Access List Commands

This module describes the Cisco IOS XR software commands used to configure IP Version 4 (IPv4) and IP Version 6 (IPv6) access lists on Cisco ASR 9000 Series Aggregation Services Routers .

An access control list (ACL) consists of one or more access control entries (ACEs) that collectively define the network traffic profile. This profile can then be referenced by Cisco IOS XR Software software features such as traffic filtering, priority or custom queueing, and dynamic access control. Each ACL includes an action element (permit or deny) and a filter element based on criteria such as source address, destination address, protocol, and protocol-specific parameters.

For detailed information about ACL concepts, configuration tasks, and examples, refer to the *IP Addresses and Services Configuration Guide for Cisco ASR 9000 Series Routers*.

atomic-disable

Allows all traffic on the interface that matches the ACL rule, while the ACL is being modified.

hardware access-list atomic-disable

Syntax Description

<none>	Allows all traffic on the interface that matches the ACL rule, while the ACL is being modified.

Command Default

None

Command Modes

Privileged Executive mode

Command History

Release	Modification
Release 6.2.1	This command was introduced.

Usage Guidelines

When atomic ACL updates are disabled, the ACL is detached, and the ACL rules are not applied during the ACE modification process. Hence, it is recommended to configure to either permit or deny all traffic until the modification is complete.

For more information, see the Atomic ACL Updates By Using the Disable Option section in the *IP Addresses and Services Configuration Guide for Cisco NCS 5000 Series Routers*.

Example

To disable atomic updates on the hardware, by permitting all packets, use the following configuration.

```
RP/0/RSP0/CPU0:router# hardware access-list atomic-disable
```

clear access-list ipv4

To clear IPv4 access list counters, use the **clear access-list ipv4** command in EXEC mode .

clear access-list ipv4 *access-list name* `[ sequence-number | hardware { ingress | egress}] [interface type interface-path-id][ location node-id | sequence number ]`

Syntax Description

access-list-name	Name of a particular IPv4 access list. The name cannot contain a spaces or quotation marks, but can include numbers.
sequence-number	(Optional) Specific sequence number with which counters are cleared for an access list. Range is 1 to 2147483644 .
hardware	Identifies the access list as an access group for an interface.
ingress	Specifies an inbound direction.
egress	Specifies an outbound direction.
interface	(Optional) Clears the interface statistics.
type	Interface type. For more information, use the question mark (?) online help function.
interface-path-id	Physical interface or virtual interface. **Note** Use the **show interfaces** command to see a list of all interfaces currently configured on the router. For more information about the syntax for the router, use the question mark (?) online help function.
location *node-id*	(Optional) Clears hardware resource counters from the designated node. The *node-id* argument is entered in the *rack/slot/module* notation.
sequence *number*	(Optional) Clears counters for an access list with a specific sequence number. Range is 1 to 2147483644 .

Command Default

The default clears the specified IPv4 access list.

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

Use the **clear access-list ipv4** command to clear counters for a specified configured access list. Use a sequence number to clear counters for an access list with a specific sequence number.

Use the **hardware** keyword to clear counters for an access list that was enabled using the **ipv4 access-group** command.

Use an asterisk (***)** in place of the *access-list-name* argument to clear all access lists.

Note An access list can be shared among multiple interfaces. Clearing hardware counters clears all counters for all interfaces that use the specified access list in a given direction (ingress or egress).

Task ID

Task ID	Operations
basic-services	read, write
acl	read, write
bgp	read, write, execute

Examples

In the following example, counters for an access list named *marketing* are cleared:

```
RP/0/RSP0/CPU0:router# show access-lists ipv4 marketing

ipv4 access-list marketing
  10 permit ip 192.168.34.0 0.0.0.255 any (51 matches)
  20 permit ip 172.16.0.0 0.0.255.255 any (26 matches)
  30 deny tcp host 172.16.0.0 eq bgp host 192.168.202.203 30 (5 matches)

RP/0/RSP0/CPU0:router# clear access-list ipv4 marketing

RP/0/RSP0/CPU0:router# show access-lists ipv4 marketing

ipv4 access-list marketing
  10 permit ip 192.168.34.0 0.0.0.255 any
  20 permit ip 172.16.0.0 0.0.255.255 any
  30 deny tcp host 172.16.0.0 eq bgp host 192.168.202.203 30
```

In the following example, counters for an access list named acl_hw_1 in the outbound direction are cleared:

```
RP/0/RSP0/CPU0:router# show access-lists ipv4 acl_hw_1 hardware egress location 0/2/cp0

ipv4 access-list acl_hw_1
  10 permit icmp 192.168.36.0 0.0.0.255 any (251 hw matches)
  20 permit ip 172.16.3.0 0.0.255.255 any (29 hw matches)
  30 deny tcp any any (58 hw matches)

RP/0/RSP0/CPU0:router# clear access-list ipv4 acl_hw_1 hardware egress location 0/2/cp0

RP/0/RSP0/CPU0:router# show access-lists ipv4 acl_hw_1 hardware egress location 0/2/cp0

ipv4 access-list acl_hw_1
  10 permit icmp 192.168.36.0 0.0.0.255 any
  20 permit ip 172.16.3.0 0.0.255.255 any
  30 deny tcp any any
```

Related Commands

Command	Description
ipv4 access-group, on page 30	Filters incoming or outgoing IPv4 traffic on an interface.
ipv4 access-list, on page 33	Defines an IPv4 access list and enters IPv4 access list configuration mode.
resequence access-list ipv4 , on page 82	Renumbers an existing statement and increments subsequent statements to allow a new IPv4 access list statements.

clear access-list ipv6

To clear IPv6 access list counters, use the **clear access-list ipv6** command in EXEC mode.

clear access-list ipv6 *access-list-name* [{*sequence-number* | **hardware** {**ingress** | **egress**}}] [**interface** *type interface-path-id*] [{**location** *node-id* | **sequence** *number*}]

Syntax Description

access-list-name	Name of a particular IPv6 access list. The name cannot contain a spaces or quotation marks, but can include numbers.
sequence-number	(Optional) Specific sequence number for a particular access control entry (ACE) with which counters are cleared for an access list. Range is 1 to 2147483644.
hardware	(Optional) Identifies the access list as an access group for an interface.
ingress	(Optional) Specifies an inbound direction.
egress	(Optional) Specifies an outbound direction.
interface	(Optional) Clears the interface statistics.
type	(Optional) Interface type. For more information, use the question mark (?) online help function.
instance *interface-path-id*	Physical interface or virtual interface. **Note** Use the **show interfaces** command to see a list of all interfaces currently configured on the router. For more information about the syntax for the router, use the question mark (?) online help function.
location *node-id*	(Optional) Clears counters for an access list enabled on a card interface. The *node-id* argument is entered in the rack/slot/module notation.
sequence *number*	(Optional) Specifies a specific sequence number that clears access list counters. Range is 1 to 2147483644.

Command Default

The default clears the specified IPv6 access list.

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

The **clear access-list ipv6** command is similar to the **clear access-list ipv4** command, except that it is IPv6-specific.

Use the **clear access-list ipv6** command to clear counters for a specified configured access list. Use a sequence number to clear counters for an access list with a specific sequence number

Use the**hardware** keyword to clear counters for an access list that was enabled using the **ipv6 access-group** command.

Use an asterisk (*****) in place of the *access-list-name* argument to clear all access lists.

Note An access list can be shared among multiple interfaces. Clearing hardware counters clears all counters for all interfaces that use the specified access list in a given direction (ingress or egress).

Task ID

Task ID	Operations
basic-services	read, write
acl	read, write
network	read, write

Examples

In the following example, counters for an access list named *marketing* are cleared:

```
RP/0/RSP0/CPU0:router# show access-lists ipv6 marketing
ipv6 access-list marketing
  10 permit ipv6 3333:1:2:3::/64 any (51 matches)
  20 permit ipv6 4444:1:2:3::/64 any (26 matches)
  30 permit ipv6 5555:1:2:3::/64 any (5 matches)
RP/0/RSP0/CPU0:router# clear access-list ipv6 marketing
RP/0/RSP0/CPU0:router# show access-lists ipv6 marketing
ipv6 access-list marketing
  10 permit ipv6 3333:1:2:3::/64 any
  20 permit ipv6 4444:1:2:3::/64 any
  30 permit ipv6 5555:1:2:3::/64 any
```

In the following example, counters for an access list named acl_hw_1 in the outbound direction are cleared:

```
RP/0/RSP0/CPU0:router# show access-lists ipv6 acl_hw_1 hardware egress location 0/2/cp0
ipv6 access-list acl_hw_1
  10 permit ipv6 3333:1:2:3::/64 any (251 hw matches)
  20 permit ipv6 4444:1:2:3::/64 any (29 hw matches)
  30 deny tcp any any (58 hw matches)
RP/0/RSP0/CPU0:router# clear access-list ipv6 acl_hw_1 hardware egress location 0/2/cp0
RP/0/RSP0/CPU0:router# show access-lists ipv6 acl_hw_1 hardware egress location 0/2/cp0
ipv6 access-list acl_hw_1
  10 permit ipv6 3333:1:2:3::/64 any
  20 permit ipv6 4444:1:2:3::/64 any
  30 deny tcp any any
```

Related Commands

Command	Description
ipv6 access-list, on page 39	Defines an IPv6 access list and enters IPv6 access list configuration mode.

copy access-list ipv4

To create a copy of an existing IPv4 access list, use the **copy access-list ipv4** command in EXEC mode.

copy access-list ipv4 *source-acl destination-acl*

Syntax Description

source-acl	Name of the access list to be copied.
destination-acl	Name of the destination access list where the contents of the *source-acl* argument is copied.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

Use the **copy access-list ipv4** command to copy a configured access list. Use the *source-acl* argument to specify the access list to be copied and the *destination-acl* argument to specify where to copy the contents of the source access list. The *destination-acl* argument must be a unique name; if the *destination-acl* argument name exists for an access list or prefix list, the access list is not copied. The **copy access-list ipv4** command checks that the source access list exists then checks the existing list names to prevent overwriting existing access lists or prefix lists.

Task ID

Task ID	Operations
acl	read, write
filesystem	execute

Examples

In the following example, a copy of access list list-1 is created:

```
RP/0/RSP0/CPU0:router# show access-lists ipv4 list-1

ipv4 access-list list-1
  10 permit tcp any any log
  20 permit ip any any
RP/0/RSP0/CPU0:router# copy access-list ipv4 list-1 list-2
RP/0/RSP0/CPU0:router# show access-lists ipv4 list-2
ipv4 access-list list-2
  10 permit tcp any any log
  20 permit ip any any
```

In the following example, copying the access list list-1 to list-3 is denied because a list-3 access list already exists:

```
RP/0/RSP0/CPU0:router# copy access-list ipv4 list-1 list-3

list-3 exists in access-list

RP/0/RSP0/CPU0:router# show access-lists ipv4 list-3

ipv4 access-list list-3
  10 permit ip any any
  20 deny tcp any any log
```

Related Commands

Command	Description
ipv4 access-list, on page 33	Defines an IPv4 access list and enters IPv4 access list configuration mode.
show access-lists ipv4 , on page 87	Displays the contents of all current IPv4 access lists.

copy access-list ipv6

To create a copy of an existing IPv6 access list, use the **copy access-list ipv6** command in EXEC mode .

copy access-list ipv6 *source-acl destination-acl*

Syntax Description

source-acl	Name of the access list to be copied.
destination-acl	Destination access list where the contents of the *source-acl* argument is copied.

Command Default

No default behavior or value

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

Use the **copy access-list ipv6** command to copy a configured access list. Use the *source-acl* argument to specify the access list to be copied and the *destination-acl* argument to specify where to copy the contents of the source access list. The *destination-acl* argument must be a unique name; if the *destination-acl* argument name exists for an access list or prefix list, the access list is not copied. The **copy access-list ipv6** command checks that the source access list exists then checks the existing list names to prevent overwriting existing access lists or prefix lists.

Task ID

Task ID	Operations
acl	read, write
filesystem	execute

Examples

In this example, a copy of access list list-1 is created:

```
RP/0/RSP0/CPU0:router# show access-lists ipv6 list-1

ipv6 access-list list-1
  10 permit tcp any any log
  20 permit ipv6 any any

RP/0/RSP0/CPU0:router# copy access-list ipv6 list-1 list-2

RP/0/RSP0/CPU0:router# show access-lists ipv6 list-2

ipv6 access-list list-2
  10 permit tcp any any log
  20 permit ipv6 any any
```

In this example, copying access list list-1 to list-3 is denied because a list-3 access list already exists:

```
RP/0/RSP0/CPU0:router# copy access-list ipv6 list-1 list-3

list-3 exists in access-list

RP/0/RSP0/CPU0:router# show access-lists ipv6 list-3
ipv6 access-list list-3
  10 permit ipv6 any any
  20 deny tcp any any log
```

Related Commands

Command	Description
ipv6 access-list, on page 39	Defines an IPv6 access list and enters IPv6 access list configuration mode.
show access-lists ipv6, on page 93	Displays the contents of all current IPv6 access lists.

deny (IPv4)

To set conditions for an IPv4 access list, use the **deny** command in access list configuration mode. There are two versions of the **deny** command: **deny** (source), and **deny** (protocol). To remove a condition from an access list, use the **no** form of this command.

[*sequence-number*] **deny** *source* [*source-wildcard*] **counter** *counter-name* [{**log** | **log-input**}]
*[sequence-number]***deny***protocol source source-wildcard destination destination-wildcard*
[**precedence***precedence*] [**dscp***dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log | log-input**] [**ttl** *ttl value [value1....value2]*] [**counter** *counter-name*]
no *sequence-number*

Internet Control Message Protocol (ICMP)
[*sequence-number*] **deny icmp** *source source-wildcard destination destination-wildcard* [*icmp-type*] [*icmp-code*] [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [{**log** | **log-input**}] [**counter** *counter-name*] [**icmp-off**]

Internet Group Management Protocol (IGMP)
[*sequence-number*] **deny igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**dscp** *value*] [**fragments**] [{**log** | **log-input**}] [**counter** *counter-name*]

User Datagram Protocol (UDP)
[*sequence-number*] **deny udp** *source source-wildcard* [*operator* {*portprotocol-port*}] *destination destination-wildcard* [*operator* {*portprotocol-port*}] [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [{**log** | **log-input**}] [**counter** *counter-name*]

Syntax Description

sequence-number	(Optional) Number of the **deny** statement in the access list. This number determines the order of the statements in the access list. The number can be from 1 to 2147483644. (By default, the first statement is number 10, and the subsequent statements are incremented by 10.) Use the **resequence access-list** command to change the number of the first statement and increment subsequent statements of a configured access list.
source	Number of the network or host from which the packet is being sent. There are three alternative ways to specify the source: • Use a 32-bit quantity in four-part dotted-decimal format. • Use the **any** keyword as an abbreviation for a *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. • Use the **host** *source* combination as an abbreviation for a *source* and *source-wildcard* of *source* 0.0.0.0.
source-wildcard	Wildcard bits to be applied to the source. There are three alternative ways to specify the source wildcard: • Use a 32-bit quantity in four-part dotted-decimal format. Place ones in the bit positions you want to ignore. • Use the **any** keyword as an abbreviation for a *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. • Use the **host** *source* combination as an abbreviation for a *source* and *source-wildcard* of *source* 0.0.0.0.

protocol	Name or number of an IP protocol. It can be one of the keywords **ahp** , **esp** , **eigrp** , **gre** , **icmp** , **igmp** , **igrp** , **ip** , **ipinip** , **nos** , **ospf** , **pim** , **pcp** , **tcp** , or **udp** , or an integer from 0 to 255 representing an IP protocol number. To match any Internet protocol (including ICMP, TCP, and UDP), use the **ip** keyword. ICMP, and TCP allow further qualifiers, which are described later in this table.
destination	Number of the network or host to which the packet is being sent. There are three alternative ways to specify the destination: • Use a 32-bit quantity in four-part dotted-decimal format. • Use the **any** keyword as an abbreviation for the *destination* and *destination-wildcard* of 0.0.0.0 255.255.255.255. • Use the **host** *destination* combination as an abbreviation for a *destination* and *destination-wildcard* of *destination* 0.0.0.0.
destination-wildcard	Wildcard bits to be applied to the destination. There are three alternative ways to specify the destination wildcard: • Use a 32-bit quantity in four-part dotted-decimal format. Place ones in the bit positions you want to ignore. • Use the **any** keyword as an abbreviation for a *destination* and *destination-wildcard* of 0.0.0.0 255.255.255.255. • Use the **host** *destination* combination as an abbreviation for a *destination* and *destination-wildcard* of *destination* 0.0.0.0.
precedence *precedence*	(Optional) Packets can be filtered by precedence level (as specified by a number from 0 to 7) or by the following names: • **routine** —Match packets with routine precedence (0) • **priority** —Match packets with priority precedence (1) • **immediate** —Match packets with immediate precedence (2) • **flash** —Match packets with flash precedence (3) • **flash-override** —Match packets with flash override precedence (4) • **critical** —Match packets with critical precedence (5) • **internet** —Match packets with internetwork control precedence (6) • **network** —Match packets with network control precedence (7)

dscp *dscp*	(Optional) Differentiated services code point (DSCP) provides quality of service control. The values for *dscp* are as follows: • **0**–**63**–Differentiated services codepoint value • **af11**—Match packets with AF11 dscp (001010) • **af12**—Match packets with AF12 dscp (001100) • **af13**—Match packets with AF13 dscp (001110) • **af21**—Match packets with AF21 dscp (010010) • **af22**—Match packets with AF22 dscp (010100) • **af23**—Match packets with AF23 dscp (010110) • **af31**—Match packets with AF31 dscp (011010) • **af32**—Match packets with AF32 dscp (011100) • **af33**—Match packets with AF33 dscp (011110) • **af41**—Match packets with AF41 dscp (100010) • **af42**—Match packets with AF42 dscp (100100) • **af43**—Match packets with AF43 dscp (100110) • **cs1**—Match packets with CS1 (precedence 1) dscp (001000) • **cs2**—Match packets with CS2 (precedence 2) dscp (010000) • **cs3**—Match packets with CS3 (precedence 3) dscp (011000) • **cs4**—Match packets with CS4 (precedence 4) dscp (100000) • **cs5**—Match packets with CS5 (precedence 5) dscp (101000) • **cs6**—Match packets with CS6 (precedence 6) dscp (110000) • **cs7**—Match packets with CS7 (precedence 7) dscp (111000) • **default**—Default DSCP (000000) • **ef**—Match packets with EF dscp (101110)
fragments	(Optional) Causes the software to examine fragments of IPv4 packets when applying this access list entry. When this keyword is specified, fragments are subject to the access list entry.
log	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the **logging console** command.) The message includes the access list number, whether the packet was permitted or denied; the protocol, whether it was TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and source and destination port numbers. The message is generated for the first packet that matches a flow, and then at 5-minute intervals, including the number of packets permitted or denied in the prior 5-minute interval.
log-input	(Optional) Provides the same function as the **log** keyword, except that the logging message also includes the input interface.
ttl	(Optional) Turns on matching against time-to-life (TTL) value.

ttl value [*value1*. . *value2*[	(Optional) TTL value used for filtering. Range is 1 to 255. If only *value* is specified, the match is against this value. If both *value1* and *value2* are specified, the packet TTL is matched against the range of TTLs between *value1* and *value2* .
icmp-off	(Optional) Turns off ICMP generation for denied packets.
icmp-type	(Optional) ICMP message type for filtering ICMP packets. Range is from 0 to 255.
icmp-code	(Optional) ICMP message code for filtering ICMP packets. Range is from 0 to 255.
igmp-type	(Optional) IGMP message type (0 to 15) or message name for filtering IGMP packets, as follows: • dvmrp • host-query • host-report • mtrace • mtrace-response • pim • precedence • trace • v2-leave • v2-report • v3-report
operator	(Optional) Operator is used to compare source or destination ports. Possible operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range). If the operator is positioned after the *source* and *source-wildcard* values, it must match the source port. If the operator is positioned after the *destination* and *destination-wildcard* values, it must match the destination port. If the operator is positioned after the **ttl** keyword, it matches the TTL value. The **range** operator requires two port numbers. All other operators require one port number.
port	Decimal number of a TCP or UDP port. A port number is a number from 0 to 65535. TCP ports can be used only when filtering TCP. UDP ports can be used only when filtering UDP.
protocol-port	Name of a TCP or UDP port. TCP and UDP port names are listed in the “Usage Guidelines” section. TCP port names can be used only when filtering TCP. UDP port names can be used only when filtering UDP.
established	(Optional) For the TCP protocol only: Indicates an established connection.

match-any	(Optional) For the TCP protocol only: Filters on any combination of TCP flags.
match-all	(Optional) For the TCP protocol only: Filters on all TCP flags.
+ | -	(Required) For the TCP protocol **match-any** , **match-all** : Prefix *flag-name* with + or - . Use the + *flag-name* argument to match packets with the TCP flag set. Use the - *flag-name* argument to match packets when the TCP flag is not set.
flag-name	(Optional) For the TCP protocol **match-any** , **match-all** . Flag names are: **ack** , **fin** , **psh** , **rst** , **syn** .
counter	(Optional) Enables accessing ACL counters using SNMP query. The **counter** *counter-name* keyword is available on Cisco ASR 9000 Enhanced Ethernet Line Cards only.
counter-name	Defines an ACL counter name.

Command Default

There is no specific condition under which a packet is denied passing the IPv4 access list.

ICMP message generation is enabled by default.

Command Modes

IPv4 access list configuration

Command History

Release	Modification
Release 3.7.2	This command was introduced.
Release 5.1.1	The optional keyword **counter** and the associated argument *counter-name* were added to the command.
Release 5.3.0	The ABF feature can be configured along with object-groups while defining an ACEs (Access Control Entry).

Usage Guidelines

Use the **deny** command following the **ipv4 access-list** command to specify conditions under which a packet cannot pass the access list.

By default, the first statement in an access list is number 10, and the subsequent statements are incremented by 10.

You can add **permit**, **deny**, or **remark** statements to an existing access list without retyping the entire list. To add a new **s**tatement anywhere other than at the end of the list, create a new statement with an appropriate entry number that falls between two existing entry numbers to indicate where it belongs.

If you want to add a statement between two consecutively numbered statements (for example, between lines 10 and 11), first use the **resequence access-list** command to renumber the first statement and increment the entry number of each subsequent statement. The *increment* argument causes new, unused line numbers between statements. Then add a new statement with the *entry-number* argument, specifying where it belongs in the access list.

The following is a list of precedence names:

- critical
- flash
- flash-override
- immediate
- internet
- network
- priority
- routine

The following is a list of ICMP message type names:

- administratively-prohibited
- alternate-address
- conversion-error
- dod-host-prohibited
- dod-net-prohibited
- echo
- echo-reply
- general-parameter-problem
- host-isolated
- host-precedence-unreachable
- host-redirect
- host-tos-redirect
- host-tos-unreachable
- host-unknown
- host-unreachable
- information-reply
- information-request
- mask-reply
- mask-request
- mobile-redirect
- net-redirect
- net-tos-redirect
- net-tos-unreachable
- net-unreachable
- network-unknown
- no-room-for-option
- option-missing
- packet-too-big
- parameter-problem
- port-unreachable
- precedence-unreachable
- protocol-unreachable
- reassembly-timeout
- redirect
- router-advertisement
- router-solicitation

- source-quench
- source-route-failed
- time-exceeded
- timestamp-reply
- timestamp-request
- traceroute
- ttl-exceeded
- unreachable

The following is a list of TCP port names that can be used instead of port numbers. Refer to the current *Assigned Numbers* RFC to find a reference to these protocols. You can find port numbers corresponding to these protocols by typing a ? in the place of a port number.

- bgp
- chargen
- cmd
- daytime
- discard
- domain
- echo
- exec
- finger
- ftp
- ftp-data
- gopher
- hostname
- ident
- irc
- klogin
- kshell
- login
- lpd
- nntp
- pim-auto-rp
- pop2
- pop3
- smtp
- sunrpc
- tacacs
- talk
- telnet
- time
- uucp
- whois
- www

The following UDP port names can be used instead of port numbers. Refer to the current *Assigned Numbers* RFC to find a reference to these protocols. You can find port numbers corresponding to these protocols by typing a ? in the place of a port number.

- biff
- bootpc
- bootps
- discard
- dnsix
- domain
- echo
- isakmp
- mobile-ip
- nameserver
- netbios-dgm
- netbios-ns
- netbios-ss
- ntp
- pim-auto-rp
- rip
- snmp
- snmptrap
- sunrpc
- syslog
- tacacs
- talk
- tftp
- time
- who
- xdmcp

Use the following flags in conjunction with the **match-any** and **match-all** keywords and the + and - signs to select the flags to display:

- ack
- fin
- psh
- rst
- syn

For example, **match-all** + *ack* + *syn* displays TCP packets with both the ack *and* syn flags set, or **match-any** + *ack* - *syn* displays the TCP packets with the ack set *or* the syn not set.

Note If any ACE in an ACL contains ABF clause, this ACL cannot be applied at any non-zero compression level.

Task ID

Task ID	Operations
ipv4	read, write
acl	read, write

Examples

This example shows how to set a deny condition for an access list named Internet filter:

```
RP/0/RSP0/CPU0:router(config)# ipv4 access-list Internetfilter
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 deny 192.168.34.0 0.0.0.255
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 deny 172.16.0.0 0.0.255.255
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 25 deny tcp host 172.16.0.0 eq bgp host
192.168.202.203 range 1300 1400
RP/0/RSP0/CPU0:router(config-ipv4-acl)# permit 10.0.0.0 0.255.255.255
```

Related Commands

Command	Description
ipv4 access-group, on page 30	Filters incoming or outgoing IPv4 traffic on an interface.
ipv4 access-list, on page 33	Defines an IPv4 access list and enters IPv4 access list configuration mode.
permit (IPv4) , on page 54	Sets the permit conditions for an IPv4 access list.
remark (IPv4) , on page 78	Inserts a helpful remark about an IPv4 access list entry.
resequence access-list ipv4 , on page 82	Changes the starting entry number of the first statement in an existing IPv4 access list, and the number by which subsequent statements are incremented.
show access-lists ipv4 , on page 87	Displays the contents of all current IPv4 access lists.

deny (IPv6)

To set deny conditions for an IPv6 access list, use the **deny** command in IPv6 access list configuration mode. To remove the deny conditions, use the **no** form of this command.

[sequence-number] **deny** *protocol* {*source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/ prefix-length*} [*operator* {*port* | *protocol-port*}] [**dscp***value*] [**routing**] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log | log-input**] [**ttl** *operator ttl value*]
no *sequence-number*

Internet Control Message Protocol (ICMP)
*[sequence-number]***deny icmp** {*source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/ prefix-length*} {*destination-ipv6-prefix/ prefix-length* | *any* | *host destination-ipv6-address ipv6-wildcard-mask/ prefix-length*} [*icmp-type*] [*icmp-code*] [**dscp** *value*] [`routing`] [**authen**] [**destopts**] [**fragments**] [**log**] [**log-input**] [**icmp-off**]

Transmission Control Protocol (TCP)
*[sequence-number]***deny tcp**{*source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/ prefix-length*} [*operator*{*port* | *protocol-port*}] {*destination-ipv6-prefix/ prefix-length* | *any* | *host destination-ipv6-address ipv6-wildcard-mask/ prefix-length*} [*operator*{*port* | *protocol* | *port*}] [**dscp***value*] [**routing**] [**authen**] [**destopts**] [**fragments**] [**established**] {**match-any | match-all | + | -**} [*flag-name*] [**log**] [**log-input**]

User Datagram Protocol (UDP)
*[sequence-number]***deny tcp**{*source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/ prefix-length*} [*operator*{*port* | *protocol-port*}] {*destination-ipv6-prefix/ prefix-length* | *any* | *host destination-ipv6-address ipv6-wildcard-mask/ prefix-length*} [*operator*{*port* | *protocol* | *port*}] [**dscp***value*] [**routing**] [**authen**] [**destopts**] [**fragments**] [**established**] [*flag-name*] [**log**] [**log-input**]

Syntax Description

sequence-number	(Optional) Number of the **deny** statement in the access list. This number determines the order of the statements in the access list. Range is 1 to 2147483644. (By default, the first statement is number 10, and the subsequent statements are incremented by 10.) Use the **resequence access-list** command to change the number of the first statement and increment subsequent statements of a configured access list.
protocol	Name or number of an Internet protocol. It can be one of the keywords **ahp** , e**igrp**, **esp** , **gre**, **icmp** , **igmp**, **igrp**, **ipinip**, **ipv6** , **nos**, **ospf**, **pcp** , **tcp** , or **udp** , or an integer in the range from 0 to 255 representing an IPv6 protocol number.
source-ipv6-prefix / prefix-length	The source IPv6 network or class of networks about which to set deny conditions. This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
any	An abbreviation for the IPv6 prefix ::/0.

host *source-ipv6-address*	Source IPv6 host address about which to set deny conditions. This *source-ipv6-address* argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
ipv6-wildcard-mask	IPv6 wildcard mask. The IPv6 wildcard mask can take any IPv6 address value which is used instead of prefix length.
operator {*port* | *protocol-port*}	(Optional) Operand that compares the source or destination ports of the specified protocol. Operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range). If the operator is positioned after the *source-ipv6-prefix / prefix-length* argument, it must match the source port. If the operator is positioned after the *destination-ipv6-prefix / prefix-length* argument, it must match the destination port. The **range** operator requires two port numbers. All other operators require one port number. The *port* argument is the decimal number of a TCP or UDP port. Range is 0 to 65535. The *protocol-port* argument is the name of a TCP or UDP port. TCP port names can be used only when filtering TCP. UDP port names can be used only when filtering UDP.
destination-ipv6-prefix / prefix-length	Destination IPv6 network or class of networks about which to set deny conditions. This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
host *destination-ipv6-address*	Destination IPv6 host address about which to set deny conditions. This *destination-ipv6-address* argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
dscp *value*	(Optional) Matches a differentiated services code point DSCP value against the traffic class value in the Traffic Class field of each IPv6 packet header. Range is 0 to 63.
routing	(Optional) Matches source-routed packets against the routing extension header within each IPv6 packet header.
authen	(Optional) Matches if the IPv6 authentication header is present.
destopts	(Optional) Matches if the IPv6 destination options header is present.
fragments	(Optional) Matches noninitial fragmented packets where the fragment extension header contains a nonzero fragment offset. The **fragments** keyword is an option only if the *operator* [*port-number*] arguments are not specified.

log	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the **logging console** command.) The message includes the access list name and sequence number, whether the packet was denied; the protocol, whether it was TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and source and destination port numbers. The message is generated for the first packet that matches, and then at 5-minute intervals, including the number of packets denied in the prior 5-minute interval.
log-input	(Optional) Provides the same function as the **log** keyword, except that the logging message also includes the input interface.
ttl	(Optional) Turns on matching against time-to-life (TTL) value.
operator	(Optional) Operand that compares the source or destination ports of the specified protocol. Operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range).
ttl value [*value1 ... value2*]	(Optional) TTL value used for filtering. Range is 1 to 255. If only *value* is specified, the match is against this value. If both *value1* and *value2* are specified, the packet TTL is matched against the range of TTLs between *value1* and *value2* .
icmp-off	(Optional) Turns off ICMP generation for denied packets
icmp-type	(Optional) ICMP message type for filtering ICMP packets. ICMP packets can be filtered by ICMP message type. Range is 0 to 255.
icmp-code	(Optional) ICMP message code for filtering ICMP packets. ICMP packets that are filtered by ICMP message type can also be filtered by the ICMP message code. Range is 0 to 255.
established	(Optional) For the TCP protocol only: Indicates an established connection.
match-any	(Optional) For the TCP protocol only: Filters on any combination of TCP flags.
match-all	(Optional) For the TCP protocol only: Filters on all TCP flags.
+ | -	(Required) For the TCP protocol **match-any** , **match-all** : Prefix *flag-name* with + or - . Use the + *flag-name* argument to match packets with the TCP flag set. Use the - *flag-name* argument to match packets when the TCP flag is not set.
flag-name	(Optional) For the TCP protocol **match-any** , **match-all** . Flag names are: **ack**, **fin**, **psh**, **rst**, **syn**.

Command Default

No IPv6 access list is defined.

ICMP message generation is enabled by default.

Command Modes

IPv6 access list configuration

Command History

Release	Modification
Release 3.7.2	This command was introduced.
Release 5.2.2	The support for IPv6 wildcard mask with a source and destination address was added.
Release 5.3.0	The ABF feature can be configured along with object-groups while defining an ACEs (Access Control Entry).

Usage Guidelines

The **deny** (IPv6) command is similar to the **deny** (IPv4) command, except that it is IPv6-specific.

Use the **deny** (IPv6) command following the **ipv6 access-list** command to define the conditions under which a packet passes the access list.

Note If any ACE in an ACL contains ABF clause, this ACL cannot be applied at any non-zero compression level.

Specifying **ipv6** for the *protocol* argument matches against the IPv6 header of the packet.

By default, the first statement in an access list is number 10, and the subsequent statements are numbered in increments of 10.

You can add permit, deny, or remark statements to an existing access list without retyping the entire list. To add a new statement anywhere other than at the end of the list, create a new statement with an appropriate entry number that falls between two existing entry numbers to indicate where it belongs.

Both the *source-ipv6-prefix/prefix-length* and *destination-ipv6-prefix/prefix-length* arguments are used for traffic filtering (the source prefix filters traffic based upon the traffic source; the destination prefix filters traffic based upon the traffic destination).

Note IPv6 prefix lists, not access lists, should be used for filtering routing protocol prefixes.

The **fragments** keyword is an option only if the *operator* [*port* | *protocol-port*] arguments are not specified.

Task ID

Task ID	Operations
acl	read, write

Examples

The following example shows how to configure the IPv6 access list named toCISCO and applies the access list to outbound traffic on GigabitEthernet interface 0/2/0/2. Specifically, the first deny entry in the list keeps all packets that have a destination TCP port number greater than 5000 from exiting out of GigabitEthernet interface 0/2/0/2. The second deny entry in the list keeps all packets that have a source UDPo port number less than 5000 from exiting out of GigabitEthernet interface 0/2/0/2. The second deny entry also logs all matches to the console. The first permit entry in the list permits all ICMP packets to exit out of GigabitEthernet interface 0/2/0/2. The second permit entry in the

list permits all other traffic to exit out of GigabitEthernet interface 0/2/0/2. The second permit entry is necessary because an implicit deny all condition is at the end of each IPv6 access list.

```
RP/0/RSP0/CPU0:router(config)# ipv6 access-list toCISCO
RP/0/RSP0/CPU0:router(config-ipv6-acl)# deny tcp any any gt 5000
RP/0/RSP0/CPU0:router(config-ipv6-acl)# deny ipv6 any lt 5000 any log
RP/0/RSP0/CPU0:router(config-ipv6-acl)# permit icmp any any
RP/0/RSP0/CPU0:router(config-ipv6-acl)# permit any any
RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/2/0/2
RP/0/RSP0/CPU0:router(config-if)# ipv6 access-group tOCISCO out
```

Related Commands

Command	Description
ipv6 access-list, on page 39	Defines an IPv6 access list and enters IPv6 access list configuration mode.
permit (IPv6) , on page 69	Sets permit conditions for an IPv6 access list.
remark (IPv6) , on page 80	Inserts a helpful remark about an IPv6 access list entry.
resequence access-list ipv6 , on page 84	Changes the starting entry number of the first statement in an existing IPv6 access list, and the number by which subsequent statements are incremented.

hardware access-list atomic disable

To disable atomic access-control list (ACL) updates, use the **hardware access-list atomic disable** command in global configuration mode. To enable atomic ACL updates, enter the **no** form of this command.

hardware access-list atomic disable
no hardware access-list atomic disable

Syntax Description

atomic	Specifies that the device performs atomic updates, which do not disrupt traffic during the update. By default, a device performs atomic ACL updates.
disable	Specifies that atomic ACL updates should be disabled.

Command Default

None.

Command Modes

Global configuration

Command History

Release	Modification
Release 6.0.1	This command was introduced.

Usage Guidelines

If a NP lacks the TCAM resources required for an atomic update, you can disable atomic updates by using the **hardware access-list atomic disable** command.

Note When atomicity is disabled, during an ACL edit there will be a duration (in milli-seconds) wherein the ACL is detached for performing this operation.

Task ID

Task ID	Operations
acl	read, write

Examples

This example shows how to disable atomic ACL updates:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router#(config)#hardware access-list atomic disable
```

hardware access-list l3-compression-optimisation

To increase the TCAM space allocated for compression fields in ACL Layer 3 compression from 70 to 76 bytes, use the **hardware access-list l3-compression-optimisation** command. Use the **no** form of this command to reverse the TCAM allocation.

hardware access-list l3-compression-optimisation

Command Default

If you do not configure the **hardware access-list l3-compression-optimisation** command, the TCAM space allocated for ACL Layer 3 compression is 70 bytes.

Command Modes

Global configuration

Command History

Release	Modification
Release 7.5.3	This command was introduced.

Usage Guidelines

By default, the TCAM space allotted in ACL for compression fields is 70 bytes and non-compression fields is 10 bytes. If you enable this command, then the TCAM space for compression field will increase to 76 bytes by assigning addition 6 bytes from non-compression fields and the TCAM space for non-compression fields will reduce to 4 bytes.

Task ID

Task ID	Operations
acl	read, write

Examples

The following example shows you how to use hardware access-list l3-compression-optimisation command:

```
Router# config
Router(config)# hardware access-list l3-compression-optimisation
Router(config)# commit
```

ipv4 access-group

To control access to an interface, use the **ipv4 access-group** command in interface configuration mode. To remove the specified access group, use the **no** form of this command.

ipv4 access-group [**common** *acl-name*] *access-list-name* {**ingress** | **egress**} [**hardware-count**] [**interface-statistics**]
[compress level *level]*
no ipv4 access-group [**common** *acl-name*] *access-list-name* {**ingress** | **egress**} [**hardware-count**] [**interface-statistics**]
[compress level *level]*

Syntax Description

access-list-name	Name of an IPv4 access list as specified by an **ipv4 access-list** command.
common *acl-name*	Specifies the common access-list name.
ingress	Filters on inbound packets.
egress	Filters on outbound packets.
hardware-count	(Optional) Specifies to access a group's hardware counters.
interface-statistics	(Optional) Specifies per-interface statistics in the hardware.
compress level *level*	Specifies ACL compression in the hardware. The available compression levels are 0, 1, and 3.

Command Default

The interface does not have an IPv4 access list applied to it.

Command Modes

Interface configuration

Command History

Release	Modification
Release 3.7.2	This command was introduced.
Release 4.3.1	The **common** keyword was added.
Release 4.3.1	The **compress level** keyword was added.

Usage Guidelines

Use the **ipv4 access-group** command to control access to an interface. To remove the specified access group, use the **no** form of the command. Use the *access-list-name* argument to specify a particular IPv4 access list.

Use the **ingress** keyword to filter on inbound packets or the**egress** keyword to filter on outbound packets. Use the *hardware-count* argument to enable hardware counters for the access group.

Permitted packets are counted only when hardware counters are enabled using the *hardware-count* argument. Denied packets are counted whether hardware counters are enabled, or not.

Filtering of MPLS packets through common ACL and interface ACL is not supported.

Restrictions for common ACLs are:

- Common ACL is supported in only ingress direction and for L3 interfaces only.
- The **interface-statistics** option is not available for common ACLs.
- The **hardware-count** option is available for only IPv4 ACLs.
- Only one common IPv4 and IPv6 ACL is supported on each line card.
- The common ACL option is not available for Ethernet Service (ES) ACLs.
- The IPv4 and IPv6 common ACL is limited to 200 Ternary Content Addressable Memory(TCAM) entries for the ASR 9000 Enhanced Ethernet line card and A9K-SIP-700 line card. Although, A9K-SIP-700 line card may support more.
- Common ACL is not supported on ASR 9000 Ethernet line card and ASR 9000 Enhanced Ethernet-TR line card.
- You can specify only common ACL or only interface ACL or both common and interface ACL in this command.
- The **compress** option is not supported for common ACLs.
- Object-groups are not supported with common ACLs.
- The **interface-statistics** and **hardware-count** options are not supported for ACLs on the A9K-SIP-700 line card.

Note For packet filtering applications using the **ipv4 access-group** command, packet counters are maintained in hardware for each direction. If an access group is used on multiple interfaces in the same direction, then packets are counted for each interface that has the *hardware-count* argument enabled.

If the access list permits the addresses, the software continues to process the packet. If the access list denies the address, the software discards the packet and returns an Internet Control Message Protocol (ICMP) host unreachable message.

If the specified access list does not exist, all packets are passed.

By default, the unique or per-interface ACL statistics are disabled.

Task ID

Task ID	Operations
acl	read, write
network	read, write

Examples

The following example shows how to apply filters on packets inbound and outbound from GigabitEthernet interface 0/2/0/2:

```
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet  0/2/0/2
RP/0/RSP0/CPU0:router(config-if)# ipv4 access-group p-ingress-filter ingress
RP/0/RSP0/CPU0:router(config-if)# ipv4 access-group p-egress-filter egress
```

The following example shows how to apply per-interface statistics in the hardware:

```
RP/0/RSP0/CPU0:router(config)# interface gigabitethernet  0/2/0/2
RP/0/RSP0/CPU0:router(config-if)# ipv4 access-group p-ingress-filter ingress
interface-statistics
```

This example shows how to configure common ACL:

```
RP/0/RSP0/CPU0:router# config
RP/0/RSP0/CPU0:router(config)# interface gigabitethernet  0/1/0/4
RP/0/RSP0/CPU0:router(config-if)# ipv4 access-group common common-acl interface-acl ingress
```

This example shows how to configure the number of fields to be compressed in hardware:

```
RP/0/RSP0/CPU0:router# config
RP/0/RSP0/CPU0:router(config)# interface gigabitethernet  0/1/0/4
RP/0/RSP0/CPU0:router(config-if)# ipv4 access-group acl1 ingress compress level 1
```

Related Commands

Command	Description
clear access-list ipv4, on page 4	Resets the IPv4 access list match counters.
deny (IPv4) , on page 14	Sets the deny conditions for an ACE of an IPv4 access list.
ipv4 access-list, on page 33	Defines an IPv4 access list and enters IPv4 access list configuration mode.
permit (IPv4) , on page 54	Sets the permit conditions for an ACE of an IPv4 access list.
show access-lists ipv4 , on page 87	Displays the contents of all current IPv4 access lists.
show ipv4 interface	Displays the usability status of interfaces configured for IPv4.

ipv4 access-list

To define an IPv4 access list by name, use the **ipv4 access-list** command in Global Configuration mode. To remove all entries in an IPv4 access list, use the **no** form of this command.

ipv4 access-list *name*

Syntax Description

name	Name of the access list. Names cannot contain a space or quotation marks.

Command Default

No IPv4 access list is defined.

Command Modes

Global Configuration mode

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

Use the **ipv4 access-list** command to configure an IPv4 access list. This command places the router in access list configuration mode, in which the denied or permitted access conditions must be defined with the **deny** or **permit** command.

Use the **resequence access-list ipv4** command if you want to add a **permit** , **deny** , or **remark** statement between consecutive entries in an existing IPv4 access list. Specify the first entry number (the *base*) and the increment by which to separate the entry numbers of the statements. The software renumbers the existing statements, thereby making room to add new statements with the unused entry numbers.

Use the **ipv4 access-group** command to apply the access list to an interface.

Task ID

Task ID	Operations
acl	read, write

Examples

The following example shows how to define a standard access list named Internetfilter:

```
Router(config)# ipv4 access-list Internetfilter
Router(config-if)# 10 permit 192.168.34.0 0.0.0.255
Router(config-if)# 20 permit 172.16.0.0 0.0.255.255
Router(config-if)# 30 permit 10.0.0.0 0.255.255.255
Router(config-if)# 39 remark Block BGP traffic from 172.16 net.
Router(config-if)# 40 deny tcp host 172.16.0.0 eq bgp host 192.168.202.203 range 1300 1400
```

Related Commands

Command	Description
show access-lists ipv4	Displays the contents of all current IPv4 access lists.

ipv4 access-list log-update rate

To specify the rate at which IPv4 access lists are logged, use the **ipv4 access-list log-update rate** command in Global Configuration mode. To return the update rate to the default setting, use the **no** form of this command.

ipv4 access-list log-update rate *rate-number*
no ipv4 access-list log-update rate *rate-number*

Syntax Description

rate-number	Rate at which IPv4 access hit logs are generated per second on the router. Range is 1 to 1000.

Command Default

Default is 1.

Command Modes

Global Configuration mode

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

The *rate-number* argument applies to all the IPv4 access-lists configured on the interfaces. That is, at any given time there can be between 1 and 1000 log entries for the system.

Task ID

Task ID	Operations
ipv4	read, write
acl	read, write

Examples

The following example shows how to configure a IPv4 access hit logging rate for the system:

```
RP/0/RSP0/CPU0:router(config)# ipv4 access-list log-update rate 10
```

ipv4 access-list log-update threshold

To specify the number of updates that are logged for IPv4 access lists, use the **ipv4 access-list log-update threshold** command in Global Configuration mode. To return the number of logged updates to the default setting, use the **no** form of this command.

ipv4 access-list log-update threshold *update-number*
no ipv4 access-list log-update threshold *update-number*

Syntax Description

update-number	Number of updates that are logged for every IPv4 access list configured on the router. Range is 0 to 2147483647.

Command Default

For IPv4 access lists, 2147483647 updates are logged.

Command Modes

Global Configuration mode

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

IPv4 access list updates are logged at 5-minute intervals, following the first logged update. Configuring a lower number of updates (a number lower than the default) is useful when more frequent update logging is desired.

Task ID

Task ID	Operations
basic-services	read, write
acl	read, write

Examples

This example shows how to configure a log threshold of ten updates for every IPv4 access list configured on the router:

```
RP/0/RSP0/CPU0:router(config)# ipv4 access-list log-update threshold 10
```

Related Commands

Command	Description
deny (IPv4) , on page 14	Sets the deny conditions for an IPv4 access list.
ipv4 access-list, on page 33	Defines an IPv4 access list and enters IPv4 access list configuration mode.

Command	Description
permit (IPv4) , on page 54	Sets the permit conditions for an IPv4 access list.
show access-lists ipv4 , on page 87	Displays the contents of all current IPv4 access lists.

ipv6 access-group

To control access to an interface, use the **ipv6 access-group** command in interface configuration mode. To remove the specified access group, use the **no** form of this command.

ipv6 access-group *access-list-name* {**ingress** | **egress**} [**interface-statistics**]

Syntax Description

access-list-name	Name of an IPv6 access list as specified by an **ipv6 access-list** command.
ingress	Filters on inbound packets.
egress	Filters on outbound packets.
interface-statistics	(Optional) Specifies per-interface statistics in the hardware.

Command Default

The interface does not have an IPv6 access list applied to it.

Command Modes

Interface configuration

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

The **ipv6 access-group** command is similar to the **ipv4 access-group** command, except that it is IPv6-specific.

Use the **ipv6 access-group** command to control access to an interface. To remove the specified access group, use the **no** form of the command. Use the *access-list-name* to specify a particular IPv6 access list. Use the **ingress** keyword to filter on inbound packets or the **egress** keyword to filter on outbound packets.

Note For packet filtering applications using the **ipv6 access-group** command, packet counters are maintained in hardware for each direction. If an access group is used on multiple interfaces in the same direction, then packets are counted for each interface.

If the access list permits the addresses, the software continues to process the packet. If the access list denies the address, the software discards the packet and returns a rate-limited Internet Control Message Protocol (ICMP) host unreachable message.

If the specified access list does not exist, all packets are passed.

By default, the unique or per-interface ACL statistics are disabled.

Task ID

Task ID	Operations
acl	read, write

Task ID	Operations
ipv6	read, write

Examples

The following example shows how to apply filters on packets inbound and outbound from GigabitEthernet interface 0/2/0/2:

```
RP/0/RSP0

/CPU0:router(config)# interface gigabitethernet 0/2/0/2
RP/0/RSP0

/CPU0:router(config-if)# ipv6 access-group p-in-filter ingress
RP/0/RSP0

/CPU0:router(config-if)# ipv6 access-group p-out-filter egress
```

The following example shows how to apply per-interface statistics in the hardware:

```
RP/0/RSP0

/CPU0:router(config)# interface gigabitethernet 0/2/0/2
RP/0/RSP0

/CPU0:router(config-if)# ipv6 access-group p-in-filter ingress interface-statistics
```

Related Commands

Command	Description
ipv6 access-list(BNG)	Defines an IPv6 access list and enters IPv6 access list configuration mode.
show ipv6 interface	Displays the usability status of interfaces configured for IPv6.

ipv6 access-list

To define an IPv6 access list and to place the router in IPv6 access list configuration mode, use the **ipv6 access-list** command in Global Configuration mode. To remove the access list, use the **no** form of this command.

ipv6 access-list *name*

Syntax Description

name	Name of the IPv6 access list. Names cannot contain a space or quotation mark, or begin with a numeric.

Command Default

No IPv6 access list is defined.

Command Modes

Global Configuration mode

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

The **ipv6 access-list** command is similar to the **ipv4 access-list** command, except that it is IPv6-specific.

The IPv6 access lists are used for traffic filtering based on source and destination addresses, IPv6 option headers, and optional, upper-layer protocol type information for finer granularity of control. IPv6 access lists are defined by using the **ipv6 access-list** command in global configuration mode and their permit and deny conditions are set by using the **deny** and **permit** commands in IPv6 access list configuration mode. Configuring the **ipv6 access-list** command places the router in IPv6 access list configuration mode—the router prompt changes to router (config-ipv6-acl)#. From IPv6 access list configuration mode, permit and deny conditions can be set for the defined IPv6 access list.

Note No more than one IPv6 access list can be applied to an interface per direction.

Note Every IPv6 access list has an implicit **deny ipv6 any any** statement as its last match condition. An IPv6 access list must contain at least one entry for the implicit **deny ipv6 any any** statement to take effect.

Note IPv6 prefix lists, not access lists, should be used for filtering routing protocol prefixes.

Use the **ipv6 access-group** interface configuration command with the *access-list-name* argument to apply an IPv6 access list to an IPv6 interface.

Note Every IPv6 ACL has implicit **permit icmp any any nd-na** , **permit icmp any any nd-ns** , and **deny ipv6 any any** statements as its last match conditions. (The former two match conditions allow for ICMPv6 neighbor discovery.) An IPv6 ACL must contain at least one entry for the implicit **deny ipv6 any any** statement to take effect. **permit icmp any any nd-na permit icmp any any nd-ns deny ipv6 any any deny ipv6 any any**

The IPv6 neighbor discovery process makes use of the IPv6 network layer service; therefore, by default, IPv6 ACLs implicitly allow IPv6 neighbor discovery packets to be sent and received on an interface. In IPv4, the Address Resolution Protocol (ARP), which is equivalent to the IPv6 neighbor discovery process, makes use of a separate data link layer protocol; therefore, by default, IPv4 ACLs implicitly allow ARP packets to be sent and received on an interface.

Task ID

Task ID	Operations
acl	read, write
ipv6	read, write

Examples

The following example shows how to configure the IPv6 access list named list2 and applies the ACL to outbound traffic on interface GigabitEthernet 0/2/0/2. Specifically, the first ACL entry keeps all packets from the network fec0:0:0:2::/64 (packets that have the site-local prefix fec0:0:0:2 as the first 64 bits of their source IPv6 address) from exiting out of interface GigabitEthernet 0/2/0/2. The second entry in the ACL permits all other traffic to exit out of interface GigabitEthernet 0/2/0/2. The second entry is necessary because an implicit deny all condition is at the end of each IPv6 ACL.

```
RP/0/RSP0

/CPU0:router(config)# ipv6 access-list list2
RP/0/RSP0

/CPU0:router(config-ipv6-acl)# 10 deny fec0:0:0:2::/64 any
RP/0/RSP0

/CPU0:router(config-ipv6-acl)# 20 permit any any

RP/0/RSP0

/CPU0:router# show ipv6 access-lists list2

ipv6 access-list list2
  10 deny ipv6 fec0:0:0:2::/64 any
  20 permit ipv6 any any

RP/0/RSP0

/CPU0:router(config)# interface gigabitethernet 0/2/0/2
RP/0/RSP0

/CPU0:router(config-if)# ipv6 access-group list2 out
```

Note IPv6 is automatically configured as the protocol type in **permit any any** and **deny any any** statements that are translated from global configuration mode to IPv6 access list configuration mode.

Note An IPv6 router does not forward to another network an IPv6 packet that has a link-local address as either its source or destination address (and the source interface for the packet is different from the destination interface for the packet).

ipv6 access-list log-update rate

To specify the rate at which IPv6 access lists are logged, use the **ipv6 access-list log-update rate** command in Global Configuration mode. To return the update rate to the default setting, use the **no** form of this command.

ipv6 access-list log-update rate *rate-number*
no ipv6 access-list log-update rate *rate-number*

Syntax Description

rate-number	Rate at which IPv6 access hit logs are generated per second on the router. Range is 1 to 1000.

Command Default

Default is 1.

Command Modes

Global Configuration mode

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

The *rate-number* argument applies to all the IPv6 access-lists configured on the interfaces. That is, at any given time there can be between 1 and 1000 log entries for the system.

Task ID

Task ID	Operations
ipv6	read, write
acl	read, write

Examples

This example shows how to configure a IPv6 access hit logging rate for the system:

```
RP/0/RSP0/CPU0:router(config)# ipv6 access-list log-update rate 10
```

ipv6 access-list log-update threshold

To specify the number of updates that are logged for IPv6 access lists (ACLs), use the **ipv6 access-list log-update threshold** command in Global Configuration mode. To return the number of logged updates to the default setting, use the **no** form of this command.

ipv6 access-list log-update threshold *update-number*
no ipv6 access-list log-update threshold *update-number*

Syntax Description

update-number	Number of updates that are logged for every IPv6 access list configured on the router. Range is 0 to 2147483647.

Command Default

For IPv6 access lists, 350000 updates are logged.

Command Modes

Global Configuration mode

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

The **ipv6 access-list log-update threshold** command is similar to the **ipv4 access-list log-update threshold** command, except that it is IPv6-specific.

IPv6 access list updates are logged at 5-minute intervals, following the first logged update. Configuring a lower number of updates (a number lower than the default) is useful when more frequent update logging is desired.

Task ID

Task ID	Operations
acl	read, write
ipv6	read, write

Examples

This example shows how to configure a log threshold of ten updates for every IPv6 access list configured on the router:

```
RP/0/RSP0/CPU0:router(config)# ipv6 access-list log-update threshold 10
```

ipv6 access-list maximum ace threshold

To set the maximum number of access control entries (ACEs) for IPv6 access lists, use the **ipv6 access-list maximum ace threshold** command in Global Configuration mode. To reset the ACE limit for IPv6 access lists, use the **no** form of this command.

ipv6 access-list maximum ace threshold *ace-number*
no ipv6 access-list maximum ace threshold *ace-number*

Syntax Description

ace-number	Maximum number of configurable ACEs allowed. Range is 50000 to 350000.

Command Default

50,000 ACEs are allowed for IPv6 access lists.

Command Modes

Global Configuration mode

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

Use the **ipv6 access-list maximum ace threshold** command to set the maximum number of configurable ACEs for IPv6 access lists. Out of resource (OOR) limits the number of ACEs that can be configured in the system. When the maximum number of configurable ACEs is reached, configuration of new ACEs is rejected.

Task ID

Task ID	Operations
acl	read, write
ipv6	read, write

Examples

This example shows how to set the maximum number of ACEs for IPv6 access lists to 75000:

```
RP/0/RSP0/CPU0:router(config)# ipv6 access-list maximum ace threshold 75000
```

Related Commands

Command	Description
show access-lists ipv6, on page 93	Displays the contents of all current IPv6 access lists.

ipv6 access-list maximum acl threshold

To set the maximum number of configurable IPv6 access control lists (ACLs), use the **ipv6 access-list maximum acl threshold** command in Global Configuration mode. To reset the IPv6 ACL limit, use the **no** form of this command.

ipv6 access-list maximum acl threshold *acl-number*
no ipv6 access-list maximum ace threshold *acl-number*

Syntax Description

acl-number	Maximum number of configurable ACLs allowed. Range is 1000 to 16000.

Command Default

1000 IPv6 ACLs can be configured.

Command Modes

Global Configuration mode

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

Use the **ipv6 access-list maximum acl threshold** command to set the maximum number of configurable IPv6 ACLs. Out of resource (OOR) limits the number of ACLs that can be configured in the system. When the limit is reached, configuration of new ACLs is rejected.

Task ID

Task ID	Operations
acl	read, write
ipv6	read, write

Examples

This example shows how to set the maximum number of configurable IPv6 ACLs to 1500:

```
RP/0/RSP0/CPU0:router(config)# ipv6 access-list maximum acl threshold 1500
```

Related Commands

Command	Description
show access-lists ipv6, on page 93	Displays the contents of all current IPv6 access lists.

interface ipv4/ipv6 access-group

To configure an interface to accept multiple IPv4 or IPv6 ACLs, use the **interface ipv4/ipv6 access-group** command in Global Configuration mode.

interface *type interface-path-id* [**ipv4 | ipv6**] **access-group common** *acl-c1* **common** *acl-c2 acl-i2 acl-i4 acl-i5* **ingress**

Syntax Description

type	Interface type. For more information, use the question mark (?) online help function.
interface -path-id	Physical interface or virtual interface. Use the **show interfaces** command to see a list of all interfaces currently configured on the router.
common *acl-c1* **common** *acl-c2*	Common ACLs, each preceded by the keyword **common**. Common ACLs are only supported in the ingress direction.
acl-i2 acl-i4 acl-i5	Interface ACLs.
ingress	Specifies an inbound direction.

Command Default

The interface does not have an IPv4/IPv6 access list applied to it.

Command Modes

Global Configuration mode

Command History

Release	Modification
Release 5.3.2	This command was introduced.

Usage Guidelines

Use the **interface ipv4/ipv6 access-group** command to configure an interface on Cisco ASR 9000 High Density 100GE Ethernet line cards (such as A9K-8x100G-LB-SE and A9K-8x100G-LB-TR) to accept up to five IPv4 and/or IPv6 ACLs in the ingress direction only. There can be any combination of common and/or interface ACLs up to a total of five ACLs.

Task ID

Task ID	Operation
acl	read, write
network	read, write
config-services	read, write

The following example shows how to apply filters on packets inbound from GigabitEthernet interface 0/1/0/0:

```
RP/0/RSP0/CPU0:router#  interface GigabitEthernet 0/1/0/0
```

```
ipv4 access-group common acl_c1 common acl_c2 acl_i2 acl_i4 acl_i5 ingress
```

The following example shows a sample configuration of multiple ACLs:

```
RP/0/RSP0/CPU0:router#  show running-config interface tenGigE 0/1/0/0/0 interface
TenGigE0/1/0/0/0
 ipv4 address 10.1.1.2 255.255.255.0
 ipv6 address 2001::33/64
 ipv4 access-group common acl_c1 common acl_c2 acl_i2 acl_i4 acl_i5 ingress
!
```

object-group network

To configure a network object group, and to enter the network object group configuration mode, use the **object-group network** command in the global configuration mode. To de-configure the network object group, use the **no** form of this command.

object-group network { **ipv4 | ipv6** } *object-group-name*
no object-group network { **ipv4 | ipv6** } *object-group-name*

Syntax Description

ipv4	Configures the operation state of an IPV4 network object group.
ipv6	Configures the operation state of an IPV6 network object group.
object-group-name	Name of the object-group.

Command Default

None

Command Modes

Global configuration

Command History

Release	Modification
Release 4.3.1	This command was introduced.
Release 5.3.0	The object-group feature can be configured along with ABF while defining an ACEs (Access Control Entry).

Usage Guidelines

Object-group is only supported on ASR 9000 Enhanced Ethernet Line Card.

Inherited object-groups up to four levels are supported in this release.

If an ACL is applied on an interface with non-zero compression level (implying it contains no ABF ACEs), a user cannot add an ACE with object-group.

Task ID

Task ID	Operation
system	read, write

Example

This example shows how to configure a network object-group, and to enter the network object-group configuration mode:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# object-group network ipv4 ipv4_type5_obj1
RP/0/RSP0/CPU0:router(config-object-group-ipv4)#
```

Related Commands

Command	Description
show object-group network, on page 98	Displays the operation state of a network object group.

object-group port

To configure a port object group, and to enter the port object group configuration mode, use the **object-group port** command in the global configuration mode. To de-configure the port object group, use the **no** form of this command.

object-group port *object-group-name*
no object-group port *object-group-name*

Syntax Description

object-group-name	Name of the object-group.

Command Default

None

Command Modes

Global configuration

Command History

Release	Modification
Release 4.3.1	This command was introduced.
Release 5.3.0	The ABF feature can be configured along with object-groups while defining an ACEs (Access Control Entry).

Usage Guidelines

Object-group is only supported on ASR 9000 Enhanced Ethernet Line Card.

Inherited object-groups upto four levels are supported in this release.

Note If any ACE in an ACL contains ABF clause, this ACL cannot be applied at any non-zero compression level.

Task ID

Task ID	Operation
system	read, write

Example

This example show how to configure a port object-group, and to enter the port object-group configuration mode:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# object-group port ipv4_type5_obj1
RP/0/RSP0/CPU0:router(config-object-group-port)#
```

Related Commands

Command	Description
show object-group port , on page 100	Displays the operation state of a port object group.

packet-length

Enables filtering of packets at an ingress/egress interface by specifying the packet length as a match condition in a IPv4/IPv6 ACL.

By using the **packet-length** condition in an ACL, IPv4 and IPv6 packets are either processed (permit statement) or dropped (deny statement).

To remove this configuration, use the **no** prefix for the command.

packet-length { **eq** *value* | **gt** *value* | **lt** *value* | **neq** *value* | **range** *lower-limit upper-limit* }

Syntax Description

packet-length eq *value*	Filters packets that have a packet length equal to the specified limit.
packet-length gt *value*	Filters packets that have a packet length greater than the specified limit.
packet-length lt *value*	Filters packets that have a packet length less than the specified limit.
packet-length neq *value*	Filters packets that have a packet length that does not match the specified limit.
packet-length range *lower-limit upper-limit*	Filters packets that have a packet length within the specified range. The IPv4/IPv6 packet length ranges from 0 to 65535.

Command Default

None

Command Modes

Access List Configuration mode

Release	Modification
Release 6.2.1	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Example

The following example shows how you can configure an IPv4 access list with the **packet-length** condition.

```
RP/0/RSP0/CPU0:router# config
RP/0/RSP0/CPU0:router(config)# ipv4 access-list pktlen-v4
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit tcp any any packet-length eq 1482
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 permit udp any any packet-length range 1400 1500
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 30 deny ipv4 any any
```

The following example shows how you can configure an IPv6 access list with the **packet-length** condition.

```
RP/0/RSP0/CPU0:router# config
RP/0/RSP0/CPU0:router(config)# ipv6 access-list pktlen-v6
RP/0/RSP0/CPU0:router(config-ipv6-acl)# 10 permit tcp any any packet-length eq 1500
```

```
RP/0/RSP0/CPU0:router(config-ipv6-acl)# 20 permit udp any any packet-length range 1500 1600
RP/0/RSP0/CPU0:router(config-ipv6-acl)# 30 deny ipv6 any any
```

For a complete configuration example, see the Configure an ACL to Filter By Packet Length section in the *Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Configuration Guide*.

permit (IPv4)

To set conditions for an IPv4 access list, use the **permit** command in access list configuration mode. There are two versions of the **permit** command: **permit** (source), and **permit** (protocol). To remove a condition from an access list, use the **no** form of this command.

[*sequence-number*] **permit** *source* [*source-wildcard*] [{**log** | **log-input**}]
[*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**capture**] [**precedence** *precedence*] [**default nexthop** [*ipv4-address1*] [*ipv4-address2*] [*ipv4-address3*]] [**dscp** *dscp*] [**fragments**] [{**log** | **log-input**}] [**nexthop** [**track** *track-name*] [*ipv4-address1*] [*ipv4-address2*] [*ipv4-address3*]] [**ttl** *ttl value* [*value1* **. . .** *value2*]][**counter** *counter-name*]
[*sequence-number*] **permit** *protocol* **net-group** *source-net-object-group-name* **port-group** *source-port-object-group-name* **net-group** *destination-net-object-group-name* **port-group** *destination-port-object-group-name* [**capture**] [**precedence** *precedence*] [**default nexthop1** [**vrf** *vrf-name*][**ipv4** *ipv4-address1*] **nexthop2**[**vrf** *vrf-name*][**ipv4** *ipv4-address2*] **nexthhop3** [**vrf** *vrf-name*][**ipv4** *ipv4-address3*]] [**dscp range** *dscp dscp*] [**fragments**] [{**log** | **log-input**}] [**nexthop** [**track** *track-name*]] [**ttl** *ttl value* [*value1* **. . .** *value2*]][**counter** *counter-name*]
no *sequence-number*

Internet Control Message Protocol (ICMP)
[*sequence-number*] **permit icmp** *source source-wildcard destination destination-wildcard* [*icmp-type*] [*icmp-code*] [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [{**log** | **log-input**}] [**icmp-off**][**counter** *counter-name*]

Internet Group Management Protocol (IGMP)
[*sequence-number*] **permit igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**dscp** *value*] [**fragments**] [{**log** | **log-input**}][**counter** *counter-name*]

User Datagram Protocol (UDP)
[*sequence-number*] **permit udp** *source source-wildcard* [*operator* {*portprotocol-port*}] *destination destination-wildcard* [*operator* {*portprotocol-port*}] [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [{**log** | **log-input**}][**counter** *counter-name*]

Syntax Description

sequence-number	(Optional) Number of the **permit** statement in the access list. This number determines the order of the statements in the access list. Range is 1 to 2147483644. (By default, the first statement is number 10, and the subsequent statements are incremented by 10.) Use the **resequence access-list** command to change the number of the first statement and increment subsequent statements of a configured access list.

source	Number of the network or host from which the packet is being sent. There are three alternative ways to specify the source: • Use a 32-bit quantity in four-part dotted-decimal format. • Use the **any** keyword as an abbreviation for a *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. • Use the **host** *source* combination as an abbreviation for a *source* and *source-wildcard* of *source* 0.0.0.0.
source-wildcard	Wildcard bits to be applied to the source. There are three alternative ways to specify the source wildcard: • Use a 32-bit quantity in four-part dotted-decimal format. Place ones in the bit positions you want to ignore. • Use the **any** keyword as an abbreviation for a *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. • Use the **host** *source* combination as an abbreviation for a *source* and *source-wildcard* of *source* 0.0.0.0.
protocol	Name or number of an IP protocol. It can be one of the keywords **ahp** , **esp** , **eigrp** , **gre** , **icmp** , **igmp** , **igrp** , **ip** , **ipinip** , **nos** , **ospf** , **pim** , **pcp** , **tcp** , or **udp** , or an integer from 0 to 255 representing an IP protocol number. To match any Internet protocol (including ICMP, TCP, and UDP), use the **ip** keyword. ICMP, and TCP allow further qualifiers, which are described later in this table.

destination	Number of the network or host to which the packet is being sent. There are three alternative ways to specify the destination: • Use a 32-bit quantity in four-part dotted-decimal format. • Use the **any** keyword as an abbreviation for the *destination* and *destination-wildcard* of 0.0.0.0 255.255.255.255. • Use the **host** *destination* combination as an abbreviation for a *destination* and *destination-wildcard* of *destination* 0.0.0.0.
destination-wildcard	Wildcard bits to be applied to the destination. There are three alternative ways to specify the destination wildcard: • Use a 32-bit quantity in four-part dotted-decimal format. Place ones in the bit positions you want to ignore. • Use the **any** keyword as an abbreviation for a *destination* and *destination-wildcard* of 0.0.0.0 255.255.255.255. • Use the **host** *destination* combination as an abbreviation for a *destination* and *destination-wildcard* of *destination* 0.0.0.0.
net-group *source-net-object-group-name*	IPv4 source network object group and group name.
port-group *source-port-object-group-name*	Source port object group and group name.
net-group*destination-net-object-group-name*	IPv4 destination network object group and group name.
port-group *destination-port-object-group-name*	Destination port object group and group name.

precedence *precedence*	(Optional) Packets can be filtered by precedence level (as specified by a number from 0 to 7) or by the following names: • **Routine** —Match packets with routine precedence (0) • **priority** —Match packets with priority precedence (1) • **immediate** —Match packets with immediate precedence (2) • **flash** —Match packets with flash precedence (3) • **flash-override** —Match packets with flash override precedence (4) • **critical** —Match packets with critical precedence (5) • **internet** —Match packets with internetwork control precedence (6) • **network** —Match packets with network control precedence (7)
default	(Optional) Specifies the default next hop for this entry. If the **default** keyword is configured, ACL-based forwarding action is taken only if the results of the PLU lookup for the destination of the packets determine a default route; that is, no specified route is determined to the destination of the packet.
capture	Captures matching traffic. When the acl command is configured on the source mirroring port, if the ACL configuration command does not use the **capture** keyword, no traffic gets mirrored. If the ACL configuration uses the **capture** keyword, but the acl command is not configured on the source port, then the whole port traffic is mirrored and the **capture** action does not have any affect.

ipv4-address1 ipv4-address2 ipv4-address3	(Optional) Uses one to three next-hop addresses. The IP address types are defined as follows: • Default IP addresses—Specifies the next-hop router in the path toward the destination in which the packets must be forwarded, if there is no explicit route for the destination address of the packet in the routing table. The first IP address that is associated with a connected interface that is currently up is used to route the packets. • Specified IP addresses—Specifies the next-hop router in the path toward the destination in which the packets must be forwarded. The first IP address that is associated with a connected interface that is currently up is used to route the packets.

dscp *dscp*	(Optional) Differentiated services code point (DSCP) provides quality of service control. The values for *dscp* are as follows: • 0–63—Differentiated services codepoint value • af11—Match packets with AF11 dscp (001010) • af12—Match packets with AF12 dscp (001100) • af13—Match packets with AF13 dscp (001110) • af21—Match packets with AF21 dscp (010010) • af22—Match packets with AF22 dscp (010100) • af23—Match packets with AF23 dscp (010110) • af31—Match packets with AF31 dscp (011010) • af32—Match packets with AF32 dscp (011100) • af33—Match packets with AF33 dscp (011110) • af41—Match packets with AF41 dscp (100010) • af42—Match packets with AF42 dscp (100100) • af43–Match packets with AF43 dscp (100110) • cs1—Match packets with CS1 (precedence 1) dscp (001000) • cs2—Match packets with CS2 (precedence 2) dscp (010000) • cs3—Match packets with CS3 (precedence 3) dscp (011000) • cs4—Match packets with CS4 (precedence 4) dscp (100000) • cs5—Match packets with CS5 (precedence 5) dscp (101000) • cs6—Match packets with CS6 (precedence 6) dscp (110000) • cs7—Match packets with CS7 (precedence 7) dscp (111000) • default—Default DSCP (000000) • ef—Match packets with EF dscp (101110)

dscp range *dscp dscp*	(Optional) Differentiated services code point (DSCP) provides quality of service control. The values for *dscp* are as follows: • 0–63—Differentiated services codepoint value • af11—Match packets with AF11 dscp (001010) • af12—Match packets with AF12 dscp (001100) • af13—Match packets with AF13 dscp (001110) • af21—Match packets with AF21 dscp (010010) • af22—Match packets with AF22 dscp (010100) • af23—Match packets with AF23 dscp (010110) • af31—Match packets with AF31 dscp (011010) • af32—Match packets with AF32 dscp (011100) • af33—Match packets with AF33 dscp (011110) • af41—Match packets with AF41 dscp (100010) • af42—Match packets with AF42 dscp (100100) • af43–Match packets with AF43 dscp (100110) • cs1—Match packets with CS1 (precedence 1) dscp (001000) • cs2—Match packets with CS2 (precedence 2) dscp (010000) • cs3—Match packets with CS3 (precedence 3) dscp (011000) • cs4—Match packets with CS4 (precedence 4) dscp (100000) • cs5—Match packets with CS5 (precedence 5) dscp (101000) • cs6—Match packets with CS6 (precedence 6) dscp (110000) • cs7—Match packets with CS7 (precedence 7) dscp (111000) • default—Default DSCP (000000) • ef—Match packets with EF dscp (101110)

fragments	(Optional) Causes the software to examine noninitial fragments of IPv4 packets when applying this access list entry. When this keyword is specified, fragments are subject to the access list entry.
log	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the **logging console** command.) The message includes the access list number, whether the packet was permitted or denied; the protocol, whether it was TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and source and destination port numbers. The message is generated for the first packet that matches a flow, and then at 5-minute intervals, including the number of packets permitted or denied in the prior 5-minute interval.
log-input	(Optional) Provides the same function as the **log** keyword, except that the logging message also includes the input interface.
nexthop1, nexthop2, nexthop3	(Optional) Forwards the specified next hop for this entry.
track *track-name*	Specifies the TRACK Name for this nexthop.
ttl	(Optional) Turns on matching against time-to-life (TTL) value.
ttl value [*value1 ... value2*]	(Optional) TTL value used for filtering. Range is 1 to 255. If only *value* is specified, the match is against this value. If both *value1* and *value2* are specified, the packet TTL is matched against the range of TTLs between *value1* and *value2* .

icmp-off	(Optional) Turns off ICMP generation for denied packets
icmp-type	(Optional) ICMP message type for filtering ICMP packets. Range is from 0 to 255.
icmp-code	(Optional) ICMP message code for filtering ICMP packets. Range is from 0 to 255.
igmp-type	(Optional) IGMP message type (0 to 15) or message name for filtering IGMP packets, as follows: • dvmrp • host-query • host-report • mtrace • mtrace-response • pim • precedence • trace • v2-leave • v2-report • v3-report
operator	(Optional) Operator is used to compare source or destination ports. Possible operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range). If the operator is positioned after the *source* and *source-wildcard* values, it must match the source port. If the operator is positioned after the *destination* and *destination-wildcard* values, it must match the destination port. If the operator is positioned after the **ttl** keyword, it matches the TTL value. The **range** operator requires two port numbers. All other operators require one port number.

port	Decimal number a TCP or UDP port. Range is 0 to 65535. TCP ports can be used only when filtering TCP. UDP ports can be used only when filtering UDP.
protocol-port	Name of a TCP or UDP port. TCP and UDP port names are listed in the “Usage Guidelines” section. TCP port names can be used only when filtering TCP. UDP port names can be used only when filtering UDP.
established	(Optional) For the TCP protocol only: Indicates an established connection.
match-any	(Optional) For the TCP protocol only: Filters on any combination of TCP flags.
match-all	(Optional) For the TCP protocol only: Filters on all TCP flags.
+ | -	(Required) For the TCP protocol **match-any** , **match-all** : Prefix *flag-name* with + or - . Use the + *flag-name* argument to match packets with the TCP flag set. Use the - *flag-name* argument to match packets when the TCP flag is not set.
flag-name	(Optional) For the TCP protocol **match-any** , **match-all** . Flag names are: **ack** , **fin** , **psh** , **rst** , **syn** .
counter	(Optional) Enables accessing ACL counters using SNMP query. The **counter** *counter-name* keyword is available on Cisco ASR 9000 Enhanced Ethernet Line Cards only.
counter-name	Defines an ACL counter name.

Command Default

There is no specific condition under which a packet is denied passing the IPv4 access list.

ICMP message generation is enabled by default.

Command Modes

IPv4 access list configuration

Command History

Release	Modification
Release 3.7.2	This command was introduced.
Release 4.0.1	The **capture** keyword was added.
Release 4.3.1	The **range** keyword for dscp and **net-group** and **port-group** keywords were added.
Release 5.1.1	The optional keyword **counter** and the associated argument *counter-name* were added to the command.
Release 5.3.0	The ABF feature can be configured along with object-groups while defining an ACEs (Access Control Entry).

Usage Guidelines

Use the **permit** command following the **ipv4 access-list** command to specify conditions under which a packet can pass the access list.

By default, the first statement in an access list is number 10, and the subsequent statements are incremented by 10.

You can add **permit**, **deny**, or **remark** statements to an existing access list without retyping the entire list. To add a new **s**tatement anywhere other than at the end of the list, create a new statement with an appropriate entry number that falls between two existing entry numbers to indicate where it belongs.

If you want to add a statement between two consecutively numbered statements (for example, between lines 10 and 11), first use the **resequence access-list** command to renumber the first statement and increment the entry number of each subsequent statement. The *increment* argument causes new, unused line numbers between statements. Then add a new statement with the *entry-number* specifying where it belongs in the access list.

Note If any ACE in an ACL contains ABF clause, this ACL cannot be applied at any non-zero compression level.

The following is a list of precedence names:

- critical
- flash
- flash-override
- immediate
- internet
- network
- priority
- routine

The following is a list of ICMP message type names:

- administratively-prohibited
- alternate-address
- conversion-error
- dod-host-prohibited
- dod-net-prohibited

- echo
- echo-reply
- general-parameter-problem
- host-isolated
- host-precedence-unreachable
- host-redirect
- host-tos-redirect
- host-tos-unreachable
- host-unknown
- host-unreachable
- information-reply
- information-request
- mask-reply
- mask-request
- mobile-redirect
- net-redirect
- net-tos-redirect
- net-tos-unreachable
- net-unreachable
- network-unknown
- no-room-for-option
- option-missing
- packet-too-big
- parameter-problem
- port-unreachable
- precedence-unreachable
- protocol-unreachable
- reassembly-timeout
- redirect
- router-advertisement
- router-solicitation
- source-quench
- source-route-failed
- time-exceeded
- timestamp-reply
- timestamp-request
- traceroute
- ttl-exceeded
- unreachable

The following is a list of TCP port names that can be used instead of port numbers. Refer to the current *Assigned Numbers* RFC to find a reference to these protocols. You can find port numbers corresponding to these protocols by typing a ? in the place of a port number.

- bgp
- chargen
- cmd

- daytime
- discard
- domain
- echo
- exec
- finger
- ftp
- ftp-data
- gopher
- hostname
- ident
- irc
- klogin
- kshell
- login
- lpd
- nntp
- pim-auto-rp
- pop2
- pop3
- smtp
- sunrpc
- tacacs
- talk
- telnet
- time
- uucp
- whois
- www

The following UDP port names can be used instead of port numbers. Refer to the current *Assigned Numbers* RFC to find a reference to these protocols. You can find port numbers corresponding to these protocols by typing a ? in the place of a port number.

- biff
- bootpc
- bootps
- discard
- dnsix
- domain
- echo
- isakmp
- mobile-ip
- nameserver
- netbios-dgm
- netbios-ns
- netbios-ss

- ntp
- pim-auto-rp
- rip
- snmp
- snmptrap
- sunrpc
- syslog
- tacacs
- talk
- tftp
- time
- who
- xdmcp

Use the following flags in conjunction with the **match-any** and **match-all** keywords and the + and - signs to select the flags to display:

- ack
- fin
- psh
- rst
- syn

For example, **match-all** +*ack* +*syn* displays TCP packets with both the ack *and* syn flags set, or **match-any** +*ack* – *syn* displays the TCP packets with the ack set *or* the syn not set.

Options such as nexthop1, nexthop2, nexthop3 are not supported with net-group configurations in an ACE.

Task ID

Task ID	Operations
ipv4	read, write
acl	read, write

Examples

The following example shows how to set a permit condition for an access list named Internetfilter:

```
RP/0/RSP0/CPU0:router(config)# ipv4 access-list Internetfilter
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit 192.168.34.0 0.0.0.255
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 permit 172.16.0.0 0.0.255.255
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 25 permit tcp host 172.16.0.0 eq bgp host
192.168.202.203 range 1300 1400
RP/0/RSP0/CPU0:router(config-ipv4-acl)# deny 10.0.0.0 0.255.255.255
```

This example shows how to configure source and destination net-groups and port-groups in an ACL:

```
RP/0/RSP0/CPU0:router#configure
```

```
RP/0/RSP0/CPU0:router(config)#ipv4 access-list acl1
RP/0/RSP0/CPU0:router(config-ipv4-acl)#10 permit tcp net-group n1 port-group p1 net-group
n2 port-group p2
```

Related Commands

Command	Description
deny (IPv4) , on page 14	Sets the conditions for an IPv4 access list.
ipv4 access-group, on page 30	Filters incoming or outgoing IPv4 traffic on an interface.
ipv4 access-list, on page 33	Defines an IPv4 access list and enters IPv4 access list configuration mode.
remark (IPv4) , on page 78	Inserts a helpful remark about an IPv4 access list entry.
resequence access-list ipv4 , on page 82	Changes the starting entry number of the first statement in an existing IPv4 access list, and the number by which subsequent statements are incremented.
show access-lists ipv4 , on page 87	Displays the contents of all current IPv4 access lists.

permit (IPv6)

To set permit conditions for an IPv6 access list, use the **permit** command in IPv6 access list configuration mode. To remove the permit conditions, use the **no** form of this command.

[sequence-number] **permit** *protocol* {*source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/prefix-length*} [*operator* {*port* | *protocol-port*} capture] [**dscp** *value*] [**routing**] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**] [**ttl** *operator ttl value*]
[**default**] **nexthop1** [**vrf** *vrf-name-1*] [**ipv6** *ipv6-address-1*] [**nexthop2** [**vrf** *vrf-name-2*] [**ipv6** *ipv6-address-2*] [**nexthop3** [**vrf** *vrf-name-3*] [**ipv6** *ipv6-address-3*]]]
counter *counter-name*
[sequence-number] **permit** *protocol* {*source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/prefix-length*} {*source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address* } [*operator* {*port* | *protocol-port*} capture] [**dscp** *value*] [**routing**] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**] [**ttl** *operator ttl value*]
[**default**] **nexthop1**[**track** *track-name-1*] [**vrf** *vrf-name-1*] [**ipv6** *ipv6-address-1*] [**nexthop2**[**track** *track-name-2*] [**vrf** *vrf-name-2*] [**ipv6** *ipv6-address-2*] [**nexthop3**[**track** *track-name-3*] [**vrf** *vrf-name-3*] [**ipv6** *ipv6-address-3*]]]
counter *counter-name*
no *sequence-number*

Internet Control Message Protocol (ICMP)
[sequence-number] **permit icmp** {*source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/prefix-length*} {*source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address* } {*destination-ipv6-prefix/ prefix-length* | *any* | *host destination-ipv6-address ipv6-wildcard-mask/prefix-length*} [*icmp-type*] [*icmp-code*] [**dscp** *value*] [routing] [**authen**] [**destopts**] [**fragments**] [**log**] [**log-input**] [**icmp-off**] [**counter** *counter-name*]

Transmission Control Protocol (TCP)
[sequence-number] **permit tcp** {*source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/prefix-length*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix/ prefix-length* | *any* | *host destination-ipv6-address ipv6-wildcard-mask/prefix-length*} [*operator* {*port* | *protocol* | *port*}] [**dscp** *value*] [**routing**] [**authen**] [**destopts**] [**fragments**] [**established**] {**match-any** | **match-all** | **+** | **-**} [*flag-name*] [**log**] [**log-input**] [**counter** *counter-name*]

User Datagram Protocol (UDP)
[sequence-number] **permit tcp** {*source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/prefix-length*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix/ prefix-length* | *any* | *host destination-ipv6-address ipv6-wildcard-mask/prefix-length*} [*operator* {*port* | *protocol* | *port*}] [**dscp** *value*] [**routing**] [**authen**] [**destopts**] [**fragments**] [**established**] [*flag-name*] [**log**] [**log-input**] [**counter** *counter-name*]

Syntax Description

sequence-number	(Optional) Number of the **permit** statement in the access list. This number determines the order of the statements in the access list. Range is from 1 to 2147483644. (By default, the first statement is number 10, and the subsequent statements are incremented by 10.) Use the **resequence access-list** command to change the number of the first statement and increment subsequent statements of a configured access list.
protocol	Name or number of an Internet protocol. It can be one of the keywords **ahp**, **eigrp**, **esp**, **gre**, **icmp**, **igmp**, **igrp**, **isinip**, **ipv6**, **nos**, **ospf**, **pcp**, **sctp**, **tcp**, or **udp**, or an integer that ranges from 0 to 255, representing an IPv6 protocol number.
source-ipv6-prefix / *prefix-length*	Source IPv6 network or class of networks about which permit conditions are to be set. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.
any	An abbreviation for the IPv6 prefix ::/0.
capture	Captures matching traffic. When the acl command is configured on the source mirroring port, if the ACL configuration command does not use the **capture** keyword, no traffic gets mirrored. If the ACL configuration uses the **capture** keyword, but the acl command is not configured on the source port, then the whole port traffic is mirrored and the **capture** action does not have any effect.

host *source-ipv6-address*	Source IPv6 host address about which to set permit conditions. This *source-ipv6-address* argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
ipv6-wildcard-mask	IPv6 wildcard mask. The IPv6 wildcard mask can take any IPv6 address value which is used instead of prefix length.
vrf *vrf-name*	Specifies VPN routing and forwarding (VRF) instance.
nexthop1, nexthop2, nexthop3	(Optional) Specifies the next hop for this entry.
track *track-name*	Specifies object tracking name for the corresponding next hop.
operator {*port* | *protocol-port*}	(Optional) Operand that compares the source or destination ports of the specified protocol. Operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range). If the operator is positioned after the *source-ipv6-prefix / prefix-length* argument, it must match the source port. If the operator is positioned after the *destination-ipv6-prefix / prefix-length* argument, it must match the destination port. The **range** operator requires two port numbers. All other operators require one port number. The *port* argument is the decimal number of a TCP or UDP port. A port number is a number whose range is from 0 to 65535. The *protocol-port* argument is the name of a TCP or UDP port. TCP port names can be used only when filtering TCP. UDP port names can be used only when filtering UDP.

destination-ipv6-prefix / *prefix-length*	Destination IPv6 network or class of networks about which permit conditions are to be set. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.
host *destination-ipv6-address*	Specifies the destination IPv6 host address about which permit conditions are to be set. This *destination-ipv6-address* argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.
dscp *value*	(Optional) Matches a differentiated services code point (DSCP) value against the traffic class value in the Traffic Class field of each IPv6 packet header. Range is from 0 to 63.
routing	(Optional) Matches source-routed packets against the routing extension header within each IPv6 packet header.
authen	(Optional) Matches if the IPv6 authentication header is present.
destopts	(Optional) Matches if the IPv6 destination options header is present.
fragments	(Optional) Matches noninitial fragmented packets where the fragment extension header contains a nonzero fragment offset. The **fragments** keyword is an option available only if the *operator* [*port-number*] arguments are not specified.

log	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the **logging console** command.) The message includes the access list name and sequence number, and whether the packet is permitted; the protocol, and whether it is TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and source and destination port numbers. The message is generated for the first matching packet, and then at 5-minute intervals, including the number of packets permitted in the prior 5-minute interval.
log-input	(Optional) Provides the same function as the **log** keyword, however, the logging message also includes the input interface.
ttl	(Optional) Turns on matching against time-to-live (TTL) value.
operator	(Optional) Operand that compares the source or destination ports of the specified protocol. Operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range).
ttl value [*value1 value2*]	(Optional) TTL value used for filtering. Range is from 1 to 255. If only *value* is specified, the match is against this value. If both *value1* and *value2* are specified, the packet TTL is matched against the range of TTLs between *value1* and *value2* .
icmp-off	(Optional) Turns off ICMP generation for denied packets.
icmp-type	(Optional) ICMP message type for filtering ICMP packets. Range is from 0 to 255.

icmp-code	(Optional) ICMP message code for filtering ICMP packets. Range is from 0 to 255.
established	(Optional) For the TCP protocol only: Indicates an established connection.
match-any	(Optional) For the TCP protocol only: Filters on any combination of TCP flags.
match-all	(Optional) For the TCP protocol only: Filters on all TCP flags.
+ | -	(Required) For the TCP protocol **match-any** , **match-all** : Prefix *flag-name* with + or - . Use the + *flag-name* argument to match packets with the TCP flag set. Use the - *flag-name* argument to match packets when the TCP flag is not set.
flag-name	(Required) For the TCP protocol **match-any**, **match-all**. Flag names are: **ack**, **fin**, **psh**, **rst**, **syn**.
counter	(Optional) Enables accessing ACL counters using SNMP query. The **counter** *counter-name* keyword is available on Cisco ASR 9000 Enhanced Ethernet Line Cards only.
counter-name	Defines an ACL counter name.

Command Default

No IPv6 access list is defined.

ICMP message generation is enabled by default.

Command Modes

IPv6 access list configuration

Command History

Release	Modification
Release 3.7.2	This command was introduced.
Release 4.0.1	The capture keyword was added.
Release 4.2.0	IPv6 support has been enabled for VRF Aware ABF.
Release 4.2.1	ACL Based Forwarding (ABF) has been enabled for Generic Routing Encapsulation (GRE) tunnel interfaces.

Release	Modification
Release 5.1	The **track** keyword was added.
Release 5.1.1	The optional keyword **counter** and the associated argument *counter-name* were added to the command.
Release 5.2.2	The support for IPv6 wildcard mask with a source and destination address was added.
Release 5.3.0	The ABF feature can be configured along with object-groups while defining an ACEs (Access Control Entry).

Usage Guidelines

The **permit** (IPv6) command is similar to the **permit** (IPv4) command, except that it is IPv6-specific.

Use the **permit** (IPv6) command following the **ipv6 access-list** command to define the conditions under which a packet passes the access list.

Specifying **ipv6** for the *protocol* argument matches against the IPv6 header of the packet.

By default, the first statement in an access list is number 10, and the subsequent statements are numbered in increments of 10.

You can add **permit**, **deny, or remark** statements to an existing access list without retyping the entire list. To add a new statement anywhere other than the end of the list, create a new statement with an appropriate entry number that falls between two existing entry numbers to indicate where it belongs.

Both the *source-ipv6-prefix/prefix-length* and *destination-ipv6-prefix/prefix-length* arguments are used for traffic filtering (the source prefix filters traffic based upon the traffic source; the destination prefix filters traffic based upon the traffic destination).

Note IPv6 prefix lists, and not access lists, should be used for filtering routing protocol prefixes.

The **fragments** keyword is an option available only if the *operator* [*port* | *protocol-port*] arguments are not specified.

ABFv4/ABFv6 for GRE tunnel interface is supported for the A9K-SIP-700 and ASR 9000 Enhanced Ethernet linecards. When ACL is configured under GRE tunnel, the incoming IPv4/IPv6 traffic will be subjected to egress ACL on the encap router. On the decap router de-capsulated packet will be processed using ingress ACL.

For the ASR 9000 Ethernet LC, ABFv4 is supported; ABFv6 is not supported.

About two thousand ACLs per box are supported for GRE tunnels.

Note If any ACE in an ACL contains ABF clause, this ACL cannot be applied at any non-zero compression level.

Task ID

Task ID	Operations
acl	read, write

Examples

This example shows how to configure the IPv6 access list named toCISCO and applies the access list to outbound traffic on GigabitEthernet interface 0/2/0/2. Specifically, the first deny entry in the list keeps all packets that have a destination TCP port number greater than 5000 from exiting out of GigabitEthernet interface 0/2/0/2. The second deny entry in the list keeps all packets that have a source UDP port number less than 5000 from exiting out of GigabitEthernet interface 0/2/0/2. The second deny entry also logs all matches to the console. The first permit entry in the list permits all ICMP packets to exit out of GigabitEthernet interface 0/2/0/2. The second permit entry in the list permits all other traffic to exit out of GigabitEthernet interface 0/2/0/2. The second permit entry is necessary because an implicit deny all condition is at the end of each IPv6 access list.

```
RP/0/RSP0/CPU0:router(config)# ipv6 access-list toCISCO
RP/0/RSP0/CPU0:router(config-ipv6-acl)# deny tcp any any gt 5000
RP/0/RSP0/CPU0:router(config-ipv6-acl)# deny ipv6 any lt 5000 any log
RP/0/RSP0/CPU0:router(config-ipv6-acl)# permit icmp any any
RP/0/RSP0/CPU0:router(config-ipv6-acl)# permit any any
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet  0/2/0/2
RP/0/RSP0/CPU0:router(config-if)# ipv6 access-group tOCISCO out
```

This example shows how to configure the IPv6 access list named v6-abf-acl and applies the access list to inbound traffic on GigabitEthernet interface 0/0/2/0.

```
RP/0/RSP0/CPU0:router(config)# ipv6 access-list  v6-abf-acl
RP/0/RSP0/CPU0:router(config-ipv6-acl)# 10 permit ipv6 any any default nexthop1 vrf vrf_A
ipv6 11::1 nexthop2 vrf vrf_B ipv6 22::2 nexthop3 vrf vrf_C ipv6 33::3
RP/0/RSP0/CPU0:router(config-ipv6-acl)# 20 permit ipv4 any any
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet  0/0/2/0
RP/0/RSP0/CPU0:router(config-if)#  ipv6 access-group v6-abf-acl ingress
```

This example shows how to configure the IPv6 access list named v6-abf-acl and applies the access list to inbound traffic on GRE tunnel interface:

```
RP/0/RSP0/CPU0:router(config)# ipv6 access-list  v6-abf-acl
RP/0/RSP0/CPU0:router(config-ipv6-acl)# 10 permit ipv6 any any default nexthop1 vrf vrf_A
ipv6 11::1 nexthop2 vrf vrf_B ipv6 22::2 nexthop3 vrf vrf_C ipv6 33::3
RP/0/RSP0/CPU0:router(config-ipv6-acl)# 20 permit ipv4 any any
RP/0/RSP0/CPU0:router(config)# interface tunnel-ip 25
RP/0/RSP0/CPU0:router(config-if)#  ipv6 access-group v6-abf-acl ingress
```

This example shows how to configure the IPv6 access list named v6-abf-acl and apply track options:

```
RP/0/RSP0/CPU0:router(config)# ipv6 access-list  v6-abf-acl
RP/0/RSP0/CPU0:router(config-ipv6-acl)# 10 permit ipv6 11::1/10 any nexthop1 track track1
ipv6 1::1 nexthop2 track track2 ipv6 2::2 nexthop3 track track3 ipv6 3::3
```

Related Commands

Command	Description
deny (IPv6) , on page 23	Sets deny conditions for an IPv6 access list.

Command	Description
ipv6 access-list, on page 39	Defines an IPv6 access list and enters IPv6 access list configuration mode.
remark (IPv6) , on page 80	Inserts a helpful remark about an IPv6 access list entry.
resequence access-list ipv6 , on page 84	Changes the starting entry number of the first statement in an existing IPv6 access list, and the number by which subsequent statements are incremented.

remark (IPv4)

To write a helpful comment (remark) for an entry in an IPv4 access list, use the **remark** command in IPv4 access list configuration mode. To remove the remark, use the **no** form of this command.

[*sequence-number*] **remark** *remark*
no *sequence-number*

Syntax Description

sequence-number	(Optional) Number of the **remark** statement in the access list. This number determines the order of the statements in the access list. Range is 1 to 2147483646. (By default, the first statement is number 10; subsequent statements are incremented by 10.)
remark	Comment that describes the entry in the access list, up to 255 characters long.

Command Default

The IPv4 access list entries have no remarks.

Command Modes

IPv4 access list configuration

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

Use the **remark** command to write a helpful comment for an entry in an IPv4 access list. To remove the remark, use the **no** form of this command.

The remark can be up to 255 characters; anything longer is truncated.

If you know the sequence number of the remark you want to delete, you can remove it by entering the **no** *sequence-number* command.

Use the **resequence access-list ipv4** command if you want to add statements to an existing access list and the sequence numbers of consecutive entries do not permit additional statements.

Task ID

Task ID	Operations
ipv4	read, write
acl	read, write

Examples

In the following example, the user1 subnet is not allowed to use outbound Telnet:

```
RP/0/RSP0/CPU0:router(config)# ipv4 access-list telnetting
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 remark Do not allow user1 to telnet out
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 deny tcp host 172.16.2.88 255.255.0.0 any eq
telnet
```

```
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 30 permit icmp any any
RP/0/RSP0/CPU0:router# show ipv4 access-list telnetting

ipv4 access-list telnetting
  0 remark Do not allow user1 to telnet out
  20 deny tcp 172.16.2.88 255.255.0.0 any eq telnet out
  30 permit icmp any any
```

Related Commands

Command	Description
deny (IPv4) , on page 14	Sets the deny conditions for an IPv4 access list.
ipv4 access-list, on page 33	Defines an IPv4 access list and enters IPv4 access list configuration mode.
permit (IPv4) , on page 54	Sets the permit conditions for an IPv4 access list
resequence access-list ipv4 , on page 82	Changes the starting entry number of the first statement in an existing IPv4 access list, and the number by which subsequent statements are incremented.
show access-lists ipv4 , on page 87	Displays the contents of all current IPv4 access lists.

remark (IPv6)

To write a helpful comment (remark) for an entry in an IPv6 access list, use the **remark** command in IPv6 access list configuration mode. To remove the remark, use the **no** form of this command.

[*sequence-number*] **remark** *remark*
no *sequence-number*

Syntax Description

sequence-number	(Optional) Number of the **remark** statement in the access list. This number determines the order of the statements in the access list. Range is 1 to 2147483646. (By default, the first statement is number 10, and the subsequent statements are incremented by 10.)
remark	Comment that describes the entry in the access list, up to 255 characters long.

Command Default

The IPv6 access list entries have no remarks.

Command Modes

IPv6 access list configuration

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

The **remark** (IPv6) command is similar to the **remark** (IPv4) command, except that it is IPv6-specific.

Use the **remark** command to write a helpful comment for an entry in an IPv6 access list. To remove the remark, use the **no** form of this command.

The remark can be up to 255 characters; anything longer is truncated.

If you know the sequence number of the remark you want to delete, you can remove it by entering the **no** *sequence-number* command.

Use the **resequence access-list ipv6** command if you want to add statements to an existing access list and the sequence numbers of consecutive entries do not permit additional statements.

Task ID

Task ID	Operations
acl	read, write

Examples

In this example, a remark is added:

```
RP/0/RSP0/CPU0:router(config)# ipv6 access-list Internetfilter
RP/0/RSP0/CPU0:router(config-ipv6-acl)# 10 permit ipv6 3333:1:2:3::/64 any
RP/0/RSP0/CPU0:router(config-ipv6-acl)# 20 permit ipv6 4444:1:2:3::/64 any
RP/0/RSP0/CPU0:router(config-ipv6-acl)# 30 permit ipv6 5555:1:2:3::/64 any
RP/0/RSP0/CPU0:router(config-ipv6-acl)# 39 remark Block BGP traffic from a given host
RP/0/RSP0/CPU0:router(config-ipv6-acl)# 40 deny tcp host 6666:1:2:3::10 eq bgp host
```

```
7777:1:2:3::20 range 1300 1400
RP/0/RSP0/CPU0:router# show ipv6 access-list Internetfilter

ipv6 access-list Internetfilter
  10 permit ipv6 3333:1:2:3::/64 any
  20 permit ipv6 4444:1:2:3::/64 any
  30 permit ipv6 5555:1:2:3::/64 any
  39 remark Block BGP traffic from a given host
  40 deny tcp host 6666:1:2:3::10 eq bgp host 7777:1:2:3::20 range host 6666:1:2:3::10 eq
bgp host 7777:1:2:3::20 range 1300 1400
```

Related Commands

Command	Description
deny (IPv6) , on page 23	Sets the deny conditions for an IPv6 access list.
ipv6 access-list, on page 39	Defines an IPv6 access list and enters IPv6 access list configuration mode.
permit (IPv6) , on page 69	Sets permit conditions for an IPv6 access list
resequence access-list ipv6 , on page 84	Changes the starting entry number of the first statement in an existing IPv6 access list, and the number by which subsequent statements are incremented.

resequence access-list ipv4

To renumber existing statements and increment subsequent statements to allow a new IPv4 access list statement (**permit**, **deny**, or **remark**) to be added, use the **resequence access-list ipv4** command in EXEC mode.

resequence access-list ipv4 *name* [*base* [*increment*]]

Syntax Description

name	Name of an IPv4 access list.
base	(Optional) Number of the first statement in the specified access list, which determines its order in the access list. Maximum value is 2147483644. Default is 10.
increment	(Optional) Number by which the base sequence number is incremented for subsequent statements. Maximum value is 2147483644. Default is 10.

Command Default

base: 10

increment: 10

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

Use the **resequence access-list ipv4** command to add a **permit**, **deny**, or **remark** statement between consecutive entries in an existing IPv4 access list. Specify the first entry number (the *base*) and the increment by which to separate the entry numbers of the statements. The software renumbers the existing statements, thereby making room to add new statements with the unused entry numbers.

Task ID

Task ID	Operations
acl	read, write

Examples

In this example, suppose you have an existing access list:

```
ipv4 access-list marketing
  1 permit 10.1.1.1
  2 permit 10.2.0.0 0.0.255.255
  3 permit tcp host 10.2.2.2 255.255.0.0 any eq telnet
```

You want to add additional entries in the access list. First you resequence the entries, renumbering the statements starting with number 20 and an increment of 5, and then you have room for four additional statements between each of the existing statements:

```
RP/0/RSP0/CPU0:router# resequence access-list ipv4 marketing 20 5
```

```
RP/0/RSP0/CPU0:router# show access-lists ipv4 marketing

ipv4 access-list marketing
  20 permit 10.1.1.1
  25 permit 10.2.0.0
  30 permit tcp host 10.2.2.2 255.255.0.0 any eq telnet
```

Now you add your new entries.

```
RP/0/RSP0/CPU0:router(config)# ipv4 access-list marketing
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 3 remark Do not allow user1 to telnet out
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 4 deny tcp host 172.16.2.88 255.255.0.0 any eq
telnet
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 29 remark Allow user2 to telnet out
RP/0/RSP0/CPU0:router# show access-lists ipv4 marketing

ipv4 access-list marketing
  3 remark Do not allow user1 to telnet out
  4 deny tcp host 171.69.2.88 255.255.0.0 any eq telnet
  20 permit 10.1.1.1
  25 permit 10.2.0.0
  29 remark Allow user2 to telnet out
  30 permit tcp host 10.2.2.2 255.255.0.0 any eq telnet
```

Related Commands

Command	Description
deny (IPv4) , on page 14	Sets the deny conditions for an IPv4 access list.
ipv4 access-list, on page 33	Defines an IPv4 access list and enters IPv4 access list configuration mode.
permit (IPv4) , on page 54	Sets the permit conditions for an IPv4 access list
remark (IPv4) , on page 78	Inserts a helpful remark about an IPv4 access list.
show access-lists ipv4 , on page 87	Displays the contents of all current IPv4 access lists.

resequence access-list ipv6

To renumber existing statements and increment subsequent statements to allow a new IPv6 access list statement (**permit**, **deny**, or **remark**) to be added, use the **resequence access-list ipv6** command in EXEC mode.

resequence access-list ipv6 *name* [*base* [*increment*]]

Syntax Description

name	Name of an IPv6 access list.
base	(Optional) Number of the first statement in the specified access list, which determines its order in the access list. Maximum value is 2147483646. Default is 10.
increment	(Optional) Number by which the base sequence number is incremented for subsequent statements. Maximum value is 2147483644. Default is 10.

Command Default

base: 10

increment: 10

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

The **resequence access-list ipv6** command is similar to the **resequence access-list ipv4** command, except that it is IPv6 specific.

Use the **resequence access-list ipv6** command to add a **permit**, **deny**, or **remark** statement between consecutive entries in an existing IPv6 access list. Specify the first entry number (the *base*) and the increment by which to separate the entry numbers of the statements. The software renumbers the existing statements, thereby making room to add new statements with the unused entry numbers.

Task ID

Task ID	Operations
acl	read, write

Examples

In the following example, suppose you have an existing access list:

```
ipv6 access-list Internetfilter
  10 permit ipv6 3333:1:2:3::/64 any
  20 permit ipv6 4444:1:2:3::/64 any
  30 permit ipv6 5555:1:2:3::/64 any
```

You want to add additional entries in the access list. First, you resequence the entries, renumbering the statements starting with number 20 and an increment of 5, and then you have room for four additional statements between each of the existing statements:

```
RP/0/RSP0/CPU0:router# resequence access-list ipv6 Internetfilter 20 5
RP/0/RSP0/CPU0:router# show access-lists ipv6 Internetfilter

ipv6 access-list Internetfilter
  20 permit ipv6 3333:1:2:3::/64 any
  25 permit ipv6 4444:1:2:3::/64 any
  30 permit ipv6 5555:1:2:3::/64 any
```

Now you add your new entries.

```
RP/0/RSP0/CPU0:router(config)# ipv6 access-list Internetfilter
RP/0/RSP0/CPU0:router(config-ipv6-acl)# 3 remark Block BGP traffic from a given host
RP/0/RSP0/CPU0:router(config-ipv6-acl)# 4 deny tcp host 6666:1:2:3::10 eq bgp host
7777:1:2:3::20 range 1300 1400
RP/0/RSP0/CPU0:router# show access-lists ipv6 Internetfilter

ipv6 access-list Internetfilter
  3 remark Block BGP traffic from a given host
  4 deny tcp host 6666:1:2:3::10 eq bgp host 7777:1:2:3::20 range 1300 1404 deny tcp host
171.69.2.88 255.255.0.0 any eq telnet
  20 permit ipv6 3333:1:2:3::/64 any
  25 permit ipv6 4444:1:2:3::/64 any
  30 permit ipv6 5555:1:2:3::/64 any
```

Related Commands

Command	Description
deny (IPv6) , on page 23	Sets the deny conditions for an IPv6 access list.
ipv6 access-list, on page 39	Defines an IPv6 access list and enters IPv6 access list configuration mode.
permit (IPv6) , on page 69	Set permit conditions for an IPv6 access list.
remark (IPv6) , on page 80	Inserts a helpful remark about an IPv6 access list entry.

show access-lists afi-all

To display the contents of current IPv4 and IPv6 access lists, use the **show access-lists afi-all** command in EXEC mode.

show access-lists afi-all

Syntax Description

This command has no keywords or arguments.

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.7.2	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
acl	read

Examples

This sample output is from the **show access-lists afi-all** command:

```
RP/0/RSP0/CPU0:router# show access-lists afi-all

ipv4 access-list crypto-1
 10 permit ipv4 65.21.21.0 0.0.0.255 65.6.6.0 0.0.0.255
 20 permit ipv4 192.168.241.0 0.0.0.255 192.168.65.0 0.0.0.255
```

show access-lists ipv4

To display the contents of current IPv4 access lists, use the **show access-lists ipv4** command in EXEC mode.

show access-lists ipv4 [{*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail interface** *type interface-path-id*] [**usage pfilter** {**resource-usage location** *node-id* | **all**}]}]

Syntax Description

access-list-name	(Optional) Name of a particular IPv4 access list. The name cannot contain spaces or quotation marks, but can include numbers.
hardware	(Optional) Identifies the access list as an access list for an interface.
ingress	(Optional) Specifies an inbound interface.
egress	(Optional) Specifies an outbound interface.
interface	(Optional) Displays interface statistics.
type	(Optional) Interface type. For more information, use the question mark (?) online help function.
interface-path-id	Physical interface or virtual interface. **Note** Use the **show interfaces** command to see a list of all interfaces currently configured on the router. For more information about the syntax for the router, use the question mark (?) online help function.
sequence *number*	(Optional) Sequence number of a particular IPv4 access list. Range is 1 to 2147483644.
resource-usage	Displays the TCAM resource usage with compression level.

location *node-id*	(Optional) Location of a particular IPv4 access list. The *node-id* argument is entered in the *rack/slot/module* notation.
summary	(Optional) Displays a summary of all current IPv4 access lists.
sequence-number	(Optional) Sequence number of a particular IPv4 access list. Range is 1 to 2147483644.
maximum	(Optional) Displays the current maximum number of configurable IPv4 access control lists (ACLs) and access control entries (ACEs).
detail interface *type* *interface-path-id*	(Optional) Displays detailed configuration of the ternary content addressable memory (TCAM) manager module of this ACL on the specified interface.
usage	(Optional) Displays the usage of the access list on a given line card.
pfilter	(Optional) Displays the packet filtering usage for the specified line card.
all	(Optional) Displays the location of all the line cards.

Command Default

The default displays all IPv4 access lists.

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.7.2	This command was introduced.
Release 4.3.1	The **resource-usage** keyword was added.
Release 5.3.2	The **detail** keyword requires an interface to be specified.

Usage Guidelines

Use the **show access-lists ipv4** command to display the contents of all IPv4 access lists. To display the contents of a specific IPv4 access list, use the *name* argument. Use the *sequence-number* argument to specify the sequence number of the access list.

Use the **hardware , ingress** or **egress** , and **location** keywords to display the access list hardware contents and counters for all interfaces that use the specified access list in a given direction (ingress or egress). To display the contents of a specific access list entry, use the **sequence** *number* keyword and argument. The access group for an interface must be configured using the **ipv4 access-group** command for access list hardware counters to be enabled.

Use the **show access-lists ipv4 summary** command to display a summary of all current IPv4 access lists. To display a summary of a specific IPv4 access list, use the *name* argument.

Use the **show access-lists ipv4 maximum detail** command to display the OOR details for IPv4 access lists. OOR limits the number of ACLs and ACEs that can be configured in the system. When the limit is reached, configuration of new ACLs or ACEs is rejected.

Use the **show access-list ipv4 usage** command to display a summary of all interfaces and access lists programmed on the specified line card.

Task ID

Task ID	Operations
acl	read

Examples

In the following example, the contents of all IPv4 access lists are displayed:

```
RP/0/RSP0/CPU0:router# show access-lists ipv4

ipv4 access-list 101
  10 deny udp any any eq ntp
  20 permit tcp any any
  30 permit udp any any eq tftp
  40 permit icmp any any
  50 permit udp any any eq domain
ipv4 access-list Internetfilter
  10 permit tcp any 172.16.0.0 0.0.255.255 eq telnet
  20 deny tcp any any
  30 deny udp any 172.18.0.0 0.0.255.255 lt 1024
  40 deny ipv4 any any log
```

In the following example, the contents of an access list named acl_hw_1 are displayed:

```
RP/0/RSP0/CPU0:router# show access-lists ipv4 acl_hw_1 hardware egress location 0/2/cp0

ipv4 access-list acl_hw_1
  10 permit icmp 192.168.36.0 0.0.0.255 any (251 hw matches)
  20 permit ip 172.16.3.0 0.0.255.255 any (29 hw matches)
  30 deny tcp any any (58 hw matches)
```

This table describes the significant fields shown in the display.

Table 1: show access-lists ipv4 hardware Field Descriptions

Field	Description
hw matches	Number of hardware matches.
ACL name	Name of the ACL programmed in hardware.

Field	Description
Sequence Number	Each ACE sequence number is programmed into hardware with all the fields that are corresponding to the values set in ACE.
Grant	Depending on the ACE rule, the grant is set to deny, permit, or both.
Logging	Logging is set to on if ACE uses a log option to enable logs.
Per ace icmp	If Per ace icmp is set to on in the hardware, ICMP is unreachable, is rate-limited, and is generated. The default is set to on.
Hits	Hardware counter for that ACE.

In the following example, a summary of all IPv4 access lists are displayed:

```
RP/0/RSP0/CPU0:router# show access-lists ipv4 summary

ACL Summary:
  Total ACLs configured: 3
  Total ACEs configured: 11
```

This table describes the significant fields shown in the display.

Table 2: show access-lists ipv4 summary Field Descriptions

Field	Description
Total ACLs configured	Number of configured IPv4 ACLs.
Total ACEs configured	Number of configured IPV4 ACEs.

In the following example, the OOR details of the IPv4 access lists are displayed:

```
RP/0/RSP0/CPU0:router# show access-lists ipv4 maximum detail

Default max configurable acls :5000
Default max configurable aces :200000
Current configured acls       :1
Current configured aces       :2
Current max configurable acls :5000
Current max configurable aces :200000
Max configurable acls         :9000
Max configurable aces         :350000
```

This table describes the significant fields shown in the display.

Table 3: show access-lists ipv4 maximum detail Command Field Descriptions

Field	Description
Default max configurable acls	Default maximum number of configurable IPv4 ACLs allowed.
Default max configurable aces	Default maximum number of configurable IPv4 ACEs allowed.
Current configured acls	Number of configured IPv4 ACLs.

Field	Description
Current configured aces	Number of configured IPv4 ACEs.
Current max configurable acls	Configured maximum number of configurable IPv4 ACLs allowed.
Current max configurable aces	Configured maximum number of configurable IPv4 ACEs allowed.
Max configurable acls	Maximum number of configurable IPv4 ACLs allowed.
Max configurable aces	Maximum number of configurable IPv4 ACEs allowed.

This example displays the packet filtering usage for the specified line card:

```
RP/0/RSP0/CPU0:router# show access-lists ipv4 usage pfilter location 0/3/CPU0

Interface : GigabitEthernet0/3/0/1
    Input Common-ACL : ipv4_c_acl  ACL : ipv4_i_acl_1
    Output ACL : ipv4_i_acl_1
```

Note To display the packet filtering usage for bundle interfaces, use the **show access-lists ipv4 usage pfilter location all** command.

This example displays the TCAM resource usage with compression level:

```
RP/0/RSP0/CPU0:router# show access-lists ipv4 acl1 hardware ingress resource-usage location
 0/3/CPU0

ACL compression level : 1
Source field Rules: 3652
Prefixes: 20929
Key Width: 189

Level :  Fields                      TCAM entries    Perf Tradeoff
1        :   S                             3652                low
```

Related Commands

Command	Description
clear access-list ipv4, on page 4	Resets the IPv4 access list match counters.
copy access-list ipv4 , on page 10	Copies an existing IPv4 access list.
deny (IPv4) , on page 14	Sets the deny conditions for an ACE of an IPv4 access list.
ipv4 access-group, on page 30	Filters incoming or outgoing IPv4 traffic on an interface.
ipv4 access-list, on page 33	Defines an IPv4 access list and enters IPv4 access list configuration mode.
permit (IPv4) , on page 54	Sets the permit conditions for an ACE of an IPv4 access list.

Command	Description
remark (IPv4) , on page 78	Inserts a helpful remark about an IPv4 access list entry.
resequence access-list ipv4 , on page 82	Changes the starting entry number of the first statement in an existing IPv4 access list, and the number by which subsequent statements are incremented.

show access-lists ipv6

To display the contents of current IPv6 access lists, use the **show access-lists ipv6** command in EXEC mode.

show access-lists ipv6 [{*access-list-name* **hardware** {**ingress**|**egress**} [**interface** *type interface-path-id*] {**sequence** *number*|**location** *node-id*}|**summary** [*access-list-name*]|*access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage pfilter** {**resource-usage location** *node-id* | **all**}]}]

Syntax Description

access-list-name	(Optional) Name of a particular IPv6 access list. The name cannot contain a spaces or quotation marks, but can include numbers.
hardware	(Optional) Identifies the access list as an access list for an interface.
ingress	(Optional) Specifies an inbound interface.
egress	Specifies an outbound interface.
interface	(Optional) Displays interface statistics.
type	(Optional) Interface type. For more information, use the question mark (?) online help function.
interface-path-id	(Optional) Either a physical interface instance or a virtual interface instance as follows: • Physical interface instance. Naming notation is *rack/slot/module/port* and a slash between values is required as part of the notation. • *rack*: Chassis number of the rack. • *slot*: Physical slot number of the modular services card or line card. • *module*: Module number. A physical layer interface module (PLIM) is always 0. • *port*: Physical port number of the interface. **Note** In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RSP0) and the module is CPU0. Example: interface MgmtEth0/RSP0/CPU0/0. • Virtual interface instance. Number range varies depending on interface type. For more information about the syntax for the router, use the question mark (?) online help function.
sequence *number*	(Optional) Sequence number of a particular IPv6 access list. Range is 1 to 2147483644.
resource-usage	Displays the TCAM resource usage with compression level.
location *node-id*	(Optional) Location of a particular IPv6 access list. The *node-id* argument is entered in the *rack/slot/module* notation.
summary	(Optional) Displays a summary of all current IPv6 access lists.

sequence-number	(Optional) Sequence number of a particular IPv6 access list. Range is 1 to 2147483644.
maximum	(Optional) Displays the current maximum number of configurable IPv6 access control lists (ACLs) and access control entries (ACEs).
detail	(Optional) Displays complete out-of-resource (OOR) details.
usage	(Optional) Displays the usage of the access list on a given line card.
pfilter	(Optional) Displays the packet filtering usage for the specified line card.
all	(Optional) Displays the location of all the line cards.

Command Default

Displays all IPv6 access lists.

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.7.2	This command was introduced.
Release 4.3.1	The **resource-usage** keyword was added.
Release 5.2.2	The show command output was updated to display IPv6 wildcard mask.

Usage Guidelines

The **show access-lists ipv6** command is similar to the **show access-lists ipv4** command, except that it is IPv6 specific.

Use the **show access-lists ipv6** command to display the contents of all IPv6 access lists. To display the contents of a specific IPv6 access list, use the *name* argument. Use the *sequence-number* argument to specify the sequence number of the access list.

Use the **hardware , ingress** or **egress**, and **location** keywords to display the access list hardware contents and counters for all interfaces that use the specified access list in a given direction (ingress or egress). To display the contents of a specific access list entry, use the **sequence** *number* keyword and argument. The access group for an interface must be configured using the **ipv6 access-group** command for access list hardware counters to be enabled.

Use the **show access-lists ipv6 summary** command to display a summary of all current IPv6 access lists. To display a summary of a specific IPv6 access list, use the *name* argument.

Use the **show access-lists ipv6 maximum detail** command to display the OOR details for IPv6 access lists. OOR limits the number of ACLs and ACEs that can be configured in the system. When the limit is reached, configuration of new ACLs or ACEs is rejected.

Use the **show access-list ipv6 ipv4 usage** command to display a summary of all interfaces and access lists programmed on the specified line card.

Task ID

Task ID	Operations
acl	read

Examples

In the following example, the IPv6 ACL is configured with the source IPv6 wildcard mask FF:0:FFFF:AA:20 and the destination wildcard mask 0:FFFF:2233::FFFF, the show command displays these wildcard mask:

```
RP/0/RSP0/CPU0:router# config
RP/0/RSP0/CPU0:router(config)# ipv6  access-list acl1
RP/0/RSP0/CPU0:router(config-ipv6-acl)# permit 1:2::3 FF:0:FFFF:AA:20:: 4:5::6
0:FFFF:2233::FFFF
RP/0/RSP0/CPU0:router(config-ipv6-acl)# commit
RP/0/RSP0/CPU0:router# show run ipv6 access-list
ipv6 access-list ACL1
 10 permit ipv6 1:2::3 ff:0:ffff:aa:20:: 4:5::6 0:ffff:2233::ffff
```

In the following example, the contents of all IPv6 access lists are displayed:

```
RP/0/RSP0/CPU0:router# show access-lists ipv6

ipv6 access-list Internetfilter
  3 remark Block BGP traffic from a given host
  4 deny tcp host 6666:1:2:3::10 eq bgp host 7777:1:2:3::20 range 1300 1404 deny tcp host
171.69.2.88 255.255.0.0 any eq telnet
  20 permit ipv6 3333:1:2:3::/64 any
  25 permit ipv6 4444:1:2:3::/64 any
  30 permit ipv6 5555:1:2:3::/64 any
ipv6 access-list marketing
  10 permit ipv6 7777:1:2:3::/64 any (51 matches)
  20 permit ipv6 8888:1:2:3::/64 any (26 matches)
  30 permit ipv6 9999:1:2:3::/64 any (5 matches)
```

In the following example, the contents of an access list named Internetfilter is displayed:

```
RP/0/RSP0/CPU0:router# show access-lists ipv6 Internetfilter

ipv6 access-list Internetfilter
  3 remark Block BGP traffic from a given host
  4 deny tcp host 6666:1:2:3::10 eq bgp host 7777:1:2:3::20 range 1300 1404 deny tcp host
171.69.2.88 255.255.0.0 any eq telnet
  20 permit ipv6 3333:1:2:3::/64 any
  25 permit ipv6 4444:1:2:3::/64 any
  30 permit ipv6 5555:1:2:3::/64 any
```

In the following example, the contents of an access list named acl_hw_1 is displayed:

```
RP/0/RSP0/CPU0:router# show access-lists ipv6 acl_hw_1 hardware egress location 0/2/cp0

ipv6 access-list acl_hw_1
  10 permit icmp any any (251 hw matches)
  20 permit ipv6 3333:1:2:3::/64 any (29 hw matches)
  30 deny tcp any any (58 hw matches)
```

This table describes the significant fields shown in the display.

Table 4: show access-lists ipv6 hardware Command Field Descriptions

Field	Description
hw matches	Number of hardware matches.

In the following example, a summary of all IPv6 access lists is displayed:

```
RP/0/RSP0/CPU0:router# show access-lists ipv6 summary

ACL Summary:
  Total ACLs configured: 3
  Total ACEs configured: 11
```

This table describes the significant fields shown in the display.

Table 5: show access-lists ipv6 summary Command Field Descriptions

Field	Description
Total ACLs configured	Number of configured IPv6 ACLs.
Total ACEs configured	Number of configured IPV6 ACEs.

In the following example, the OOR details of the IPv6 access lists are displayed:

```
RP/0/RSP0/CPU0:router# show access-lists ipv6 maximum detail

Default max configurable acls :1000
Default max configurable aces :50000
Current configured acls       :1
Current configured aces       :2
Current max configurable acls :1000
Current max configurable aces :50000
Max configurable acls         :2000
Max configurable aces         :100000
```

This example displays the packet filtering usage for the specified line card:

```
RP/0/RSP0/CPU0:router# show access-lists ipv6 usage pfilter location 0/3/CPU0

Interface : GigabitEthernet0/3/0/1
    Input Common-ACL : ipv6_c_acl  ACL : ipv6_i_acl_1
    Output ACL : ipv6_i_acl_1
```

This example displays the TCAM resource usage with compression level:

```
RP/0/RSP0/CPU0:router# show access-lists ipv6 acl1 hardware ingress resource-usage location
 0/0/CPU0

NP                     : 0
Rules (ACE)            : 16
```

```
ACL compression level  : 1
Fields compressed      : SrcIP
TCAM Entries  used     : 383  ( 16k total)
TCAM Key Width         : 640 ( 128 total for compressed fields)
Fields           Prefix count          Bit width/rounded
~~~~~~~          ~~~~~~~~~~~~          ~~~~~~~~~~~~~~~~~~
SourceIP                    43          5/8   (of max 128)
```

Related Commands

Command	Description
copy access-list ipv6, on page 12	Copies an existing IPv6 access list.
deny (IPv6) , on page 23	Sets the deny conditions for an IPv6 access list.
ipv6 access-list, on page 39	Defines an IPv6 access list and enters IPv6 access list configuration mode.
permit (IPv6) , on page 69	Set permit conditions for an IPv6 access list.
remark (IPv6) , on page 80	Inserts a helpful remark about an IPv6 access list entry.
resequence access-list ipv6 , on page 84	Changes the starting entry number of the first statement in an existing IPv4 access list, and the number by which subsequent statements are incremented.

show object-group network

To display the operation state of a network object group, use the **show object-group network** command in EXEC mode.

show object-group network { **ipv4 | ipv6** } *object-group-name*

Syntax Description

ipv4	Displays the operation state of an IPV4 network object group.
ipv6	Displays the operation state of an IPV6 network object group.
object-group-name	Name of the object-group.

Command Default

None

Command Modes

EXEC mode

Command History

Release	Modification
Release 4.3.1	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operation
root-system	read
system	read

Example

This example shows how to display the operation state of an IPV4 network object group:

```
RP/0/RSP0/CPU0:router# show object-group network ipv4 ipv4_type5_obj1

50.0.0.0/16
50.1.0.0/16
50.2.0.0/16
50.3.0.0/16
50.4.0.0/16
host 40.0.0.1
host 40.0.0.2
host 40.0.0.3
host 40.0.0.4
host 40.0.0.5
object-group ipv4_type1_obj1
range 60.0.0.1 60.0.1.100
!
```

This example shows how to display the operation state of an IPV6 network object group:

```
RP/0/RSP0/CPU0:router# show object-group network ipv6 ipv6_type5_obj1

50::/120
50::100/120
50::200/120
50::300/120
50::400/120
host 40::1
host 40::2
host 40::3
host 40::4
host 40::5
object-group ipv6_type2_obj1
range 60::10 60::20
!
```

Related Commands

Command	Description
show object-group port , on page 100	Displays the operation state of a port object group.

show object-group port

To display the operation state of a port object group, use the **show object-group port** command in EXEC mode.

show object-group port *object-group-name*

Syntax Description

object-group-name	Name of the object-group.

Command Default

None

Command Modes

EXEC mode

Command History

Release	Modification
Release 4.3.1	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operation
root-system	read
system	read

Example

This example shows how to display the operation state of a port object group:

```
RP/0/RSP0/CPU0:router# show object-group port port_type4_obj1

object-group port port_type4_obj1
eq 40
object-group port_type1_obj1
range 50 60
!
```

Related Commands

Command	Description
show object-group network, on page 98	Displays the operation state of a network object group.