

Configuring MAC ACLs

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

This chapter contains the following sections:

- MAC ACLs, on page 1
- Guidelines for configuring MAC ACLs, on page 2
- Default settings for MAC ACLs, on page 3
- Configuring MAC ACLs, on page 3
- MAC ACL configuration verification commands, on page 12
- MAC ACL statistics monitoring and clearing, on page 13
- Examples of MAC ACL configuration, on page 13
- Additional references for MAC ACLs, on page 14

MAC ACLs

A MAC access control list (ACL)

- filters network traffic using information in the Layer 2 header.
- works like an IP access control list (ACL) and supports virtualization
- applies filtering based on MAC addresses instead of IP addresses.

MAC packet classification

Media Access Control packet characteristics include the following:

- is identified and filtered using Media Access Control access control lists (MAC ACLs) at Layer 2 interfaces,
- Packets can be classified to affect how traffic, including both Internet Protocol (IP) and non-Internet Protocol (non-IP), is handled.
- This enables differentiated control over interface access for security and policy enforcement.

Media Access Control (MAC) packet classification determines whether a Media Access Control access control list (MAC ACL) on a Layer 2 interface applies to all incoming traffic, including Internet Protocol (IP) traffic, or applies only to non-Internet Protocol (non-IP) traffic.

Table 1: MAC packet classification

MAC Packet Classification State	Effect on the interface
Enabled	• A Media Access Control access control list (MAC ACL) that is on the interface applies to all traffic entering the interface, including Internet Protocol (IP) traffic. • You can apply an IP port ACL to the interface.
Disabled	• A MAC ACL that is on the interface applies only to non-IP traffic entering the interface. • You can apply an Internet Protocol (IP) port access control list (ACL) to the interface.

Guidelines for configuring MAC ACLs

Follow these guidelines and review the limitations when you configure MAC ACLs.

MAC ACL best practices for the supported platforms

- MAC packet classification is not supported when MAC ACLs are used as match criteria for QoS policies on Nexus 9300 Series switches with forty-gigabit (40-gigabit) [43.6 billion bits per second] uplink ports.
- Beginning NX-OS Release 9.3(2), you can configure a user-defined MAC address limit in the range of 16 to 256.
- Nexus 93600CD-GX switches do not support breakout on ports 1 through 24.
- Beginning with NX-OS Release 10.4(1)F, a new ACE keyword (all) is provided for applying the MAC ACL rule priority over SUP rule on Nexus 9300- FX/FX2/FX3/GX/GX2 and Cisco Nexus 9500 with X97160YC-EX, 9700- FX/GX line cards.

MAC ACL general best practices

- MAC ACLs control only ingress traffic.
- If you apply too many ACL entries, your configuration might be rejected.
- MAC packet classification is not supported when a MAC ACL is applied as part of a VACL.
- When you set a user-defined MAC limit using the **mac address-table limit <16-256> user-defined** command, the FHRP group limit is automatically adjusted so that the sum of user-defined MAC limits and FHRP limits equals 490. For example, if you set the user-defined MAC limit to 100, the FHRP limit is reduced to 390.
- A MAC access list applied to an interface will not block Bridge Protocol Data Unit (BPDU) traffic, such as Spanning Tree Protocol BPDUs

Default settings for MAC ACLs

Use this table to find the default settings for MAC ACL parameters.

Table 2: Default MAC ACLs parameters

Parameters	Default
MAC ACLs	By default, no MAC ACLs are configured.
ACL rules	Implicit rules are applied to all access control lists.

Configuring MAC ACLs

Explains configuring MAC ACLs.

Create a MAC ACL

You define and configure a MAC access control list to filter traffic based on MAC addresses.

You can use this task to create and configure a MAC access control list (ACL) that includes permit or deny rules in the command-line interface (CLI).

Before you begin

Access the global configuration mode on the NX-OS device.

Procedure

Step 1 Enter global configuration mode by using the **configure terminal** command .

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Create the MAC ACL and enter ACL configuration mode by using the **mac access-list** *name* command.

Example:

```
switch(config)#mac access-list acl-mac-01
switch(config-mac-acl)#
```

Notice
The names **dynamic** , expanded, and summary are reserved for system-defined access lists. Do not use these names for user-defined ACLs, as this can cause conflicts when displaying or verifying your configuration.

Step 3 Create a rule in the MAC access control list by using the {**permit** | **deny**} *source destination-protocol* command.

Example:

```
switch(config-mac-acl)#100 permit mac 00c0.4f00.0000 0000.00ff.ffff any 0x0806
```

The **permit** and **deny** commands support many ways of identifying traffic.

Step 4 (Optional) Specify that the device collects global statistics for packets matching the ACL rules by using the **statistics per-entry** command.

Example:

```
switch(config-mac-acl)#statistics per-entry
```

Step 5 (Optional) Display the MAC ACL configuration by using the **show mac access-lists** *name* command.

Example:

```
switch#show mac access-lists acl-mac-01
```

Step 6 (Optional) Save the configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

You create and configure the MAC access control list with the specified rules. Apply the list to interfaces or verify its configuration.

Configure a UDF-Based MAC ACL

You can configure UDF-based MAC access lists (ACLs) for the Cisco Nexus 9300 Series switches. This feature enables the device to match on user-defined fields (UDFs) and apply MAC ACLs to the matching packets.

Starting with Cisco NX-OS Release 9.3(3), you can configure UDF-based access lists (ACLs) on Cisco Nexus 9364C-GX, Cisco Nexus 9316D-GX, and Cisco Nexus 93600CD-GX switches.

Procedure

Step 1 Enter global configuration mode by using the command **configure terminal** and define the UDF using the command **udf** *udf-name offset-base offset length* .

Example:

```
switch#configure terminal
switch(config)#udf pktoff10 packet-start 10 2
```

The following defines the UDF.

- *udf-name* —Specifies the name of the UDF. You can enter up to sixteen alphanumeric characters for the name.
- *offset-base* —Specifies the UDF offset base as follows: {**packet-start**} .
- *offset* —Specifies the number of bytes offset from the offset base.
- *length* —Specifies the number of bytes from the offset. Only one or two bytes are supported. To match additional bytes, you must define multiple UDFs.

You can define multiple UDFs; however, Cisco recommends defining only the required UDFs.

Step 2 Attach the UDFs to the ing-ifacl TCAM region for IPv4 or IPv6 port ACLs. To do this, use the } **hardware access-list tcam region ing-ifacl qualify** {**udf** *udf-name*} command.

Example:

Note
This configuration takes effect only after you enter copy running-config startup-config and reload the device, or reload all line card modules where supported. Multiple TCAM changes can be completed before a single reload.

Example:

```
switch#hardware access-list tcam region ing-ifacl qualify udf pktoff10
```

Up to eighteen UDFs are supported.

Note
When the UDF qualifier is added, the TCAM region goes from single wide to double wide. Ensure there is enough free space 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
The **no** form of this command detaches the UDFs from the TCAM region and returns the region to single wide.

Step 3 Required: Ensure the configuration is saved after reboots and restarts, copy the running configuration to the startup configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

Step 4 Required: Reload the device by using the **reload** command.

Example:

```
switch#reload
```

Note
The UDF configuration takes effect after you enter the **copy running-config startup-config** and **reload** commands, and then reload the device.

Step 5 Create a MAC access control list (ACL) and enter MAC ACL configuration mode by using the **mac access-list** *udf-acl* command.

Example:

```
switch(config)#mac access-list udfacl
switch(config-acl)#
```

Step 6 Configure the MAC ACL to match on UDFs using the **permit mac** *source destination* **udf** *udf-name value mask* command.

Example:

```
switch(config-mac-acl)#permit mac any any udf pktoff10 0x1234 0xffff
```

Configure the MAC ACL to match on UDFs together with the current access control entries (ACEs) for the outer packet fields. The *value* and *mask* arguments can range from 0x0 to 0xffff.

A single ACL can have ACEs with or without UDFs. Each ACE can match a different UDF field, or all ACEs can match the same UDF or UDFs.

Step 7 Enter interface configuration mode for a Layer 2 port-channel interface using the **interface port-channel** *channel-number* command.

Example:

```
switch(config)#interface port-channel 5
switch(config-if)#
```

Step 8 Apply the UDF-based MAC ACL to the interface by using the **mac port access-group** *udf-access-list* command.

Example:

```
switch(config-if)#mac port access-group udfacl
```

Step 9 (Optional) Copy the running configuration to the startup configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

You can configure UDF-based MAC access lists (ACLs) and apply them to matching packets.

Modify a MAC ACL

You can use this task to modify a MAC ACL on your device

Modify a MAC ACL by completing these steps:

Before you begin

Use the **show mac access-lists** command with the **summary** keyword. Find the interfaces that have a MAC ACL configured.

Procedure

Step 1 Enter the global configuration mode by using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Enter MAC ACL configuration mode for the target ACL by using the **mac access-list** *name* command.

Example:

```
switch(config)#mac access-list acl-mac-01
switch(config-mac-acl)#
```

Step 3 (Optional) Create a rule in the MAC ACL by using the [*sequence-number*] {**permit** | **deny**} *source destination-protocol* command.

Example:

```
switch(config-mac-acl)#100 permit mac 00c0.4f00.0000 0000.00ff.ffff any 0x0806
```

You can specify a position for the rule in the ACL by using a sequence number. If you do not provide a sequence number, the rule is added to the end of the list.

Use the **permit** and **deny** commands to identify traffic.

Step 4 (Optional) Remove the specific rule by sequence number or criteria by using the **no** {*sequence-number* | {**permit** | **deny**} *source destination-protocol*} command.

Example:

```
switch(config-mac-acl)#no 80
```

The **permit** and **deny** commands support many ways of identifying traffic.

Step 5 (Optional) Use the [**no**] **statistics per-entry** command to specify the device that maintains global statistics for packets matching the rules in the ACL

Example:

```
switch(config-mac-acl)#statistics per-entry
```

Use the **no statistics per-entry** command to stop the device from maintaining global statistics for packets that match the rules in the ACL.

Step 6 (Optional) You can display the updated MAC ACL configuration by using the **show mac access-lists** *name* command.

Example:

```
switch#show mac access-lists acl-mac-01
```

Step 7 (Optional) (Optional) Save the configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

You modified the specified MAC ACL rule on your device.

What to do next

Verify that you modified the ACL and confirm that your device functions as expected.

Change sequence numbers in a MAC ACL

You can change all the sequence numbers assigned to rules in a MAC ACL. Resequencing is useful when you need to insert rules into an ACL and there are not enough available sequence numbers.

Procedure

Step 1 Enter global configuration mode by using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Assign sequence numbers to the rules contained in the ACL by using the **resequence mac access-list** *name starting-sequence-number increment* command.

Example:

```
switch(config)#resequence mac access-list acl-mac-01 100 10
```

The first rule receives the starting sequence number that you specify. Each subsequent rule receives a number greater than the previous rule. The difference in numbers is determined by the increment number that you specify.

Step 3 (Optional) Display the MAC ACL configuration by using the **show mac access-lists** *name* command.

Example:

```
switch#show mac access-lists acl-mac-01
```

Step 4 (Optional) Copy the running configuration to the startup configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

Remove a MAC ACL

Remove an Access Control List (ACL) from the running configuration to stop specific traffic filtering.

Perform this task when you no longer need an ACL or plan to replace it on the device.

Before you begin

- You need administrative privileges to modify the device configuration.
- Identify the name of the ACL you want to remove.

Procedure

Step 1 Enter global configuration mode by using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Remove the MAC ACL by name from the running configuration using the **no mac access-list** *name* command.

Example:

```
switch(config)#no mac access-list acl-mac-01
switch(config)#
```

Step 3 (Optional) Display the MAC ACL configuration by using the **show mac access-lists** *name* **summary** command.

Example:

```
switch#show mac access-lists acl-mac-01 summary
```

If the ACL remains applied to an interface, the command lists the interfaces.

Step 4 (Optional) Copy the running configuration to the startup configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

The switch removes the specified ACL from the running configuration and from the startup configuration if you save the changes.

What to do next

- Verify that network traffic behaves as expected after removing the ACL.
- Remove the ACL from any interface where it is still applied.

Apply a MAC ACL as a port ACL

You can apply a MAC ACL as a port ACL to these interface types:

- Layer 2 Ethernet interfaces
- Layer 2 port-channel interfaces

Before you begin

Confirm that the required ACL is present and configured to filter traffic appropriately for your use case.

Procedure

Step 1 Enter global configuration mode by using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Enter one of these commands:

- **interface ethernet** *slot* / *port*
- **interface port-channel** *channel-number*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Example:

```
switch(config)#interface port-channel 5
switch(config-if)#
```

- Enter interface configuration mode for a Layer 2 or Layer 3 interface.
- Enter interface configuration mode for a Layer 2 or Layer 3 port-channel interface.

Step 3 Applies a MAC ACL to the interface by using the **mac port access-group** *access-list* command.

Example:

```
switch(config-if)#mac port access-group acl-01
```

Step 4 (Optional) Display the ACL configuration by using the **show running-config aclmgr** command.

Example:

```
switch#show running-config aclmgr
```

Step 5 (Optional) Copies the running configuration to the startup configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

Apply a MAC ACL as a VACL

You can apply a MAC ACL as a VACL.

Enable or disable MAC packet classification

You can enable or disable MAC packet classification on a Layer 2 interface.

Before you begin

The interface must be configured as a Layer 2 interface.

Note If the interface is configured with the **ip port access-group** command or the **ipv6 port traffic-filter** command, you cannot enable MAC packet classification until you remove the **ip port access-group** and **ipv6 port traffic-filter** commands from the interface configuration.

Procedure

Step 1 Enters the global configuration mode by using the **configure terminal** command.

Example:

```
switch#configure terminal
switch(config)#
```

Step 2 Enter one of the following commands:

- **interface ethernet** *slot* / *port*
- **interface port-channel** *channel-number*

Example:

```
switch(config)#interface ethernet 2/1
switch(config-if)#
```

Example:

```
switch(config)#interface port-channel 5
switch(config-if)#
```

- Enter interface configuration mode for an Ethernet interface, or
- Enter interface configuration mode for a port-channel interface.

Step 3 Enable MAC packet classification on the interface by using the command [**no**] **mac packet-classify** .

Example:

```
switch(config-if)#mac packet-classify
```

Use the **no** form of the command to disable MAC packet classification on the interface.

Step 4 (Optional) Enter one of the following commands:

- **show running-config interface ethernet** *slot* / *port*
- **show running-config interface port-channel** *channel-number*

Example:

```
switch#show running-config interface ethernet 2/1
```

Example:

```
switch#show running-config interface port-channel 5
```

- Display the running configuration of the Ethernet interface, or
- Display the running configuration of the port-channel interface.

Step 5 (Optional) Copy the running configuration to the startup configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

Prioritize a MAC ACL rule over a SUP rule

In Cisco NX-OS Release 10.4(1)F, a new ACE keyword (all) is supported for MAC ACLs. This keyword increases the priority of the ACL rule to 0 (the highest), giving it precedence over any other SUP ACL rule that matches the same criteria.

Procedure

Step 1 Enter global configuration mode by using the **configure terminal** command.

Example:

```
switch#configure terminal
```

Step 2 Create a MAC ACL and enter ACL configuration mode by using the **mac access-list** *name* command.

Example:

```
switch(config)#mac access-list acl-mac-01
 switch(config-acl)#
```

The *name* argument can be up to 64 characters.

Step 3 Create a rule in a MAC ACL using the {**permit** | **deny**} *source destination-protocol* **all** command.

Example:

```
switch(config-mac-acl)#100 permit 00c0.4f00.0000 0000.00ff.ffff any all
```

The **all** keyword prioritizes the MAC 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)#
```

Enter configuration mode for the interface type that you specified.

Step 5 Apply a MAC ACL to the interface or port channel by using the **mac port access-group** *access-list* **in** command.

Example:

```
switch(config-if)#mac port access-group acl-mac-01 in
```

Port ACLs support only inbound filtering. You can apply one port ACL to each interface.

Step 6 (Optional) Display the ACL configuration by using the **show running-config aclmgr** command.

Example:

```
switch#show running-config aclmgr
```

Step 7 (Optional) Copy the running configuration to the startup configuration by using the **copy running-config startup-config** command.

Example:

```
switch#copy running-config startup-config
```

MAC ACL configuration verification commands

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

Table 3: Commands to verify MAC ACL configuration

Command	Purpose
show mac access-lists	Displays the MAC ACL configuration.

Command	Purpose
show running-config aclmgr [**all**]	Displays the ACL configuration, including MAC ACLs and the interfaces to which MAC 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 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.

MAC ACL statistics monitoring and clearing

Use these commands to monitor or clear MAC ACL statistics on the device.

Table 4: Commands for monitoring and clearing MAC ACL statistics

Command	Purpose
show mac access-lists	This command displays the MAC ACL configuration. If the MAC ACL includes the **statistics per-entry** command, the output of **show mac access-lists** command also includes the number of packets that have matched each rule.
clear mac access-list counters	Clears statistics for MAC ACLs.

Examples of MAC ACL configuration

MAC ACLs (Media Access Control Lists) filter network traffic based on source and destination MAC addresses and are applied to Layer 2 interfaces.

```
mac access-list acl-mac-01
permit 00c0.4f00.0000 0000.00ff.ffff any 0x0806
interface ethernet 2/1
mac port access-group acl-mac-01
```

This configuration:

- Creates a MAC ACL named acl-mac-01.
- Permits Address Resolution Protocol (ARP) packets from the specified MAC address range.

- Applies the ACL to Ethernet interface 2/1.

Additional references for MAC ACLs

Use this table for related documents for Media Access Control (MAC) access control lists (ACLs).

Table 5: Related documents

Related Topic	Document Title
TAP aggregation	Configuring TAP Aggregation and MPLS Stripping