

Configure Access Control List

This chapter describes the Access Control List (ACL) and the procedures to configure ACL.

- Understand Access Control Lists, on page 2
- How an ACL Works, on page 3
- Apply ACLs, on page 5
- Configure an Ingress IPv4 ACL on Management Ethernet Interface, on page 5
- Configure an Egress IPv4 ACL on the Management Ethernet Interface, on page 6
- Configure an Ingress IPv6 ACL on the Management Ethernet Interface, on page 8
- Configure an Egress IPv6 ACL on the Management Ethernet Interface, on page 9
- Configure Extended Access Lists, on page 10
- Modify ACLs, on page 11

Understand Access Control Lists

Table 1: Feature History

Feature Name	Release Information	Feature Description
ACL on Management Port	Cisco IOS XR Release 7.11.1	Access Control List (ACL) feature enables you to permit or deny specific devices to connect to the management port and access NCS 1010 devices. This control enhances network security. Both IPv4 and IPv6 ACLs are supported on the management port. Commands added: • **ipv4-access-list** • **ipv4-access-group** • **show access-lists-ipv4** • **ipv6-access-list** • **ipv6-access-group** • **show access-lists-ipv6**

Access Control Lists (ACLs) perform packet filtering to control the packets that move through the network. These controls allow to limit the network traffic and restrict the access of users and devices to the network. ACLs have many uses, and therefore many commands accept a reference to an access list in their command syntax. An ACL consists of one or more access control entries (ACE) that collectively define the network traffic profile. Access control entries (ACE) are entries in an ACL that describe the access rights related to a particular security identifier or user.

There are 2 types of ACLs:

- Standard ACLs-Verifies only the source IP address of the packets. Traffic is controlled by the comparison of the address or prefix configured in the ACL, with the source address found in the packet.
- Extended ACLs-Verifies more than just the source address of the packets. Attributes such as destination address, specific IP protocols, User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) port numbers, Differentiated Services Code Point (DSCP), and so on are validated. Traffic is controlled by a comparison of the attributes stated in the ACL with those in the incoming or outgoing packets.

Purpose of ACLs

ACLs allow you to perform the following:

- Filter incoming or outgoing packets on an interface.
- Restrict the contents of routing updates.

- Limit debug output that is based on an address or protocol.
- Control vty access.

How an ACL Works

An ACL is a sequential list consisting of permit and deny statements that apply to IP addresses and upper-layer IP protocols. The ACL has a name by which it is referenced. Many software commands accept an ACL as part of their syntax.

An ACL can be configured and named; however, it does not take effect until the ACL is referenced by a command that accepts an ACL. Multiple commands can reference the same ACL. An ACL can control traffic arriving at the router or leaving the router, but not traffic originating at the router.

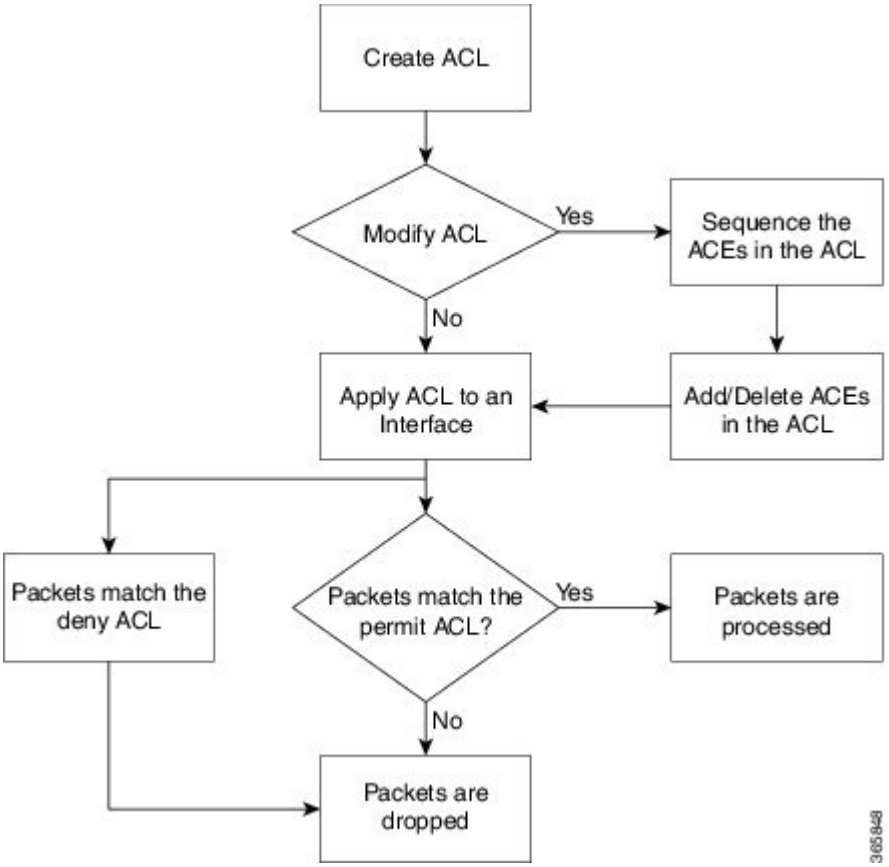
Source address and destination address are two of the most typical fields in an IP packet on which to base an ACL. Specify source addresses to control packets from certain networking devices or hosts. Specify destination addresses to control packets that are sent to certain networking devices or hosts.

You can also filter packets on the basis of transport layer information, such as whether the packet is a TCP, UDP, Internet Control Message Protocol (ICMP), or Internet Group Management Protocol (IGMP) packet.

ACL Workflow

The following image illustrates the workflow of an ACL.

Figure 1: ACL Workflow



Helpful Hints for Creating ACLs

Consider the following when creating ACLs:

- Create the access list before applying it to an interface.
- Organize your access list so that more specific references in a network or subnet appear before more general ones.
- To make the purpose of individual statements more easily understood at a glance, you can write a helpful remark before or after any statement.

Guidelines and Restrictions for Configuring ACLs

You must be aware of the following restrictions for configuring ACLs.

- Modifying an ACL when it is attached to the interface is supported.
- You can configure an ACL name with a maximum of 64 characters.
- You can configure an ACL name to comprise of only letters and numbers.

Apply ACLs

After you create an ACL, you must reference the ACL to make it work. ACL can be applied on *either* outbound or inbound interfaces. This section describes guidelines on how to accomplish this task for both terminal lines and network interfaces.

For inbound ACLs, after receiving a packet, Cisco IOS XR software checks the source address of the packet against the ACL. If the ACL permits the address, the software continues to process the packet. If the ACL rejects the address, the software discards the packet and returns an ICMP host unreachable message. The ICMP message is configurable.

For outbound ACLs, after receiving and routing a packet to a controlled interface, the software checks the source address of the packet against the ACL. If the ACL permits the address, the software sends the packet. If the ACL rejects the address, the software discards the packet and returns an ICMP host unreachable message.

When you apply an ACL that has not yet been defined to an interface, the software acts as if the ACL has not been applied to the interface and accepts all packets. Note this behavior if you use undefined ACLs as a means of security in your network.

Configure an Ingress IPv4 ACL on Management Ethernet Interface

Use the following configuration to configure an ingress IPv4 ACL on mgmtEth interface.

```
/* Configure mgmtEth interface with an IPv4 address */
RP/0/RP0/CPU0:ios#configure
Thu Oct 19 17:30:23.719 UTC
RP/0/RP0/CPU0:ios(config)#
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:ios(config-if)#interface mgmtEth 0/RP0/CPU0/2
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 198.51.100.247 255.255.255.0
RP/0/RP0/CPU0:ios(config-if)#no shutdown
RP/0/RP0/CPU0:ios(config-if)#commit
Thu Oct 19 17:31:25.127 UTC
RP/0/RP0/CPU0:ios(config-if)#exit


/* Verify if the interface is up */
RP/0/RP0/CPU0:ios(config)#do show ipv4 interface brief
Thu Oct 19 17:32:10.998 UTC

Interface                      IP-Address      Status          Protocol Vrf-Name
Loopback0                      192.0.2.89      Up              Up       default
GigabitEthernet0/0/0/0         198.51.100.1    Up              Up       default
GigabitEthernet0/0/0/2         198.51.100.10   Up              Up       default
MgmtEth0/RP0/CPU0/0            198.51.100.247  Up              Up       default
PTP0/RP0/CPU0/0                unassigned      Shutdown        Down     default
MgmtEth0/RP0/CPU0/1            192.0.2.121     Up              Up       default
PTP0/RP0/CPU0/1                unassigned      Shutdown        Down     default
MgmtEth0/RP0/CPU0/2            192.0.2.1       Down            Down     default


/* Configure an IPv4 ingress ACL */
RP/0/RP0/CPU0:ios(config)# ipv4 access-list V4-ACL-INGRESS
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 10 permit tcp 192.0.2.2 255.255.255.0 any
```

```
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 20 deny udp any any
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 30 permit ipv4 192.0.2.64 255.255.255.0 any
RP/0/RP0/CPU0:ios(config-ipv4-acl)# commit
Thu Oct 19 18:31:25.127 UTC


/* Verify the ingress ACL creation */
RP/0/RP0/CPU0:ios(config)# do show access-lists ipv4
Thu Oct 19 18:32:25.127 UTC
...
ipv4 access-list V4-ACL-INGRESS
 10 permit tcp 192.0.2.2 255.255.255.0 any
 20 deny udp any any
 30 permit ipv4 192.0.2.64 255.255.255.0 any


/* Apply the ingress ACL to the mgmtEth interface */
RP/0/RP0/CPU0:ios(config)# interface mgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:ios(config-if)# ipv4 access-group V4-ACL-INGRESS ingress
RP/0/RP0/CPU0:ios(config-if)#commit
Thu Oct 19 18:34:25.127 UTC
RP/0/RP0/CPU0:ios(config-if)#exit


/* Verify if the ingress ACL has been successfully applied to the interface */

RP/0/RP0/CPU0:ios#show ipv4 interface mgmtEth 0/RP0/CPU0/0
Fri Oct 20 05:07:06.383 UTC
MgmtEth0/RP0/CPU0/0 is Up, ipv4 protocol is Up
  Vrf is default (vrfid 0x60000000)
  Internet address is 198.51.100.247/16
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  common access list is not set, access list is V4-ACL-INGRESS
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000
```

You have successfully configured an IPv4 ingress ACL on the mgmtEth interface.

Configure an Egress IPv4 ACL on the Management Ethernet Interface

Use the following configuration to configure an egress IPv4 ACL on the mgmtEth interface.

```
/* Configure mgmtEth interface with an IPv4 address */
RP/0/RP0/CPU0:ios#configure
Thu Oct 19 17:30:23.719 UTC
RP/0/RP0/CPU0:ios(config)#
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 198.51.100.247 255.255.255.0
RP/0/RP0/CPU0:ios(config-if)#no shutdown
RP/0/RP0/CPU0:ios(config-if)#commit
Thu Oct 19 17:31:25.127 UTC
RP/0/RP0/CPU0:ios(config-if)#exit
```

```
/* Verify if the interface is up */
RP/0/RP0/CPU0:ios(config)#do show ipv4 interface brief
Thu Oct 19 17:32:10.998 UTC

Interface                      IP-Address      Status          Protocol Vrf-Name
Loopback0                      192.0.2.89      Up              Up       default
GigabitEthernet0/0/0/0         198.51.100.1    Up              Up       default
GigabitEthernet0/0/0/2         198.51.100.10   Up              Up       default
MgmtEth0/RP0/CPU0/0            198.51.100.247  Up              Up       default
PTP0/RP0/CPU0/0                unassigned      Shutdown        Down     default
MgmtEth0/RP0/CPU0/1            192.0.2.121     Up              Up       default
PTP0/RP0/CPU0/1                unassigned      Shutdown        Down     default
MgmtEth0/RP0/CPU0/2            192.0.2.1       Down            Down     default


/* Configure an IPv4 egress ACL */
RP/0/RP0/CPU0:ios(config)# ipv4 access-list V4-ACL-EGRESS
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 10 permit ipv4 203.0.113.1 255.255.255.0 192.0.2.1
0.255.255.255
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 20 deny udp any any
RP/0/RP0/CPU0:ios(config-ipv4-acl)# commit
Thu Oct 19 18:31:25.127 UTC


/* Verify the egress ACL creation */
RP/0/RP0/CPU0:ios(config)# do show access-lists ipv4
Thu Oct 19 18:32:25.127 UTC
ipv4 access-list V4-ACL-EGRESS
 10 permit ipv4 203.0.113.1 255.255.255.0 192.0.2.1 255.255.255.0
 20 deny ipv4 any any
...


/* Apply the egress ACL to the mgmtEth interface */
RP/0/RP0/CPU0:ios(config)# interface mgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:ios(config-if)# ipv4 access-group V4-ACL-EGRESS egress
RP/0/RP0/CPU0:ios(config-if)# commit
Thu Jul 11 09:19:49.569 UTC
RP/0/RP0/CPU0:ios(config-if)# exit


/* Verify if the egress ACL has been successfully applied to the interface */

RP/0/RP0/CPU0:ios#show ipv4 interface mgmtEth 0/RP0/CPU0/0
Fri Oct 20 05:07:06.383 UTC
MgmtEth0/RP0/CPU0/0 is Up, ipv4 protocol is Up
  Vrf is default (vrfid 0x60000000)
  Internet address is 198.51.100.247/16
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is V4-ACL-EGRESS
  Inbound  common access list is not set, access list is not set
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000
```

You have successfully configured an IPv4 egress ACL on the mgmtEth interface.

Configure an Ingress IPv6 ACL on the Management Ethernet Interface

Use the following configuration to configure an ingress IPv6 ACL on the mgmtEth interface.

```
/* Configure a mgmtEth interface with an IPv6 address */
RP/0/RP0/CPU0:ios#
RP/0/RP0/CPU0:ios#configure
Fri Oct 20 05:25:58.785 UTC
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/1
RP/0/RP0/CPU0:ios(config-if)#ipv6 address 2001::1/64
RP/0/RP0/CPU0:ios(config-if)#no shutdown
RP/0/RP0/CPU0:ios(config-if)#commit
Fri Oct 20 05:26:13.669 UTC
RP/0/RP0/CPU0:ios(config-if)#exit


/* Verify if the interface is up */
RP/0/RP0/CPU0:ios(config)#do show ipv6 interface mgmtEth 0/RP0/CPU0/1 brief
Fri Oct 20 05:26:52.612 UTC
MgmtEth0/RP0/CPU0/1    [Up/Up]
    fe80::3afd:f8ff:fe66:872
    2001::1


/* Configure an IPv6 ingress ACL */
RP/0/RP0/CPU0:ios(config)#ipv6 access-list V6-INGRESS-ACL
RP/0/RP0/CPU0:ios(config-ipv6-acl)#10 permit ipv6 any any
RP/0/RP0/CPU0:ios(config-ipv6-acl)#20 deny udp any any
RP/0/RP0/CPU0:ios(config-ipv6-acl)#commit
Fri Oct 20 05:28:46.664 UTC
RP/0/RP0/CPU0:ios(config-ipv6-acl)#exit


/* Verify the ingress ACL creation */
RP/0/RP0/CPU0:ios(config)#do show access-lists ipv6
Fri Oct 20 05:29:01.125 UTC
ipv6 access-list V6-INGRESS-ACL
 10 permit ipv6 any any
 20 deny udp any any


/* Apply the ingress ACL to the HundredGigE interface */
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/1
RP/0/RP0/CPU0:ios(config-if)#ipv6 access-group V6-INGRESS-ACL ingress
RP/0/RP0/CPU0:ios(config-if)#commit
Fri Oct 20 05:37:32.738 UTC
RP/0/RP0/CPU0:ios(config-if)#exit


/* Verify if the ingress ACL has been successfully applied to the interface */
RP/0/RP0/CPU0:ios(config)#do show ipv6 interface mgmtEth 0/RP0/CPU0/1
Fri Oct 20 05:38:00.753 UTC
MgmtEth0/RP0/CPU0/1 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
  IPv6 is enabled, link-local address is fe80::3afd:f8ff:fe66:872
  Global unicast address(es):
    2001::1, subnet is 2001::/64
  Joined group address(es): ff02::1:ff00:1 ff02::1:ff66:872 ff02::2
      ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
```

```
ICMP unreachables are enabled
ND DAD is enabled, number of DAD attempts 1
ND reachable time is 0 milliseconds
ND cache entry limit is 1000000000
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 160 to 240 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
Outgoing access list is not set
Inbound  common access list is not set, access list is V6-INGRESS-ACL
Table Id is 0xe0800000
Complete protocol adjacency: 0
Complete glean adjacency: 0
Incomplete protocol adjacency: 0
Incomplete glean adjacency: 0
Dropped protocol request: 0
Dropped glean request: 0
RA DNS Server Address Count: 0
RA DNS Search list Count: 0
```

You have successfully configured an IPv6 ingress ACL on the mgmtEth interface.

Configure an Egress IPv6 ACL on the Management Ethernet Interface

Use the following configuration steps to configure an egress IPv6 ACL on the mgmtEth interface.

```
/* Configure a mgmtEth interface with an IPv6 address */
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/1
RP/0/RP0/CPU0:ios(config-if)#ipv6 address 2001::1/64
RP/0/RP0/CPU0:ios(config-if)#no shutdown
RP/0/RP0/CPU0:ios(config-if)#commit
Thu Jan 25 11:41:25.778 IST
Router(config-if)# exit


/* Verify if the interface is up */
Router(config)# do show ipv6 interface brief
Thu Jul 11 09:47:50.812 UTC
HundredGigE 0/0/0/0 [Up/Up]
    fe80::bd:b9ff:fea9:5606
    1001::1
HundredGigE 0/0/0/1 [Up/Up]
    fe80::23:e9ff:fea8:a44e
    2001::1


/* Configure an IPv6 egress ACL */
Router(config)# ipv6 access-list V6-EGRESS-ACL
Router(config-ipv6-acl)# 10 permit ipv6 any any
Router(config-ipv6-acl)# 20 deny udp any any
Router(config-ipv6-acl)# commit
Thu Jul 11 09:50:40.566 UTC
Router(config-ipv6-acl)# exit


/* Verify the egress ACL creation */
RP/0/RP0/CPU0:ios(config)#do show ipv6 interface mgmtEth 0/RP0/CPU0/1 brief
Fri Oct 20 05:26:52.612 UTC
MgmtEth0/RP0/CPU0/1    [Up/Up]
```

```
    fe80::3afd:f8ff:fe66:872
    2001::1
…

/* Apply the egress ACL to the mgmtEth interface */
Router(config)# interface mgmtEth 0/RP0/CPU0/1
Router(config-if)# ipv6 access-group V6-EGRESS-ACL egress
Router(config-if)# commit
Thu Jul 11 09:52:57.751 UTC
Router(config-if)# exit


/* Verify if the egress ACL has been successfully applied to the interface */
RP/0/RP0/CPU0:ios(config)#do show ipv6 interface mgmtEth 0/RP0/CPU0/1
Fri Oct 20 05:38:00.753 UTC
MgmtEth0/RP0/CPU0/1 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
  IPv6 is enabled, link-local address is fe80::3afd:f8ff:fe66:872
  Global unicast address(es):
    2001::1, subnet is 2001::/64
  Joined group address(es): ff02::1:ff00:1 ff02::1:ff66:872 ff02::2
      ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND cache entry limit is 1000000000
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 160 to 240 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
  Outgoing access list is V6-EGRESS-ACL
  Inbound  common access list is not set, access list is not set
  Table Id is 0xe0800000
  Complete protocol adjacency: 0
  Complete glean adjacency: 0
  Incomplete protocol adjacency: 0
  Incomplete glean adjacency: 0
  Dropped protocol request: 0
  Dropped glean request: 0
  RA DNS Server Address Count: 0
  RA DNS Search list Count: 0
...
```

You have successfully configured an IPv6 egress ACL on the mgmtEth interface.

Configure Extended Access Lists

Use Extended Access Lists to verify more than just the source address of the packets. Attributes such as destination address, specific IP protocols, UDP or TCP port numbers, DSCP, and so on are validated. Traffic is controlled by a comparison of the attributes stated in the ACL with those in the incoming or outgoing packets.

To configure Extended Access Lists, you must create an access list and specify the condition to allow or deny the network traffic.

```
/* Enter the global configuration mode and create the access list*/
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ipv4 access-list acl_1
RP/0/RP0/CPU0:ios(config-ipv4-acl)#10 remark Do not allow user1 to telnet out
```

```
/*Specify the condition to allow or deny the network traffic.*/
RP/0/RP0/CPU0:ios(config-ipv4-acl)#10 permit 172.16.0.0 0.0.255.255
RP/0/RP0/CPU0:ios(config-ipv4-acl)#20 deny 192.168.34.0 0.0.0.255
RP/0/RP0/CPU0:ios(config-ipv4-acl)#commit
```

Running Configuration

```
RP/0/RP0/CPU0:ios#show running-config
Fri Oct 20 06:21:11.024 UTC
!! Building configuration...
!! IOS XR Configuration 24.1.1.23I
!! Last configuration change at Fri Oct 20 06:19:08 2023 by cisco

!
ipv4 access-list acl_1
 10 permit ipv4 172.16.0.0 0.0.255.255 any
 20 deny ipv4 192.168.34.0 0.0.0.255 any
!
```

Verification

```
RP/0/RP0/CPU0:ios#show access-lists ipv4 acl_1
Fri Oct 20 06:22:17.223 UTC
ipv4 access-list acl_1
 10 permit ipv4 172.16.0.0 0.0.255.255 any
 20 deny ipv4 192.168.34.0 0.0.0.255 any
```

Modify ACLs

This section describes a sample configuration to modify ACLs.

```
*/ Create an Access List*/
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ipv4 access-list acl_1

*/Add entries (ACEs) to the ACL*/
RP/0/RP0/CPU0:ios(config-ipv4-acl)#10 permit ip host 10.3.3.3 host 172.16.5.34
RP/0/RP0/CPU0:ios(config-ipv4-acl)#20 permit icmp any any
RP/0/RP0/CPU0:ios(config-ipv4-acl)#30 permit tcp any host 10.3.3.3
RP/0/RP0/CPU0:ios(config-ipv4-acl)#end


*/Verify the entries of the ACL*/:
Router#show access-lists ipv4 acl_1
ipv4 access-list acl_1
10 permit ip host 10.3.3.3 host 172.16.5.34
20 permit icmp any any
30 permit tcp any host 10.3.3.3


*/Add new entries, one with a sequence number "15" and another without a sequence number
to the ACL. Delete an entry with the sequence number "30":*/
RP/0/RP0/CPU0:ios(config)#ipv4 access-list acl_1
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 15 permit 10.5.5.5 0.0.0.255
RP/0/RP0/CPU0:ios(config-ipv4-acl)# no 30
RP/0/RP0/CPU0:ios(config-ipv4-acl)# permit 10.4.4.4 0.0.0.255
RP/0/RP0/CPU0:ios(config-ipv4-acl)# commit
```

/When an entry is added without a sequence number, it is automatically given a sequence number that puts it at the end of the access list. Because the default increment is 10, the entry will have a sequence number 10 higher than the last entry in the existing access list/

```
*/Verify the entries of the ACL:*/
RP/0/RP0/CPU0:ios#show access-lists ipv4 acl_1
ipv4 access-list acl_1
 10 permit ipv4 host 10.3.3.3 host 172.16.5.34

15 permit 10.5.5.5 0.0.0.255---*/newly added ACL (with the sequence number)*/
20 permit icmp any any
30 permit ipv4 10.4.4.0 0.0.0.255 any ---*/newly added ACL (without the sequence number)*/

*/The entry with the sequence number 30, that is, "30 permit tcp any host 10.3.3.3" is
deleted from the ACL*/
```

You have successfully modified ACLs in operation.