database archive through dns

deny

To set conditions in a named IP access list or object group access control list (OGACL) that will deny packets, use the **deny** configuration command in the appropriate configuration mode. To remove a deny condition from an IP access list or OGACL, use the **no** form of this command.

deny *protocol* {*src-addr src-wildcard*| **object-group** *object-group-name*| **any**| **host** {*addr*| *name*}} {*dest-addr dest-wildcard*| **any**| **eq** *port*| **gt** *port*| **host** {*addr*| *name*}| **lt** *port*| **neq** *port*| **portgroup** *srcport-groupname*| **object-group** *dest-addr-groupname*| **range** *port*| [**dscp** *type*| **fragments**| **option** *option*| **precedence** *precedence*| **log**| **log-input**| **time-range** *time-range-name*| **tos** *tos*| **ttl** *ttl-value*]}

no deny *protocol* {*src-addr src-wildcard*| **object-group** *object-group-name*| **any**| **host** {*addr*| *name*}} {*dest-addr dest-wildcard*| **any**| **eq** *port*| **gt** *port*| **host** {*addr*| *name*}| **lt** *port*| **neq** *port*| **portgroup** *srcport-groupname*| **object-group** *dest-addr-groupname*| **range** *port*| [**dscp** *type*| **fragments**| **option** *option*| **precedence** *precedence*| **log**| **log-input**| **time-range** *time-range-name*| **tos** *tos*| **ttl** *ttl-value*]}

Syntax Description

protocol	Name or number of a protocol; valid values are **eigrp**, **gre**, **icmp**, **igmp**, **igrp**, **ip**, **ipinip**, **nos**, **ospf**, **tcp**, or **udp**, or an integer in the range 0 to 255 representing an IP protocol number. To match any Internet protocol (including Internet Control Message Protocol (ICMP), TCP, and User Datagram Protocol (UDP), use the keyword **ip**. See the "Usage Guidelines" section for additional qualifiers.
src-addr	Number of the source network or host from which the packet is being sent in a 32-bit quantity in four-part, dotted-decimal format.
src-wildcard	Wildcard bits to be applied to source network in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore.
object-group *object-group-name*	Specifies the source or destination name of the object group.
any	Specifies any source or any destination host as an abbreviation for the *source-addr*or *destination-addr value*and the *source-wildcard* or *destination-wildcard* value of 0.0.0.0 255.255.255.255.
host *addr*	Specifies the source or destination address of a single host.
host *name*	Specifies the source or destination name of a single host.
tcp	Specifies the TCP protocol.

udp	Specifies the UDP protocol.
object-group *source-addr-group-name*	Specifies the source address group name.
destination-addr	Number of the network or host to which the packet is being sent in a 32-bit quantity in four-part, dotted-decimal format.
destination-wildcard	Wildcard bits to be applied to the destination in a 32-bit quantity in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore.
eq *port*	Matches only packets on a given port number; see the "Usage Guidelines" section for valid values.
gt *port*	Matches only the packets with a greater port number; see the "Usage Guidelines" section for valid values.
lt *port*	Matches only the packets with a lower port number; see the "Usage Guidelines" section for valid values.
neq *port*	Matches only the packets that are not on a given port number; see the "Usage Guidelines" section for valid values.
portgroup *srcport-group-name*	Specifies the source port object group name.
object-group *dest-addr-group-name*	Specifies the destination address group name.
portgroup *destport-group-name*	Specifies the destination port object group name.
dscp *type*	(Optional) Matches the packets with the given Differentiated Services Code Point (DSCP) value; see the "Usage Guidelines" section for valid values.
fragments	(Optional) Applies the access list entry to noninitial fragments of packets; the fragment is either permitted or denied accordingly. For more details about the **fragments** keyword, see the "Access List Processing of Fragments" and "deny, on page 2" sections in the "Usage Guidelines" section.
option *option*	(Optional) Matches the packets with the given IP options value number; see the "Usage Guidelines" section for valid values.
precedence *precedence*	(Optional) Specifies the precedence filtering level for packets; valid values are a number from 0 to 7 or by a name. See the "Usage Guidelines" section for a list of valid names.

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 for a standard list includes the access list number, whether the packet was permitted or denied, the source address, and the number of packets. The message for an extended list includes the access list number; whether the packet was permitted or denied; the protocol; whether the protocol was TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and source and destination port numbers. For both standard and extended lists, the message is generated for the first packet that matches, and then at 5-minute intervals, including the number of packets permitted or denied in the prior 5-minute interval. The logging facility might drop some logging message packets if there are too many to be handled or if there is more than one logging message to be handled in 1 second. This behavior prevents the router from reloading because of too many logging packets. Therefore, the logging facility should not be used as a billing tool or an accurate source of the number of matches to an access list.
log-input	(Optional) Matches the log against this entry, including the input interface.
time-range *time-range-name*	(Optional) Specifies a time-range entry name.
tos *tos*	(Optional) Specifies the service filtering level for packets; valid values are a number from 0 to 15 or by a name as listed in the "Usage Guidelines" section of the **access-list**(IP extended) command.
option option	(Optional) Matches packets with the IP options value; see the "Usage Guidelines" section for the valid values.
fragments	(Optional) Applies the access list entry to noninitial fragments of packets; the fragment is either permitted or denied accordingly. For more details about the **fragments** keyword, see the deny, on page 2 and "deny, on page 2" sections in the "Usage Guidelines" section.

ttl *ttl-value*	(Optional) Matches packets with a given Time-to-live (ttl) value.

Command Default

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

Command Modes

Standard access-list configuration (config-std-nacl) Extended access-list configuration (config-ext-nacl)

Command History

Release	Modification
12.4(20)T	This command was introduced.

Usage Guidelines

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

The **portgroup** keyword appears only when you configure an extended ACL.

The *address* or *object-group-name* value is created using the **object-group** command.

The **object-group** *object-group-name*keyword and argument allow you to create logical groups of users (or servers), which you can use to define access policy using ACLs. For example, with one ACL entry you can permit the object group named engineering to access all engineering servers. Otherwise, you would need one ACL entry for every person in the engineering group.

If the operator is positioned after the *source-addr* and *source-wildcard* values, it must match the source port.

If the operator is positioned after the *destination-addr* and *destination-wildcard* values, it must match the destination port.

If you are entering the port number of a TCP or UDP port, you can enter the decimal number or name of a TCP or UDP port. A port number is a number from 0 to 65535. TCP and UDP port names are listed in the "Usage Guidelines" section of the **access-list**(IP extended) command. TCP port names can be used only when filtering TCP. UDP port names can be used only when filtering UDP.

The valid values for the **dscp** *type* keyword and argument are as follows:

- 0 to 63--Differentiated services code point 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** --Matches the patches with the AF23 dscp (010110).
- **af31** --Matches the patches with the AF31 dscp (011010).
- **af32** --Matches the patches with the AF32 dscp (011100).

- **af33** --Matches the patches with the AF33 dscp (011110).
- **af41** --Matches the patches with the AF41 dscp (100010).
- **af42** --Matches the patches with the AF42 dscp (100100).
- **af43** --Matches the patches with the AF43 dscp (100110).
- **cs1** --Matches the patches with the CS1 (precedence 1) dscp (001000).
- **cs2** --Matches the patches with the CS2 (precedence 2) dscp (010000).
- **cs3** --Matches the patches with the CS3 (precedence 3) dscp (011000).
- **cs4** --Matches the patches with the CS4 (precedence 4) dscp (100000).
- **cs5** --Matches the patches with the CS5 (precedence 5) dscp (101000).
- **cs6** --Matches the patches with the CS6 (precedence 6) dscp (110000).
- **cs7** --Matches the patches with the CS7 (precedence 7) dscp (111000).
- **default** --Matches the patches with the default dscp (000000).
- **ef** --Matches the patches with the EF dscp (101110).

The valid values for the **eq** *port* keyword and argument are as follows:

- 0 to 65535--Port number.
- **bgp** --Border Gateway Protocol (179).
- **chargen** --Character generator (19).
- **cmd** --Remote commands (rcmd, 514).
- **daytime** --Daytime (13).
- **discard** --Discard (9).
- **domain** --Domain Name Service (53).
- **echo** --Echo (7).
- **exec** --Exec (rsh, 512).
- **finger** --Finger (79).
- **ftp** --File Transfer Protocol (21).
- **ftp-data** --FTP data connections (20).
- **gopher** --Gopher (70).
- **hostname** --NIC hostname server (101).
- **ident** --Ident Protocol (113).
- **irc** --Internet Relay Chat (194).
- **klogin** --Kerberos login (543).
- **kshell** --Kerberos shell (544).
- **login** --Login (rlogin, 513).

- **lpd** --Printer service (515).
- **nntp** --Network News Transport Protocol (119).
- **pim-auto-rp** --PIM Auto-RP (496).
- **pop2** --Post Office Protocol v2 (109).
- **pop3** --Post Office Protocol v3 (110).
- **smtp** --Simple Mail Transport Protocol (25).
- **sunrpc** --Sun Remote Procedure Call (111).
- **syslog** --Syslog (514).
- **tacacs** --TAC Access Control System (49).
- **talk** --Talk (517).
- **telnet** --Telnet (23).
- **time** --Time (37).
- **uucp** --Unix-to-Unix Copy Program (540).
- **whois** --Nicname (43).
- **www** --World Wide Web (HTTP, 80).

The valid values for the **gt** *port* keyword and argument are as follows:

- 0-65535--Port number.
- **biff** --Biff (mail notification, comsat, 512).
- **bootpc** --Bootstrap Protocol (BOOTP) client (68).
- **bootps** --Bootstrap Protocol (BOOTP) server (67).
- **discard** --Discard (9).
- **dnsix** --DNSIX security protocol auditing (195).
- **domain** --Domain Name Service (DNS, 53).
- **echo** --Echo (7).
- **isakmp** --Internet Security Association and Key Management Protocol (500).
- **mobile-ip** --Mobile IP registration (434).
- **nameserver** --IEN116 name service (obsolete, 42).
- **netbios-dgm** --NetBios datagram service (138).
- **netbios-ns** --NetBios name service (137).
- **netbios-ss** --NetBios session service (139).
- **non500-isakmp** --Internet Security Association and Key Management Protocol (4500).
- **ntp** --Network Time Protocol (123).
- **pim-auto-rp** --PIM Auto-RP (496).

- **rip** --Routing Information Protocol (router, in.routed, 520).
- **snmp** --Simple Network Management Protocol (161).
- **snmptrap** --SNMP Traps (162).
- sunrpc--Sun Remote Procedure Call (111).
- **syslog** --System Logger (514).
- **tacacs** --TAC Access Control System (49).
- **talk** --Talk (517).
- **tftp** --Trivial File Transfer Protocol (69).
- **time** --Time (37).
- **who** --Who service (rwho, 513).
- **xdmcp** --X Display Manager Control Protocol (177).

The valid values for the **lt** *port* keyword and argument are as follows:

- 0-65535--Port number.
- **biff** --Biff (mail notification, comsat, 512).
- **bootpc** --Bootstrap Protocol (BOOTP) client (68).
- **bootps** --Bootstrap Protocol (BOOTP) server (67).
- **discard** --Discard (9).
- **dnsix** --DNSIX security protocol auditing (195).
- **domain** --Domain Name Service (DNS, 53).
- **echo** --Echo (7).
- **isakmp** --Internet Security Association and Key Management Protocol (500).
- **mobile-ip** --Mobile IP registration (434).
- **nameserver** --IEN116 name service (obsolete, 42).
- **netbios-dgm** --NetBios datagram service (138).
- **netbios-ns** --NetBios name service (137).
- **netbios-ss** --NetBios session service (139).
- **non500-isakmp** --Internet Security Association and Key Management Protocol (4500).
- **ntp** --Network Time Protocol (123).
- **pim-auto-rp** --PIM Auto-RP (496).
- **rip** --Routing Information Protocol (router, in.routed, 520).
- **snmp** --Simple Network Management Protocol (161).
- **snmptrap** --SNMP Traps (162).
- **sunrpc** --Sun Remote Procedure Call (111).

- **syslog** --System Logger (514).
- **tacacs** --TAC Access Control System (49).
- **talk** --Talk (517).
- **tftp** --Trivial File Transfer Protocol (69).
- **time** --Time (37).
- **who** --Who service (rwho, 513).
- **xdmcp** --X Display Manager Control Protocol (177).

The valid values for the **neg** *port* keyword and argument are as follows:

- 0 to 65535--Port number.
- **biff** --Biff (mail notification, comsat, 512).
- **bootpc** --Bootstrap Protocol (BOOTP) client (68).
- **bootps** --Bootstrap Protocol (BOOTP) server (67).
- **discard** --Discard (9).
- **dnsix** --DNSIX security protocol auditing (195).
- **domain** --Domain Name Service (DNS, 53).
- **echo** --Echo (7).
- **isakmp** --Internet Security Association and Key Management Protocol (500).
- **mobile-ip** --Mobile IP registration (434).
- **nameserver** --IEN116 name service (obsolete, 42).
- **netbios-dgm** --NetBios datagram service (138).
- **netbios-ns** --NetBios name service (137).
- **netbios-ss** --NetBios session service (139).
- **non500-isakmp** --Internet Security Association and Key Management Protocol (4500).
- **ntp** --Network Time Protocol (123).
- **pim-auto-rp** --PIM Auto-RP (496).
- **rip** --Routing Information Protocol (router, in.routed, 520).
- **snmp** --Simple Network Management Protocol (161).
- **snmptrap** --SNMP Traps (162).
- **sunrpc** --Sun Remote Procedure Call (111).
- **syslog** --System Logger (514).
- **tacacs** --TAC Access Control System (49).
- **talk** --Talk (517).
- **tftp** --Trivial File Transfer Protocol (69).

- **time** --Time (37).
- **who** --Who service (rwho, 513).
- **xdmcp** --X Display Manager Control Protocol (177).

The valid values for the **option** *option* keyword and argument are as follows:

- 0 to 255--IP Options value.
- **add-ext** --Matches the packets with Address Extension Option (147).
- **any-options** --Matches the packets with ANY Option.
- **com-security** --Matches the packets with Commercial Security Option (134).
- **dps** --Matches the packets with Dynamic Packet State Option (151).
- **encode** --Matches the packets with Encode Option (15).
- **eool** --Matches the packets with End of Options (0).
- **ext-ip** --Matches the packets with the Extended IP Option (145).
- **ext-security** --Matches the packets with the Extended Security Option (133).
- **finn** --Matches the packets with the Experimental Flow Control Option (205).
    - **imitd**--Matches the packets with IMI Traffic Desriptor Option (144).
    - **lsr**--Matches the packets with Loose Source Route Option (131).
    - **match-all**--Matches the packets if all specified flags are present.
    - **match-any**--Matches the packets if any specified flag is present.
    - **mtup**--Matches the packets with MTU Probe Option (11).
    - **mtur**--Matches the packets with MTU Reply Option (12).
    - **no-op**--Matches the packets with No Operation Option (1).
    - **psh**--Match the packets on the PSH bit.
    - **nsapa**--Matches the packets with NSAP Addresses Option (150).
    - **reflect**--Creates reflexive access list entry.
    - **record-route**--Matches the packets with Record Route Option (7).
    - **rst**--Matches the packets on the RST bit.
    - **router-alert**--Matches the packets with Router Alert Option (148).
    - **sdb**--Matches the packets with Selective Directed Broadcast Option (149).
    - **security**--Matches the packets with Basic Security Option (130).
    - **ssr**--Matches the packets with Strict Source Routing Option (137).
    - **stream-id**--Matches the packets with Stream ID Option (136).
    - **syn**--Match the packets on the SYN bit.
- **timestamp** --Matches the packets with the Time Stamp Option (68).

- **traceroute** --Matches the packets with the Trace Route Option (82).
- **ump** --Matches the packets with the Upstream Multicast Packet Option (152).
- **visa** --Matches the packets with the Experimental Access Control Option (142).
- **zsu** --Matches the packets with the Experimental Measurement Option (10).

The valid values for the **tos** *value* keyword and argument are as follows:

- 0 to 15--Type of service value.
- **max-reliability** --Matches the packets with the maximum reliable ToS (2).
- **max-throughput** --Matches the packets with the maximum throughput ToS (4).
- **min-delay** --Matches the packets with the minimum delay ToS (8).
- **min-monetary-cost** --Matches packets with the minimum monetary cost ToS (1).
- **normal** --Matches the packets with the normal ToS (0).

Access List or OGACL Processing of Fragments

The behavior of access-list entries regarding the use or lack of the **fragments** keyword are summarized in the table below:

Table 1: Access list or OGACL Processing of Fragments

If the Access-List Entry Has...	Then...

...no **fragments** keyword (the default behavior), and assuming all of the access-list entry information matches,	For an access-list entry containing only Layer 3 information:
	• The entry is applied to nonfragmented packets, initial fragments, and noninitial fragments.
	For an access list entry containing Layer 3 and Layer 4 information:
	• The entry is applied to nonfragmented packets and initial fragments:
	  • If the entry is a **permit**statement, the packet or fragment is permitted.
	  • If the entry is a **deny**statement, the packet or fragment is denied.
	• The entry is also applied to noninitial fragments in the following manner. Because noninitial fragments contain only Layer 3 information, only the Layer 3 portion of an access-list entry can be applied. If the Layer 3 portion of the access-list entry matches, and
	  • If the entry is a **permit**statement, the noninitial fragment is permitted.
	  • If the entry is a **deny** statement, the next access-list entry is processed.
	Note The **deny** statements are handled differently for noninitial fragments versus nonfragmented or initial fragments.

...the **fragments** keyword, and assuming all of the access-list entry information matches,	**Note** The access-list entry is applied only to noninitial fragments. The **fragments** keyword cannot be configured for an access-list entry that contains any Layer 4 information.

Be aware that you should not simply add the **fragments** keyword to every access list entry because the first fragment of the IP packet is considered a nonfragment and is treated independently of the subsequent fragments. An initial fragment will not match an access list **permit** or **deny** entry that contains the **fragments**keyword, the packet is compared to the next access list entry, and so on, until it is either permitted or denied by an access list entry that does not contain the **fragments** keyword. Therefore, you may need two access list entries for every **deny** entry. The first **deny** entry of the pair will not include the **fragments** keyword, and applies to the initial fragment. The second **deny** entry of the pair will include the **fragments** keyword and applies to the

subsequent fragments. In the cases where there are multiple **deny** access-list entries for the same host but with different Layer 4 ports, a single **deny** access-list entry with the **fragments** keyword for that host is all that needs to be added. Thus all the fragments of a packet are handled in the same manner by the access list.

Packet fragments of IP datagrams are considered individual packets and each counts individually as a packet in access list accounting and access list violation counts.

Note The **fragments**keyword cannot solve all cases involving access lists and IP fragments.

Fragments and Policy Routing

Fragmentation and the fragment control feature affect policy routing if the policy routing is based on the **match ip address**s command and the access list had entries that match on Layer 4 through 7 information. It is possible that noninitial fragments pass the access list and are policy routed, even if the first fragment was not policy routed or the reverse.

By using the **fragments** keyword in access list entries as described earlier, a better match between the action taken for initial and noninitial fragments can be made and it is more likely policy routing will occur as intended.

The **portgroup** *srcport-groupname or* **portgroup** *destport-groupname*keywords and arguments allow you to create an object group based on a source or destination group.

Examples

The following example creates an access list that denies all TCP packets:

```
Router> enable
Router# configure terminal
Router(config)# ip access-list extended my_ogacl_policy
Router(config-ext-nacl)# deny tcp any any
Router(config-ext-nacl)# exit
Router(config)# exit
```

Related Commands

Command	Description
ip access-group	Applies an ACL or OGACL to an interface or a service policy map.
ip access-list	Defines an IP access list or OGACL by name or number.
object-group network	Defines network object groups for use in OGACLs.
object-group service	Defines service object groups for use in OGACLs.
permit	Sets conditions in a named IP access list or OGACL that will permit packets.
show ip access-list	Displays the contents of IP access lists or OGACLs.

Command	Description
show object-group	Displays information about object groups that are configured.

deny (IP)

To set conditions in a named IP access list that will deny packets, use the **deny** command in access list configuration mode. To remove a deny condition from an access list, use the **no** form of this command.

[*sequence-number*] **deny** *source* [*source-wildcard*]

[*sequence-number*] **deny** *protocol source source-wildcard destination destination-wildcard* [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] **[log]** [**time-range** *time-range-name*] **[fragments]**

no *sequence-number*

no deny *source* [*source-wildcard*]

no deny *protocol source source-wildcard destination destination-wildcard*

Internet Control Message Protocol (ICMP)

[*sequence-number*] **deny icmp** *source source-wildcard destination destination-wildcard* [*icmp-type* [*icmp-code*]| *icmp-message*] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] **[log]** [**time-range** *time-range-name*] **[fragments]**

Internet Group Management Protocol (IGMP)

[*sequence-number*] **deny igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] **[log]** [**time-range** *time-range-name*] **[fragments]**

Transmission Control Protocol (TCP)

[sequence-number] deny tcp *source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established** {**match-any**| **match-all**} {+-} *flag-name*| **precedence** *precedence*| **tos** *tos*| **ttl** *operator value*| **log**| **time-range** *time-range-name*| **fragments**]

User Datagram Protocol (UDP)

[*sequence-number*] **deny udp** *source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] **[log]** [**time-range** *time-range-name*] **[fragments]**

Syntax Description

sequence-number	(Optional) Sequence number assigned to the deny statement. The sequence number causes the system to insert the statement in that numbered position in the access list.

source	Number of the network or host from which the packet is being sent. There are three alternative ways to specify the source: • Use a 32-bit quantity in four-part dotted-decimal format. • Use the **any**keyword as an abbreviation for a *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. • Use **host** *source*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 1s in the bit positions that 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 **host** *source*as an abbreviation for a *source*and *source-wildcard*of *source*0.0.0.0.
protocol	Name or number of an Internet protocol. The *protocol* argument can be one of the keywords **eigrp**, **gre**, **icmp**, **igmp**, **ip**, **ipinip**, **nos**, **ospf**, **tcp**, or **udp**, or an integer in the range from 0 to 255 representing an Internet protocol number. To match any Internet protocol (including ICMP, TCP, and UDP), use the **ip**keyword. **Note** When the **icmp**, **igmp**, **tcp,** and **udp** keywords are entered, they must be followed with the specific command syntax that is shown for the ICMP, IGMP, TCP, and UDP forms of the **deny** command.
icmp	Denies only ICMP packets. When you enter the **icmp** keyword, you must use the specific command syntax shown for the ICMP form of the **deny** command.
igmp	Denies only IGMP packets. When you enter the **igmp** keyword, you must use the specific command syntax shown for the IGMP form of the **deny** command.
tcp	Denies only TCP packets. When you enter the **tcp** keyword, you must use the specific command syntax shown for the TCP form of the **deny** command.

udp	Denies only UDP packets. When you enter the **udp** keyword, you must use the specific command syntax shown for the UDP form of the **deny** command.
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 **host** *destination* 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 1s in the bit positions that 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 **host** *destination* as an abbreviation for a *destination*and *destination-wildcard* of *destination* 0.0.0.0.
option *option-name*	(Optional) Packets can be filtered by IP Options, as specified by a number from 0 to 255 or by the corresponding IP Option name, as listed in the table in the “Usage Guidelines” section.
precedence *precedence*	(Optional) Packets can be filtered by precedence level, as specified by a number from 0 to 7 or by a name.
tos *tos*	(Optional) Packets can be filtered by type of service (ToS) level, as specified by a number from 0 to 15, or by a name as listed in the “Usage Guidelines” section of the **access-list**(IP extended) command.

ttl *operator value*	(Optional) Compares the TTL value in the packet to the TTL value specified in this **deny** statement. • The *operator* can be **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), or **range** (inclusive range). • The *value* can range from 0 to 255. • If the operator is **range**, specify two values separated by a space. • For Release 12.0S, if the operator is **eq** or **neq**, only one TTL value can be specified. • For all other releases, if the operator is **eq** or **neq**, as many as 10 TTL values can be specified, separated by a space. If the TTL in the packet matches just one of the possibly 10 values, the entry is considered to be matched.
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.)
time-range *time-range-name*	(Optional) Name of the time range that applies to this **deny** statement. The name of the time range and its restrictions are specified by the **time-range** and **absolute** or **periodic** commands, respectively.
fragments	(Optional) The access list entry applies to noninitial fragments of packets; the fragment is either permitted or denied accordingly. For more details about the **fragments** keyword, see the "deny (IP), on page 15" and "deny (IP), on page 15" sections in the "Usage Guidelines" section.
icmp-type	(Optional) ICMP packets can be filtered by ICMP message type. The type is a number from 0 to 255.
icmp-code	(Optional) ICMP packets that are filtered by ICMP message type can also be filtered by the ICMP message code. The code is a number from 0 to 255.
icmp-message	(Optional) ICMP packets can be filtered by an ICMP message type name or an ICMP message type and code name. The possible names are listed in the "Usage Guidelines" section of the **access-list**(IP extended) command.

igmp-type	(Optional) IGMP packets can be filtered by IGMP message type or message name. A message type is a number from 0 to 15. IGMP message names are listed in the "Usage Guidelines" section of the **access-list**(IP extended) command.
operator	(Optional) Compares source or destination ports. Operators include **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 arguments, it must match the source port. If the operator is positioned after the destination and destination-wildcard arguments, it must match the destination port. The range operator requires two port numbers. Up to ten port numbers can be entered for the **eq** (equal) and **neq** (not equal) operators. All other operators require one port number.
port	(Optional) The decimal number or name of a TCP or UDP port. A port number is a number from 0 to 65535. TCP and UDP port names are listed in the "Usage Guidelines" section of the **access-list**(IP extended) command. 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. A match occurs if the TCP datagram has the ACK or RST bit set. The nonmatching case is that of the initial TCP datagram to form a connection. **Note** The **established** keyword can be used only with the old command-line interface (CLI) format. To use the new CLI format, you must use the **match-any** or **match-all** keywords followed by the **+** or **-** keywords and *flag-name* argument.

match-any | **match-all**	(Optional) For the TCP protocol only: A match occurs if the TCP datagram has certain TCP flags set or not set. You use the **match-any**keyword to allow a match to occur if any of the specified TCP flags are present, or you can use the **match-all**keyword to allow a match to occur only if all of the specified TCP flags are present. You must follow the **match-any**and **match-all**keywords with the **+**or **-**keyword and the *flag-name*argument to match on one or more TCP flags.
+ | **-** *flag-name*	(Optional) For the TCP protocol only: The **+** keyword allows IP packets if their TCP headers contain the TCP flags that are specified by the *flag-name* argument. The **-** keyword filters out IP packets that do not contain the TCP flags specified by the *flag-name* argument. You must follow the **+** and **-** keywords with the *flag-name* argument. TCP flag names can be used only when filtering TCP. Flag names for the TCP flags are as follows: **urg**, **ack**, **psh**, **rst**, **syn**, and **fin**.

Command Default

There are no specific conditions under which a packet is denied passing the named access list.

Command Modes

Access list configuration

Command History

Release	Modification
11.2	This command was introduced.
12.0(1)T	The **time-range** *time-range-name*keyword and argument were added.
12.0(11)	The **fragments**keyword was added.
12.2(13)T	The igrp keyword was removed because the IGRP protocol is no longer available in Cisco IOS software.
12.2(14)S	The *sequence-number*argument was added.
12.2(15)T	The *sequence-number*argument was added.
12.3(4)T	The **option** *option-name* keyword and argument were added. The **match-any**, **match-all,** **+**,and **-**keywords and the *flag-name* argument were added.
12.3(7)T	Command functionality was modified to allow up to ten port numbers to be added after the **eq** and **neq** operators so that an access list entry can be created with noncontiguous ports.

Release	Modification
12.4(2)T	The **ttl** *operator value* keyword and arguments were added.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use this command following the **ip access-list** command to specify conditions under which a packet cannot pass the named access list.

The **time-range** keyword allows you to identify a time range by name. The **time-range**, **absolute**, and **periodic** commands specify when this **deny** statement is in effect.

log Keyword

A log 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, and then at 5-minute intervals, including the number of packets permitted or denied in the prior 5-minute interval.

Use the **ip access-list log-update** command to generate logging messages when the number of matches reaches a configurable threshold (rather than waiting for a 5-minute-interval). See the **ip access-list log-update** command for more information.

The logging facility might drop some logging message packets if there are too many to be handled or if there is more than one logging message to be handled in 1 second. This behavior prevents the router from crashing because of too many logging packets. Therefore, the logging facility should not be used as a billing tool or an accurate source of the number of matches to an access list.

If you enable Cisco Express Forwarding (CEF) and then create an access list that uses the **log** keyword, the packets that match the access list are not CEF-switched. They are fast-switched. Logging disables CEF.

Access List Filtering of IP Options

Access control lists can be used to filter packets with IP Options to prevent routers from being saturated with spurious packets containing IP Options. To see a complete table of all IP Options, including ones currently not in use, refer to the latest Internet Assigned Numbers Authority (IANA) information that is available from its URL: www.iana.org.

Cisco IOS software allows you to filter packets according to whether they contain one or more of the legitimate IP Options by entering either the IP Option value or the corresponding name for the *option-name* argument as shown in the table below.

Table 2: IP Option Values and Names

IP Option Value or Name	Description
0 to 255	IP Options values.

IP Option Value or Name	Description
add-ext	Match packets with Address Extension Option (147).
any-options	Match packets with any IP Option.
com-security	Match packets with Commercial Security Option (134).
dps	Match packets with Dynamic Packet State Option (151).
encode	Match packets with Encode Option (15).
eool	Match packets with End of Options (0).
ext-ip	Match packets with Extended IP Options (145).
ext-security	Match packets with Extended Security Option (133).
finn	Match packets with Experimental Flow Control Option (205).
imitd	Match packets with IMI Traffic Descriptor Option (144).
lsr	Match packets with Loose Source Route Option (131).
mtup	Match packets with MTU Probe Option (11).
mtur	Match packets with MTU Reply Option (12).
no-op	Match packets with No Operation Option (1).
nsapa	Match packets with NSAP Addresses Option (150).
psh	Matches the packets on the PSH bit.
record-route	Match packets with Router Record Route Option (7).
reflect	Creates reflexive access list entry.
rst	Matches the packets on the RST bit.
router-alert	Match packets with Router Alert Option (148).
sdb	Match packets with Selective Directed Broadcast Option (149).
security	Match packets with Base Security Option (130).

IP Option Value or Name	Description
ssr	Match packets with Strict Source Routing Option (137).
stream-id	Match packets with Stream ID Option (136).
syn	Matches the packets on the SYN bit.
timestamp	Match packets with Time Stamp Option (68).

Filtering IP Packets Based on TCP Flags

The access list entries that make up an access list can be configured to detect and drop unauthorized TCP packets by allowing only the packets that have very specific groups of TCP flags set or not set. Users can select any desired combination of TCP flags with which to filter TCP packets. Users can configure access list entries in order to allow matching on a flag that is set and on a flag that is not set. Use the **+** and **-** keywords with a flag name to specify that a match is made based on whether a TCP header flag has been set. Use the **match-any** and **match-all** keywords to allow the packet if any or all, respectively, of the flags specified by the **+** or **-** keyword and *flag-name* argument have been set or not set.

Access List Processing of Fragments

The behavior of access list entries regarding the use or lack of use of the **fragments** keyword can be summarized as follows:

If the Access-List Entry Has...	Then...

...no **fragments** keyword (the default behavior), and assuming all of the access-list entry information matches,	For an access list entry that contains only Layer 3 information:
	• The entry is applied to nonfragmented packets, initial fragments, and noninitial fragments.
	
	For an access list entry that contains Layer 3 and Layer 4 information:
	• The entry is applied to nonfragmented packets and initial fragments.
	  • If the entry is a **permit**statement, then the packet or fragment is permitted.
	  • If the entry is a **deny**statement, then the packet or fragment is denied.
	• The entry is also applied to noninitial fragments in the following manner. Because noninitial fragments contain only Layer 3 information, only the Layer 3 portion of an access list entry can be applied. If the Layer 3 portion of the access list entry matches, and
	  • If the entry is a **permit**statement, then the noninitial fragment is permitted.
	  • If the entry is a **deny** statement, then the next access list entry is processed.
	
	Note The **deny** statements are handled differently for noninitial fragments versus nonfragmented or initial fragments.

...the **fragments** keyword, and assuming all of the access-list entry information matches,	The access list entry is applied only to noninitial fragments. The **fragments** keyword cannot be configured for an access list entry that contains any Layer 4 information.

Be aware that you should not add the **fragments** keyword to every access list entry because the first fragment of the IP packet is considered a nonfragment and is treated independently of the subsequent fragments. An initial fragment will not match an access list **permit** or **deny** entry that contains the **fragments**keyword. The packet is compared to the next access list entry, and so on, until it is either permitted or denied by an access list entry that does not contain the **fragments** keyword. Therefore, you may need two access list entries for every **deny** entry. The first **deny** entry of the pair will not include the **fragments** keyword and applies to the initial fragment. The second **deny** entry of the pair will include the **fragments** keyword and applies to the subsequent fragments. In the cases in which there are multiple **deny** access list entries for the same host but with different Layer 4 ports, a single **deny** access list entry with the **fragments** keyword for that host is all that needs to be added. Thus all the fragments of a packet are handled in the same manner by the access list.

Packet fragments of IP datagrams are considered individual packets, and each counts individually as a packet in access list accounting and access list violation counts.

Note The **fragments**keyword cannot solve all cases that involve access lists and IP fragments.

Fragments and Policy Routing

Fragmentation and the fragment control feature affect policy routing if the policy routing is based on the **match ip address** command and the access list has entries that match on Layer 4 through 7 information. It is possible that noninitial fragments pass the access list and are policy-routed, even if the first fragment is not policy-routed.

By using the **fragments** keyword in access list entries as described earlier, a better match between the action taken for initial and noninitial fragments can be made, and it is more likely that policy routing will occur as intended.

Creating an Access List Entry with Noncontiguous Ports

For Cisco IOS Release 12.3(7)T and later releases, you can specify noncontiguous ports on the same access control entry, which greatly reduces the number of access list entries required for the same source address, destination address, and protocol. If you maintain large numbers of access list entries, we recommend that you consolidate them when possible by using noncontiguous ports. You can specify up to ten port numbers following the **eq** and **neq** operators.

Examples

The following example sets conditions for a standard access list named Internetfilter:

```
ip access-list standard Internetfilter
 deny 192.168.34.0  0.0.0.255
 permit 172.16.0.0  0.0.255.255
 permit 10.0.0.0  0.255.255.255
! (Note: all other access implicitly denied.)
```

The following example denies HTTP traffic on Monday through Friday from 8:00 a.m. to 6:00 p.m.:

```
time-range no-http
 periodic weekdays 8:00 to 18:00
!
ip access-list extended strict
 deny tcp any any eq http time-range no-http
!
interface ethernet 0
 ip access-group strict in
```

The following example adds an entry with the sequence number 25 to extended IP access list 150:

```
ip access-list extended 150
 25 deny ip host 172.16.3.3 host 192.168.5.34
```

The following example removes the entry with the sequence number 25 from the extended access list example shown above:

```
 no 25
```

The following example sets a deny condition for an extended access list named filter2. The access list entry specifies that a packet cannot pass the named access list if it contains the Strict Source Routing IP Option, which is represented by the IP option value ssr.

```
ip access-list extended filter2
 deny ip any any option ssr
```

The following example sets a deny condition for an extended access list named kmdfilter1. The access list entry specifies that a packet cannot pass the named access list if the RST and FIN TCP flags have been set for that packet:

```
ip access-list extended kmdfilter1
 deny tcp any any match-any +rst +fin
```

The following example shows several **deny** statements that can be consolidated into one access list entry with noncontiguous ports. The **show access-lists** command is entered to display a group of access list entries for the access list named abc.

```
Router# show access-lists abc
Extended IP access list abc
 10 deny tcp any eq telnet any eq 450
 20 deny tcp any eq telnet any eq 679
 30 deny tcp any eq ftp any eq 450
 40 deny tcp any eq ftp any eq 679
```

Because the entries are all for the same **deny** statement and simply show different ports, they can be consolidated into one new access list entry. The following example shows the removal of the redundant access list entries and the creation of a new access list entry that consolidates the previously displayed group of access list entries:

```
ip access-list extended abc
 no 10
 no 20
 no 30
 no 40
 deny tcp any eq telnet ftp any eq 450 679
```

The following examples shows the creation of the consolidated access list entry:

```
Router# show access-lists abc
Extended IP access list abc
 10 deny tcp any eq telnet ftp any eq 450 679
```

The following access list filters IP packets containing Type of Service (ToS) level 3 with TTL values 10 and 20. It also filters IP packets with a TTL greater than 154 and applies that rule to noninitial fragments. It permits IP packets with a precedence level of flash and a TTL not equal to 1, and sends log messages about such packets to the console. All other packets are denied.

```
ip access-list extended canton
 deny ip any any tos 3 ttl eq 10 20
 deny ip any any ttl gt 154 fragments
 permit ip any any precedence flash ttl neq 1 log
```

Related Commands

Command	Description
absolute	Specifies an absolute time when a time range is in effect.
access-list (IP extended)	Defines an extended IP access list.
access-list (IP standard)	Defines a standard IP access list.
ip access-group	Controls access to an interface.
ip access-list	Defines an IP access list by name.

Command	Description
ip access-list log-update	Sets the threshold number of packets that cause a logging message.
ip access-list resequence	Applies sequence numbers to the access list entries in an access list.
ip options	Drops or ignores IP Options packets that are sent to the router.
logging console	Sends system logging (syslog) messages to all available TTY lines and limits messages based on severity.
match ip address	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, or performs policy routing on packets.
periodic	Specifies a recurring (weekly) time range for functions that support the time-range feature.
permit (IP)	Sets conditions under which a packet passes a named IP access list.
remark	Writes a helpful comment (remark) for an entry in a named IP access list.
show access-lists	Displays a group of access-list entries.
show ip access-list	Displays the contents of all current IP access lists.
time-range	Specifies when an access list or other feature is in effect.

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.

deny *protocol* {*source-ipv6-prefix*/*prefix-length*| **any**| **host** *source-ipv6-address*| **auth**} [*operator* [*port-number*]] {*destination-ipv6-prefix*/*prefix-length*| **any**| **host** *destination-ipv6-address*| **auth**} [*operator* [*port-number*]] [**dest-option-type** [*doh-number*| *doh-type*]] [**dscp** *value*] [**flow-label** *value*] **[fragments] [hbh] [log] [log-input] [mobility]** [**mobility-type** [*mh-number*| *mh-type*]] **[routing]** [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] **[undetermined-transport]**

no deny *protocol* {*source-ipv6-prefix*/*prefix-length*| **any**| **host** *source-ipv6-address*| **auth**} [*operator* [*port-number*]] {*destination-ipv6-prefix*/*prefix-length*| **any**| **host** *destination-ipv6-address*| **auth**} [*operator* [*port-number*]] [**dest-option-type** [*doh-number*| *doh-type*]] [**dscp** *value*] [**flow-label** *value*] **[fragments] [hbh] [log] [log-input] [mobility]** [**mobility-type** [*mh-number*| *mh-type*]] **[routing]** [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] **[undetermined-transport]**

Internet Control Message Protocol

deny icmp {*source-ipv6-prefix*/*prefix-length*| **any**| **host** *source-ipv6-address*| **auth**} [*operator* [*port-number*]] {*destination-ipv6-prefix*/*prefix-length*| **any**| **host** *destination-ipv6-address*| **auth**} [*operator* [*port-number*]] [*icmp-type* [*icmp-code*]| *icmp-message*] [**dest-option-type** [*doh-number*| *doh-type*]] [**dscp** *value*] [**flow-label** *value*] **[fragments] [hbh] [log] [log-input] [mobility]** [**mobility-type** [*mh-number*| *mh-type*]] **[routing]** [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]

Transmission Control Protocol

deny tcp {*source-ipv6-prefix*/*prefix-length*| **any**| **host** *source-ipv6-address*| **auth**} [*operator* [*port-number*]] {*destination-ipv6-prefix*/*prefix-length*| **any**| **host** *destination-ipv6-address*| **auth**} [*operator* [*port-number*]] **[ack]** [**dest-option-type** [*doh-number*| *doh-type*]] [**dscp** *value*] **[established] [fin]** [**flow-label** *value*] **[fragments] [hbh] [log] [log-input] [mobility]** [**mobility-type** [*mh-number*| *mh-type*]] [**neq** {*port*| *protocol*}] **[psh]** [**range** {*port*| *protocol*}] **[routing]** [**routing-type** *routing-number*] **[rst]** [**sequence** *value*] **[syn]** [**time-range** *name*] **[urg]**

User Datagram Protocol

deny udp {*source-ipv6-prefix*/*prefix-length*| **any**| **host** *source-ipv6-address*| **auth**} [*operator* [*port-number*]] {*destination-ipv6-prefix*/*prefix-length*| **any**| **host** *destination-ipv6-address*| **auth**} [*operator* [*port-number*]] [**dest-option-type** [*doh-number*| *doh-type*]] [**dscp** *value*] [**flow-label** *value*] **[fragments] [hbh] [log] [log-input] [mobility]** [**mobility-type** [*mh-number*| *mh-type*]] [**neq** {*port*| *protocol*}] [**range** {*port*| *protocol*}] **[routing]** [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]

Syntax Description

protocol	Name or number of an Internet protocol. It can be one of the keywords **ahp**, **esp**, **icmp**, **ipv6**, **pcp**, **sctp**, **tcp**, **udp**, or **hbh**, 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*	The 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.
operator [*port-number*]	(Optional) Specifies an 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-length* argument, it must match the destination port. The **range** operator requires two port numbers. All other operators require one port number. The optional *port-number* argument is a decimal number or the name of a TCP or UDP port. A port number is a number from 0 to 65535. 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*	The 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*	The 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.

auth	Allows matching traffic against the presence of the authentication header in combination with any protocol.
dest-option-type	(Optional) Matches IPv6 packets against the hop-by-hop option extension header within each IPv6 packet header.
doh-number	(Optional) Integer in the range from 0 to 255 representing an IPv6 destination option extension header.
doh-type	(Optional) Destination option header types. The possible destination option header type and its corresponding *doh-number* value are home-address—201.
dscp *value*	(Optional) Matches a differentiated services code point value against the traffic class value in the Traffic Class field of each IPv6 packet header. The acceptable range is from 0 to 63.
flow-label *value*	(Optional) Matches a flow label value against the flow label value in the Flow Label field of each IPv6 packet header. The acceptable range is from 0 to 1048575.
fragments	(Optional) Matches non-initial fragmented packets where the fragment extension header contains a non-zero fragment offset. The **fragments** keyword is an option only if the *operator* [*port-number*] arguments are not specified.
hbh	(Optional) Specifies a hop-by-hop options header.
log	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the **logging console** command.) The message includes the access list name and sequence number, whether the packet was denied; the protocol, whether it was TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and source and destination port numbers. The message is generated for the first packet that matches, and then at 5-minute intervals, including the number of packets denied in the prior 5-minute interval.

log-input	(Optional) Provides the same function as the **log** keyword, except that the logging message also includes the input interface.
mobility	(Optional) Extension header type. Allows matching of any IPv6 packet including a mobility header, regardless of the value of the mobility-header-type field within that header.
mobility-type	(Optional) Mobility header type. Either the *mh-number* or *mh-type* argument must be used with this keyword.
mh-number	(Optional) Integer in the range from 0 to 255 representing an IPv6 mobility header type.
mh-type	(Optional) Name of a mobility header type. Possible mobility header types and their corresponding *mh-number* value are as follows: • 0—bind-refresh • 1—hoti • 2—coti • 3—hot • 4—cot • 5—bind-update • 6—bind-acknowledgment • 7—bind-error
routing	(Optional) Matches source-routed packets against the routing extension header within each IPv6 packet header.
routing-type	(Optional) Allows routing headers with a value in the type field to be matched independently. The *routing-number* argument must be used with this keyword.
routing-number	Integer in the range from 0 to 255 representing an IPv6 routing header type. Possible routing header types and their corresponding *routing-number* value are as follows: • 0—Standard IPv6 routing header • 2—Mobile IPv6 routing header

sequence *value*	(Optional) Specifies the sequence number for the access list statement. The acceptable range is from 1 to 4294967295.
time-range *name*	(Optional) Specifies the time range that applies to the deny statement. The name of the time range and its restrictions are specified by the **time-range** and **absolute** or **periodic** commands, respectively.
undetermined-transport	(Optional) Matches packets from a source for which the Layer 4 protocol cannot be determined. The **undetermined-transport** keyword is an option only if the *operator* [*port-number*] arguments are not specified.
icmp-type	(Optional) Specifies an ICMP message type for filtering ICMP packets. ICMP packets can be filtered by ICMP message type. The ICMP message type can be a number from 0 to 255, some of which include the following predefined strings and their corresponding numeric values: • 144—dhaad-request • 145—dhaad-reply • 146—mpd-solicitation • 147—mpd-advertisement
icmp-code	(Optional) Specifies an 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. The code is a number from 0 to 255.
icmp-message	(Optional) Specifies an ICMP message name for filtering ICMP packets. ICMP packets can be filtered by an ICMP message name or ICMP message type and code. The possible names are listed in the "Usage Guidelines" section.
ack	(Optional) For the TCP protocol only: acknowledgment (ACK) bit set.
established	(Optional) For the TCP protocol only: Indicates an established connection. A match occurs if the TCP datagram has the ACK or RST bits set. The nonmatching case is that of the initial TCP datagram to form a connection.

fin	(Optional) For the TCP protocol only: Fin bit set; no more data from sender.
neq {*port* | *protocol*}	(Optional) Matches only packets that are not on a given port number.
psh	(Optional) For the TCP protocol only: Push function bit set.
range {*port* | *protocol*}	(Optional) Matches only packets in the range of port numbers.
rst	(Optional) For the TCP protocol only: Reset bit set.
syn	(Optional) For the TCP protocol only: Synchronize bit set.
urg	(Optional) For the TCP protocol only: Urgent pointer bit set.

Command Default

No IPv6 access list is defined.

Command Modes

IPv6 access list configuration (config-ipv6-acl)#

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.4(2)T	The *icmp-type* argument was enhanced. The **dest-option-type**, **mobility**, **mobility-type**, and **routing-type** keywords were added. The *doh-number*, *doh-type*, *mh-number*, *mh-type*, and *routing-number* arguments were added.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(25)SG	This command was integrated into Cisco IOS Release 12.2(25)SG.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 Aggregation Series Routers.

Release	Modification
12.4(20)T	The **auth** keyword was added.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.
15.2(3)T	This command was modified. Support was added for the **hbh** keyword.
15.1(1)SY	This command was integrated into Cisco IOS Release 15.1(1)SY.
Cisco IOS XE Release 3.2SE	This command was integrated into Cisco IOS XE Release 3.2SE.

Usage Guidelines

The **deny** (IPv6) command is similar to the **deny** (IP) 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 or to define the access list as a reflexive access list.

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

By 1default, 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**, **remark**, or **evaluate** 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.

In Cisco IOS Release 12.2(2)T or later releases, 12.0(21)ST, and 12.0(22)S, IPv6 access control lists (ACLs) are defined and their deny and permit conditions are set by using the **ipv6 access-list** command with the **deny** and **permit** keywords in global configuration mode. In Cisco IOS Release 12.0(23)S or later releases, IPv6 ACLs are defined by using the **ipv6 access-list** command in global configuration mode and their permit and deny conditions are set by using the **deny** and **permit** commands in IPv6 access list configuration mode. Refer to the **ipv6 access-list** command for more information on defining IPv6 ACLs.

Note In Cisco IOS Release 12.0(23)S or later releases, 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. 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.

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-number*] arguments are not specified.

The **undetermined-transport** keyword is an option only if the *operator* [*port-number*] arguments are not specified.

The following is a list of ICMP message names:

- beyond-scope
- destination-unreachable
- echo-reply
- echo-request
- header
- hop-limit
- mld-query
- mld-reduction
- mld-report
- nd-na
- nd-ns
- next-header
- no-admin
- no-route
- packet-too-big
- parameter-option
- parameter-problem
- port-unreachable
- reassembly-timeout
- renum-command
- renum-result
- renum-seq-number
- router-advertisement
- router-renumbering
- router-solicitation
- time-exceeded
- unreachable

Examples

The following example configures the IPv6 access list named toCISCO and applies the access list to outbound traffic on Ethernet interface 0. Specifically, the first deny entry in the list keeps all packets that have a destination

TCP port number greater than 5000 from exiting out of Ethernet interface 0. The second deny entry in the list keeps all packets that have a source UDP port number less than 5000 from exiting out of Ethernet interface 0. The second deny also logs all matches to the console. The first permit entry in the list permits all ICMP packets to exit out of Ethernet interface 0. The second permit entry in the list permits all other traffic to exit out of Ethernet interface 0. The second permit entry is necessary because an implicit deny all condition is at the end of each IPv6 access list.

```
ipv6 access-list toCISCO
 deny tcp any any gt 5000
 deny ::/0 lt 5000 ::/0 log
 permit icmp any any
 permit any any
interface ethernet 0
 ipv6 traffic-filter toCISCO out
```

The following example shows how to allow TCP or UDP parsing although an IPsec AH is present:

```
IPv6 access list example1
    deny tcp host 2001::1 any log sequence 5
    permit tcp any any auth sequence 10
    permit udp any any auth sequence 20
```

Related Commands

Command	Description
ipv6 access-list	Defines an IPv6 access list and enters IPv6 access list configuration mode.
ipv6 traffic-filter	Filters incoming or outgoing IPv6 traffic on an interface.
permit (IPv6)	Sets permit conditions for an IPv6 access list.
show ipv6 access-list	Displays the contents of all current IPv6 access lists.

dialer aaa

To allow a dialer to access the authentication, authorization, and accounting (AAA) server for dialing information, use the dialer aaa command in interface configuration mode. To disable this function, use the no form of this command.

dialer aaa [**password** *string*| **suffix** *string*]

no dialer aaa [**password** *string*| **suffix** *string*]

Syntax Description

password *string*	(Optional) Defines a nondefault password for authentication. The password string can be a maximum of 128 characters.
suffix *string*	(Optional) Defines a suffix for authentication. The suffix string can be a maximum of 64 characters.

Command Default

This feature is not enabled by default.

Command Modes

Interface configuration

Command History

Release	Modification
12.0(3)T	This command was introduced.
12.1(5)T	The password and **suffix** keywords were added.
12.2(33)SRA	This command was integrated into Cisco IOS release 12.(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

This command is required for large scale dial-out and Layer 2 Tunneling Protocol (L2TP) dial-out functionality. With this command, you can specify a suffix, a password, or both. If you do not specify a password, the default password will be “cisco.”

Note Only IP addresses can be specified as usernames for the **dialer aaa suffix** command.

Examples

This example shows a user sending out packets from interface Dialer1 with a destination IP address of 10.1.1.1. The username in the access-request message is “10.1.1.1@ciscoDoD” and the password is “cisco.”

```
interface dialer1
 dialer aaa
 dialer aaa suffix @ciscoDoD password cisco
```

Related Commands

Command	Description
accept dialout	Accepts requests to tunnel L2TP dial-out calls and creates an accept-dialout VPDN subgroup.
dialer congestion-threshold	Specifies congestion threshold in connected links.
dialer vpdn	Enables a Dialer Profile or DDR dialer to use L2TP dial-out.