

Access List Commands

Note All commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router that is introduced from Cisco IOS XR Release 6.3.2. References to earlier releases in Command History tables apply to only the Cisco NCS 5500 Series Router.

Note

- Starting with Cisco IOS XR Release 6.6.25, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 560 Series Routers.
- Starting with Cisco IOS XR Release 6.3.2, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router.
- References to releases before Cisco IOS XR Release 6.3.2 apply to only the Cisco NCS 5500 Series Router.
- Cisco IOS XR Software Release 7.0.1 specific updates are not applicable for the following variants of Cisco NCS 540 Series Routers:
  - N540-28Z4C-SYS-A
  - N540-28Z4C-SYS-D
  - N540X-16Z4G8Q2C-A
  - N540X-16Z4G8Q2C-D
  - N540X-16Z8Q2C-D
  - N540-12Z20G-SYS-A
  - N540-12Z20G-SYS-D
  - N540X-12Z16G-SYS-A
  - N540X-12Z16G-SYS-D

This module describes the Cisco IOS XR software commands used to configure IP Version 4 (IPv4) and IP Version 6 (IPv6) access lists.

For detailed information about ACL concepts, configuration tasks, and examples, refer to the *IP Addresses and Services Configuration Guide for Cisco NCS 5500 Series RoutersIP Addresses and Services Configuration Guide for Cisco NCS 540 Series RoutersIP Addresses and Services Configuration Guide for Cisco NCS 560 Series Routers*.

- acl compress, on page 3
- acl egress layer3 interface-based, on page 5
- acl-permit, on page 6
- acl ipv6 ext-header, on page 8
- acl-prefix percent, on page 10
- clear access-list ipv4, on page 12
- clear access-list ipv6, on page 14
- common-acl, on page 16
- copy access-list ipv4 , on page 17
- copy access-list ipv6, on page 19
- deny (IPv4) , on page 21
- deny (IPv6) , on page 32
- dont-fragment, on page 37
- enable-set-ttl, on page 39
- first-fragment, on page 42
- fragment-offset, on page 43
- fragment-type, on page 44
- hw-module profile acl ipv6 single-pass-egress-acl, on page 46
- interface-based, on page 47
- ipv4 access-group, on page 49
- ipv4 access-list, on page 51
- ipv4 access-list log-update rate , on page 53
- ipv4 access-list log-update threshold , on page 54
- ipv6 access-group, on page 55
- ipv6 access-list, on page 57
- ipv6 access-list log-update rate, on page 60
- ipv6 access-list log-update threshold , on page 61
- ipv6 access-list maximum ace threshold, on page 62
- is-fragment, on page 63
- last-fragment, on page 64
- packet-length, on page 65
- permit (IPv4) , on page 67
- permit (IPv6) , on page 86
- remark (IPv4) , on page 95
- remark (IPv6) , on page 97
- ttl-match, on page 99
- tx-scale-enhanced acl-permit, on page 102
- set qos-group, on page 104
- set ttl, on page 106
- show access-lists afi-all, on page 107
- show access-lists ipv4 , on page 108
- show access-lists ipv6, on page 113

acl compress

To load the compression ACL database profile instead of the ACL database profile, use the **acl {ingress | egress} compress enable** option with the **hw-module** command in the global configuration mode.

hw-module profile acl { { **ingress** | **egress** } **compress enable** [**location** *location*] | **egress layer 3** }

Syntax Description

hw-module	Configures the hardware module.
profile	Configures the profile of the hardware module.
acl ingress	Configures the Ingress ACL profile.
compress enable	Enables the compression ACL database profile on the line card.
location *location*	Configures the location of the ACL.
egress layer3	Configure the egress compress ACL profile for layer3 (L3) traffic.

Command Default

If you do not configure the **acl ingress compress enable** command, the ACL database profile is loaded by default on the Cisco NCS-57B1-5DSE and Cisco NCS-57C3-MODS-SYS routers and, NC57-18DD-SE, and NC57-36H-SE line cards..

Command History

Release	Modification
Release 7.0.2	This command was introduced for the **acl ingress** compress option.

Usage Guidelines for acl ingress compress enable Command

The compression ACL database profile is loaded for the Cisco NCS-57B1-5DSE and Cisco NCS-57C3-MODS-SYS routers and, NC57-18DD-SE, and NC57-36H-SE line cards only after you execute the **acl ingress compress enable** command and reboot the line cards.

Table 1: Task ID

Task ID	Operations
configuration	read, write
root-lr	read, write

Examples

The following example shows you how to configure the **acl ingress compress** command:

```
Router# configure
Router(config)# hw-module profile acl ingress compress enable location 0/6/CPU0
Mon Feb  3 09:35:31.640 PST
In order to activate/deactivate Ingress ACL profile, you must manually reload the chassis/line
 card(s).
Router(config)#commit
Mon Feb  3 09:35:35.355 PST
Router#(config)#exit
Router#reload location 0/6/CPU0
Mon Feb  3 09:36:49.892 PST

Proceed with reload?  [confirm] yes
Router#
```

acl egress layer3 interface-based

To enable a Layer3 ACL over BVI interfaces in the egress direction, use the **acl egress layer3 interface-based** command in the global configuration mode.

hw-module profile acl egress layer3 interface-based

Syntax Description

This command has no keywords or arguments.

Command Default

None

Command Modes

XR Configuration

Command History

Release	Modification
Release 7.3.1	This command was introduced.

Usage Guidelines

Once this profile is enabled, egress ACL will not work on any non-BVI interface.

For this configuration to take effect, you must reload all line cards on the system.

Table 2: Task ID

Task ID	Operations
configuration	read, write
root-lr	read, write

Examples

The following example shows you how to enable a Layer3 ACL over BVI interfaces in the egress direction.

```
Router# configure
Router(config)# hw-module profile acl egress layer3 interface-based
Router(config)# commit
```

acl-permit

To get the permitted statistics of the routing traffic that are allowed by an ACL, use the **acl-permit** command. Statistics of the routing sessions that are not allowed by an ACL are enabled by default.

hw-module profile stats acl-permit
no hw-module profile stats acl-permit

Syntax Description

hw-module	Configures the hardware module.
profile	Configures the profile of the hardware module.
stats	Configures the statistics profile.
acl-permit	Enables the statistics of the routing traffic that are permitted by an ACL.

Command Default

If you do not configure the **acl-permit** command, the statistics for the routing traffic permitted by an ACL are not enabled.

Command History

Release	Modification
Release 6.2.1	This command was introduced.

Usage Guidelines

- The permit statistics of the routing traffic allowed by an ACL are available only for NCS 5500 routers after you execute the **acl-permit** command and reboot the line cards.
- QoS stats are not supported (disabled) when acl-permit stats are enabled.
- You need not configure this command for NC57-24DD and NC57-18DD-SE line cards because both the permitted and denied statistics of the routing traffic that are allowed by an ACL are available by default for these line cards.

Table 3: Task ID

Task ID	Operations
configuration	read, write
root-lr	read, write

Examples

The following example shows you how to configure the acl-permit command:

```
Router# configure
Router(config)# hw-module profile stats acl-permit
Tuw Aug  14 15:31:47.505 UTC
In order to activate/deactivate this stats profile, you must manually reload the chassis/all
 line cards
Router(config)# commit
Tue Aug 14 15:31:50.103 UTC
LC/0/4/CPU0:Aug 14 15:31:50.218 UTC: fia_driver[245]:
%FABRIC-FIA_DRVR-4-STATS_HW_PROFILE_MISMATCH : Mismatch found, reload LC to activate the
new stats profile
Router(config)#
```

acl ipv6 ext-header

To permit the IPV6 extension header packets, use the **acl IPv6 ext-header** command.

hw-module profile acl ipv6 ext-header permit

no hw-module profile acl ipv6 ext-header permit

Syntax Description

hw-module	Configures the hardware module.
profile	Configures the profile of the hardware module.
acl	Configures the ACL profile.
ipv6	Configures the IPv6 protocol.
ext-header	Configures the IPv6 extension header.
permit	Permits the IPv6 extension header packets.

Command Default

By default, the control plane CPU filters the packets and applies security ACLs, when the following IPv6 extensions headers are included:

- Hop-by-Hop
- Destination-Options
- Routing
- Fragment
- Mobility
- Host-Identity

Filtering of the packets in control plane CPU reduces the packet rate to 100 packets/sec and later leads to packet drop.

Command History

Release	Modification
Release 6.6.3	This command was introduced.

Usage Guidelines

Use this command, if you don’t want to filter packets with extension headers and process the packets at line rate. This command allows you to permit all the packets with extension headers and bypass security ACLs.

Table 4: Task ID

Task ID	Operations
configuration	read, write
root-lr	read, write

Examples

The following example shows you how to configure the **ext-header permit** command:

```
Router# configure
Router(config)# hw-module profile acl IPv6 ext-header permit
Router(config)# commit
```

acl-prefix percent

To allocate a certain percentage of external TCAM of the NC55-24x100G-SE and NC55-24H12F-SE line cards for use by a compressed ACL, use the **acl-prefix percent** command.

Note You need not configure this command to support ACL with compression on NC57-24DD and NC57-18DD-SE line cards.

hw-module profile tcam acl-prefix percent percent *value*
no hw-module profile tcam acl-prefix percent percent *value*

Syntax Description

hw-module	Configures the hardware module.
profile	Configures the profile of the hardware module.
tcam	Configures the profile for TCAM LC cards.
acl-prefix	Configures the ACL table.
percent	Configures the percentage of TCAM on the LCs that will be used by a compressed ACL.
value	Configures the value of the percentage.

Command Default

None

Command History

Release	Modification
Release 6.3.2	This command was introduced.

Usage Guidelines

After you execute this command, you must reboot the LCs.

Table 5: Task ID

Task ID	Operations
configuration	read, write
root-lr	read, write

Examples

The following example shows you how to configure the **acl-prefix percent** command:

```
Router# configure
Router(config)# hw-module profile tcam acl-prefix percent 30
Router(config)# commit
Thu Aug  9 13:07:41.401 UTC
LC/0/4/CPU0:Aug  9 13:07:41.539 UTC: fia_driver[209]:
%FABRIC-FIA_DRVR-3-ERR_HW_PROFILE_SOC_PROPERTY_MISMATCH : Mismatch found, reload LC to get
 the most recent config updated
Router(config)#
```

clear access-list ipv4

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

clear access-list ipv4 *access-list name* [*sequence-number* | **ingress**] [**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.
ingress	Specifies an inbound direction.
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

XR EXEC mode

Command History

Release	Modification
Release 6.0	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 an asterisk (***)** in place of the *access-list-name* argument to clear all access lists.

Task ID

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

Task ID	Operations
bgp	read, write, execute

Examples

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

```
RP/0/RP0/CPU0:router# show access-lists ipv4 marketing

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

RP/0/RP0/CPU0:router# clear access-list ipv4 marketing

RP/0/RP0/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
```

clear access-list ipv6

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

clear access-list ipv6 *access-list-name* [*sequence-number* | **ingress**] [**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.
ingress	(Optional) Specifies an inbound direction.
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

Command History

Release	Modification
Release 6.0	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 an asterisk **(*)** in place of the *access-list-name* argument to clear all access lists.

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/# 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
RP/0/# clear access-list ipv6 marketing
RP/0/# 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
```

common-acl

To enable IPv4 or IPv6 common ACLs in an ingress direction on the TCAM of a router, use the **common-acl** option with the **hw-module** command in the XR Config modeglobal configuration mode.

hw-module profile tcam format access-list { **ipv4** | **ipv6** } **common-acl**

Syntax Description

common-acl	Enables you to configure common ACLs.

Command Default

None

Command Modes

XR Config mode

Command History

Release	Modification
Release 7.0.1	The command was introduced.

Usage Guidelines

A reboot of the line card is required after entering the **hw-module profile** command to activate the command.

Configuring Common ACLs for IPv4 and IPv6 ACLs Using User-Defined TCAM Keys

Enable the use of common ACL when IPv4 and IPv6 User-Defined TCAM Keys are used instead of the default TCAM Keys. The following configuration describes how you can enable a common ACL in the IPv4 UDK.

```
/* Configure a common IPv4 acl, common-1, in the global configuration mode by using the
hw-module command */
Router(config)# hw-module profile tcam format access-list ipv4 src-addr dst-addr src-port
dst-port proto tcp-flags frag-bit common-acl location 0/7/CPU0
```

The following configuration describes how you can enable a common ACL in the IPv6 UDK.

```
/* Configure a common IPv6 acl, common-1, in the global configuration mode by using the
hw-module command */
Router(config)# hw-module profile tcam format access-list ipv6 src-addr src-port dst-addr
next-hdr tcp-flags payload-length common-acl location 0/7/CPU0
```

copy access-list ipv4

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

None

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 6.0	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/RP0/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/RP0/CPU0:router# copy access-list ipv4 list-1 list-2
RP/0/RP0/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/RP0/CPU0:router# copy access-list ipv4 list-1 list-3

list-3 exists in access-list

RP/0/RP0/CPU0:router# show access-lists ipv4 list-3

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

copy access-list ipv6

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

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

Command History

Release	Modification
Release 6.0	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/# show access-lists ipv6 list-1

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

RP/0/# copy access-list ipv6 list-1 list-2

RP/0/# 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/# copy access-list ipv6 list-1 list-3

list-3 exists in access-list

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

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* { *port protocol-port* }] *destination destination-wildcard* [*operator* { *port protocol-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.)
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** , **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. **Note** Filtering on AHP protocol is not supported.
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 log-message also includes the input interface.
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** , **urg**.
counter	(Optional) Enables accessing ACL counters using SNMP query.
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 7.6.1	The **log-input** option was introduced.
Release 6.0	This command was introduced.

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 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.

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:

```
Router(config)# ipv4 access-list Internetfilter
Router(config-ipv4-acl)# 10 deny 192.168.34.0 0.0.0.255
Router(config-ipv4-acl)# 20 deny 172.16.0.0 0.0.255.255
Router(config-ipv4-acl)# 25 deny tcp host 172.16.0.0 eq bgp host 192.168.202.203 range 1300
 1400
Router(config-ipv4-acl)# permit 10.0.0.0 0.255.255.255
```

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**] [**hop-by-hop**] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**] [**ttl** *operator ttl value*] [**icmp-off**]
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] [**hop-by-hop**] [**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**] [**hop-by-hop**] [**authen**] [**destopts**] [**fragments**] [**established**] {**match-any | match-all | + | -**}[*flag-name*] [**log**] [**log-input**] [**icmp-off**]

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**] [**hop-by-hop**] [**authen**] [**destopts**] [**fragments**] [**established**][*flag-name*] [**log**] [**log-input**] [**icmp-off**]

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.)
protocol	Name or number of an Internet protocol. It can be one of the keywords **ahp** , **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.
hop-by-hop	(Optional) Supports Jumbo-grams. With the Router Alert option, it is an integral part in the operation of Multicast Listener Discovery (MLD). Router Alert [3] is an integral part in the operations of IPv6 Multicast through MLD and RSVP for IPv6.
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 log-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**, **urg**.

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 7.6.1	The **log-input** option was introduced.
Release 6.5.1	Added the hop-by-hop option.
Release 6.0	This command was introduced.

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 apply the access list to the traffic entering the HundredGigE interface 0/2/0/2. Specifically, the deny entry in the list keeps all packets that have a destination TCP port number greater than 5000 from entering the HundredGigE interface 0/2/0/2. The permit entry in the list permits all ICMP packets to enter the HundredGigE interface 0/2/0/2.

```
Router(config)# ipv6 access-list toCISCO
```

```
Router(config-ipv6-acl)# deny tcp any any gt 5000
Router(config-ipv6-acl)# permit icmp any any
Router(config)#  interface HundredGigE 0/2/0/2
Router(config-if)# ipv6 access-group tOCISCO ingress
```

The following example shows how to configure the IPv6 access list named toCISCO and apply the access list to the traffic entering theHundredGigE interface 0/2/0/2. Specifically, the deny entry in the list keeps all packets that have a hop-by-hop optional field from entering the HundredGigE interface 0/2/0/2.

```
Router(config)# ipv6 access-list toCISCO
Router(config-ipv6-acl)# deny ipv6 any any hop-by-hop
Router(config)#  interface HundredGigE 0/2/0/2
Router(config-if)# ipv6 access-group tOCISCO ingress
```

dont-fragment

To configure an access list to match on the **dont-fragment** flag.

fragment-type dont-fragment {**capture** | **counter** | **default** | **first-fragment** | **is-fragment** | **last-fragment** | **log** | **log-input** | **set** | **udf** | **<none>**}

Syntax Description

capture	ACL matches on the **dont-fragment** flag, and captures the matched packet.
counter	ACL matches on the **dont-fragment** flag, and displays the counter for the matches.
default	ACL matches on the **dont-fragment** flag, and uses specified default next hop.
first-fragment	ACL matches on the **dont-fragment** flag, and then matches on the **first-fragment** flag.
is-fragment	ACL matches on the **dont-fragment** flag, and then matches on the **is-fragment** flag.
last-fragment	ACL matches on the **dont-fragment** flag, and then matches on the **last-fragment** flag.
log	ACL matches on the **dont-fragment** flag and logs the matches.
log-input	ACL matches on the **dont-fragment** flag and logs the matches, incuding on the input interface.
set	ACL matches on the **dont-fragment** flag and sets a particular action on the matches.
udf	ACL matches on the **dont-fragment** flag, and sets the user-defined fields for the matches.

Command Default

None

Command Modes

ACL configuration mode

Command History

Release	Modification
Release 6.3.2	This command was introduced.

Usage Guidelines

This command is supported only for IPv4 ACLs.

Example

Use the following sample configuration to match on the **dont-fragment** flag.

```
/* Enter the global configuraton mode and configure an IPv4 access list */
Router# config
Router(config)# ipv4 access-list TEST
Router(config-ipv4-acl)# 10 permit tcp any any

/* Configure an ACE to match on the dont-fragment flag (indicates a non-fragmented packet)
 and forward the packet to the default (pre-configured) next hop  */
Router(config-ipv4-acl)# 20 permit tcp any any fragment-type dont-fragment default
```

```
Router(config-ipv4-acl)# commit
```

enable-set-ttl

To enable ACLs to set or rewrite a TTL value, use the **enable-set-ttl** option with the **hw-module** command in the global configuration mode.

hw-module profile tcam format access-list ipv4 src-addr src-port enable-set-ttl
hw-module profile tcam format access-list ipv4 dst-addr dst-port enable-set-ttl
hw-module profile tcam format access-list ipv6 src-addr src-port next-hdr enable-set-ttl
hw-module profile tcam format access-list ipv6 dst-addr dst-port src-port next-hdr enable-set-ttl

Syntax Description

dst-addr	Destination address. 32 bit qualifier for IPv4 ACLs and 128-bit qualifier for IPv6 ACLs.
dst-port	Destination L4 Port. 16-bit qualifier
frag-bit	Fragmentation bit for IPv4 ACLs. 1-bit qualifier
enable-capture	Enables ACL-based traffic mirroring and disables ACL logging..
enable-set-ttl	Enables the setting or rewriting of the TTL field.
interface-based	Configures ACLs to be unique for an interface.
location	Specifies location of an access list.
next-hdr	Specifies the next header of IPv6 access list, which is an 8-bit qualifier. This option is mandatory.
packet-length	Specifies packet length for IPv4 ACLs, which is a 10-bit qualifier.
payload-length	Specifies payload length for IPv6 ACLs, which is a 16-bit qualifier.
port-range	Specifies IPv4 port range qualifier, 24-bit qualifier
precedence	Specifies DSCP precedence. 10-bit qualifier
proto	Specifies protocol type. 8-bit qualifier
src-addr	Specifies source address. 32-bit qualifier for IPv4 ACLs and 128-bit qualifier for IPv6 ACLs.
src-port	Specifies source L4 port. 16-bit qualifier
tcp-flags	Specifies TCP Flags. 6-bit qualifier for IPv4 ACLs and 8-bit qualifier for IPv6 ACLs.
traffic-class	Specifies traffic class for IPv6 ACLs, which is an 8-bit qualifier.
ttl-match	Enables ACLs to match on specified TTL value.
udf1	Specifies user-defined filter.

udf2	Specifies user-defined filter.
udf3	Specifies user-defined filter.
udf4	Specifies user-defined filter.
udf5	Specifies user-defined filter.
udf6	Specifies user-defined filter.
udf7	Specifies user-defined filter.
udf8	Specifies user-defined filter.

Command Default None

Command Modes Global configuration mode

Command History

Release	Modification
Release 6.3.2	This command was introduced.

Usage Guidelines If you use either **src-port**, **dst-port**, or **port-range** as one of the optional keywords while setting or modifying the TTL values, you must also use **frag-bit** as one of the other optional keywords to avoid the following error message:



Note A reboot of the line card is required after entering the **hw-module profile** command to activate the command.

```
A SysDB client requested a function that the server or EDM does not currently support:
fragment_bit must be included, if any of the following are include: src-port, dst-port,
port-range, or tcp-flags
```

Enabling TTL Matching and Rewriting for IPv4 ACLs

The following configuration describes how you can enable TTL Matching and Rewriting for IPv4 ACLs.

```
/* Enable TTL matching and rewriting in the global configuration mode by using the hw-module
 command */
Router(config)# hw-module profile tcam format access-list ipv4 dst-addr dst-port proto
port-range enable-set-ttl ttl-match
```

For complete ACL configuration, see the Configuring TTL Matching and Rewriting for IPv4 ACLs section in the *IP Addresses and Services Configuration Guide for NCS 5500 Series Routers*

Enabling TTL Matching and Rewriting for IPv6 ACLs

The following configuration describes how you can enable TTL Matching and Rewriting for IPv4 ACLs.

```
/* Enable TTL matching and rewriting in the global configuration mode by using the hw-module
 command */
Router(config)# hw-module profile tcam format access-list ipv6 dst-addr dst-port src-port
next-hdr enable-set-ttl ttl-match
```

For complete ACL configuration, see the Configuring TTL Matching and Rewriting for IPv6 ACLs section in the *IP Addresses and Services Configuration Guide for NCS 5500 Series Routers*

first-fragment

To configure an ACL to match on the **first-fragment** flag.

fragment-type first-fragment {**capture** | **counter** | **default** | **log** | **log-input** | **set** | **udf** | **<none>**}

Syntax Description

capture	ACL matches on the **first-fragment** flag, and captures the matched packet.
counter	ACL matches on the **first-fragment** flag, and displays the counter for the matches.
default	ACL matches on the **first-fragment** flag, and uses specified default next hop.
log	ACL matches on the **first-fragment** flag and logs the matches.
log-input	ACL matches on the **first-fragment** flag and logs the matches, incuding on the input interface.
set	ACL matches on the **first-fragment** flag and sets a particular action on the matches.
udf	ACL matches on the **first-fragment** flag, and sets the user-defined fields for the matches.

Command Default

None

Command Modes

ACL configuration mode.

Command History

Release	Modification
Release 7.5.1	Added support for IPv6 ACLs.
Release 6.3.2	This command was introduced.

Usage Guidelines

This command is supported for IPv4 and IPv6 ACLs.

Example

Use the following sample configuration to match on the **first-fragment** flag.

```
/* Enter the global configuraton mode and configure an IPv4 access list */
Router# config
Router(config)# ipv4 access-list TEST
Router(config-ipv4-acl)# 10 permit tcp any any

/* Configure an ACE to match on the first-fragment flag (indicates the first fragment of a
 fragmented packet)
 and forward the packet to a next hop of 20.20.20.1  */
Router(config-ipv4-acl)# 40 permit ospf any any fragment-type first-fragment nexthop1 ipv4
 20.20.20.1
Router(config-ipv4-acl)# commit
```

fragment-offset

To enable packet filtering at an ingress or egress interface by specifying fragment-offset as a match condition in an IPv4 or IPv6 ACL, use the **fragment-offset** option in **permit** or **deny** command in IPv4 or IPv6 access-list configuration mode. To disable this feature, use the **no** form of this command.

fragment-offset {**eq** *value* | **gt** *value* | **lt** *value* | **neq** *value* | **range** *lower-limit upper-limit*}

Syntax Description

fragment-offset eq *value*	Filters packets that have a fragment offset equal to the specified limit.
fragment-offset gt *value*	Filters packets that have a fragment offset greater than the specified limit.
fragment-offset lt *value*	Filters packets that have a fragment offset less than the specified limit.
fragment-offset neq *value*	Filters packets that have a fragment offset that does not match the specified limit.
fragment-offset range *lower-limit upper-limit*	Filters packets that have a fragment offset within the specified range.

Command Default

None

Command Modes

IPv4 or IPv6 Access List Configuration mode

Release	Modification
Release 6.2.2	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Example

This example shows how to configure an IPv4 access list to filter packets by the fragment-offset condition:

```
RP/0/RP0/CPU0:router# config
RP/0/RP0/CPU0:router(config)# ipv4 access-list fragment-offset-acl
RP/0/RP0/CPU0:router(config-ipv4-acl)# 10 permit ipv4 any any fragment-offset range 300 400
```

fragment-type

To configure an access list to match on the type of fragment.

fragment-type {**dont-fragment** | **first-fragment** | **is-fragment** | **last-fragment**}

Syntax Description

dont-fragment	ACL matches on the **dont-fragment** flag
first-fragment	ACL matches on the **first-fragment** flag
is-fragment	ACL matches on the **is-fragment** flag
last-fragment	ACL matches on the **last-fragment** flag

Command Default

None

Command Modes

ACL configuration mode

Command History

Release	Modification
Release 6.3.2	This command was introduced.

Usage Guidelines

This command is supported only for IPv4 access lists.

Example

Use the following sample configuration to configure an ACL to match on the type of fragment..

```
/* Enter the global configuraton mode and configure an IPv4 access list */
Router# config
Router(config)# ipv4 access-list TEST
Router(config-ipv4-acl)# 10 permit tcp any any

/* Configure an ACE to match on the dont-fragment flag (indicates a non-fragmented packet)
 and forward the packet to the default (pre-configured) next hop  */
Router(config-ipv4-acl)# 20 permit tcp any any fragment-type dont-fragment default

/* Configure an ACE to match on the is-fragment flag (indicates a fragmented packet)
 and forward the packet to a next hop of 10.10.10.1  */
Router(config-ipv4-acl)# 30 permit udp any any fragment-type is-fragment nexthop1 ipv4
10.10.10.1

/* Configure an ACE to match on the first-fragment flag (indicates the first fragment of a
 fragmented packet)
 and forward the packet to a next hop of 20.20.20.1  */
Router(config-ipv4-acl)# 40 permit ospf any any fragment-type first-fragment nexthop1 ipv4
 20.20.20.1


/* Configure an ACE to match on the last-fragment flag (indicates the last fragment of a
fragmented packet)
 and forward the packet to a next hop of 30.30.30.1  */
```

```
Router(config-ipv4-acl)# 50 permit icmp any any fragment-type last-fragment nexthop1 ipv4
30.30.30.1
Router(config-ipv4-acl)# commit
```

hw-module profile acl ipv6 single-pass-egress-acl

To configure single-pass on IPv6 Egress ACL use the **hw-module profile acl ipv6 single-pass-egress acl** command in XR config mode. To remove the configuration, use the **no** form of the command.

This command has no keywords or arguments.

Command Default

None

Command Modes

XR Config Mode

Command History

Release	Modification
Release 7.10.1	This command was introduced.

Usage Guidelines

- You must manually reload the router after configuring this command.
- By default, Cisco NC57 line cards process packets in a single-pass. So, this feature is not applicable to NCS 5700 Series Routers and Cisco NCS 5500 series routers that have the Cisco NC57 line cards installed and operating in the native or compatibility mode.

Task ID

Task ID	Operations
configuration	read, write
root-lr	read, write

Example

The following example shows how to configure single-pass IPv6 egress ACL:

```
Router# configure terminal
Router(config)# hw-module profile acl ipv6 single-pass-egress-acl
Router(config)# commit
```

interface-based

To configure ACLs that are unique for an interface, use the **interface-based** option with the **hw-module** command in the global configuration mode.

hw-module profile tcam format access-list ipv4 src-addr src-port dst-addr dst-port interface-based
hw-module profile tcam format access-list ipv6 src-addr src-port dst-addr dst-port next-hdr interface-based

Syntax Description

dst-addr	Destination address. 32 bit qualifier for IPv4 ACLs and 128-bit qualifier for IPv6 ACLs.
dst-port	Destination L4 Port. 16-bit qualifier
frag-bit	Fragmentation bit for IPv4 ACLs. 1-bit qualifier
enable-capture	Enables ACL-based traffic mirroring and disables ACL logging.
enable-set-ttl	Enables the setting or rewriting of an ACL.
interface-based	Configures ACLs to be unique for an interface.
location	Specifies location of an access list.
next-hdr	Specifies the next header of IPv6 access list, which is an 8-bit qualifier. This option is mandatory.
packet-length	Specifies packet length for IPv4 ACLs, which is a 10-bit qualifier.
payload-length	Specifies payload length for IPv6 ACLs, which is a 16-bit qualifier.
port-range	Specifies IPv4 port range qualifier, 24-bit qualifier
precedence	Specifies DSCP precedence. 10-bit qualifier
proto	Specifies protocol type. 8-bit qualifier
src-addr	Specifies source address. 32-bit qualifier for IPv4 ACLs and 128-bit qualifier for IPv6 ACLs.
src-port	Specifies source L4 port. 16-bit qualifier
tcp-flags	Specifies TCP Flags. 6-bit qualifier for IPv4 ACLs and 8-bit qualifier for IPv6 ACLs.
traffic-class	Specifies traffic class for IPv6 ACLs, which is an 8-bit qualifier.
ttl-match	Enables ACLs to match on specified TTL value.
udf1	Specifies user-defined filter.

udf2	Specifies user-defined filter.
udf3	Specifies user-defined filter.
udf4	Specifies user-defined filter.
udf5	Specifies user-defined filter.
udf6	Specifies user-defined filter.
udf7	Specifies user-defined filter.
udf8	Specifies user-defined filter.

Command Default

None

Command Modes

Global configuration

Command History

Release	Modification
6.3.2	This command was introduced.

Usage Guidelines

ACLs that are shared across interfaces and use the same TCAM space are known as shared ACLs. However, you can configure only 31 unique, shared ACLs. To configure more unique ACLs, ACL sharing must be disabled by using the **interface-based** command. By making the ACLs unique for an interface, you can configure more than 31 ACLs.

Enabling interface-based IPv4 ACLs

```
/* Enable interface-based, unique IPv4 ACLs */
Router(config)# hw-module profile tcam format access-list ipv4 src-addr src-port dst-addr
dst-port interface-based
```

For complete ACL configuration, see the Configuring TTL Matching for IPv4 ACLs section in the *IP Addresses and Services Configuration Guide for NCS 5500 Series Routers*

Enabling interface-based IPv6 ACLs

```
/* Enable interface-based, unique IPv6 ACLs */
Router(config)# hw-module profile tcam format access-list ipv6 src-addr src-port dst-addr
dst-port next-hdr interface-based
```

For complete ACL configuration, see the Configuring TTL Matching for IPv6 ACLs section in the *IP Addresses and Services Configuration Guide for NCS 5500 Series Routers*

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 *access-list-name* [**common** *common-acl-name*] { **ingress** | **egress** } [**compress level** *compression-level*] [**interface-statistics**] [**hardware-count**]

Syntax Description

access-list-name	Name of an IPv4 access list as specified by an **ipv4 access-list** command.
common	Configures common ACLs.
ingress	Filters on inbound packets.
egress	Filters on outbound packets.
compress level *compression-level*	Configures compression level for interface ACLs. Compression level values range from zero to three.
interface-statistics	Configures the logging of per interface statistics.
hardware-count	Configures the logging of count of filtered packets.

Command Default

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

Command Modes

Interface configuration

Command History

Release	Modification
Release 6.0	This command was introduced.
Release 7.2.1	Support to configure multiple ACLs was added.

Usage Guidelines

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 interface ACL is not supported.



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.

You can configure common ACLs only in the ingress direction. You cannot configure compression levels for common ACLs.

Task ID

Task ID	Operations
acl	read, write
network	read, write

Examples

The following example shows how to apply filters on packets from HundredGigE interface 0/2/0/2:

```
Router(config)# interface  HundredGigE 0/2/0/2
Router(config-if)# ipv4 access-group p-ingress-filter ingress
```

ipv4 access-list

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

ipv4 access-list [*name* | **icmp-off**]
no ipv4 access-list [*name* | **icmp-off**]

Syntax Description

name	Name of the access list. Names cannot contain a space or quotation marks.
icmp-off	Disables generating the ICMP unreachable messages for packets dropped by deny ACEs in the router.

Command Default

No IPv4 access list is defined.

Command Modes

XR Config mode

Command History

Release	Modification
Release 7.5.1	Support for icmp-off option was introduced.
Release 6.0	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 **ipv4 access-group** command to apply the access list to an interface.

The maximum number of supported port ranges including both IPv4 and IPv6 must not exceed 23. That is, if a configuration that supports 23 unique ranges for IPv4 and 23 unique ranges for IPv6 is applied together, then it results in invalid configuration and causes OOR (out-of-resource) condition.

Task ID

Task ID	Operations
acl	read, write

Examples

This example shows how to define a standard access list named Internetfilter and disable ICMP Unreachable messages at global configuration:

```
Router(config)# ipv4 access-list Internetfilter
Router(config-ipv4-acl)# 10 permit 192.168.34.0 0.0.0.255
Router(config-ipv4-acl)# 20 permit 172.16.0.0 0.0.255.255
Router(config-ipv4-acl)# 30 permit 10.0.0.0 0.255.255.255
Router(config-ipv4-acl)# 39 remark Block BGP traffic from 172.16 net.
```

```
Router(config-ipv4-acl)# 40 deny tcp host 172.16.0.0 eq bgp host 192.168.202.203 range 1300
 1400

Router(config)# ipv4 access-list icmp-off
```

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 XR Config 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

XR Config mode

Command History

Release	Modification
Release 6.0	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/RP0/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 XR Config 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

XR Config mode

Command History

Release	Modification
Release 6.0	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/RP0/CPU0:router(config)# ipv4 access-list log-update threshold 10
```

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* [**common** *common-acl-name*] { **ingress** | **egress** } [**compress level** *compression-level*] [**interface-statistics**] [**hardware-count**]

Syntax Description

access-list-name	Name of an IPv4 access list as specified by an **ipv4 access-list** command.
common	Configures common ACLs.
ingress	Filters on inbound packets.
egress	Filters on outbound packets.
compress level *compression-level*	Configures compression level for interface ACLs. Compression level values range from zero to three.
interface-statistics	Configures the logging of per interface statistics.
hardware-count	Configures the logging of count of filtered packets.

Command Default

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

Command Modes

Interface configuration

Command History

Release	Modification
Release 6.0	This command was introduced.

Usage Guidelines

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 interface ACL is not supported.

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 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.

You can configure common ACLs only in the ingress direction. You cannot configure compression levels for common ACLs.

Task ID

Task ID	Operations
acl	read, write
ipv6	read, write

Examples

The following example shows how to apply filters on packets from HundredGigE interface 0/2/0/2:

```
Router(config)# interface  HundredGigE 0/2/0/2
Router(config-if)# ipv6 access-group p-ingress-filter ingress
```

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 interface configuration mode. To remove the access list, use the **no** form of this command.

ipv6 access-list [*name* | **icmp-off**]
no ipv6 access-list [*name* | **icmp-off**]

Syntax Description

name	Name of the IPv6 access list. Names cannot contain a space or quotation mark, or begin with a numeric.
icmp-off	Disables generating the ICMP unreachable messages for packets dropped by deny ACEs in the router.

Command Default

No IPv6 access list is defined.

Command Modes

Interface configuration

Command History

Release	Modification
Release 7.5.1	Support for icmp-off option was introduced.
Release 6.0	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 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.

See the “Examples” section for an example of a translated IPv6 access control list (ACL) configuration.



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 An IPv6 access list applied to an interface with the **ipv6 access-group** command filters traffic that is forwarded, not originated, by the router.

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.

The maximum number of supported port ranges including both IPv4 and IPv6 must not exceed 23. That is, if a configuration that supports 23 unique ranges for IPv4 and 23 unique ranges for IPv6 is applied together, then it results in invalid configuration and causes OOR (out-of-resource) condition.

Task ID

Task ID	Operations
acl	read, write
ipv6	read, write

Examples

This example shows how to configure the IPv6 access list named list2 and applies the ACL to traffic on interface HundredGigE 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 HundredGigE 0/2/0/2. The second entry in the ACL permits all other traffic to exit out of interface HundredGigE 0/2/0/2. The second entry is necessary because an implicit deny all condition is at the end of each IPv6 ACL.

```
Router(config)# ipv6 access-list list2
Router(config-ipv6-acl)# 10 deny fec0:0:0:2::/64 any
Router(config-ipv6-acl)# 20 permit any any

Router# show ipv6 access-lists list2
```

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

Router(config)# interface HundredGigE 0/2/0/2
```

Note IPv6 is automatically configured as the protocol type in **permit any any** and **deny any any** statements that are translated from 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).

This example shows how to disable ICMP Unreachable messages at global configuration:

```
Router(config)# ipv6 access-list icmp-off
```

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 . 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

Command History

Release	Modification
Release 6.0	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/(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 . 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

Command History

Release	Modification
Release 6.0	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/(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 . 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

Command History

Release	Modification
Release 6.0	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:

```
Router(config)# ipv6 access-list maximum ace threshold 75000
```

is-fragment

To configure an ACL to match on the **is-fragment** flag.

fragment-type is-fragment {**capture** | **counter** | **default** | **log** | **log-input** | **set** | **udf** | **<none>**}

Syntax Description

capture	ACL matches on the **is-fragment** flag, and captures the matched packet.
counter	ACL matches on the **is-fragment** flag, and displays the counter for the matches.
default	ACL matches on the **is-fragment** flag, and uses specified default next hop.
log	ACL matches on the **is-fragment** flag and logs the matches.
log-input	ACL matches on the **is-fragment** flag and logs the matches, incuding on the input interface.
set	ACL matches on the **is-fragment** flag and sets a particular action on the matches.
udf	ACL matches on the **is-fragment** flag, and sets the user-defined fields for the matches.

Command Default

None

Command Modes

ACL configuration mode.

Command History

Release	Modification
Release 7.5.1	Added support for IPv6 ACLs.
Release 6.3.2	This command was introduced.

Usage Guidelines

This command is supported for IPv4 and IPv6 ACLs.

Example

Use the following sample configuration to match on the **is-fragment** flag.

```
/* Enter the global configuraton mode and configure an IPv4 access list */
Router# config
Router(config)# ipv4 access-list TEST
Router(config-ipv4-acl)# 10 permit tcp any any

/* Configure an ACE to match on the is-fragment flag (indicates a fragmented packet)
 and forward the packet to a next hop of 10.10.10.1  */
Router(config-ipv4-acl)# 30 permit udp any any fragment-type is-fragment nexthop1 ipv4
10.10.10.1
Router(config-ipv4-acl)# commit
```

last-fragment

To configure an access list to match on the **last-fragment** flag.

fragment-type last-fragment {**capture** | **counter** | **default** | **log** | **log-input** | **set** | **udf** | **<none>**}

Syntax Description

capture	ACL matches on the **last-fragment** flag, and captures the matched packet.
counter	ACL matches on the **last-fragment** flag, and displays the counter for the matches.
default	ACL matches on the **last-fragment** flag, and uses specified default next hop.
log	ACL matches on the **last-fragment** flag and logs the matches.
log-input	ACL matches on the **last-fragment** flag and logs the matches, incuding on the input interface.
set	ACL matches on the **dont-fragment** flag and sets a particular action on the matches.
udf	ACL matches on the **last-fragment** flag, and sets the user-defined fields for the matches.

Command Default

None

Command Modes

ACL configuration mode.

Command History

Release	Modification
Release 6.3.2	This command was introduced.

Usage Guidelines

This command is supported only for IPv4 ACLs.

Example

Use the following sample configuration to match on the **last-fragment** flag.

```
/* Enter the global configuraton mode and configure an IPv4 access list */
Router# config
Router(config)# ipv4 access-list TEST
Router(config-ipv4-acl)# 10 permit tcp any any

/* Configure an ACE to match on the last-fragment flag (indicates the last fragment of a
fragmented packet)
 and forward the packet to a next hop of 30.30.30.1  */
Router(config-ipv4-acl)# 50 permit icmp any any fragment-type last-fragment nexthop1 ipv4
30.30.30.1
Router(config-ipv4-acl)# commit
```

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/RP0/CPU0:router# config
RP/0/RP0/CPU0:router(config)# ipv4 access-list pktlen-v4
RP/0/RP0/CPU0:router(config-ipv4-acl)# 10 permit tcp any any packet-length eq 1482
RP/0/RP0/CPU0:router(config-ipv4-acl)# 20 permit udp any any packet-length range 1400 1500
RP/0/RP0/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/RP0/CPU0:router# config
RP/0/RP0/CPU0:router(config)# ipv6 access-list pktlen-v6
RP/0/RP0/CPU0:router(config-ipv6-acl)# 10 permit tcp any any packet-length eq 1500
```

```
RP/0/RP0/CPU0:router(config-ipv6-acl)# 20 permit udp any any packet-length range 1500 1600
RP/0/RP0/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* [**precedence** *precedence*] [**nexthop** [*ipv4-address1*] [*ipv4-address2*] [*ipv4-address3*]] [**dscp** *dscp* **bitmask**] [**fragments**] [**log** | **log-input**] [**nexthop** [**track** *track-name*] [*ipv4-address1*] [*ipv4-address2*] [*ipv4-address3*] [**ttl** *ttl value* [*value1* . . . *value2*]] [**counter** *counter-name*]
police *rate*
capture
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**][**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.)

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** , **gre** , **icmp** , **igmp** , **igrp** , **ip** , **ipinip** , **nos** , **ospf** , **pim** , **pcp** , **sctp** , **tcp** , or **udp** , or an integer from 0 to 255 representing an IP protocol number. To match any Internet protocol (including ICMP, SCTP TCP, and UDP), use the **ip** keyword. ICMP, and TCP allow further qualifiers, which are described later in this table. **Note** Filtering on AHP protocol is not supported.
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.
nexthop1, nexthop2, nexthop3	Specifies the next hop for this entry. **Note** You must specify the VRF for all nexthops unless the nexthop is in the default VRF.

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)
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.

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 log-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-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** , **urg** .
counter	(Optional) Enables accessing ACL counters using SNMP query.
counter-name	Defines an ACL counter name.
police	(Optional) Enables traffic policing for the ACE.
rate	Specify the policing rate in bps, kbps, mbps, or gbps.

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 7.6.1	The following options were introduced: • **log-input** • **police**
Release 7.5.4	**bitmask** keyword was introduced.
Release 6.3.2	The **vrf** option for **nexthop** was made mandatory.
Release 6.0	This command was introduced.

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.



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.

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:

```
Router(config)# ipv4 access-list Internetfilter
Router(config-ipv4-acl)# 10 permit 192.168.34.0 0.0.0.255
Router(config-ipv4-acl)# 20 permit 172.16.0.0 0.0.255.255
Router(config-ipv4-acl)# 25 permit tcp host 172.16.0.0 eq bgp host 192.168.202.203 range
1300 1400
Router(config-ipv4-acl)# deny 10.0.0.0 0.255.255.255
```

This example shows how you can configure DSCP bitmask on ingress ERSPAN.

```
Router# config
Router(config)# ipv4 access-list acl1
Router(config-ipv4-acl)# 10 permit ipv4 host 192.0.2.1 any dscp af22 bitmask 0x3f
Router(config-ipv4-acl)# commit
Router(config-ipv4-acl)# exit
Router(config)# interface HundredGigE0/0/0/6
Router(config-if)# ipv4 address 192.0.2.51 255.255.255.0
Router(config-if)# monitor-session TEST ethernet direction rx-only port-level acl ipv4 acl1
Router(config-if)# commit
```

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** *source* { *source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/prefix-length* } [*operator* { *port* | *protocol-port* }] `capture`] [**dscp** *value*] [**routing**] [**hop-by-hop**] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**] [**ttl** *operator ttl value*]
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* }] [**dscp** *value* [**bitmask** *value*] [**routing**] [**hop-by-hop**] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**] [**ttl** *operator ttl value*]
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*]]] [**police** *rate*]
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**] [**hop-by-hop**] [**authen**] [**destopts**] [**fragments**] [**log** | **log-input**] [**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**] [**hop-by-hop**] [**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**] [**hop-by-hop**] [**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.)
protocol	Name or number of an Internet protocol. It can be one of the keywords **ahp**, **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	Specifies the next hop for this entry. **Note** You must specify the VRF for all nexthops unless the nexthop is in the default VRF.
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.
hop-by-hop	(Optional) Supports Jumbo-grams. With the Router Alert option, it is an integral part in the operation of Multicast Listener Discovery (MLD). Router Alert [3] is an integral part in the operations of IPv6 Multicast through MLD and RSVP for IPv6.
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, except that the log-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-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**, **urg**.
counter	(Optional) Enables accessing ACL counters using SNMP query.
counter-name	Defines an ACL counter name.
police	(Optional) Enables traffic policing for the ACE.
rate	Specify the policing rate in bps, kbps, mbps, or gbps.

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 7.6.1	The following options were introduced: • **log-input** • **police**
Release 7.5.4	**bitmask** keyword was introduced.

Release	Modification
Release 6.3.2	The **vrf** option for **nexthop** was made mandatory.
Release 6.0	This command was introduced.

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.

Task ID

Task ID	Operations
acl	read, write

Examples

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

```
Router(config)# ipv6 access-list v6-abf-acl
Router(config-ipv6-acl)# 10 permit ipv6 any any
Router(config-ipv6-acl)# 20 permit ipv4 any any
Router(config)# interface HundredGigE 0/0/2/0
Router(config-if)# ipv6 access-group v6-abf-acl ingress
```

The following example shows how to configure the IPv6 access list named toCISCO and apply the access list to the traffic entering theHundredGigE interface 0/2/0/2. Specifically, the permit entry in

the list allows all packets that have a hop-by-hop optional field from entering the HundredGigE interface 0/2/0/2.

```
Router(config)# ipv6 access-list toCISCO
Router(config-ipv6-acl)# permit ipv6 any any hop-by-hop
Router(config)#  interface HundredGigE 0/2/0/2
Router(config-if)# ipv6 access-group tOCISCO ingress
```

This example shows how to configure the IPv6 access list named Test with ACL-based policing applied to each ACEs.

```
Router(config)# ipv6 access-list Test
Router(config-ipv6-acl)# 10 permit fec0:0:0:2::/64 any police 10 gbps
Router(config-ipv6-acl)# 20 permit any any police 1274 kbps

Router# show ipv6 access-lists Test hardware ingress location 0/1/CPU0
10 permit fec0:0:0:2::/64 any (Accepted: 24303 packets, Dropped: 0 packets)
20 permit any any (Accepted: 13 packets, Dropped: 0 packets)
```

The following example shows how you can configure DSCP bitmask on ingress ERSPAN.

```
Router# config
Router(config)# ipv6 access-list acl1
Router(config-ipv6-acl)# 10 permit ipv6 host 2001:DB8::2/32 any dscp 33 bitmask 0x3f
Router(config-ipv6-acl)# commit
Router(config-ipv6-acl)# exit
Router(config)# interface HundredGigE 0/0/10/3
Router(config-if)# ipv6 address 2001:DB8::1/32
Router(config-if)# monitor-session TEST ethernet direction rx-only port-level acl ipv6 acl1
Router(config-if)# commit
```

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 6.0	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.

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/RP0/CPU0:router(config)# ipv4 access-list telnetting
RP/0/RP0/CPU0:router(config-ipv4-acl)# 10 remark Do not allow user1 to telnet out
RP/0/RP0/CPU0:router(config-ipv4-acl)# 20 deny tcp host 172.16.2.88 255.255.0.0 any eq
telnet
RP/0/RP0/CPU0:router(config-ipv4-acl)# 30 permit icmp any any
RP/0/RP0/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
```

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 6.0	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.

Task ID

Task ID	Operations
acl	read, write

Examples

In this example, a remark is added:

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

ttl-match

To enable ACLs to match on a specified TTL value, use the **ttl-match** option with the **hw-module** command in the global configuration mode.

hw-module profile tcam format access-list ipv4 src-addr src-port enable-set-ttl ttl-match
hw-module profile tcam format access-list ipv4 dst-addr dst-port enable-set-ttl ttl-match
hw-module profile tcam format access-list ipv6 src-addr src-port next-hdr enable-set-ttl ttl-match
hw-module profile tcam format access-list ipv6 dst-addr dst-port src-port next-hdr enable-set-ttl ttl-match

Syntax Description

dst-addr	Destination address. 32 bit qualifier for IPv4 ACLs and 128-bit qualifier for IPv6 ACLs.
dst-port	Destination L4 Port. 16-bit qualifier
frag-bit	Fragmentation bit for IPv4 ACLs. 1-bit qualifier
enable-capture	Enables ACL-based traffic mirroring and disables ACL logging..
enable-set-ttl	Enables the setting or rewriting of an ACL.
interface-based	Configures ACLs to be unique for an interface.
location	Specifies location of an access list.
next-hdr	Specifies the next header of IPv6 access list, which is an 8-bit qualifier. This option is mandatory.
packet-length	Specifies packet length for IPv4 ACLs, which is a 10-bit qualifier.
payload-length	Specifies payload length for IPv6 ACLs, which is a 16-bit qualifier.
port-range	Specifies IPv4 port range qualifier, 24-bit qualifier
precedence	Specifies DSCP precedence. 10-bit qualifier
proto	Specifies protocol type. 8-bit qualifier
src-addr	Specifies source address. 32-bit qualifier for IPv4 ACLs and 128-bit qualifier for IPv6 ACLs.
src-port	Specifies source L4 port. 16-bit qualifier. This is a mandatory option.
tcp-flags	Specifies TCP Flags. 6-bit qualifier for IPv4 ACLs and 8-bit qualifier for IPv6 ACLs.
traffic-class	Specifies traffic class for IPv6 ACLs, which is an 8-bit qualifier.

ttl-match	Enables ACLs to match on specified TTL value.
udf1	Specifies user-defined filter.
udf2	Specifies user-defined filter.
udf3	Specifies user-defined filter.
udf4	Specifies user-defined filter.
udf5	Specifies user-defined filter.
udf6	Specifies user-defined filter.
udf7	Specifies user-defined filter.
udf8	Specifies user-defined filter.

Command Default

None

Command Modes

Global configuration mode

Command History

Release	Modification
6.3.2	This command was introduced.

Usage Guidelines

Using TTL matching for ACLs is known to have the following limitations.

- TTL matching is supported only for ingress ACLs.
- TTL rewrite using the set ttl command, cannot be used with ACL logging.
- If a TTL rewrite is applied to the outer IPv4/IPv6 header of an IP-in-IP header, then when the outer IPv4/IPv6 header is decapsulated, (by GRE decapsulation) the TTL rewrite is also applied to the inner IPv4/IPv6 header.

Enabling TTL Matching and Rewriting for IPv4 ACLs

The following configuration describes how you can enable TTL Matching and Rewriting for IPv4 ACLs.

```
/* Enable TTL matching and rewriting in the global configuration mode by using the hw-module
 command */
Router(config)# hw-module profile tcam format access-list ipv4 dst-addr dst-port proto
port-range enable-set-ttl ttl-match
```

For complete ACL configuration, see the Configuring TTL Matching and Rewriting for IPv4 ACLs section in the *IP Addresses and Services Configuration Guide for NCS 5500 Series Routers*

Enabling TTL Matching and Rewriting for IPv6 ACLs

The following configuration describes how you can enable TTL Matching and Rewriting for IPv4 ACLs.

```
/* Enable TTL matching and rewriting in the global configuration mode by using the hw-module
 command */
Router(config)# hw-module profile tcam format access-list ipv6 dst-addr dst-port src-port
next-hdr enable-set-ttl ttl-match
```

For complete ACL configuration, see the Configuring TTL Matching and Rewriting for IPv6 ACLs section in the *IP Addresses and Services Configuration Guide for NCS 5500 Series Routers*

tx-scale-enhanced acl-permit

To get the permitted statistics of the routing traffic that are allowed by an ACL for increasing the Tx scale, use the **tx-scale-enhanced acl-permit** command. Statistics of the routing sessions that are not allowed by an ACL are enabled by default.

hw-module profile stats tx-scale-enhanced acl-permit

Syntax Description

hw-module	Configures the hardware module.
profile	Configures the profile of the hardware module.
stats	Configures the statistics profile.
tx-scale-enhanced	Increases the Tx scale.
acl-permit	Enables the statistics of the routing traffic that are permitted by an ACL.

Command Default

If you do not configure the **tx-scale-enhanced acl-permit** command, the statistics for the routing traffic permitted by an ACL are not enabled.

Command History

Release	Modification
Release 6.2.1	This command was introduced.

Usage Guidelines

- The permit statistics of the routing traffic allowed by an ACL are available only for NCS 5500 routers after you execute the **tx-scale-enhanced acl-permit** command and reboot the line cards.
- QoS stats are not supported (disabled) when acl-permit stats are enabled.
- You need not configure this command for NC57-24DD and NC57-18DD-SE line cards because both the permitted and denied statistics of the routing traffic that are allowed by an ACL are available by default for these line cards.

Task ID

Task ID	Operations
configuration	read, write
root-lr	read, write

Examples

The following example shows you how to configure the**tx-scale-enhanced acl-permit** command:

```
Router# configure
Router(config)# hw-module profile stats tx-scale-enhanced acl-permit
Tuw Aug  14 15:31:47.505 UTC
In order to activate/deactivate this stats profile, you must manually reload the chassis/all
 line cards
Router(config)# commit
Tue Aug 14 15:31:50.103 UTC
LC/0/4/CPU0:Aug 14 15:31:50.218 UTC: fia_driver[245]:
%FABRIC-FIA_DRVR-4-STATS_HW_PROFILE_MISMATCH : Mismatch found, reload LC to activate the
new stats profile
Router(config)#
```

set qos-group

To set the quality of service (QoS) group identifiers on packets, use the **set qos-group** command in policy map class configuration mode. To leave the QoS group values unchanged, use the **no** form of this command.

set qos-group *qos-group-value*
no set qos-group *qos-group-value*

Syntax Description

qos-group-value	QoS group ID. An integer from 1 to 7, to be marked on the packet. The *qos-group-value* is used to select a CoSQ and eventually to a VOQ

Command Default

No group ID is specified.

Command Modes

Policy map class configuration

Command History

Release	Modification
Release 6.0	This command was introduced.

Usage Guidelines

The **set qos-group** command is supported only in the ingress direction.

The **set qos-group** will be used as internal priority to choose the queue on the egress port.

In the ingress policy-map, in order to designate the traffic class to a certain CoSQ other than CoSQ 0, the class-map needs to have an explicit set qos-group x statement, where 'x' is the CoSQ in the range of 0 to 7. The default COSQ is 0. In the egress policy-map, a class-map with a corresponding match qos-group x will allow further Quality of Service actions to be applied to the traffic class. For example,

```
class-map prec1
  match prec 1

policy-map test-ingress
  class prec1
    set qos-group 1
    police rate percent 50


class-map qg1
  match qos-group 1


policy-map test-egress
  class qg1
    shape average percent 70
```

Task ID

Task ID	Operations
qos	read, write

Examples

This example sets the QoS group to 5 for packets that match the MPLS experimental bit 1:

```
Router(config)# class-map class1
Router(config-cmap)# match mpls experimental topmost 1
Router(config-cmap)# exit

Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# set qos-group 5
Router(config-pmap-c)# exit
Router(config-pmap)# exit

Router(config)# HundredGigE interface 0/1/0/0
Router(config-if)# service-policy input policy1
```

set ttl

To set or rewrite the TTL field, use the **set ttl** command in global configuration mode.

set ttl *value*

Syntax Description

value	Value of TTL to be set. Range: 0-255

Command Default

No group ID is specified.

Command Modes

Global configuration

Command History

Release	Modification
Release 6.0	This command was introduced.

Task ID

Task ID	Operations
ttl	read, write

Usage Guidelines

Using TTL matching for ACLs is known to have the following limitations.

- TTL matching is supported only for ingress ACLs.
- TTL rewrite using the set ttl command, cannot be used with ACL logging.
- If a TTL rewrite is applied to the outer IPv4/IPv6 header of an IP-in-IP header, then when the outer IPv4/IPv6 header is decapsulated, (by GRE decapsulation) the TTL rewrite is also applied to the inner IPv4/IPv6 header.

Setting the TTL value to less than 50 for an ACL:

The following example describes how you can set TTL values for IPv4 ACLs.

```
/* Enter the global configuration mode and configure an IPv4 access list */
Router# config
Router(config)# ipv4 access-list abc
Router(config-ipv4-acl)# 20 permit tcp any any

/* Set the ACL with an either greater than (gt) or lesser  than (lt)  TTL value. The range
 is 0-255 */
Router(config-ipv4-acl)# 20 permit tcp any any ttl lt 50 set
Router(config-ipv4-acl)# commit
```

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 XR EXEC mode.

show access-lists afi-all

Syntax Description

This command has no keywords or arguments.

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 6.0	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/RP0/CPU0:router# show access-lists afi-all

ipv4 access-list test_ipv4
 10 permit ipv4 any any
 20 deny tcp any eq 2000 any eq 2000
 30 permit tcp any eq 3000 any eq 3000
ipv6 access-list test_ipv6
 10 permit ipv6 any any
 20 permit tcp any eq 3000 any eq 3000
```

show access-lists ipv4

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

show access-lists ipv4 [*access-list-name* **hardware** { **ingress** | **verify** } [**interface** *type interface-path-id*] { **sequence** *number* | **location** *node-id* } | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage pfilter** { **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.
verify	(Optional) Verifies the ACL configured. **Note** The **verify** keyword is not supported on NC57-24DD and NC57-18DD-SE line cards.
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.
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	(Optional) Displays TCAM entries.
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

XR EXEC mode

Command History

Release	Modification
Release 7.9.1	The ACL counters displays statistics in bytes.
Release 7.6.1	Added counters for packets allowed and dropped according to policing rate per ACE.
Release 6.0	This command was introduced.

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** 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. 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-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:

```
Router# show access-lists ipv4


ipv4 access-list test_ipv4
 10 permit ipv4 any any
 20 deny tcp any eq 2000 any eq 2000
 30 permit tcp any eq 3000 any eq 3000
```

In the following example, the contents of an access list named Test that has ACL-based policing is configured is displayed:

```
Router(config)# show ipv4 access-list Test hardware ingress location 0/1/CPU0
10 permit 192.168.34.0 0.0.0.255 (Accepted: 130 packets, Dropped: 0 packets)
20 permit 172.16.0.0 0.0.255.255 (Accepted: 1005 packets, Dropped: 0 packets)
30 permit 10.0.0.0 0.255.255.255 (Accepted: 10303 packets, Dropped: 7 packets)
```

This table describes the significant fields shown in the display.

Table 6: show access-lists ipv4 hardware Field Descriptions

Field	Description
ACL name	Name of the ACL programmed in hardware.
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:

```
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 7: 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.

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

```
Router# show access-lists ipv4 usage pfilter location 0/RP0/CPU0


 Interface : TenGigE0/0/0/10/0
Input ACL : Common-ACL : N/A ACL : test_ipv4
Output ACL : N/A
```

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 ACL contents:

```
Router# show access-lists IPv4-ABF hardware ingress location 0/6/CPU0

Wed Feb 19 13:36:26.663 PST
ipv4 access-list IPv4-ABF
100 permit tcp host 27.0.0.2 any eq 8080 (6854367 matches) (next-hop: addr=21.0.0.2, vrf
name=vrf1)
110 permit tcp any eq https any (6858321 matches) (next-hop: addr=200.1.1.2, vrf name=vrf2)
120 permit ipv4 any any (6940396 matches) (next-hop: addr=50.0.0.1, vrf name=default)
```

In the following example, the statistics IPv4 access lists are displayed in bytes and packet counts:

```
Router:ios# show access-lists ipv4 ac hardware ingress location 0/0/CPU0
ipv4 access-list ac
 10 permit ipv4 any 2.2.0.0 0.0.255.255 dscp af11 (477 matches) (30528 byte matches)
 20 permit ipv4 any 2.2.0.0 0.0.255.255 police 5 gbps (Accepted: 464 matches, Dropped: 0)
(Accepted: 29696 byte matches, Dropped: 0 bytes)
```

In the following example, the IPv4 access list is displayed using **detail** keyword:

```
Router# show access-lists ipv4 objv4acl hardware ingress detail location 0/0/CPU0
objv4acl Details:
Sequence Number: 10
```

```
NPU ID: 0
Number of DPA Entries: 1
ACL ID: 2
ACE Action: PERMIT
ACE Logging: DISABLED
ABF Action: 0(ABF_NONE)
Hit Packet Count: 477 Byte Count: 30528
Source Address: 0.0.0.1 (Mask 255.255.255.254)
Destination Address: 0.0.0.1 (Mask 255.255.255.254)
DPA Entry: 1
        Entry Index: 0
        DPA Handle: 0x8E08F0A8
        DSCP: 0x28 (Mask 0xFC)
Sequence Number: IMPLICIT DENY
NPU ID: 0
Number of DPA Entries: 1
ACL ID: 2
ACE Action: DENY
ACE Logging: DISABLED
ABF Action: 0(ABF_NONE)
Hit Packet Count: 0 Byte Count: 0
Source Address: 0.0.0.2 (Mask 255.255.255.253)
Destination Address: 0.0.0.2 (Mask 255.255.255.253)
DPA Entry: 1
        Entry Index: 0
        DPA Handle: 0x8E08F390
```

show access-lists ipv6

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

show access-lists ipv6 [*access-list-name* **hardware** { **ingress** | **verify** } [**interface** *type interface-path-id*] { **sequence** *number* | **location** *node-id* } | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage pfilter** { **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.
verify	Verifies the ACL configured. **Note** The **verify** keyword is not supported on NC57-24DD and NC57-18DD-SE line cards.
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.

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 TCAM entries.
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

Command History

Release	Modification
Release 7.9.1	The ACL counters displays statistics in bytes.
Release 7.6.1	Added counters for packets allowed and dropped according to policing rate per ACE.
Release 6.0	This command was introduced.

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** 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. 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-list ipv6 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:

```
Router# config
Router(config)# ipv6  access-list acl1
Router(config-ipv6-acl)# permit 1:2::3 FF:0:FFFF:AA:20:: 4:5::6 0:FFFF:2233::FFFF
Router(config-ipv6-acl)# commit
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:

```
Router# show access-lists ipv6

ipv6 access-list test_ipv6
 10 permit ipv6 any any
 20 permit tcp any eq 3000 any eq 3000
```

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

```
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 Test that has ACL-based policing is configured is displayed:

```
Router# show ipv6 access-lists Test hardware ingress location 0/1/CPU0
10 permit fec0:0:0:2::/64 any (Accepted: 24303 packets, Dropped: 0 packets)
20 permit any any (Accepted: 13 packets, Dropped: 0 packets)
```

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

```
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 8: 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.

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

```
Router# show access-lists ipv6 usage pfilter location 0/0/CPU0

Interface : TenGigE0/0/0/10/0
    Input  ACL : Common-ACL : N/A  ACL : test_ipv6
    Output ACL : N/A
```

In the following example, the statistics IPv6 access lists are displayed in bytes and packet counts:

```
Router# show ipv6 access-lists Test hardware ingress location 0/1/CPU0
ipv6 access-list Test
10 permit fec0:0:0:2::/64 any (24303 matches) (2459695 byte matches)
20 permit any any (13 matches) (246 byte matches)
```

In the following example, the IPv6 access list is displayed using **detail** keyword:

```
Router# show access-lists ipv6 v6t1 hardware ingress detail location 0/0/CPU0
v6t1 Details:
Sequence Number: 10
NPU ID: 0
Number of DPA Entries: 1
ACL ID: 1
ACE Action: PERMIT
ACE Logging: DISABLED
ABF Action: 0(ABF_NONE)
Hit Packet Count: 0 Byte Count: 0
Source Address: 0:0:0:0::
 Source Address Mask: 0:0:0:0::
Destination Address: 2222:0:0:0::
 Destination Address Mask: ffff:ffff:ffff:ffff::
DPA Entry: 1
        Entry Index: 0
        DPA Handle: 0x8E3000A8
        DSCP: 0x28 (Mask 0xFC)
Sequence Number: 20
NPU ID: 0
Number of DPA Entries: 1
ACL ID: 1
ACE Action: PERMIT
ACE Logging: DISABLED
ABF Action: 0(ABF_NONE)
Hit Packet Count: 0 Byte Count: 0
TCP Flags: 0x01 (Mask 0x01)
Protocol: 0x06 (Mask 0xFF)
Source Address: 0:0:0:0::
 Source Address Mask: 0:0:0:0::
Destination Address: 2222:0:0:0::
 Destination Address Mask: ffff:ffff:ffff:ffff::
DPA Entry: 1
        Entry Index: 0
        DPA Handle: 0x8E300390
Sequence Number: IMPLICIT NDNA PERMIT
NPU ID: 0
Number of DPA Entries: 1
ACL ID: 1
ACE Action: PERMIT
ACE Logging: DISABLED
ABF Action: 0(ABF_NONE)
Hit Packet Count: 0 Byte Count: 0
Protocol: 0x3A (Mask 0xFF)
```

```
Source Address: 0:0:0:0::
 Source Address Mask: 0:0:0:0::
Destination Address: 0:0:0:0::
 Destination Address Mask: 0:0:0:0::
DPA Entry: 1
        Entry Index: 0
        DPA Handle: 0x8E300678
Sequence Number: IMPLICIT NDNS PERMIT
NPU ID: 0
Number of DPA Entries: 1
ACL ID: 1
ACE Action: PERMIT
ACE Logging: DISABLED
ABF Action: 0(ABF_NONE)
Hit Packet Count: 0 Byte Count: 0
Protocol: 0x3A (Mask 0xFF)
Source Address: 0:0:0:0::
 Source Address Mask: 0:0:0:0::
Destination Address: 0:0:0:0::
 Destination Address Mask: 0:0:0:0::
DPA Entry: 1
        Entry Index: 0
        DPA Handle: 0x8E300960
Sequence Number: IMPLICIT DENY
NPU ID: 0
Number of DPA Entries: 1
ACL ID: 1
ACE Action: DENY
ACE Logging: DISABLED
ABF Action: 0(ABF_NONE)
Hit Packet Count: 0 Byte Count: 0
Source Address: 0:0:0:0::
 Source Address Mask: 0:0:0:0::
Destination Address: 0:0:0:0::
 Destination Address Mask: 0:0:0:0::
DPA Entry: 1
        Entry Index: 0
        DPA Handle: 0x8E300C48
```