Configuring 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.

This chapter includes the following sections:

Information 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 by default; however, you can configure Layer 2 interfaces to apply MAC ACLs to all traffic.
Security-group ACLs (SGACLs)	The device applies SGACLs to traffic tagged by Cisco TrustSec.

IP and MAC ACLs have the following types of applications:

Port ACL	Filters Layer 2 traffic
Router ACL	Filters Layer 3 traffic
VLAN ACL	Filters VLAN traffic

This table summarizes the applications for security ACLs.

Table 1: 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 • IPv6 ACLs • MAC ACLs
Router ACL	• VLAN interfaces **Note** You must enable VLAN interfaces globally before you can configure a VLAN interface. For more information, see the Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 4.2. • Physical Layer 3 interfaces	• IPv4 ACLs • IPv6 ACLs **Note** MAC ACLs are supported on Layer 3 interfaces only if you enable MAC packet classification.

Application	Supported Interfaces	Types of ACLs Supported
	• Layer 3 Ethernet subinterfaces • Layer 3 Ethernet port-channel interfaces • Layer 3 Ethernet port-channel subinterfaces • Tunnels • Management interfaces	
VLAN ACL	• VLANs	• IPv4 ACLs • IPv6 ACLs • MAC ACLs

Related Topics

- Information About MAC ACLs
- Information About VLAN ACLs
- Information About MAC ACLs
- SGACLs and SGTs

Order of ACL Application

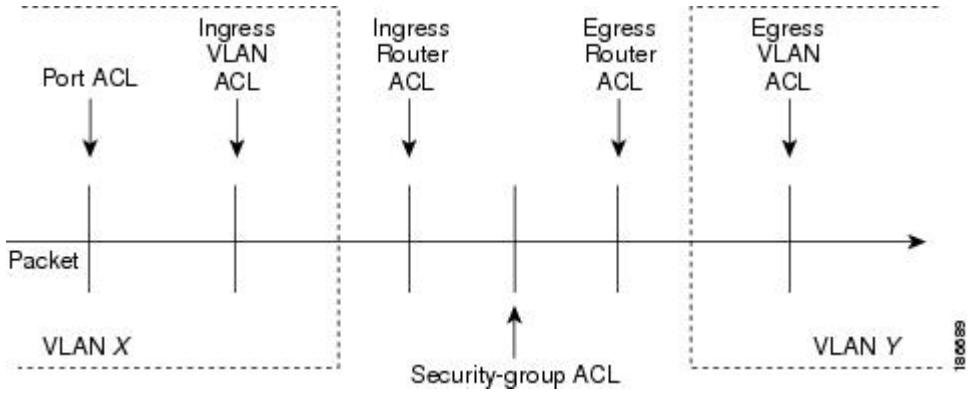
When the device processes a packet, it determines the forwarding path of the packet. The path determines which ACLs that the device applies to the traffic. The device applies the ACLs in the following order:

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

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

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

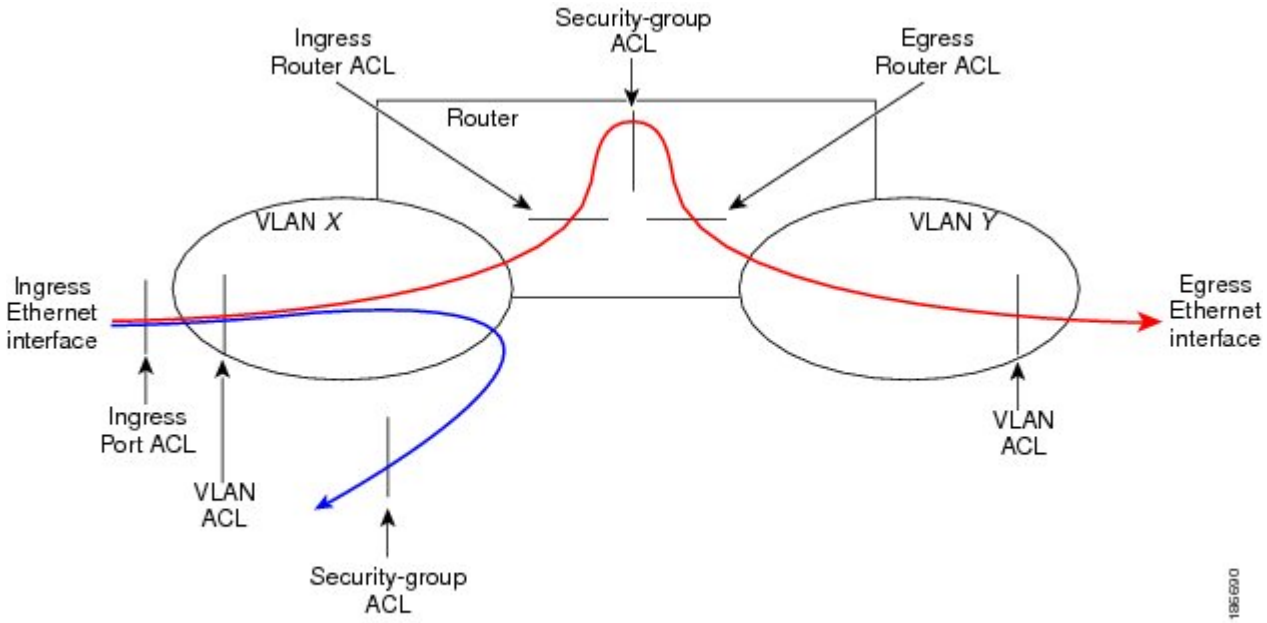
Figure 1: Order of ACL Application



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.

Figure 2: ACLs and Packet Flow



Related Topics

- SGACLs and SGTs

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. For information about every option, see the applicable **permit** and **deny** commands in the Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.2.

Protocols

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. For example, you can use 115 to specify Layer 2 Tunneling Protocol (L2TP) traffic.

For a list of the protocols that each type of ACL supports by name, see the applicable **permit** and **deny** commands in the Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.2.

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, IPv6, or MAC ACLs. For information about specifying the source and destination, see the applicable **permit** and **deny** commands in the Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.2.

Implicit Rules

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 rules:

```
permit icmp any any nd-na
permit icmp any any nd-ns
permit icmp any any router-advertisement
permit icmp any any router-solicitation
deny ipv6 any any
```

Unless you configure an IPv6 ACL with a rule that denies ICMPv6 neighbor discovery messages, the first four rules ensure that the device permits neighbor discovery advertisement and solicitation messages. The fifth rule ensures that the device denies unmatched IPv6 traffic.

Note If you explicitly configure an IPv6 ACL with a **deny ipv6 any any** rule, the implicit permit rules can never permit traffic. If you explicitly configure a **deny ipv6 any any** rule but want to permit ICMPv6 neighbor discovery messages, explicitly configure a rule for all five implicit IPv6 ACL rules.

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
  - Authentication Header Protocol
  - Enhanced Interior Gateway Routing Protocol (EIGRP)
  - Encapsulating Security Payload
  - General Routing Encapsulation (GRE)
  - KA9Q NOS-compatible IP-over-IP tunneling
  - Open Shortest Path First (OSPF)
  - Payload Compression Protocol
  - Protocol-independent multicast (PIM)
  - 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
  - Authentication Header Protocol
  - Encapsulating Security Payload
  - Payload Compression Protocol
  - Stream Control Transmission Protocol (SCTP)
  - SCTP, TCP, and UDP ports
  - ICMP types and codes
  - IGMP types
  - Flow label
  - 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
  - VLAN ID
  - Class of Service (CoS)

For information about all filtering options available in rules, see the applicable **permit** and **deny** commands in the Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.2.

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. The device stores operator-operand couples in registers called logical operator units (LOUs). Cisco Nexus 7000-series devices support 104 LOUs.

The LOU usage for each type of operator is as follows:

eq	Is never stored in an LOU
gt	Uses 1/2 LOU
lt	Uses 1/2 LOU
neq	Uses 1/2 LOU
range	Uses 1 LOU

The following guidelines determine when the devices store operator-operand couples in LOUs:

- If the operator or operand differs from other operator-operand couples that are used in other rules, the couple is stored in an LOU.

  For example, the operator-operand couples "gt 10" and "gt 11" would be stored separately in half an LOU each. The couples "gt 10" and "lt 10" would also be stored separately.

- Whether the operator-operand couple is applied to a source port or a destination port in the rule affects LOU usage. Identical couples are stored separately when one of the identical couples is applied to a source port and the other couple is applied to a destination port.

  For example, if a rule applies the operator-operand couple "gt 10" to a source port and another rule applies a "gt 10" couple to a destination port, both couples would also be stored in half an LOU, resulting in the use of one whole LOU. Any additional rules using a "gt 10" couple would not result in further LOU usage.

Logging

You can enable the device to create an informational log message for packets that match a rule. The log message contains the following information about the packet:

- Protocol
- Status of whether the packet is a TCP, UDP, or ICMP packet, or if the packet is only a numbered packet.
- Source and destination address
- Source and destination port numbers, if applicable

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, 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.

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, page 25
- Implicit Rules, page 5

Atomic ACL Updates

By default, when a supervisor module of a Cisco Nexus 7000 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.

Note The **hardware access-list update** command is available in the default VDC only but applies to all VDCs.

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
```

VTY Support

Cisco NX-OS does not support applying an ACL directly to a VTY line; however, you can use control plane policing (CoPP) to filter VTY traffic. To do so, you must define two ACLs for use with filtering VTY traffic: one ACL that permits traffic that you want to allow and another ACL that permits traffic that you want to drop. Then you can configure CoPP to transmit the packets that are permitted by the ACL that matches desirable traffic and to drop the packets that are permitted by the ACL that matches undesirable traffic.

In the following example, the ACL copp-system-acl-allow explicitly allows Telnet, SSH, SNMP, NTP, RADIUS, and TACACS+ traffic that is inbound from the 10.30.30.0/24 network and allows any traffic outbound from the device to the 10.30.30.0/24 network. The copp-system-acl-deny explicitly allows all traffic. The policing policies are configured to transmit the traffic permitted by the copp-system-acl-allow ACL and to drop the traffic permitted by the copp-system-acl-deny ACL.

```
ip access-list copp-system-acl-allow
  10 remark ### ALLOW TELNET from 10.30.30.0/24
  20 permit tcp 10.30.30.0/24 any eq telnet
  30 permit tcp 10.30.30.0/24 any eq 107
  40 remark ### ALLOW SSH from 10.30.30.0/24
  50 permit tcp 10.30.30.0/24 any eq 22
  60 remark ### ALLOW SNMP from 10.30.30.0/24
  70 permit udp 10.30.30.0/24 any eq snmp
  80 remark ### ALLOW TACACS from 10.30.30.0/24
  90 permit tcp 10.30.30.0/24 any eq tacacs
  100 remark ### ALLOW RADIUS from 10.30.30.0/24
  110 permit udp 10.30.30.0/24 any eq 1812
  120 permit udp 10.30.30.0/24 any eq 1813
  130 permit udp 10.30.30.0/24 any eq 1645
  140 permit udp 10.30.30.0/24 any eq 1646
  150 permit udp 10.30.30.0/24 eq 1812 any
  160 permit udp 10.30.30.0/24 eq 1813 any
  170 permit udp 10.30.30.0/24 eq 1645 any
  180 permit udp 10.30.30.0/24 eq 1646 any
  190 remark ### ALLOW NTP from 10.30.30.0/24
  200 permit udp 10.30.30.0/24 any eq ntp
  210 remark ### ALLOW ALL OUTBOUND traffic TO 10.30.30.0/24
  220 permit ip any 10.30.30.0/24
  statistics  # keep statistics on matches
ip access-list copp-system-acl-deny
  10 remark ### this is a catch-all to match any other traffic
  20 permit ip any any
  statistics  # keep statistics on matches
class-map type control-plane match-any copp-system-class-management-allow
  match access-group name copp-system-acl-allow
```

```
class-map type control-plane match-any copp-system-class-management-deny
  match access-group name copp-system-acl-deny
policy-map type control-plane copp-system-policy
    class copp-system-class-management-allow
        police cir 60000 kbps bc 250 ms conform transmit violate drop
    class copp-system-class-management-deny
        police cir 60000 kbps bc 250 ms conform drop violate drop
control-plane
  service-policy input copp-system-policy
```

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. For more information about Session Manager, see the Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 4.2.

Virtualization Support for IP ACLs

The following information applies to IP and MAC ACLs used in virtual device contexts (VDCs):

- ACLs are unique per VDC. You cannot use an ACL that you created in one VDC in a different VDC.
- Because ACLs are not shared by VDCs, you can reuse ACL names in different VDCs.
- The device does not limit ACLs or rules on a per-VDC basis.
- Configuring atomic ACL updates must be performed in the default VDC but applies to all VDCs.

Licensing Requirements for IP ACLs

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	No license is required to use IP ACLs. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For an explanation of the Cisco NX-OS licensing scheme, see the *Cisco Nexus 7000 Series NX-OS Licensing Guide, Release 4.2*.

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:

- 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 is especially useful for ACLs that include more than about 1000 rules. For more information about Session Manager, see the Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.2.
- In most cases, 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 a large number of 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 (additional 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.

  Note Prior to Cisco NX-OS Release 4.2(3), ACL logging does not support ACL processing that occurs on the supervisor module.

- When you apply an ACL that uses time ranges, the device updates the ACL entries on the affected I/O modules whenever a time range 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 7000 Series NX-OS Interfaces Configuration Guide, Release 4.2.

Default Settings for IP ACLs

This table lists the default settings for IP ACL parameters.

Table 2: Default IP ACL Parameters

Parameters	Default
IP ACLs	No IP ACLs exist by default
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, page 5

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 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 is especially useful for ACLs that include more than about 1000 rules. For more information about Session Manager, see the Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.2.

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 destination*
5. (Optional) **statistics per-entry**
6. (Optional) Enter one of the following commands:
   - **show ip access-lists** *name*
   - **show ipv6 access-lists** *name*
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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	**fragments** {**permit-all** | **deny-all**} **Example:** `switch(config-acl)# fragments permit-all`	(Optional) 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 destination* **Example:** `switch(config-acl)# permit ip 192.168.2.0/24 any`	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 more information, see the Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.2.
Step 5	**statistics per-entry** **Example:** `switch(config-acl)# statistics per-entry`	(Optional) Specifies that the device maintains global statistics for packets that match the rules in the ACL.
Step 6	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`	(Optional) Displays the IP ACL configuration.
Step 7	**copy running-config startup-config** **Example:** `switch(config-acl)# copy running-config startup-config`	(Optional) 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. 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 is especially useful for ACLs that include more than about 1000 rules. For more information about Session Manager, see the Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.2.

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

	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*	Enters IP ACL configuration mode for the ACL that you specify by name.

	Command or Action	Purpose

	Example:
	`switch(config)# ip access-list acl-01`
	`switch(config-acl)#`	

Step 3	[*sequence-number*] {**permit** | **deny**} *protocol source destination* **Example:** `switch(config-acl)# 100 permit ip 192.168.2.0/24 any`	(Optional) 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. For more information, see the Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.2.
Step 4	[**no**] **fragments** {**permit-all** | **deny-all**} **Example:** `switch(config-acl)# fragments permit-all`	(Optional) 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	**no** {*sequence-number* | {**permit** | **deny**} *protocol source destination*} **Example:** `switch(config-acl)# no 80`	(Optional) Removes the rule that you specified from the IP ACL. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.2.
Step 6	[**no**] **statistics per-entry** **Example:** `switch(config-acl)# statistics per-entry`	(Optional) 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	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`	(Optional) Displays the IP ACL configuration.
Step 8	**copy running-config startup-config** **Example:** `switch(config-acl)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Related Topics

- Changing Sequence Numbers in an IP ACL, page 19

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 is especially useful for ACLs that include more than about 1000 rules. For more information about Session Manager, see the Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 4.2.

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

	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	**show ip access-lists** *name* **Example:** `switch(config)# show ip access-lists acl-01`	(Optional) Displays the IP ACL configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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

	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.

	Command or Action	Purpose
Step 3	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`	(Optional) Displays the IP ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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 and subinterfaces
- VLAN interfaces
- Tunnels
- Management interfaces

ACLs applied to these interface types are considered router ACLs.

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*[**.** *number*]
   - **interface tunnel** *tunnel-number*
   - **interface vlan** *vlan-ID*
   - **interface mgmt** *port*
3. Enter one of the following commands:
   - **ip access-group** *access-list* {**in** | **out**}
   - **ipv6 traffic-filter** *access-list* {**in** | **out**}
4. (Optional) **show running-config aclmgr**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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*[**.** *number*]
	• **interface tunnel** *tunnel-number*
	• **interface vlan** *vlan-ID*
	• **interface mgmt** *port*
	
	Example:
	`switch(config)# interface tunnel 13`
	`switch(config-if)#`
		Enters configuration mode for the interface type that you specified.

Step 3	Enter one of the following commands: • **ip access-group** *access-list* {**in**	**out**} • **ipv6 traffic-filter** *access-list* {**in**	**out**}	Applies an IPv4 or IPv6 ACL to the Layer 3 interface for traffic flowing in the direction specified. You can apply one router ACL per direction.

	Command or Action	Purpose
	Example: `switch(config-if)# ip access-group acl-120 out`	
Step 4	**show running-config aclmgr** **Example:** `switch(config-if)# show running-config aclmgr`	(Optional) Displays the ACL configuration.
Step 5	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Related Topics

- Creating an IP ACL, page 15

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.

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.

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.

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

	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	**show running-config aclmgr** **Example:** `switch(config-if)# show running-config aclmgr`	(Optional) Displays the ACL configuration.
Step 5	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Related Topics

- Creating an IP ACL, page 15
- 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

Verifying IP ACL Configurations

To display IP ACL configuration information, perform one of the following tasks. For detailed information about the fields in the output from these commands, see the Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.2.

Command	Purpose
show running-config aclmgr	Displays the ACL configuration, including IP ACL configuration and interfaces that IP ACLs are applied to.
show ip access-lists	Displays the IPv4 ACL configuration.
show ipv6 access-lists	Displays the IPv6 ACL configuration.
show running-config interface	Displays the configuration of an interface to which you have applied an ACL.

Monitoring and Clearing IP ACL Statistics

To monitor or clear IP ACL statistics, use one of the commands in this table. For detailed information about these commands, see the Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.2.

Command	Purpose
show ip access-lists	Displays IPv4 ACL configuration. If the IPv4 ACL includes the **statistics per-entry** command, then 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
```

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 7000 Series NX-OS System Management Configuration Guide, Release 4.2.

Creating and Changing an IPv4 Address Object Group

You can create and change an IPv4 address group object.

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 network-wildcard*
   - [*sequence-number*] *IPv4-address*/*prefix-len*
4. Enter one of the following commands:
   - **no** [*sequence-number*]
   - **no host** *IPv4-address*
   - **no** *IPv4-address network-wildcard*
   - **no** *IPv4-address*/*prefix-len*
5. (Optional) **show object-group** *name*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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 network-wildcard* • [*sequence-number*] *IPv4-address*/*prefix-len* **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.

	Command or Action	Purpose
Step 4	Enter one of the following commands: • **no** [*sequence-number*] • **no host** *IPv4-address* • **no** *IPv4-address network-wildcard* • **no** *IPv4-address*/*prefix-len* **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	**show object-group** *name* **Example:** `switch(config-ipaddr-ogroup)# show object-group ipv4-addr-group-13`	(Optional) Displays the object group configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config-ipaddr-ogroup)# copy running-config startup-config`	(Optional) 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. **config t**
2. **object-group ipv6 address** *name*
3. Enter one of the following commands:
   - [*sequence-number*] **host** *IPv6-address*
   - [*sequence-number*] *IPv6-address*/*prefix-len*
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

	Command or Action	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`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* **Example:** `switch(config-ipv6addr-ogroup)# host 2001:db8:0:3ab0::1`	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 specify a network of hosts.
Step 4	Enter one of the following commands: • **no** *sequence-number* • **no host** *IPv6-address* • **no** *IPv6-address*/*prefix-len* **Example:** `switch(config-ipv6addr-ogroup)# no host 2001:db8:0:3ab0::1`	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.
Step 5	**show object-group** *name* **Example:** `switch(config-ipv6addr-ogroup)# show object-group ipv6-addr-group-A7`	(Optional) Displays the object group configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config-ipv6addr-ogroup)# copy running-config startup-config`	(Optional) 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

	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.

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 the port number that you specify only. • **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 number 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.

	Command or Action	Purpose
Step 5	**show object-group** *name* **Example:** `switch(config-port-ogroup)# show object-group NYC-datacenter-ports`	(Optional) Displays the object group configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config-port-ogroup)# copy running-config startup-config`	(Optional) 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

	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 object group that you specified.
Step 3	**show object-group** **Example:** `switch(config)# show object-group`	(Optional) Displays all object groups. The removed object group should not appear.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Verifying the Object-Group Configuration

To display object-group configuration information, perform one of the following tasks:

Command	Purpose
show object-group	Displays the object-group configuration.
show running-config aclmgr	Displays ACL configuration, including object groups.

For detailed information about the fields in the output from these commands, see the Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.2.

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 7000 Series NX-OS System Management Configuration Guide, Release 4.2.

Creating a Time Range

You can create a time range on the device and add rules to it.

Before You Begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

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

	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	[*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time* **Example:** `switch(config-time-range)# periodic monday 00:00:00 to friday 23:59:59`	(Optional) 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	[*sequence-number*] **periodic** *list-of-weekdays time* **to** *time* **Example:** `switch(config-time-range)# periodic weekdays 06:00:00 to 20:00:00`	(Optional) 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	[*sequence-number*] **absolute start** *time date* [**end** *time date*] **Example:** `switch(config-time-range)# absolute start 1:00 15 march 2008`	(Optional) 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	[*sequence-number*] **absolute** [start *time date*] **end** *time date* **Example:** `switch(config-time-range)# absolute end 23:59:59 31 december 2008`	(Optional) 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	**show time-range** *name* **Example:** `switch(config-time-range)# show time-range workday-daytime`	(Optional) Displays the time-range configuration.

	Command or Action	Purpose
Step 8	**copy running-config startup-config** **Example:** `switch(config-time-range)# copy running-config startup-config`	(Optional) 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.

Before You Begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

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

	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.

	Command or Action	Purpose
Step 3	[*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time* **Example:** `switch(config-time-range)# periodic monday 00:00:00 to friday 23:59:59`	(Optional) 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	[*sequence-number*] **periodic** *list-of-weekdays time* **to** *time* **Example:** `switch(config-time-range)# 100 periodic weekdays 05:00:00 to 22:00:00`	(Optional) 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	[*sequence-number*] **absolute start** *time date* [**end** *time date*] **Example:** `switch(config-time-range)# absolute start 1:00 15 march 2008`	(Optional) 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	[*sequence-number*] **absolute** [**start** *time date*] **end** *time date* **Example:** `switch(config-time-range)# absolute end 23:59:59 31 december 2008`	(Optional) 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	**no** {*sequence-number* | **periodic** *arguments* . . . | **absolute** *arguments*. . .} **Example:** `switch(config-time-range)# no 80`	(Optional) Removes the specified rule from the time range.
Step 8	**show time-range** *name* **Example:** `switch(config-time-range)# show time-range workday-daytime`	(Optional) Displays the time-range configuration.
Step 9	**copy running-config startup-config** **Example:** `switch(config-time-range)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Related Topics

- Changing Sequence Numbers in a Time Range, page 36

Removing a Time Range

You can remove a time range from the device.

Before You Begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

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

	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.
Step 3	**show time-range** **Example:** `switch(config-time-range)# show time-range`	(Optional) Displays the configuration for all time ranges. The removed time range should not appear.
Step 4	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) 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.

Before You Begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

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

	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	**show time-range** *name* **Example:** `switch(config)# show time-range daily-workhours`	(Optional) Displays the time-range configuration.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) 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. For detailed information about the fields in the output from these commands, see the Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.2.

Command	Purpose
show time-range	Displays the time-range configuration.

Command	Purpose
show running-config aclmgr	Displays ACL configuration, including all time ranges.

Additional References for IP ACLs

Related Documents

Related Topic	Document Title
IP ACL commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.2
Object group commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.2
Time range commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	Cisco Nexus 7000 Series NX-OS Security Command Reference, Release 4.2

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

Feature History for IP ACLs

This table lists the release history for this feature.

Table 3: Feature History for IP ACLs

Feature Name	Releases	Feature Information
ACL logging	4.2(3)	Support was added for logging of packets sent to the supervisor module for ACL processing.
IP ACLs	4.2(1)	Support was added for MAC packet classification on Layer 2 interfaces.