware access-list tcam region** {**ing-ifacl** | **ing-l2-span-filter** | **ing-l3-span-filter**} **qualify v6udf** *v6udf-names* commands to configure IPv6 UDF-based ERSPAN. For more information and configuration instructions, see the latest *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.
Step 3	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.
Step 4	(Optional) **show hardware access-list tcam region** **Example:** `switch(config)# show hardware access-list tcam region`	Displays the TCAM sizes that will be applicable on the next reload of the device.
Step 5	**hardware access-list tcam label vrf-nat** **Example:** `switch(config)# hardware access-list tcam label vrf-nat`	Configures the ITD NAT with VRF. **Note** Beginning with Cisco NX-OS Release 10.3(1)F, this command is supported on Cisco Nexus 9300-GX switches.
Step 6	**reload** **Example:** `switch(config)# reload`	Reloads the device. **Note** The new size values are effective only after you enter **copy running-config startup-config** + **reload** or reload all line card modules.

Example

The following example shows how to change the size of the n9k-arp-acl TCAM region on a Cisco Nexus NFE-enabled device:

```
switch(config)#hardware access-list tcam region n9k-arp-acl 256switch(config)#copy r s
switch(config)# reload
Configuring storm-control-cpu:
switch (config)# interface ethernet 1/10switch
switch (config-if)# storm-control-cpu arp rate 150
switch (config)# show access-list storm-control-cpu arp-stats interface ethernet 1/10
```

```
slot  1
```

The following example shows how to change the size of the RACL TCAM region on a Cisco Nexus 9500 Series switch:

```
switch(config)# hardware access-list tcam region racl 256
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy run start' and 'reload'
switch(config)# copy running-config startup-config
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

This example shows how to display the TCAM region sizes to verify your changes:

```
switch(config)# show hardware access-list tcam region
         TCAM Region Sizes:

                             IPV4 PACL [ifacl] size =    512
                        IPV6 PACL [ipv6-ifacl] size =    0
                          MAC PACL [mac-ifacl] size =    0
                           IPV4 Port QoS [qos] size =    256
                      IPV6 Port QoS [ipv6-qos] size =    0
                        MAC Port QoS [mac-qos] size =    0
                    FEX IPV4 PACL [fex-ifacl] size =    0
                FEX IPV6 PACL [fex-ipv6-ifacl] size =    0
                  FEX MAC PACL [fex-mac-ifacl] size =    0
                   FEX IPV4 Port QoS [fex-qos] size =    0
               FEX IPV6 Port QoS [fex-ipv6-qos] size =    0
                 FEX MAC Port QoS [fex-mac-qos] size =    0
                             IPV4 VACL [vacl] size =    512
                        IPV6 VACL [ipv6-vacl] size =    0
                          MAC VACL [mac-vacl] size =    0
                         IPV4 VLAN QoS [vqos] size =    0
                    IPV6 VLAN QoS [ipv6-vqos] size =    0
                      MAC VLAN QoS [mac-vqos] size =    0
                             IPV4 RACL [racl] size =    512
                        IPV6 RACL [ipv6-racl] size =    0
               IPV4 Port QoS Lite [qos-lite] size =    0
           FEX IPV4 Port QoS Lite [fex-qos-lite] size =    0
              IPV4 VLAN QoS Lite [vqos-lite] size =    0
               IPV4 L3 QoS Lite [l3qos-lite] size =    0
                     Egress IPV4 QoS [e-qos] size =    0
                Egress IPV6 QoS [e-ipv6-qos] size =    0
                  Egress MAC QoS [e-mac-qos] size =    0
                     Egress IPV4 VACL [vacl] size =    512
                Egress IPV6 VACL [ipv6-vacl] size =    0
                  Egress MAC VACL [mac-vacl] size =    0
                   Egress IPV4 RACL [e-racl] size =    256
              Egress IPV6 RACL [e-ipv6-racl] size =    0
          Egress IPV4 QoS Lite [e-qos-lite] size =    0
                        IPV4 L3 QoS [l3qos] size =    0
                   IPV6 L3 QoS [ipv6-l3qos] size =    0
                     MAC L3 QoS [mac-l3qos] size =    0
                              Ingress System size =  256
                               Egress System size =  256
                                 SPAN [span] size =  256
                           Ingress COPP [copp] size =  256
                 Ingress Flow Counters [flow] size =    0
                   Egress Flow Counters [e-flow] size = 0
                   Ingress SVI Counters [svi] size =    0
                          Redirect [redirect] size =  512
                    NS IPV4 Port QoS [ns-qos] size =  256
```

```
      NS IPV6 Port QoS [ns-ipv6-qos] size =    0
        NS MAC Port QoS [ns-mac-qos] size =    0
          NS IPV4 VLAN QoS [ns-vqos] size =  256
     NS IPV6 VLAN QoS [ns-ipv6-vqos] size =    0
      NS MAC VLAN QoS [ns-mac-vqos] size =    0
          NS IPV4 L3 QoS [ns-l3qos] size =  256
     NS IPV6 L3 QoS [ns-ipv6-l3qos] size =    0
       NS MAC L3 QoS [ns-mac-l3qos] size =    0
VPC Convergence [vpc-convergence] size =  256
   IPSG SMAC-IP bind table [ipsg] size =    0
 Ingress ARP-Ether ACL [arp-ether] size =    0
ranger+ IPV4 QoS Lite [rp-qos-lite] size =    0
          ranger+ IPV4 QoS [rp-qos] size =  256
     ranger+ IPV6 QoS [rp-ipv6-qos] size =  256
       ranger+ MAC QoS [rp-mac-qos] size =  256
                      NAT ACL[nat] size =    0
                          Mpls ACL size =    0
                Ingress IPv4 N3K QoS size =    0
                Ingress IPv6 N3K QoS size =    0
                          MOD RSVD size =    0
             sFlow ACL [sflow] size =    0
               mcast bidir ACL size =    0
                          Openflow size =    0
```

This example shows how to revert to the default RACL TCAM region size:

```
switch(config)# no hardware profile tcam region racl 512
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy run start' and 'reload'
switch(config)# copy running-config startup-config
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

Using Templates to Configure ACL TCAM Region Sizes

You can use create and apply custom templates to configure ACL TCAM region sizes.

For all Cisco Nexus 9200, 9300, and 9500 Series switches, you can use this procedure or the Configuring ACL TCAM Region Sizes procedure to configure ACL TCAM region sizes. However, NFE2-enabled devices (such as the X9432C-S 100G line card and the C9508-FM-S fabric module) do not support the **hardware access-list tcam region** command and must use a template to configure the ACL TCAM region size.

Note

- Once you apply a TCAM template, the **hardware access-list tcam region** command will not work. You must uncommit the template in order to use the command.
- For information on configuring QoS TCAM carving, see the *Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide*.
- The TCAM profile template is not supported on the C9508-FM-S fabric module.

SUMMARY STEPS

1. **configure terminal**

2. [**no**] **hardware profile tcam resource template** *template-name* **ref-template** {**nfe** | **nfe2** | {**l2-l3** | **l3**}}
3. (Optional) *region tcam-size*
4. **exit**
5. [**no**] **hardware profile tcam resource service-template** *template-name*
6. (Optional) **show hardware access-list tcam template** {**all** | **nfe** | **nfe2** | **l2-l3** | **l3** | *template-name*}
7. (Optional) **copy running-config startup-config**
8. **reload**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	[**no**] **hardware profile tcam resource template** *template-name* **ref-template** {**nfe**	**nfe2**	{**l2-l3**	**l3**}} **Example:** `switch(config)# hardware profile tcam resource template SR_MPLS_CARVE ref-template nfe2` `switch(config-tcam-temp)#`	Creates a template for configuring ACL TCAM region sizes. **nfe**—The default TCAM template for Network Forwarding Engine (NFE)-enabled Cisco Nexus 9300 and 9500 Series. **nfe2**—The default TCAM template for NFE2-enabled Cisco Nexus 9500 Series devices. **l2-l3**—The default TCAM template for Layer 2 and Layer 3 configurations. **l3**—The default TCAM template for Layer 3 configurations.
Step 3	(Optional) *region tcam-size* **Example:** `switch(config-tcam-temp)# mpls 256`	Adds any desired TCAM regions and their sizes to the template. Enter this command for each region you want to add to the template. For the list of available regions, see Configuring ACL TCAM Region Sizes.
Step 4	**exit** **Example:** `switch(config-tcam-temp)# exit` `switch(config#)`	Exits the TCAM template configuration mode.
Step 5	[**no**] **hardware profile tcam resource service-template** *template-name* **Example:** `switch(config)# hardware profile tcam resource service-template SR_MPLS_CARVE`	Applies the custom template to all line cards and fabric modules.
Step 6	(Optional) **show hardware access-list tcam template** {**all**	**nfe**	**nfe2**	**l2-l3**	**l3**	*template-name*} **Example:**	Displays the configuration for all TCAM templates or for a specific template.

	Command or Action	Purpose
	`switch(config)# show hardware access-list tcam template SR_MPLS_CARVE`	
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.
Step 8	**reload** **Example:** `switch(config)# reload`	Reloads the device. **Note** The configuration is effective only after you enter **copy running-config startup-config** + **reload**.

Configuring TCAM Carving

The default TCAM region configuration varies by platform and does not accommodate all TCAM regions. To enable any desired regions, you must decrease the TCAM size of one region and then increase the TCAM size for the desired region.

Note For information on configuring QoS TCAM carving, see the *Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide*.

Note Beginning with Cisco NX-OS Release 10.3(1)F, the following TCAM limitation applies on Cisco Nexus 9800 platform switches:

- TCAM carving is not supported. However you can view the currently allocated TCAM for each individual feature. To view the currently allocated TCAM, use the **show hardware access-list resource utilization** command.
- Central TCAM is supported. However it is shared among both ingress and egress ACLs.

The following tables list the default sizes for the ingress and egress TCAM regions on different platforms.

Table 9: Default TCAM Region Configuration (Ingress) - For Cisco Nexus 9500 Series Switches

Region Name	Size	Width	Total Size
IPv4 RACL	1536	1	1536
IPv4 Layer 3 QoS	256	2	512
SPAN	256	1	256
CoPP	256	2	512
System	256	2	512
Redirect	256	1	256

Region Name	Size	Width	Total Size
vPC convergence	512	1	512
			4K

Table 10: Default TCAM Region Configuration (Egress) - For Cisco Nexus 9500 Series Switches

Region Name	Size	Width	Total Size
IPv4 RACL	768	1	768
System	256	1	256
			1K

Table 11: Default TCAM Size - For Cisco Nexus 9504 and 9508 Platform switches

Region	Size
MAC PACL [mac-ifacl]	1952
IPV6 Port QoS [ipv6-qos]	256
PV6 L3 QoS [ipv6-l3qos]	256
SPAN [span]	96
Ingress CoPP [copp]	128
Redirect IPv4	2048
Redirect IPv6	2048

Table 12: Default TCAM Region Configuration (Ingress) - For Cisco Nexus 9300-FX Series Switches

Region Name	Size	Width	Total Size
IPv4 RACL	2304	1	2304
Layer 2 QoS	256	1	256
Layer 3/VLAN QoS	512	1	512
System	512	1	512
Layer 2 SPAN filter	256	1	256
Layer 3 SPAN filter	256	1	256
SPAN	512	1	512
NetFlow/Analytics filter	512	1	512
			5K

Table 13: Default TCAM Region Configuration (Ingress) - For Cisco Nexus 9300-EX Series Switches

Region Name	Size	Width	Total Size
IPv4 RACL	1792	1	1792
Layer 2 QoS	256	1	256
Layer 3/VLAN QoS	512	1	512
System	512	1	512
Layer 2 SPAN ACL	256	1	256
Layer 3/VLAN SPAN ACL	256	1	256
SPAN	512	1	512
			4K

Table 14: Default TCAM Region Configuration (Egress) - For Cisco Nexus 9300-EX Series Switches

Region Name	Size	Width	Total Size
IPv4 RACL	1792	1	1792
System	256	1	256
			2K

Table 15: Default TCAM Region Configuration (Ingress) - For Cisco Nexus 9300 Series Switches

Region Name	Size	Width	Total Size
IPv4 port ACL	512	1	512
IPv4 port QoS	256	2	512
IPv4 VACL	512	1	512
IPv4 RACL	512	1	512
SPAN	256	1	256
CoPP	256	2	512
IPv4 port QoS for ACI leaf line card	256	1	256
IPv4 VLAN QoS for ACI leaf line card	256	1	256
IPv4 Layer 3 QoS for ACI leaf line card	256	1	256
System	256	2	512
Redirect	512	1	512
vPC convergence	256	1	256

Region Name	Size	Width	Total Size
			4K

Table 16: Default TCAM Region Configuration (Egress) - For Cisco Nexus 9300 Series Switches

Region Name	Size	Width	Total Size
IPv4 VACL	512	1	512
IPv4 RACL	256	1	256
System	256	1	256
			1K

Table 17: Default TCAM Region Configuration (Ingress) - For Layer 2-to-Layer 3 Configurations on Cisco Nexus 9200 Series Switches

Region Name	Size	Width	Total Size
Ingress NAT	0	1	0
Ingress port ACL	256	1	256
Ingress VACL	256	1	256
Ingress RACL	1536	1	1536
Ingress Layer 2 QoS	256	1	256
Ingress Layer 3 VLAN QoS	256	1	256
Ingress supervisor	512	1	512
Ingress Layer 2 ACL SPAN	256	1	256
Ingress Layer 3 ACL SPAN	256	1	256
Port-based SPAN	512	1	512
			4096

Table 18: Default TCAM Region Configuration (Egress) - For Layer 2-to-Layer 3 Configurations on Cisco Nexus 9200 Series Switches

Region Name	Size	Width	Total Size
Egress VACL	256	1	256
Egress RACL	1536	1	1536
Egress supervisor	256	1	256
			2048

Table 19: Default TCAM Region Configuration (Ingress) - For Layer 3 Configurations on Cisco Nexus 9200 Series Switches

Region Name	Size	Width	Total Size
Ingress NAT	0	1	0
Ingress port ACL	0	1	0
Ingress VACL	0	1	0
Ingress RACL	1792	1	1792
Ingress Layer 2 QoS	256	1	256
Ingress Layer 3 VLAN QoS	512	1	512
Ingress supervisor	512	1	512
Ingress Layer 2 ACL SPAN	256	1	256
Ingress Layer 3 ACL SPAN	256	1	256
Port-based SPAN	512	1	512
			4096

Table 20: Default TCAM Region Configuration (Egress) - For Layer 3 Configurations on Cisco Nexus 9200 Series Switches

Region Name	Size	Width	Total Size
Egress VACL	0	1	0
Egress RACL	1792	1	1792
Egress supervisor	256	1	256
			2048

The following example sets the IPv6 RACL TCAM size to 256 on a Cisco Nexus 9500 Series switch. An IPv6 RACL of size 256 takes 512 entries because IPv6 is double wide.

Note Follow a similar procedure to modify the TCAM settings for a different region or to modify the TCAM settings on a different device.

To set the size of the ingress IPv6 RACL TCAM region on a Cisco Nexus 9500 Series switch, perform one of two options.

Option #1

Reduce the ingress IPv4 RACL by 1024 entries (1536 - 1024 = 512) and add an ingress IPv6 RACL with 512 entries—This option is preferred.

```
switch(config)# hardware access-list tcam region racl 512
Warning: Please reload the linecard for the configuration to take effect
switch(config)# hardware access-list tcam region ipv6-racl 256
Warning: Please reload the linecard for the configuration to take effect
```

Table 21: Updated TCAM Region Configuration After Reducing the IPv4 RACL (Ingress)

Region Name	Size	Width	Total Size
IPv4 RACL	1024	1	1024
IPv6 RACL	256	2	1024[2]
IPv4 Layer 3 QoS	256	2	512
SPAN	256	1	256
CoPP	256	2	512
System	256	2	512
Redirect	256	1	256
vPC convergence	512	1	512
			4K

[2] 2 x 512 entry slices are allocated due to the non-availability of 256 entry slices.

Option #2

Remove IPv4 Layer 3 QoS by reducing its size to 0 and add an ingress IPv6 RACL—This option is available if you are not using IPv4 Layer 3 QoS.

```
switch(config)# hardware access-list tcam region l3qos 0
Warning: Please reload the linecard for the configuration to take effect
switch(config)# hardware access-list tcam region ipv6-racl 256
Warning: Please reload the linecard for the configuration to take effect
```

Table 22: Updated TCAM Region Configuration After Removing Layer 3 QoS (Ingress)

Region Name	Size	Width	Total Size
IPv4 RACL	1536	1	1536
IPv6 RACL	256	2	512
IPv4 Layer 3 QoS	0	2	0
SPAN	256	1	256
CoPP	256	2	512
System	256	2	512
Redirect	256	1	256
vPC convergence	512	1	512
			4K

To enable an egress IPv6 RACL of size 256, reduce the egress IPv4 RACL to 256 and add the egress IPv6 RACL:

```
switch(config)# hardware access-list tcam region e-racl 256
Warning: Please reload the linecard for the configuration to take effect
switch(config)# hardware access-list tcam region e-ipv6-racl 256
```

```
Warning: Please reload the linecard for the configuration to take effect
```

Table 23: Default TCAM Region Configuration After Reducing the IPv4 RACL (Egress)

Region Name	Size	Width	Total Size
IPv4 RACL	256	1	256
IPv6 RACL	256	2	512
System	256	1	256
			1K

Table 24: Default TCAM Size - For Cisco Nexus 9800 Platform switches

Feature name	Size (Unidimensional)
Ingress RACLv4	9216
Ingress QoSv4	
Ingress SPAN filter v4	
Egress RACLv4	
Ingress SUP	
Ingress RACLv6	4608
Ingress QoSv6	
Ingress SPAN filter v6	
Egress RACL v6	

Note Each IPv6 ACL is limited to 1,000 ACEs. This applies to all IPv6 ACLs (RACL, QoS or SPAN filter). No such limitation applies for IPv4 ACL.

After you adjust the TCAM region sizes, enter the **show hardware access-list tcam region** command to display the TCAM sizes that will be applicable on the next reload of the device.

Attention To keep all modules synchronized, you must reload all line card modules or enter **copy running-config startup-config** + **reload** to reload the device. Multiple TCAM region configurations require only a single reload. You can wait until you complete all of your TCAM region configurations before you reload the device.

Depending on the configuration, you might exceed the TCAM size or run out of slices.

If you exceed the 4K ingress limit for all TCAM regions when you configure a TCAM region, the following message appears:

```
ERROR: Aggregate TCAM region configuration exceeded the available Ingress TCAM space. Please
 re-configure.
```

If you exceed the number of slices, the following message appears:

```
ERROR: Aggregate TCAM region configuration exceeded the available Ingress TCAM slices.
Please re-configure.
```

If you exceed the 1K egress limit for all TCAM regions when you configure a TCAM region, the following message appears:

```
ERROR: Aggregate TCAM region configuration exceeded the available Egress TCAM space. Please
 re-configure.
```

If TCAM for a particular feature is not configured and you try to apply a feature that requires TCAM carving, the following message appears:

```
 ERROR: Module x returned status: TCAM region is not configured. Please configure TCAM
region and retry the command.
```

Note The default redirect TCAM region size of 256 might not be sufficient if you are running many BFD or DHCP relay sessions. To accommodate more BFD or DHCP relay sessions, you might need to increase the TCAM size to 512 or greater.

Note "e-racl" tcam region size can be maximum of 16K when we have at least one "N9K-X9624D-R2" line card on a N9K-C9508 (Fretta) system.

Related Topics

Configuring ACL TCAM Region Sizes, on page 184

Configuring UDF-Based Port ACLs

You can configure UDF-based port ACLs for Cisco Nexus 9200, 9300, and 9300-EX Series switches. This feature enables the device to match on user-defined fields (UDFs) and to apply the matching packets to an IPv4 port ACL.

You can configure UDF-based port IPv6 ACLs for Cisco Nexus 9300-EX switches. This feature enables the device to match on the new UDFs and to apply the matching packets to an IPv6 port ACL.

SUMMARY STEPS

1. **configure terminal**
2. **udf** *udf-name offset-base offset length*
3. **hardware access-list tcam region ing-ifacl qualify** {**udf** *udf-name* | **v6udf** *v6udf-name*}
4. **copy running-config startup-config**
5. **reload**
6. **ip access-list** *udf-acl*
7. Enter one of the following commands:
   - **permit udf** *udf-name value mask*
   - **permit ip** *source destination* **udf** *udf-name value mask*

8. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**udf** *udf-name offset-base offset length* **Example:** `switch(config)# udf pktoff10 packet-start 10 2` **Example:** `switch(config)# udf pktoff10 header outer l3 20 2`	Defines the UDF as follows: • *udf-name*—Specifies the name of the UDF. You can enter up to 16 alphanumeric characters for the name. • *offset-base*—Specifies the UDF offset base as follows, where **header** is the packet header to consider for the offset: {**packet-start** | **header** {**outer** | **inner** {**l3** | **l4**}}}. • *offset*—Specifies the number of bytes offset from the offset base. To match the first byte from the offset base (Layer 3/Layer 4 header), configure the offset as 0. • *length*—Specifies the number of bytes from the offset. Only 1 or 2 bytes are supported. To match additional bytes, you must define multiple UDFs. You can define multiple UDFs, but Cisco recommends defining only required UDFs.
Step 3	**hardware access-list tcam region ing-ifacl qualify** {**udf** *udf-name* | **v6udf** *v6udf-name*} **Example:** `switch(config)# hardware access-list tcam region ing-ifacl qualify udf pktoff10`	Attaches the UDFs to the ing-ifacl TCAM region, which applies to IPv4 or IPv6 port ACLs. The number of UDFs that can be attached to a TCAM region varies by platform. You can attach up to 2 UDFs for Cisco Nexus 9200 switches, up to 8 UDFs for Cisco Nexus 9300 switches, and up to 18 UDFs for IPv4 port ACLs or 7 UDFs for IPv6 port ACLs for Cisco Nexus 9300-EX switches. **Note** When the UDF qualifier is added, the TCAM region goes from single wide to double wide. Make sure enough free space is available; otherwise, this command will be rejected. If necessary, you can reduce the TCAM space from unused regions and then re-enter this command. For more information, see Configuring ACL TCAM Region Sizes. **Note**

	Command or Action	Purpose
		The **no** form of this command detaches the UDFs from the TCAM region and returns the region to single wide.
Step 4	Required: **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 5	Required: **reload** **Example:** `switch(config)# reload`	Reloads the device. **Note** Your UDF configuration is effective only after you enter **copy running-config startup-config** + **reload**.
Step 6	**ip access-list** *udf-acl* **Example:** `switch(config)# ip access-list udfacl` `switch(config-acl)#`	Creates an IPv4 access control list (ACL) and enters IP access list configuration mode.
Step 7	Enter one of the following commands: • **permit udf** *udf-name value mask* • **permit ip** *source destination* **udf** *udf-name value mask* **Example:** `switch(config-acl)# permit udf pktoff10 0x1234 0xffff` **Example:** `switch(config-acl)# permit ip any any udf pktoff10 0x1234 0xffff`	Configures the ACL to match only on UDFs (example 1) or to match on UDFs along with the current access control entries (ACEs) for the outer packet fields (example 2). The range for the *value* and *mask* arguments is from 0x0 to 0xffff. A single ACL can have ACEs with and without UDFs together. Each ACE can have different UDF fields to match, or all ACEs can match for the same list of UDFs.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Applying an IP ACL as a Router ACL

You can apply an IPv4 or IPv6 ACL to any of the following types of interfaces:

- Physical Layer 3 interfaces and subinterfaces
- Layer 3 Ethernet port-channel interfaces
- VLAN interfaces
- Management interfaces

ACLs applied to these interface types are considered router ACLs.

Note Egress router ACLs are not supported on Cisco Nexus 9300 Series switch uplink ports.

Before you begin

Ensure that the ACL you want to apply exists and that it is configured to filter traffic in the manner that you need for this application.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **interface ethernet** *slot*/*port*[**.** *number*]
   - **interface port-channel** *channel-number*
   - **interface vlan** *vlan-id*
   - **interface mgmt** *port*
3. (Optional) **encapsulation dot1q 21**
4. Enter one of the following commands:
   - **ip access-group** *access-list* {**in** | **out**}
   - **ipv6 traffic-filter** *access-list* {**in** | **out**}
5. **ip access-list match-local-traffic**
6. (Optional) **show running-config aclmgr**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	Enter one of the following commands: • **interface ethernet** *slot*/*port*[**.** *number*] • **interface port-channel** *channel-number* • **interface vlan** *vlan-id* • **interface mgmt** *port* **Example:** `switch(config)# interface ethernet 2/3` `switch(config-if)#`	Enters configuration mode for the interface type that you specified.

	Command or Action	Purpose
	`switch(config)# interface ethernet 2/3.1` `switch(config-if)#`	
Step 3	(Optional) **encapsulation dot1q 21** **Example:** `switch(config-if)# encapsulation dot1q 21` `switch(config-if)#`	**Note** This command is required only for Layer 3 subinterfaces.
Step 4	Enter one of the following commands: • **ip access-group** *access-list* {**in** | **out**} • **ipv6 traffic-filter** *access-list* {**in** | **out**} **Example:** `switch(config-if)# ip access-group acl1 in`	Applies an IPv4 or IPv6 ACL to the Layer 3 interface and subinterfaces for traffic flowing in the direction specified. You can apply one router ACL per direction.
Step 5	**ip access-list match-local-traffic** **Example:** `switch(config-if)# ip access-list match-local-traffic`	Lists the matching traffic which is generated locally. It does not affect transit traffic through the switch.
Step 6	(Optional) **show running-config aclmgr** **Example:** `switch(config-if)# show running-config aclmgr`	Displays the ACL configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Creating an IP ACL, on page 176

Applying an IP ACL as a Port ACL

You can apply an IPv4 or IPv6 ACL to a Layer 2 interface, which can be a physical port or a port channel. ACLs applied to these interface types are considered port ACLs.

Note If the interface is configured with the **mac packet-classify** command, you cannot apply an IP port ACL to the interface until you remove the **mac packet-classify** command from the interface configuration.

Before you begin

Ensure that the ACL you want to apply exists and that it is configured to filter traffic in the manner that you need for this application.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **interface ethernet** *slot/port*
   - **interface port-channel** *channel-number*
3. Enter one of the following commands:
   - **ip port access-group** *access-list* **in**
   - **ipv6 port traffic-filter** *access-list* **in**
4. (Optional) **show running-config aclmgr**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	Enter one of the following commands: • **interface ethernet** *slot/port* • **interface port-channel** *channel-number* **Example:** `switch(config)# interface ethernet 2/3` `switch(config-if)#`	Enters configuration mode for the interface type that you specified.
Step 3	Enter one of the following commands: • **ip port access-group** *access-list* **in** • **ipv6 port traffic-filter** *access-list* **in** **Example:** `switch(config-if)# ip port access-group acl-l2-marketing-group in`	Applies an IPv4 or IPv6 ACL to the interface or port channel. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface.
Step 4	(Optional) **show running-config aclmgr** **Example:** `switch(config-if)# show running-config aclmgr`	Displays the ACL configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Creating an IP ACL, on page 176
Enabling or Disabling MAC Packet Classification

Applying an IP ACL as a VACL

You can apply an IP ACL as a VACL.

Related Topics

Configuring VACLs

Applying an IP ACL Rule Prioritization over SUP Rule

Beginning with Cisco NX-OS Release 10.4(1)F, a new ACE keyword (all) is supported for IP or IPv6 ACL which would increase the priority of ACL rule to 0 (highest) over any other SUP ACL rule that also matches on the same criteria.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **ip access-list** *name*
   - **ipv6 access-list** *name*
3. [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ip-prefix* | *source-ip-mask*} {*destination-ip-prefix* | *destination-ip-mask*} **all**
4. Enter one of the following commands:
   - **interface ethernet** *slot*/*port*
   - **interface port-channel** *channel-number*
5. Enter one of the following commands:
   - **ip port access-group** *access-list* **in**
   - **ipv6 port traffic-filter** *access-list* **in**
6. (Optional) **show running-config aclmgr**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	Enter one of the following commands: • **ip access-list** *name* • **ipv6 access-list** *name* **Example:** `switch(config)# ip access-list acl-01` `switch(config-acl)#`	Creates the IP or IPv6 ACL and enters the ACL configuration mode. The *name* argument can be up to 64 characters.
Step 3	[*sequence-number*] {**permit** | **deny**} *protocol* {*source-ip-prefix* | *source-ip-mask*} {*destination-ip-prefix* | *destination-ip-mask*} **all** **Example:** For IP `switch(config-acl)# permit ip 192.168.2.0/24 any all` For IPv6 `switch(config-ipv6-acl)# 10 permit ipv6 1::1 2::2 3::3 4::4 all`	Creates a rule in the IP or IPv6 ACL with an **all** keyword to prioritize the ACL rule over the SUP rule.
Step 4	Enter one of the following commands: • **interface ethernet** *slot*/*port* • **interface port-channel** *channel-number* **Example:** `switch(config)# interface ethernet 2/3` `switch(config-if)#`	Enters configuration mode for the interface type that you specified.
Step 5	Enter one of the following commands: • **ip port access-group** *access-list* **in** • **ipv6 port traffic-filter** *access-list* **in** **Example:** For IP `switch(config-if)# ip port access-group acl-01 in`	Applies an IPv4 or IPv6 ACL to the interface or port channel. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface.
Step 6	(Optional) **show running-config aclmgr** **Example:** `switch(config-if)# show running-config aclmgr`	Displays the ACL configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring ACL Logging

To configure the ACL logging process, you first create the access list, then enable filtering of traffic on an interface using the specified ACL, and finally configure the ACL logging process parameters.

SUMMARY STEPS

1. **configure terminal**
2. **ip access-list** *name*
3. {**permit** | **deny**} **ip** *source-address destination-address* **log**
4. **exit**
5. **interface ethernet** *slot/port*
6. **ip access-group** *name* **in**
7. **exit**
8. **logging ip access-list cache interval** *interval*
9. **logging ip access-list cache entries** *number-of-flows*
10. **logging ip access-list cache threshold** *threshold*
11. **logging ip access-list detailed**
12. **hardware rate-limiter access-list-log** *packets*
13. **acllog match-log-level** *severity-level*
14. (Optional) **logging ip access-list include sgt**
15. (Optional) **logging ip access-list include mac**
16. (Optional) **show logging ip access-list cache** [**detail**]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**ip access-list** *name* **Example:** `switch(config)# ip access-list logging-test` `switch(config-acl)#`	Creates an IPv4 ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.
Step 3	{**permit** | **deny**} **ip** *source-address destination-address* **log** **Example:** `switch(config-acl)# permit ip any 10.30.30.0/24 log`	Creates an ACL rule that permits or denies IPv4 traffic matching its conditions. To enable the system to generate an informational logging message about each packet that matches the rule, you must include the **log** keyword. The *source-address* and *destination-address* arguments can be the IP address with a network wildcard, the IP address and variable-length subnet mask, the host address, or **any** to designate any address.

	Command or Action	Purpose
Step 4	**exit** **Example:** `switch(config-acl)# exit` `switch(config)#`	Updates the configuration and exits IP ACL configuration mode.
Step 5	**interface ethernet** *slot/port* **Example:** `switch(config)# interface ethernet 1/1` `switch(config-if)#`	Enters interface configuration mode.
Step 6	**ip access-group** *name* **in** **Example:** `switch(config-if)# ip access-group logging-test in`	Enables the filtering of IPv4 traffic on an interface using the specified ACL. You can apply an ACL to inbound traffic.
Step 7	**exit** **Example:** `switch(config-if)# exit` `switch(config)#`	Updates the configuration and exits interface configuration mode.
Step 8	**logging ip access-list cache interval** *interval* **Example:** `switch(config)# logging ip access-list cache interval 490`	Configures the log-update interval (in seconds) for the ACL logging process. The default value is 300 seconds. The range is from 5 to 86400 seconds.
Step 9	**logging ip access-list cache entries** *number-of-flows* **Example:** `switch(config)# logging ip access-list cache entries 8001`	Specifies the maximum number of flows to be monitored by the ACL logging process. The default value is 8000. The range of values supported is from 0 to 1048576.
Step 10	**logging ip access-list cache threshold** *threshold* **Example:** `switch(config)# logging ip access-list cache threshold 490`	If the specified number of packets is logged before the expiry of the alert interval, the system generates a syslog message.
Step 11	**logging ip access-list detailed** **Example:** `switch(config)# logging ip access-list detailed`	Enables the following information to be displayed in the output of the **show logging ip access-list cache** command: the access control entry (ACE) sequence number, ACE action, ACL name, ACL direction, ACL filter type, and ACL applied interface.
Step 12	**hardware rate-limiter access-list-log** *packets* **Example:** `switch(config)# hardware rate-limiter access-list-log 200`	Configures rate limits in packets per second for packets copied to the supervisor module for ACL logging. The range is from 0 to 30000.

	Command or Action	Purpose
Step 13	**acllog match-log-level** *severity-level* **Example:** `switch(config)# acllog match-log-level 5`	Specifies the minimum severity level to log ACL matches. The default is 6 (informational). The range is from 0 (emergency) to 7 (debugging).
Step 14	(Optional) **logging ip access-list include sgt** **Example:** `switch(config)# logging ip access-list include sgt`	Enables the following SGACL information to be displayed in the output of the **show logging ip access-list cache** command: the security group tag (SGT), destination group tag (DGT), source MAC (SMAC) and destination MAC (DMAC).
Step 15	(Optional) **logging ip access-list include mac** **Example:** `switch(config)# logging ip access-list include mac`	Enables to include MAC address as part of the ACLLOG entry to be displayed in the output of the **show logging ip access-list cache** command: source MAC (SMAC) and destination MAC (DMAC). This command can be configured along with detailed option (**logging ip access-list detailed**), or without detailed option.
Step 16	(Optional) **show logging ip access-list cache** [**detail**] **Example:** `switch(config)# show logging ip access-list cache`	Displays information on the active logged flows, such as source IP and destination IP addresses, source port and destination port information, source interfaces. No other information of active flows will be displayed specifically all the unsupported options. If you entered the **logging ip access-list detailed** command, the output also includes the following information: the access control entry (ACE) sequence number, ACE action, ACL name, ACL direction, ACL filter type, and ACL applied interface. **Note** • The following commands are mutually exclusive: • **logging ip access-list detailed** • **logging ip access-list include sgt** • The output format of the **show logging ip access-list cache** [**detail**] command will be based on the chosen optional configurations.

Configuring ACLs Using HTTP Methods to Redirect Requests

You can configure ACLs to intercept and redirect specific HTTP methods to a server that is connected to a specific port.

The following HTTP methods can be redirected:

- connect

- delete
- get
- head
- post
- put
- trace

Before you begin

Enable the double-wide TCAM for the IFACL region using the **hardware access-list tcam region ifacl 512 double-wide** command. This command applies to the global configuration. Reload the switch for this configuration to take into effect.

SUMMARY STEPS

1. **configure terminal**
2. **ip access-list** *name*
3. [*sequence-number*] **permit** *protocol source destination* **http-method** *method* [**tcp-option-length** *length*] [**redirect** *interface*]
4. (Optional) **show ip access-lists** *name*
5. (Optional) **show run interface** *interface slot/port*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**ip access-list** *name* **Example:** `switch(config)# ip access-list acl-01` `switch(config-acl)#`	Creates the IP ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.
Step 3	[*sequence-number*] **permit** *protocol source destination* **http-method** *method* [**tcp-option-length** *length*] [**redirect** *interface*] **Example:** `switch(config-acl)# permit tcp 1.1.1.1/32 any http-method get`	Configures the ACL to redirect specific HTTP methods to a server. The following HTTP methods are supported: • connect—Matches HTTP packets with the CONNECT method [0x434f4e4e] • delete—Matches HTTP packets with the DELETE method [0x44454c45]

	Command or Action	Purpose
		• get—Matches HTTP packets with the GET method [0x47455420] • head—Matches HTTP packets with the HEAD method [0x48454144] • post—Matches HTTP packets with the POST method [0x504f5354] • put—Matches HTTP packets with the PUT method [0x50555420] • trace—Matches HTTP packets with the TRACE method [0x54524143] The **tcp-option-length** option specifies the length of the TCP options header in the packets. You can configure up to four TCP option lengths (in multiples of four bytes) in the access control entries (ACEs). The *length* range is from 0 to 40. If you do not configure this option, the length is specified as 0, and only packets without the TCP options header can match the ACE. This option allows the HTTP method to be matched even on packets that have a variable-length TCP options header. The **redirect** option redirects an HTTP method to a server that is connected to a specific port. The HTTP redirect feature does not work on Layer 3 ports.
Step 4	(Optional) **show ip access-lists** *name* **Example:** `switch(config-acl)# show ip access-lists acl-01`	Displays the IP ACL configuration.
Step 5	(Optional) **show run interface** *interface slot/port* **Example:** `switch(config-acl)# show run interface ethernet 2/2`	Displays the interface configuration.

Example

The following example specifies a length for the TCP options header in the packets and redirects the post HTTP method to a server that is connected to port channel 4001:

```
switch(config)# ip access-list http-redirect-acl
switch(config-acl)# 10 permit tcp any any http-method get tcp-option-length 4 redirect
port-channel4001
switch(config-acl)# 20 permit tcp any any http-method post redirect port-channel4001
switch(config-acl)# statistics per-entry
switch(config)# interface Ethernet 1/33
switch(config-if)# ip port access-group http-redirect-acl in
```

Configuring an ACL for IPv6 Extension Headers

This procedure applies only to the following devices:

- Cisco Nexus 9504 and 9508 modular chassis with these line cards: N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX, and N9K-X96136YC-R
- Cisco Nexus 3600 Platform Switches (N3K-C36180YC-R and N3K-C3636C-R)

Beginning with Cisco NX-OS Release 9.3(7), if you configure an IPv6 ACL on the devices listed here, you must include a new rule for the disposition of IPv6 packets that include extension headers. For more information about IPv6 extension headers, see "Simplified IPv6 Packet Header" in NX-OS Release 9.3(x) or later of the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*.

Note The permit or deny rule that you choose in this procedure is applied to any IPv6 packet with at least one extension header regardless of any other ACL rule that matches the packet's other fields.

SUMMARY STEPS

1. **configure terminal**
2. **ipv6 access-list** *name*
3. **extension-header** {**permit-all** | **deny-all**}

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**ipv6 access-list** *name* **Example:** `switch(config)# ipv6 access-list acl-01` `switch(config-acl)#`	Creates the IPv6 ACL and enters ACL configuration mode.
Step 3	**extension-header** {**permit-all**	**deny-all**} **Example:** `switch(config-acl)# extension-header permit-all` `switch(config-acl)#`	Choose the desired action for matched packets: • **permit-all** — Any IPv6 packet with at least one extension header is permitted. • **deny-all** — Any IPv6 packet with at least one extension header is dropped.

Verifying the IP ACL Configuration

To display IP ACL configuration information, perform one of the following tasks.

Command	Purpose
show hardware access-list tcam region	Displays the TCAM sizes that will be applicable on the next reload of the device.
show hardware access-list tcam template {**all** | **nfe** | **nfe2** | **l2-l3** | **l3** | *template-name*}	Displays the configuration for all TCAM templates or for a specific template. **nfe**—The default TCAM template for Network Forwarding Engine (NFE)-enabled Cisco Nexus 9300 and 9500 Series devices. **nfe2**—The default TCAM template for NFE2-enabled Cisco Nexus 9500 devices. **l2-l3**—The default TCAM template for Layer 2 and Layer 3 configurations. **l3**—The default TCAM template for Layer 3 configurations.
show ip access-lists	Displays the IPv4 ACL configuration.
show ipv6 access-lists	Displays the IPv6 ACL configuration.
show logging ip access-list cache [**detail**]	Displays information on the active logged flows, such as source IP and destination IP addresses, source port and destination port information, and source interfaces. No other information of active flows will be displayed specifically all the unsupported options. If you entered the **logging ip access-list detailed** command, the output also includes the following information: the access control entry (ACE) sequence number, ACE action, ACL name, ACL direction, ACL filter type, and ACL applied interface.

Command	Purpose
show logging ip access-list status	Displays the deny maximum flow count, the current effective log interval, and the current effective threshold value.
show running-config acllog	Displays the ACL log running configuration.
show running-config aclmgr [**all**]	Displays the ACL running configuration, including the IP ACL configuration and the interfaces to which IP ACLs are applied. **Note** This command displays the user-configured ACLs in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration.
show startup-config acllog	Displays the ACL log startup configuration.
show startup-config aclmgr [**all**]	Displays the ACL startup configuration. **Note** This command displays the user-configured ACLs in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration.

Command	Purpose
show hardware access-list interface *ethernet X/Y* **input entries detail**	Displays the hardware ACL interface input entries' detail. **Note** On platforms other than 9500-R, even when the entry is expanded, the display shows the range as x y. **Sample output for 9500-R:** `permit tcp 100.1.1.0/24 eq 10006 100.1.1.0/24 eq 0x4e24/fffe [0]` **Sample output for 9300-FX3S:** `permit tcp 100.1.1.0/24 eq 10006 100.1.1.0/24 range 20004 20005 routeable 0x1 [0]`

Monitoring and Clearing IP ACL Statistics

To monitor or clear IP ACL statistics, use one of the commands in this table.

Command	Purpose
show ip access-lists	Displays the IPv4 ACL configuration. If the IPv4 ACL includes the **statistics per-entry** command, the **show ip access-lists** command output includes the number of packets that have matched each rule.
show ipv6 access-lists	Displays IPv6 ACL configuration. If the IPv6 ACL includes the **statistics per-entry** command, then the **show ipv6 access-lists** command output includes the number of packets that have matched each rule.
clear ip access-list counters	Clears statistics for all IPv4 ACLs or for a specific IPv4 ACL.
clear ipv6 access-list counters	Clears statistics for all IPv6 ACLs or for a specific IPv6 ACL.

Configuration Examples for IP ACLs

The following example shows how to create an IPv4 ACL named acl-01 and apply it as a port ACL to Ethernet interface 2/1, which is a Layer 2 interface:

```
ip access-list acl-01
  permit ip 192.168.2.0/24 any
interface ethernet 2/1
  ip port access-group acl-01 in
```

The following example shows how to create an IPv6 ACL named acl-120 and apply it as a router ACL to Ethernet interface 2/3, which is a Layer 3 interface:

```
ipv6 access-list acl-120
  permit tcp 2001:0db8:85a3::/48 2001:0db8:be03:2112::/64
  permit udp 2001:0db8:85a3::/48 2001:0db8:be03:2112::/64
  permit tcp 2001:0db8:69f2::/48 2001:0db8:be03:2112::/64
  permit udp 2001:0db8:69f2::/48 2001:0db8:be03:2112::/64
interface ethernet 2/3
  ipv6 traffic-filter acl-120 in
```

The following example shows how to create a VTY ACL named single-source and apply it on input IP traffic over the VTY line. This ACL allows all TCP traffic through and drops all other IP traffic:

```
ip access-list single-source
  permit tcp 192.168.7.5/24 any
  exit
  line vty
  ip access-class single-source in
  show ip access-lists
```

The following example shows how to configure IPv4 ACL logging:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# ip access-list logging-test
switch(config-acl)# permit ip any 2001:DB8:1::1/64 log
switch(config-acl)# exit
switch(config)# interface ethernet 1/1
switch(config-if)# ip access-group logging-test in
switch(config-if)# exit
switch(config)# logging ip access-list cache interval 400
switch(config)# logging ip access-list cache entries 100
switch(config)# logging ip access-list cache threshold 900
switch(config)# hardware rate-limiter access-list-log 200
switch(config)# acllog match-log-level 5
```

The following example shows how to configure a UDF-based port ACL:

```
switch# configure terminal
switch(config)# hardware access-list tcam region ing-ifacl 256
switch(config)# udf pktoff10 packet-start 10 2
switch(config)# udf pktoff20 packet-start 10 1
switch(config)# hardware access-list tcam region ing-ifacl qualify udf pktoff10 pktoff20

switch# configure terminal
switch(config)# ip access-list udfacl
switch(config-acl)# statistics per-entry
switch(config-acl)# 10 permit ip any any udf pktoff10 0x1234 0xffff

switch# configure terminal
switch(config)# interface Ethernet1/1
switch(config-if)# ip port access-group udfacl in
switch(config-if)# switchport
switch(config-if)# no shutdown
```

The following example shows the configuration of the **route-tag default-route**:

```
switch(config)# ip access-list global-acl
switch(config-acl)# 10 permit ip any any route-tag default-route
switch(config-acl)# exit

switch(config)#class-map type qos global
switch(config-cmap-qos)#match access-group name global-acl
```

```
switch(config)#class-map type qos domestic
switch(config-cmap-qos)#match access-group name domestic-acl

switch(config)#policy-map type qos pmap
switch(config-pmap)#class global
switch(config-pmap-c)#police cir 100 mbps bc 200 ms conform transmit violate drop
switch(config-pmap)#class domestic
switch(config-pmap-c)#police cir 200 mbps bc 200 ms conform transmit violate drop

switch(config)#interface ethernet1/12
switch(config-if)#service-policy type qos input pmap

switch(config)# show running-config ipqos
!Running configuration last done at: Tue Jun 13 10:08:38 2023
!Time: Tue Jun 13 10:10:05 2023
version 10.4(2) Bios:version 01.08
class-map type qos match-all global
match access-group name global-acl
class-map type qos match-all domestic
match access-group name domestic-acl
policy-map type qos pmap
class global
police cir 100 mbps bc 200 ms conform transmit violate drop
class domestic
police cir 200 mbps bc 200 ms conform transmit violate drop
```

About System ACLs

You can configure system ACLs on Cisco Nexus 9500 Series switches with -R and -RX line cards. With system ACLs, you can now configure a Layer 2 port ACL (PACL) on all the ports with the same access-list in the switch. Configuring system ACLs reduces the TCAM usage and also brings down the time and memory usage while the policy is being applied or modified.

See the following guidelines and limitations for configuring system ACLs:

- The system PACL is supported for Layer 2 interface only.
- Up to 10K ACEs are supported with all other basic features for the switch to come up on Cisco Nexus 9500 Series switches with -R line cards. The hardware capacity on Cisco Nexus 9500 Series switches with -RX line cards is 64K ACEs.
- You can also configure system ACLs on Cisco Nexus 3600 platform switches with N3K-C3636C-R and N3K-C36180YC-R line cards.
- Configuring IPv4 PACL TCAM region (ifacl) with anything more than the total physical TCAM capacity of -R line cards of 12k results in the power down of -R line cards only.
- ACE statistics are not yet supported for the system ACLs.
- IPv6 is not yet supported in the system ACLs.
- System ACLs are not supported on the breakout port.
- For quality of service, ACL, or TCAM carving configuration on Cisco Nexus Series switches with -R series line cards, see the Cisco Nexus 3600 NX-OS Quality of Service Configuration Guide, Release 7.x for more information.
- The non-atomic update either drops or permits all the traffic. By default, the non-atomic update drops all the traffic until the ACL update completes. The non-atomic ACL update behavior can be controlled

using the **hardware access-list update default-result permit** CLI command. This CLI works only for physical ports. See the following example:

```
hardware access-list update default-result permit     => #Allows all the traffic during
 ACL updates. There may be < 10secs traffic drop.
no hardware access-list update default-result permit   => #This is the default behavior.
 It denies all the traffic during ACL updates.
```

- In Cisco NX-OS Release 9.2(2) and earlier releases, although the atomic ACL update is not supported on Cisco Nexus -R series line cards, the non-atomic update **hardware access-list update default-result** is supported on the Cisco Nexus -R series line cards.

Carving a TCAM Region

Before configuring the system ACLs, carve the TCAM region first. Note that for configuring the ACLs less than 1k, you do not need to carve the TCAM region. See the Configuring ACL TCAM Region Sizes, on page 184 section for more information.

Note Beginning with Cisco NX-OS Release 7.0(3)F3(4) or a later release, you can configure PACL IPv4, RACL IPv4, and RACL IPv6 beyond 12k.

Configuring System ACLs

After an IPv4 ACL is created, configure the system ACL.

Before you begin

Create an IPv4 ACL on the device. See Creating an IP ACL, on page 176 for more information.

SUMMARY STEPS

1. **config t**
2. **system acl**
3. **ip port access-group** <*pacl name*> **in**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**config t**	Enters the configuration mode.
Step 2	**system acl**	Configures the system ACL.
Step 3	**ip port access-group** <*pacl name*> **in**	Applies a Layer 2 PACL to the interface. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface.

Configuration and Show Command Examples for the System ACLs

See the following configuration examples for the system ACL show commands.

Configuring system PACL with 1K scale [using default TCAM]

See the following example for configuring system PACL with 1K scale [Using default TCAM].

Step 1: Create PACL.

```
config t
ip access-list PACL-DNA
    10 permit ip 1.1.1.1/32 any
    20 permit tcp 3.0.0.0/8 255.0.0.0 eq 1500
    25 deny udp any any eq 500
    26 deny tcp any eq 490 any
    …... ….

    1000 deny any any
```

Step 2: Apply PACL into system level.

```
configuration terminal
system acl
    ip port access-group PACL-DNA in
```

To validate the system ACLs that are configured on the switch, use the **sh run aclmgr | sec system** command:

```
switch# sh run aclmgr | sec system
system acl
 ip port access-group test in
switch#
```

To validate the PACLs that are configured on the switch, use the **sh ip access-lists <name> [summary]** command:

```
switch# sh ip access-lists test

IP access list test
        10 deny udp any any eq 27
        20 permit ip 1.1.1.1/32 100.100.100.100/32
        30 permit ip 1.2.1.1/32 100.100.100.100/32
        40 permit ip 1.3.1.1/32 100.100.100.100/32
        50 permit ip 1.4.1.1/32 100.100.100.100/32
        60 permit ip 1.5.1.1/32 100.100.100.100/32
        70 permit ip 1.6.1.1/32 100.100.100.100/32
        80 permit ip 1.7.1.1/32 100.100.100.100/32
        90 permit ip 1.8.1.1/32 100.100.100.100/32

switch# sh ip access-lists test summary
IPV4 ACL test
        Total ACEs Configured: 12279
        Configured on interfaces:
        Active on interfaces:
                - ingress
                - ingress
```

```
switch#
```

To validate PACL IPv4 (ifacl) TCAM region size, use the **show hardware access-list tcam region** command:

```
switch# show hardware access-list tcam region
*********************************WARNING*********************************
*****************The output shows NFE tcam region info******************
***Please refer to 'show hardware access-list tcam template' for NFE2***
*************************************************************************
                              IPV4 PACL [ifacl] size = 12280
                          IPV6 PACL [ipv6-ifacl] size =    0
                            MAC PACL [mac-ifacl] size =    0
                             IPV4 Port QoS [qos] size =  640
                         IPV6 Port QoS [ipv6-qos] size =  256
                           MAC Port QoS [mac-qos] size =    0
                       FEX IPV4 PACL [fex-ifacl] size =    0
                   FEX IPV6 PACL [fex-ipv6-ifacl] size =    0
                     FEX MAC PACL [fex-mac-ifacl] size =    0
                      FEX IPV4 Port QoS [fex-qos] size =    0
                  FEX IPV6 Port QoS [fex-ipv6-qos] size =    0
                    FEX MAC Port QoS [fex-mac-qos] size =    0
                                 IPV4 VACL [vacl] size =    0
                            IPV6 VACL [ipv6-vacl] size =    0
                              MAC VACL [mac-vacl] size =    0
                             IPV4 VLAN QoS [vqos] size =    0
                         IPV6 VLAN QoS [ipv6-vqos] size =    0
                           MAC VLAN QoS [mac-vqos] size =    0
                                 IPV4 RACL [racl] size =    0
                            IPV6 RACL [ipv6-racl] size =  128
                   IPV4 Port QoS Lite [qos-lite] size =    0
            FEX IPV4 Port QoS Lite [fex-qos-lite] size =    0
                  IPV4 VLAN QoS Lite [vqos-lite] size =    0
                   IPV4 L3 QoS Lite [l3qos-lite] size =    0
                        Egress IPV4 QoS [e-qos] size =    0
                    Egress IPV6 QoS [e-ipv6-qos] size =    0
                      Egress MAC QoS [e-mac-qos] size =    0
                         Egress IPV4 VACL [vacl] size =    0
                    Egress IPV6 VACL [ipv6-vacl] size =    0
                      Egress MAC VACL [mac-vacl] size =    0
                       Egress IPV4 RACL [e-racl] size =    0
                  Egress IPV6 RACL [e-ipv6-racl] size =    0
              Egress IPV4 QoS Lite [e-qos-lite] size =    0
                                IPV4 L3 QoS [l3qos] size =  640
                          IPV6 L3 QoS [ipv6-l3qos] size =  256
                            MAC L3 QoS [mac-l3qos] size =    0
                                    Ingress System size =    0
                                     Egress System size =    0
                                       SPAN [span] size =   96
                                Ingress COPP [copp] size =  128
                     Ingress Flow Counters [flow] size =    0
switch#
```

To view ACL related tech support information, use the **show tech-support aclmgr** and **show tech-support aclqos** commands.

```
show tech-support aclmgr
show tech-support aclqos
```

Configuring Object Groups

You can use object groups to specify source and destination addresses and protocol ports in IPv4 ACL and IPv6 ACL rules.

Session Manager Support for Object Groups

Session Manager supports the configuration of object groups. This feature allows you to create a configuration session and verify your object group configuration changes prior to committing them to the running configuration. For more information about Session Manager, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.

Creating and Changing an IPv4 Address Object Group

You can create and change an IPv4 address group object.

Note Beginning Cisco Nexus Release 7.0(3)I5(2), the **no host IPv4-address** command is not supported. With the DME support, deletion without the no sequence command is not supported.

SUMMARY STEPS

1. **configure terminal**
2. **object-group ip address** *name*
3. Enter one of the following commands:
   - [*sequence-number*] **host** *IPv4-address*
   - [*sequence-number*] *IPv4-address*/*prefix-len*
   - [*sequence-number*] *IPv4-address network-wildcard*
4. Enter one of the following commands:
   - **no** [*sequence-number*]
   - **no host** *IPv4-address*
   - **no** *IPv4-address*/*prefix-len*
   - **no** *IPv4-address network-wildcard*
5. (Optional) **show object-group** *name*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**object-group ip address** *name* **Example:** `switch(config)# object-group ip address ipv4-addr-group-13` `switch(config-ipaddr-ogroup)#`	Creates the IPv4 address object group and enters IPv4 address object-group configuration mode.
Step 3	Enter one of the following commands: • [*sequence-number*] **host** *IPv4-address* • [*sequence-number*] *IPv4-address*/*prefix-len* • [*sequence-number*] *IPv4-address network-wildcard* **Example:** `switch(config-ipaddr-ogroup)# host 10.99.32.6`	Creates an entry in the object group. For each entry that you want to create, use the **host** command and specify a single host, or omit the **host** command to specify a network of hosts. You can specify a prefix length for an IPv4 object group, which matches only on the first contiguous bits, or you can specify a wildcard mask, which matches on any bit in the address.
Step 4	Enter one of the following commands: • **no** [*sequence-number*] • **no host** *IPv4-address* • **no** *IPv4-address*/*prefix-len* • **no** *IPv4-address network-wildcard* **Example:** `switch(config-ipaddr-ogroup)# no host 10.99.32.6`	Removes an entry in the object group. For each entry that you want to remove from the object group, use the **no** form of the **host** command.
Step 5	(Optional) **show object-group** *name* **Example:** `switch(config-ipaddr-ogroup)# show object-group ipv4-addr-group-13`	Displays the object group configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-ipaddr-ogroup)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Creating and Changing an IPv6 Address Object Group

You can create and change an IPv6 address group object.

SUMMARY STEPS

1. **configure terminal**
2. **object-group ipv6 address** *name*
3. Enter one of the following commands:
   - [*sequence-number*] **host** *IPv6-address*
   - [*sequence-number*] *IPv6-address*/*prefix-len*
   - [*sequence-number*] *IPv6-address network-wildcard*
4. Enter one of the following commands:
   - **no** *sequence-number*
   - **no host** *IPv6-address*
   - **no** *IPv6-address*/*prefix-len*
5. (Optional) **show object-group** *name*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**object-group ipv6 address** *name* **Example:** `switch(config)# object-group ipv6 address ipv6-addr-group-A7` `switch(config-ipv6addr-ogroup)#`	Creates the IPv6 address object group and enters IPv6 address object-group configuration mode.
Step 3	Enter one of the following commands: • [*sequence-number*] **host** *IPv6-address* • [*sequence-number*] *IPv6-address*/*prefix-len* • [*sequence-number*] *IPv6-address network-wildcard* **Example:** `switch(config-ipv6addr-ogroup)# host 2001:db8:0:3ab0::1` **Example:** `switch(config-ipv6addr-ogroup)# 10 1::1 2::2`	Creates an entry in the object group. For each entry that you want to create, use the **host** command and specify a single host, or omit the **host** command to specify a network of hosts. You can specify a prefix length for an IPv6 object group, which matches only on the first contiguous bits, or you can specify a wildcard mask, which matches on any bit in the address. IPv6 wildcard masks are supported for Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2/FXP switches and the Cisco Nexus 9364C switch.
Step 4	Enter one of the following commands: • **no** *sequence-number* • **no host** *IPv6-address*	Removes an entry from the object group. For each entry that you want to remove from the object group, use the **no** form of the **host** command.

	Command or Action	Purpose
	• **no** *IPv6-address/prefix-len* **Example:** `switch(config-ipv6addr-ogroup)# no host 2001:db8:0:3ab0::1`	
Step 5	(Optional) **show object-group** *name* **Example:** `switch(config-ipv6addr-ogroup)# show object-group ipv6-addr-group-A7`	Displays the object group configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-ipv6addr-ogroup)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Creating and Changing a Protocol Port Object Group

You can create and change a protocol port object group.

SUMMARY STEPS

1. **configure terminal**
2. **object-group ip port** *name*
3. [*sequence-number*] *operator port-number* [*port-number*]
4. **no** {*sequence-number* | *operator port-number* [*port-number*]}
5. (Optional) **show object-group** *name*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**object-group ip port** *name* **Example:** `switch(config)# object-group ip port NYC-datacenter-ports` `switch(config-port-ogroup)#`	Creates the protocol port object group and enters port object-group configuration mode.

	Command or Action	Purpose
Step 3	[*sequence-number*] *operator port-number* [*port-number*] **Example:** `switch(config-port-ogroup)# eq 80`	Creates an entry in the object group. For each entry that you want to create, use one of the following operator commands: • **eq**—Matches only the port number that you specify. • **gt**—Matches port numbers that are greater than (and not equal to) the port number that you specify. • **lt**—Matches port numbers that are less than (and not equal to) the port number that you specify. • **neq**—Matches all port numbers except for the port number that you specify. • **range**—Matches the range of port numbers between and including the two port numbers that you specify. **Note** The **range** command is the only operator command that requires two *port-number* arguments.
Step 4	**no** {*sequence-number* | *operator port-number* [*port-number*]} **Example:** `switch(config-port-ogroup)# no eq 80`	Removes an entry from the object group. For each entry that you want to remove, use the **no** form of the applicable operator command.
Step 5	(Optional) **show object-group** *name* **Example:** `switch(config-port-ogroup)# show object-group NYC-datacenter-ports`	Displays the object group configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-port-ogroup)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Removing an Object Group

You can remove an IPv4 address object group, an IPv6 address object group, or a protocol port object group.

SUMMARY STEPS

1. **configure terminal**
2. **no object-group** {**ip address** | **ipv6 address** | **ip port**} *name*
3. (Optional) **show object-group**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**no object-group** {**ip address** | **ipv6 address** | **ip port**} *name* **Example:** `switch(config)# no object-group ip address ipv4-addr-group-A7`	Removes the specified object group.
Step 3	(Optional) **show object-group** **Example:** `switch(config)# show object-group`	Displays all object groups. The removed object group should not appear.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the Object-Group Configuration

To display object-group configuration information, enter one of the following commands:

Command	Purpose
show object-group	Displays the object-group configuration.
show {**ip** | **ipv6**} **access-lists** *name* [**expanded**]	Displays expanded statistics for the ACL configuration.
show running-config aclmgr	Displays the ACL configuration, including object groups.

Configuring Time-Ranges

Session Manager Support for Time-Ranges

Session Manager supports the configuration of time ranges. This feature allows you to create a configuration session and verify your time-range configuration changes prior to committing them to the running configuration. For more information about Session Manager, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.

Creating a Time-Range

You can create a time range on the device and add rules to it.

SUMMARY STEPS

1. **configure terminal**
2. **time-range** *name*
3. (Optional) [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time*
4. (Optional) [*sequence-number*] **periodic** *list-of-weekdays time* **to** *time*
5. (Optional) [*sequence-number*] **absolute start** *time date* [**end** *time date*]
6. (Optional) [*sequence-number*] **absolute** [start *time date*] **end** *time date*
7. (Optional) **show time-range** *name*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**time-range** *name* **Example:** `switch(config)# time-range workday-daytime` `switch(config-time-range)#`	Creates the time range and enters time-range configuration mode.
Step 3	(Optional) [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time* **Example:** `switch(config-time-range)# periodic monday 00:00:00 to friday 23:59:59`	Creates a periodic rule that is in effect for one or more contiguous days between and including the specified start and end days and times.
Step 4	(Optional) [*sequence-number*] **periodic** *list-of-weekdays time* **to** *time* **Example:** `switch(config-time-range)# periodic weekdays 06:00:00 to 20:00:00`	Creates a periodic rule that is in effect on the days specified by the *list-of-weekdays* argument between and including the specified start and end times. The following keywords are also valid values for the *list-of-weekdays* argument: • **daily** —All days of the week. • **weekdays** —Monday through Friday. • **weekend** —Saturday through Sunday.
Step 5	(Optional) [*sequence-number*] **absolute start** *time date* [**end** *time date*]	Creates an absolute rule that is in effect beginning at the time and date specified after the **start** keyword. If you

	Command or Action	Purpose
	Example: `switch(config-time-range)# absolute start 1:00 15 march 2013`	omit the **end** keyword, the rule is always in effect after the start time and date have passed.
Step 6	(Optional) [*sequence-number*] **absolute** [start *time date*] **end** *time date* **Example:** `switch(config-time-range)# absolute end 23:59:59 31 may 2013`	Creates an absolute rule that is in effect until the time and date specified after the **end** keyword. If you omit the **start** keyword, the rule is always in effect until the end time and date have passed.
Step 7	(Optional) **show time-range** *name* **Example:** `switch(config-time-range)# show time-range workday-daytime`	Displays the time-range configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config-time-range)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Changing a Time-Range

You can add and remove rules in an existing time range. You cannot change existing rules. Instead, to change a rule, you can remove it and recreate it with the desired changes.

If you need to add more rules between existing rules than the current sequence numbering allows, you can use the **resequence** command to reassign sequence numbers.

SUMMARY STEPS

1. **configure terminal**
2. **time-range** *name*
3. (Optional) [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time*
4. (Optional) [*sequence-number*] **periodic** *list-of-weekdays time* **to** *time*
5. (Optional) [*sequence-number*] **absolute start** *time date* [**end** *time date*]
6. (Optional) [*sequence-number*] **absolute** [**start** *time date*] **end** *time date*
7. (Optional) **no** {*sequence-number* | **periodic** *arguments . . .* | **absolute** *arguments. . .*}
8. (Optional) **show time-range** *name*
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**time-range** *name* **Example:** `switch(config)# time-range workday-daytime` `switch(config-time-range)#`	Enters time-range configuration mode for the specified time range.
Step 3	(Optional) [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time* **Example:** `switch(config-time-range)# periodic monday 00:00:00 to friday 23:59:59`	Creates a periodic rule that is in effect for one or more contiguous days between and including the specified start and end days and times.
Step 4	(Optional) [*sequence-number*] **periodic** *list-of-weekdays time* **to** *time* **Example:** `switch(config-time-range)# 100 periodic weekdays 05:00:00 to 22:00:00`	Creates a periodic rule that is in effect on the days specified by the *list-of-weekdays* argument between and including the specified start and end times. The following keywords are also valid values for the *list-of-weekdays* argument: • **daily** —All days of the week. • **weekdays** —Monday through Friday. • **weekend** —Saturday through Sunday.
Step 5	(Optional) [*sequence-number*] **absolute start** *time date* [**end** *time date*] **Example:** `switch(config-time-range)# absolute start 1:00 15 march 2013`	Creates an absolute rule that is in effect beginning at the time and date specified after the **start** keyword. If you omit the **end** keyword, the rule is always in effect after the start time and date have passed.
Step 6	(Optional) [*sequence-number*] **absolute** [**start** *time date*] **end** *time date* **Example:** `switch(config-time-range)# absolute end 23:59:59 31 may 2013`	Creates an absolute rule that is in effect until the time and date specified after the **end** keyword. If you omit the **start** keyword, the rule is always in effect until the end time and date have passed.
Step 7	(Optional) **no** {*sequence-number* | **periodic** *arguments . . .* | **absolute** *arguments. . .*} **Example:** `switch(config-time-range)# no 80`	Removes the specified rule from the time range.

	Command or Action	Purpose
Step 8	(Optional) **show time-range** *name* **Example:** `switch(config-time-range)# show time-range workday-daytime`	Displays the time-range configuration.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config-time-range)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Changing Sequence Numbers in a Time Range, on page 233

Removing a Time-Range

You can remove a time range from the device.

Before you begin

Ensure that you know whether the time range is used in any ACL rules. The device allows you to remove time ranges that are used in ACL rules. Removing a time range that is in use in an ACL rule does not affect the configuration of interfaces where you have applied the ACL. Instead, the device considers the ACL rule using the removed time range to be empty.

SUMMARY STEPS

1. **configure terminal**
2. **no time-range** *name*
3. (Optional) **show time-range**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**no time-range** *name* **Example:** `switch(config)# no time-range daily-workhours`	Removes the time range that you specified by name.

	Command or Action	Purpose
Step 3	(Optional) **show time-range** **Example:** `switch(config-time-range)# show time-range`	Displays the configuration for all time ranges. The removed time range should not appear.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Changing Sequence Numbers in a Time Range

You can change all the sequence numbers assigned to rules in a time range.

SUMMARY STEPS

1. **configure terminal**
2. **resequence time-range** *name starting-sequence-number increment*
3. (Optional) **show time-range** *name*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**resequence time-range** *name starting-sequence-number increment* **Example:** `switch(config)# resequence time-range daily-workhours 100 10` `switch(config)#`	Assigns sequence numbers to the rules contained in the time range, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify.
Step 3	(Optional) **show time-range** *name* **Example:** `switch(config)# show time-range daily-workhours`	Displays the time-range configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the Time-Range Configuration

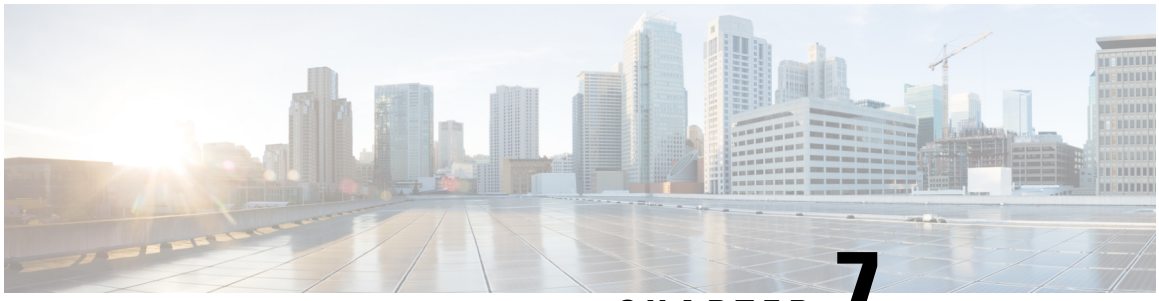
To display time-range configuration information, perform one of the following tasks.

Command	Purpose
show time-range	Displays the time-range configuration.
show running-config aclmgr	Displays ACL configuration, including all time ranges.

Additional References for IP ACLs

Related Documents

Related Topic	Document Title
TAP aggregation	Configuring TAP Aggregation and MPLS Stripping



CHAPTER 7

Configure QoS Classification

This chapter describes how to configure QoS Classification on Cisco NX-OS devices.

About Classification

Classification is the separation of packets into traffic classes. You configure the device to take a specific action on the specified classified traffic, such as policing or marking down, or other actions.

You can create class maps to represent each traffic class by matching packet characteristics with the classification criteria in the following table:

Table 25: Classification Criteria

Classification Criteria	Description
CoS	Class of service (CoS) field in the IEEE 802.1Q header.
IP precedence	Precedence value within the type of service (ToS) byte of the IP header.
Differentiated Services Code Point (DSCP)	DSCP value within the DiffServ field of the IP header.
ACL	IP, IPv6, or MAC ACL name.
Packet length	Size range of Layer 3 packet lengths. **Note** Match on packet-length is not supported on Cisco Nexus 9300 and 9800 Series switches.

Classification Criteria	Description
IP RTP	Identify applications using Real-time Transport Protocol (RTP) by UDP port number range.

You can specify multiple match criteria, you can choose to not match on a particular criterion, or you can determine the traffic class by matching any or all criteria.

Note However, if you match on an ACL, no other match criteria, except the packet length, can be specified in a match-all class. In a match-any class, you can match on ACLs and any other match criteria.

Traffic that fails to match any class in a QoS policy map is assigned to a default class of traffic called class-default. The class-default can be referenced in a QoS policy map to select this unmatched traffic.

You can reuse class maps when defining the QoS policies for different interfaces that process the same types of traffic.

Prerequisites for Classification

Classification has the following prerequisites:

- You must be familiar with using modular QoS CLI.
- You are logged on to the device.

Guidelines and Limitations for Classification

Configuration guidelines and limitations

- The **show** commands with the **internal** keyword are not supported.
- QoS policy will not be effective for fragmented packets. Fragmented packets will be forwarded to the default queue.
- When the **destination interface sup-eth0** CLI command is configured, the following system log message is displayed: `Enabling span destination to SUP will affect ingress QoS classification.`
- Matching the packets based on DSCP, CoS, or precedence in Cisco Nexus 9300-EX platform switches, the TCAM entries for both IPv4 (single-wide is one entry) and IPv6 (double-wide are two entries) are installed in the hardware. For example, if you match DSCP 4, three entries are installed in the hardware, one entry for IPv4 and two entries for IPv6.
- You can specify a maximum of 1024 match criteria in a class map.
- You can configure a maximum of 128 classes for use in a single policy map.
- When you match on an ACL, the only other match you can specify is the Layer 3 packet length in a match-all class.

- The **match-all** option in the **class-map type qos match-all** command is not supported. The match criteria of this command becomes the same as in the **class-map type qos match-any** command. The **class-map type qos match-all** command yields the same results as the **class-map type qos match-any** command.
- The **match-all** option is not supported in CoPP class-map and it always defaults to the **match-any** option.
- You can classify traffic on Layer 2 ports that are based on either the port policy or VLAN policy of the incoming packet but not both. If both are present, the device acts on the port policy and ignores the VLAN policy.
- When a Cisco Nexus Fabric Extender (FEX) is connected and in use, do not mark data traffic with a CoS value of 7. CoS 7 is reserved for control traffic transiting the Fabric Extender.
- Control traffic (control frames) from the switch to the FEX are marked with a CoS value of 7 and are limited to a jumbo MTU frame size of 2344 bytes.
- Jumbo ping (MTU of 2400 or greater) from a switch supervisor with a COS of 7, to a FEX host, fails because the control queue on a FEX supports an MTU limited to 2240.
- QoS classification policies are not supported under system QoS for Layer 2 switch ports. However, you can configure a QoS policy to classify the incoming traffic based on CoS/DSCP and map it to different queues. The QoS policy must be applied under all the interfaces that require the classification.
- A QoS policy with a MAC-based ACL as a match in the class map does not work for IPv6 traffic. For QoS, IPv6 traffic must be matched based on IPv6 addresses and not on MAC addresses.
- As a best practice, avoid having a voice VLAN configuration where an access VLAN is same as the voice VLAN.

  The following are alternative approaches:

  - If a separate dot1p tag (cos) value is not required for voice traffic, use the **switchport voice vlan untagged** command.

    ```
    switch(config)# interface ethernet 1/1
    switch(config-if)# switchport access vlan 20
    switch(config-if)# switchport voice vlan untagged
    ```

  - If a separate cos value is required for voice traffic, use the **switchport voice vlan dot1p** command.

    ```
    switch(config)# interface ethernet 1/1
    switch(config-if)# switchport access vlan 20
    switch(config-if)# switchport voice vlan dot1p
    ```

- MPLS packets with a NULL label on transit nodes, receive an MPLS classification that is based on its NULL label EXP.
- Ingress DROP_ACL_DROP is seen with Cisco Nexus 9272Q, 9236C, and 92160YC-X switches on an ASIC during congestion. However, these drops do not impact the performance of the switch.

Unsupported features

- PVLANs do not provide support for PVLAN QoS.
- QoS classification is not supported on the FEX interfaces ingressing the VXLAN traffic. This limitation is applicable to all Cisco Nexus 9000 series switches.

- For VXLAN, the following Cisco Nexus platforms do not support QoS policies for traffic from the network to access direction (decapsulation path) as ingress policy on the uplink interface:
  - Cisco Nexus 9300 and 9500 platform switches.
  - Cisco Nexus 9200 and 9300-EX platform switches; and Cisco Nexus 93180YC-EX and 93108TC-EX switches; and the Cisco Nexus 9732C-EX line card.
  - Cisco Nexus 9230QC, 9272Q, 9232C, 9236C, and 92300YC switches; and Cisco Nexus 9160YC-X switches.
- Cisco Nexus 9504 and Cisco Nexus 9508 switches with the specified line cards do not support QoS match acl with fragments:
  - Cisco Nexus 96136YC-R
  - Cisco Nexus 9636C-RX
  - Cisco Nexus 9636Q-R
  - Cisco Nexus 9636C-R
- A QoS policy that references an ACL that contains a match for ICMP type or code is not supported.

Supported features and platforms

- For VXLAN, the following Cisco Nexus platforms support QoS policies for traffic in the network to host direction (decapsulation path) as egress policy on both the port and VLAN:
  - Cisco Nexus 9300 and 9500 platform switches.
  - Cisco Nexus 9200 and 9300-EX platform switches; Cisco Nexus 93180YC-EX and 93108TC-EX switches; and the Cisco Nexus 9732C-EX line card.
  - The preceding is not supported for the following hardware: Cisco Nexus 9230QC, 9272Q, 9232C, 9236C, and 92300YC switches; and Cisco Nexus 9160YC-X switches.
- FEX QoS policy supports FEX host interfaces (HIF).
  - QoS TCAM carving is supported on ALE (Application Leaf Engine) enabled switches.
  - Only system level policies are supported.
  - Match on CoS is supported.
  - Match on QoS-group is supported.
- A QoS Policy that references an ACL that contains a match for TCP flags is only supported on the following Cisco Nexus 9000 series switches:
  - Cisco Nexus 9200 platform switches
  - Cisco Nexus 9300-EX platform switches
  - Cisco Nexus 9300-FX platform switches
  - Cisco Nexus 9300-GX platform switches

- Cisco Nexus 9500 platform switches with Cisco Nexus 97xx-EX and 97xx-FX line cards
- Beginning with Cisco NX-OS Release 10.2(1q)F, QoS Classification is supported on the N9K-C9332D-GX2B platform switches.

- From Cisco NX-OS Release 10.4(1)F, QoS classification is supported on Cisco Nexus C9348GCFX3 and Cisco C9348GC-FX3PH switches.

Note QoS classification is not supported for ports 41-48 on Cisco Nexus C9348GC-FX3PH switch.

- From Cisco NX-OS Release 10.4(2)F, QoS classification (ACL) is supported on Cisco Nexus C93108TC-FX3 switches.

Guidelines and limitations for Cisco Nexus 9800 Series switches

- Beginning with Cisco NX-OS Release 10.3(1)F, QoS classification (ACL) is supported on the Cisco Nexus 9808 platform switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, QoS classification (ACL) is supported on the Cisco Nexus 9804 platform switches.
- Cisco Nexus 9808/9804 platform switches have the following limitations for SUP QoS ACL support:
  - Egress type QoS policy is not supported.
  - Policer re-marking is not supported for exceed-action and violate-action.
  - The **match cos** and **set cos** commands are not supported.
  - Max burst values are supported for 16 configs. QoS and CoPP shares these burst configs. CoPP reserves 8, and QoS will have remaining 8.
  - ACL counters are not available for the policer. The **show system internal access-list interface eth <> input entries** command will not show counters if it has policer.
  - 2-rate 3-color (2R3C) policing support is provided only for confirm action transmit and exceed action transmit.
  - Match on packet-length is not supported.
- Beginning with Cisco NX-OS Release 10.4(1)F, System level ingress QoS policy (classification and remarking) is supported on Cisco Nexus 9808/9804 platform switches. However, policer is not supported at system level QoS.
- Beginning with Cisco NX-OS Release 10.5(3)F, QoS classification policy is supported for system QoS on Cisco Nexus 9800 Series switches with N9K-X9836DM-A and N9K-X98900CD-A line cards with following capabilities:
  - Classify the incoming traffic based on CoS or DSCP and map it to different queues.
  - Remark DSCP values.

- System qos policy applies to traffic coming in from all front panel ports (both Layer 2 and Layer 3)

Guidelines and limitations for Cisco Nexus N9364E-SG2 switches

- Beginning with Cisco NX-OS Release 10.5(3)F, Cisco Nexus N9364E-SG2-Q and N9364E-SG2-O switches support QoS classification policies for system QoS with these capabilities:
  - System QoS supports DSCP matching for IPv4/IPv6 and COS matching for non-IP traffic.
  - When a system QoS policy is configured, the default qos-map profile table entries are updated to match the policy settings.
  - System QoS supports setting the qos-group, DSCP for IP packets, and COS for non-IP traffic.
  - The System QoS feature does not support collecting, displaying, or analyzing traffic statistics managed by QoS policies.

Guidelines and limitations for Cisco Nexus N9324C-SE1U switches

- Beginning with Cisco NX-OS Release 10.5(3s), Cisco Nexus N9324C-SE1U switches support QoS classification policies for system QoS with these capabilities:
  - System QoS supports DSCP matching for IPv4/IPv6.
  - When a system QoS policy is configured, the default qos-map profile table entries are updated to match the policy settings.
  - System QoS supports setting the qos-group and DSCP for IP packets.
  - The System QoS feature does not support collecting, displaying, or analyzing traffic statistics managed by QoS policies.

Configuring Traffic Classes

Configuring ACL Classification

You can classify traffic by matching packets based on an existing access control list (ACL). Traffic is classified by the criteria defined in the ACL. The permit and deny ACL keywords are ignored in the matching; even though a match criteria in the access-list has a deny action, it is still used for matching for this class.

Note Use the **class-map class_acl** command to display the ACL class-map configuration.

SUMMARY STEPS

1. **configure terminal**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*

3. **match access-group name** *acl-name*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-name* **Example:** `switch(config)# class-map class_acl`	Creates or accesses the class map named class-name and enters class-map mode. The class map name can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters. (**match-any** is the default when no option is selected and multiple match statements are entered.)
Step 3	**match access-group name** *acl-name* **Example:** `switch(config-cmap-qos)# match access-group name my_acl`	Configures the traffic class by matching packets based on the *acl-name*. The **permit** and **deny** ACL keywords are ignored in the matching.

Examples: Configuring ACL Classification

To prevent packets from being matched by the QoS class-map, you must explicitly specify the packets you want to match with permit statements. The *implicit* default deny statement at the end of the ACL will filter out the remainder. Any *explicit* deny statements configured inside the access list of a QoS class map will be ignored in the matching and treated as an explicit permit statement as shown in the examples below.

The following examples, A1, B1, and C1, all produce the same QoS matching results:

- A1

```
ip access-list extended A1
  permit ip 10.1.0.0 0.0.255.255 any
  permit ip 172.16.128.0 0.0.1.255 any
  permit ip 192.168.17.0 0.0.0.255 any
```

- B1

```
ip access-list extended B1
  permit ip 10.1.0.0 0.0.255.255 any
  deny ip 172.16.128.0 0.0.1.255 any  /* deny is interpreted as a permit */
  permit ip 192.168.17.0 0.0.0.255 any
```

- C1

```
ip access-list extended C1
  deny ip 10.1.0.0 0.0.255.255 any   /* deny is interpreted as a permit */
 deny ip 172.16.128.0 0.0.1.255 any  /* deny is interpreted as a permit */
```

```
deny ip 192.168.17.0 0.0.0.255 any  /* deny is interpreted as a permit */
```

Adding an explicit DENY ALL at the end of a QoS matching ACL causes the QoS ACL to permit all traffic.

The following examples, D1 and E1, produce the same QoS matching results:

- D1

```
ip access-list extended D1
  permit ip 10.1.0.0 0.0.255.255 any
  permit ip 172.16.128.0 0.0.1.255 any
  permit ip 192.168.17.0 0.0.0.255 any
 deny ip 0.0.0.0 255.255.255.255 any  /* deny is interpreted as a permit */
```

Note The last line in the example effectively becomes a PERMIT ALL statement and results in the QoS ACL to permit all packets.

- E1

```
ip access-list extended E1
    permit ip 0.0.0.0 255.255.255.255 any
```

Configuring a DSCP Wildcard Mask

Use the DSCP wildcard mask feature to classify multiple DSCP values from a set of IP flows recognized by an ACL and the DSCP value. Classification of IP information and DSCP values occurs in a more granular way by using multiple parameters. With this granularity, you can treat these flows by policing them to protest the rest of the traffic, or assign them to a qos-group for further QoS operations.

Note Only Cisco Nexus 9300-EX/FX/FX2/FX3 platform switches support the DSCP wildcard mask feature.

SUMMARY STEPS

1. **configure terminal**
2. **ip access-list** *acl-name*
3. [*sequence-number*] { **permit** | **deny** } *protocol* { *source-ip-prefix* | *source-ip-mask* } { *destination-ip-prefix* | *destination-ip-mask* } [**dscp** *dscp-value* [*dscp-mask*]]
4. **exit**
5. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
6. **match access-list** *acl-name*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**ip access-list** *acl-name* **Example:** `switch(config)# ip access-list acl-01` `switch(config-acl)`	Enters the ACL configuration mode and creates an ACL with the entered name.
Step 3	[*sequence-number*] { **permit** | **deny** } *protocol* { *source-ip-prefix* | *source-ip-mask* } { *destination-ip-prefix* | *destination-ip-mask* } [**dscp** *dscp-value* [*dscp-mask*]] **Example:** `switch(config-acl)# 10 permit ip 10.1.1.1/24 20.1.1.2/24 dscp 33 30`	Creates an ACL entry that matches or filters traffic that is based on a DSCP wildcard bit mask. The *sequence-number* argument can be a whole number from 1 through 4294967295. **dscp**: Match packets with a specific DSCP value. *dscp-mask*: Configures the DSCP wildcard mask which matches on any bit in the DSCP value to filter traffic. Range is from 0 to 0x3F.
Step 4	**exit** **Example:** `switch(config-acl)# exit` `switch(config)#`	Exits ACL configuration mode and enters global configuration mode.
Step 5	**class-map** [**type qos**] [**match-any** | **match-all**] *class-name* **Example:** `switch(config)# class-map type qos match-any class_dscp_mask` `switch(config-cmap-qos)#`	Creates or accesses the class map that is named by the *class-name* variable and enters the class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters.
Step 6	**match access-list** *acl-name* **Example:** `switch(config-cmap-qos)# match access-list acl-01` `switch(config-cmap-qos)#`	Configures the traffic class by matching packets that are based on the IP access list.

Example

In the following example, an ACL looks at traffic that is sent from subnet 10.1.1.0 to subnet 20.1.1.0. The ACL also checks for traffic with DSCP 33, and any subsequent DSCP values from 33 through 63, with a mask value of 30. The ACL is set to a class map that is matching this ACL for further QoS operations.

```
switch# configure terminal
switch(config)# ip access-list acl-01
switch(config-acl)# 10 permit ip 10.1.1.1/24 20.1.1.2/24 dscp 33 dscp-mask 30
switch(config-acl)# exit
switch(config)# class-map type qos match-any class_dscp_mask
switch(config-cmap-qos)# match access-list acl-01
```

Configuring DSCP Classification

You can classify traffic based on the DSCP value in the DiffServ field of the IP header. The standard DSCP values are listed in the following table:

Table 26: Standard DSCP Values

Value	List of DSCP Values
af11	AF11 dscp (001010)—decimal value 10
af12	AF12 dscp (001100)—decimal value 12
af13	AF13 dscp (001110)—decimal value 14
af21	AF21 dscp (010010)—decimal value 18
af22	AF22 dscp (010100)—decimal value 20
af23	AF23 dscp (010110)—decimal value 22
af31	AF31 dscp (011010)—decimal value 26
af32	AF40 dscp (011100)—decimal value 28
af33	AF33 dscp (011110)—decimal value 30
af41	AF41 dscp (100010)—decimal value 34
af42	AF42 dscp (100100)—decimal value 36
af43	AF43 dscp (100110)—decimal value 38
cs1	CS1 (precedence 1) dscp (001000)—decimal value 8
cs2	CS2 (precedence 2) dscp (010000)—decimal value 16
cs3	CS3 (precedence 3) dscp (011000)—decimal value 24
cs4	CS4 (precedence 4) dscp (100000)—decimal value 32
cs5	CS5 (precedence 5) dscp (101000)—decimal value 40
cs6	CS6 (precedence 6) dscp (110000)—decimal value 48
cs7	CS7 (precedence 7) dscp (111000)—decimal value 56
default	Default dscp (000000)—decimal value 0

Value	List of DSCP Values
ef	EF dscp (101110)—decimal value 46

SUMMARY STEPS

1. **configure terminal**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
3. **match** [**not**] **dscp** *dscp-values*
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**class-map** [**type qos**] [**match-any**	**match-all**] *class-name* **Example:** `switch(config)# class-map class_dscp`	Creates or accesses the class map named class-name and enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters.
Step 3	**match** [**not**] **dscp** *dscp-values* **Example:** `switch(config-cmap-qos)# match dscp af21, af32`	Configures the traffic class by matching packets based on dscp-values. The standard DSCP values are shown in the following table. Use the **not** keyword to match on values that do not match the specified range.
Step 4	**exit** **Example:** `switch(config-cmap-qos)# exit` `switch(config)#`	Exits global class-map queuing mode and enters global configuration mode.
Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Example

This example shows how to display the DSCP class-map configuration:

```
switch# show class-map class_dscp
```

Configuring IP Precedence Classification

You can classify traffic based on the precedence value in the type of service (ToS) byte field of the IP header. The precedence values are listed in the following:

Table 27: Precedence Values

Value	List of Precedence Values
0-7	IP precedence value
critical	Critical precedence (5)
flash	Flash precedence (3)
flash-override	Flash override precedence (4)
immediate	Immediate precedence (2)
internet	Internetwork control precedence (6)
network	Network control precedence (7)
priority	Priority precedence (1)
routine	Routine precedence (0)

SUMMARY STEPS

1. **configure terminal**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
3. **match** [**not**] **precedence** *precedence-values*
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**class-map** [**type qos**] [**match-any**	**match-all**] *class-name* **Example:**	Creates or accesses the class map named class-name and then enters class-map mode. The class-map name can

	Command or Action	Purpose
	`switch(config)# class-map class_ip_precedence`	contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters.
Step 3	**match** [**not**] **precedence** *precedence-values* **Example:** `switch(config-cmap-qos)# match precedence 1-2, 5-7`	Configures the traffic class by matching packets based on *precedence-values*. Values are shown in the following table. Use the **not** keyword to match on values that do not match the specified range.
Step 4	**exit** **Example:** `switch(config-cmap-qos)# exit` `switch(config)#`	Exits global class-map queuing mode and enters global configuration mode.
Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Example

This example shows how to display the IP precedence class-map configuration:

```
switch# show class-map class_ip_precedence
```

Configuring Protocol Classification

For Layer 3 protocol traffic, you can use the ACL classification match.

Table 28: match Command Protocol Arguments

Argument	Description
arp	Address Resolution Protocol (ARP)
bridging	Bridging
cdp	Cisco Discovery Protocol (CDP)
dhcp	Dynamic Host Configuration (DHCP)
isis	Intermediate system to intermediate system (IS-IS)

SUMMARY STEPS

1. **configure terminal**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
3. **match** [**not**] **protocol** {**arp** | **bridging** | **cdp** | **dhcp** | **isis**}
4. **exit**

5. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-name* **Example:** `switch(config)# class-map class_protocol`	Creates or accesses the class map named class-name and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters.
Step 3	**match** [**not**] **protocol** {**arp** | **bridging** | **cdp** | **dhcp** | **isis**} **Example:** `switch(config-cmap-qos)# match protocol isis`	Configures the traffic class by matching packets based on the specified protocol. Use the **not** keyword to match on protocols that do not match the protocol specified.
Step 4	**exit** **Example:** `switch(config-cmap-qos)# exit` `switch(config)#`	Exits global class-map queuing mode and enters global configuration mode.
Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Example

This example shows how to display the protocol class-map configuration:

```
switch# show class-map class_protocol
```

Configuring Layer 3 Packet Length Classification

You can classify Layer 3 traffic based on various packet lengths.

Note This feature is designed for IP packets only.

SUMMARY STEPS

1. **configure terminal**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
3. **match** [**not**] **packet length** *packet-length-list*
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-name* **Example:** `switch(config)# class-map class_packet_length`	Creates or accesses the class map named class-name and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters.
Step 3	**match** [**not**] **packet length** *packet-length-list* **Example:** `switch(config-cmap-qos)# match packet length min 2000`	Configures the traffic class by matching packets based on various packet lengths (bytes). Values can range from 1 to 9198. Use the **not** keyword to match on values that do not match the specified range. **Note** This command is not supported on Cisco Nexus 9300 and 9800 Series switches.
Step 4	**exit** **Example:** `switch(config-cmap-qos)# exit` `switch(config)#`	Exits global class-map queuing mode and enters global configuration mode.
Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Example

This example shows how to display the packet length class-map configuration:

```
switch# show class-map class_packet_length
```

Configuring CoS Classification

You can classify traffic based on the class of service (CoS) in the IEEE 802.1Q header. This 3-bit field is defined in IEEE 802.1p to support QoS traffic classes. CoS is encoded in the high order 3 bits of the VLAN ID Tag field and is referred to as user_priority.

SUMMARY STEPS

1. **configure terminal**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
3. **match** [**not**] **cos** *cos-list*
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**class-map** [**type qos**] [**match-any**	**match-all**] *class-name* **Example:** `switch(config)# class-map class_cos`	Creates or accesses the class map named class-name and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters.
Step 3	**match** [**not**] **cos** *cos-list* **Example:** `switch(config-cmap-qos)# match cos 4,5-6`	Configures the traffic class by matching packets based on the list of CoS values. Values can range from 0 to 7. Use the **not** keyword to match on values that do not match the specified range. **Note** When a Cisco Nexus Fabric Extender (FEX) is connected and in use, data traffic should not be marked with a CoS value of 7. CoS 7 is reserved for control traffic transiting the Fabric Extender.
Step 4	**exit** **Example:** `switch(config-cmap-qos)# exit` `switch(config)#`	Exits global class-map queuing mode and enters global configuration mode.
Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Example

This example shows how to display the CoS class-map configuration:

```
switch# show class-map class_cos
```

Configuring CoS Classification for FEX

Note The CoS Classification for FEX feature is not supported on the Cisco Nexus 9508 switch (NX-OS 7.0(3)F3(3)).

You can classify traffic based on the class of service (CoS) for a FEX.

Before you begin

Before configuring the FEX, enable **feature-set fex**.

SUMMARY STEPS

1. **configure terminal**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
3. **match** [**not**] **cos** *cos-list*
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**class-map** [**type qos**] [**match-any**	**match-all**] *class-name* **Example:** `switch(config)# class-map class_cos`	Creates or accesses the class map named class-name and then enters class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters.
Step 3	**match** [**not**] **cos** *cos-list* **Example:** `switch(config-cmap-qos)# match cos 4,5-6`	Configures the traffic class by matching packets based on the list of CoS values. Values can range from 0 to 7. Use the **not** keyword to match on values that do not match the specified range. **Note** When a Cisco Nexus Fabric Extender (FEX) is connected and in use, data traffic should not be marked with a CoS

	Command or Action	Purpose
		value of 7. CoS 7 is reserved for control traffic transiting the Fabric Extender.
Step 4	**exit** **Example:** `switch(config-cmap-qos)# exit` `switch(config)#`	Exits global class-map queuing mode and enters global configuration mode.
Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Example

This example shows how to configure the CoS class-map configuration:

```
switch# conf t
switch(config)# class-map type qos match-all cos6
switch(config-cmap-qos)# match cos 6
switch(config)# class-map type qos match-all cos1
switch(config-cmap-qos)# match cos 1
switch(config)# class-map type qos match-all cos2
switch(config-cmap-qos)# match cos 2
switch(config)# class-map type qos match-all cos3
switch(config-cmap-qos)# match cos 3
switch(config)# class-map type qos match-all cos0
switch(config-cmap-qos)# match cos 0
```

Configuring IP RTP Classification

The IP Real-Time Transport Protocol (RTP) is a transport protocol for real-time applications that transmit data such as audio or video (RFC 3550). Although RTP does not use a common TCP or UDP port, you typically configure RTP to use ports 16384 to 32767. UDP communications uses an even-numbered port and the next higher odd-numbered port is used for RTP Control Protocol (RTCP) communications.

Cisco Nexus 9000 Series switches support the transport of RDMA over Converged Ethernet (RoCE) v1 and v2 protocols. RoCE uses a UDP port.

When defining a match statement in a **type qos class-map**, to match with upper layer protocols and port ranges (UDP/TCP/RTP, among others), the system cannot differentiate, for example, between UDP traffic and RTP traffic in the same port range. The system classifies both traffic types the same. For better results, you must engineer the QoS configurations to match the traffic types present in the environment.

SUMMARY STEPS

1. **configure terminal**
2. **class-map** [**type qos**] [**match-any** | **match-all**] *class-name*
3. **match** [**not**] **ip rtp** *udp-port-value*

4. **match** [**not**] **ip roce** *udp-port-value*
5. **exit**
6. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**class-map** [**type qos**] [**match-any** | **match-all**] *class-name* **Example:** `switch(config)# class-map class_rtp`	Creates or accesses a class map and then enters the class-map mode. The class-map name can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters.
Step 3	**match** [**not**] **ip rtp** *udp-port-value* **Example:** `switch(config-cmap-qos)# match ip rtp 2000-2100, 4000-4100`	Configures the traffic class by matching packets that are based on a range of lower and upper UDP port numbers, targeting applications using RTP. Values can range from 2000 to 65535. Use the **not** keyword to match on values that do not match the specified range.
Step 4	**match** [**not**] **ip roce** *udp-port-value* **Example:** `switch(config-cmap-qos)# match ip roce 3000-3100, 6000-6100`	Configures the traffic class by matching packets that are based on a range of lower and upper UDP port numbers, targeting applications using RoCE. Values can range from 2000 to 65535. Use the **not** keyword to match on values that do not match the specified range. **Note** If ip roce and ip rtp are configured to match with the same port number, only ip rtp is displayed when you use the **show policy-map interface** *interface-type***type qos** command.. When you use the help string for both the RTP and RoCE, the recommended range is displayed but you are allowed to specify the value outside the recommended range as well (based on your requirement).
Step 5	**exit** **Example:** `switch(config-cmap-qos)# exit` `switch(config)#`	Exits global class-map queuing mode and enters global configuration mode.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves the running configuration to the startup configuration.

Example

This example shows how to display the RTP class-map configuration:

```
switch# show class-map class_rtp
```

Commands for classification configuration verification

Use the following commands to verify the classification configuration:

Command	Purpose
show class-map	Displays all class maps.
show ip access-lists *name*	Displays the IPv4 ACL configuration.
show ipv6 access-lists *name*	Displays the IPv6 ACL configuration.

Configuration Examples for Classification

The following example shows how to configure classification for two classes of traffic:

```
class-map class_dscp
match dscp af21, af32
exit
class-map class_cos
match cos 4, 5-6
exit
```

The following example shows how to configure system QoS:

```
class-map type qos match-all match-dscp-cs1
  match dscp 8
class-map type qos match-all match-dscp-cs2
  match dscp 16
class-map type qos match-all match-dscp-cs3
  match dscp 24
class-map type qos match-all match-dscp-cs4
  match dscp 32
class-map type qos match-all match-dscp-cs5
  match dscp 40
class-map type qos match-all match-dscp-cs6
  match dscp 48
class-map type qos match-all match-dscp-cs7
  match dscp 56

policy-map type qos system-level-policy1
  class match-dscp-cs1
    set qos-group 1
  class match-dscp-cs2
    set qos-group 2
  class match-dscp-cs3
    set qos-group 3
  class match-dscp-cs4
    set qos-group 4
  class match-dscp-cs5
```

```
    set qos-group 5
  class match-dscp-cs6
    set qos-group 6
  class match-dscp-cs7
    set qos-group 7

switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# system qos
switch(config-sys-qos)# service-policy type qos input system-level-policy1
switch(config-sys-qos)# end
switch#
```