IP Access List Entry Sequence Numbering

Users can apply sequence numbers to **permit** or **deny** statements and also reorder, add, or remove such statements from a named IP access list. This feature makes revising IP access lists much easier. Prior to this feature, users could add access list entries to the end of an access list only; therefore needing to add statements anywhere except the end required reconfiguring the access list entirely.

Feature History for the IP Access List Entry Additions Feature

Release	Modification
12.2(14)S	This feature was introduced.
12.2(15)T	This feature was integrated into Cisco IOS Release 12.2(15)T.
12.3(2T	This feature was integrated into Cisco IOS Release 12.3(2)T.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for IP Access List Entry Sequence Numbering

- This feature does not support dynamic, reflexive, or firewall access lists.

- This feature does not support old-style numbered access lists, which existed before named access lists. Keep in mind that you can name an access list with a number, so numbers are allowed when they are entered in the standard or extended named access list (NACL) configuration mode.

Information About IP Access Lists

Before you resequence or add entries to an IP access list, you should understand the following concepts:

Purpose of IP Access Lists

Access lists perform packet filtering to control which packets move through the network and where. Such control can help limit network traffic and restrict the access of users and devices to the network. Access lists have many uses, and therefore many commands accept a reference to an access list in their command syntax. Access lists can be used to do the following:

- Filter incoming packets on an interface.

- Filter outgoing packets on an interface.

- Restrict the contents of routing updates.

- Limit debug output based on an address or protocol.

- Control virtual terminal line access.

- Identify or classify traffic for advanced features, such as congestion avoidance, congestion management, and priority and custom queuing.

- Trigger dial-on-demand routing (DDR) calls.

How an IP Access List Works

An access list is a sequential list consisting of at least one **permit** statement and possibly one or more **deny** statements that apply to IP addresses and possibly upper-layer IP protocols. The access list has a name by which it is referenced. Many software commands accept an access list as part of their syntax.

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

IP Access List Process and Rules

- The software tests the source or destination address or the protocol of each packet being filtered against the conditions in the access list, one condition (**permit** or **deny** statement) at a time.

- If a packet does not match an access list statement, the packet is then tested against the next statement in the list.

- If a packet and an access list statement match, the rest of the statements in the list are skipped and the packet is permitted or denied as specified in the matched statement. The first entry that the packet matches determines whether the software permits or denies the packet. That is, after the first match, no subsequent entries are considered.

- If the access list denies the address or protocol, the software discards the packet and returns an ICMP Host Unreachable message.

- If no conditions match, the software drops the packet. This is because each access list ends with an unwritten or implicit **deny** statement. That is, if the packet has not been permitted by the time it was tested against each statement, it is denied.

- The access list must contain at least one **permit** statement or else all packets are denied.

- Because the software stops testing conditions after the first match, the order of the conditions is critical. The same **permit** or **deny** statements specified in a different order could result in a packet being passed under one circumstance and denied in another circumstance.

- If an access list is referenced by name in a command, but the access list does not exist, all packets pass.

- Only one access list per interface, per protocol, per direction is allowed.

- Inbound access lists process packets arriving at the router. Incoming packets are processed before being routed to an outbound interface. An inbound access list is efficient because it saves the overhead of routing lookups if the packet is to be discarded because it is denied by the filtering tests. If the packet is permitted by the tests, it is then processed for routing. For inbound lists, **permit** means continue to process the packet after receiving it on an inbound interface; **deny** means discard the packet.

- Outbound access lists process packets before they leave the router. Incoming packets are routed to the outbound interface and then processed through the outbound access list. For outbound lists, **permit** means send it to the output buffer; **deny** means discard the packet.

Helpful Hints for Creating IP Access Lists

- Create the access list before applying it to an interface. An interface with an empty access list applied to it permits all traffic.

- Another reason to configure an access list before applying it is because if you applied a nonexistent access list to an interface and then proceed to configure the access list, the first statement is put into effect, and the implicit **deny** statement that follows could cause you immediate access problems.

- Because the software stops testing conditions after it encounters the first match (to either a **permit** or **deny** statement), you will reduce processing time and resources if you put the statements that packets are most likely to match at the beginning of the access list. Place more frequently occurring conditions before less frequent conditions.

- Organize your access list so that more specific references in a network or subnet appear before more general ones.

- In order to make the purpose of individual statements more easily understood at a glance, you can write a helpful remark before or after any statement.

Source and Destination Addresses

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

Wildcard Mask and Implicit Wildcard Mask

Address filtering uses wildcard masking to indicate to the software whether to check or ignore corresponding IP address bits when comparing the address bits in an access list entry to a packet being submitted to the access list. By carefully setting wildcard masks, an administrator can select single or several IP addresses for permit or deny tests.

Wildcard masking for IP address bits uses the number 1 and the number 0 to specify how the software treats the corresponding IP address bits. A wildcard mask is sometimes referred to as an inverted mask because a 1 and 0 mean the opposite of what they mean in a subnet (network) mask.

- A wildcard mask bit 0 means *check* the corresponding bit value.
- A wildcard mask bit 1 means *ignore* that corresponding bit value.

If you do not supply a wildcard mask with a source or destination address in an access list statement, the software assumes a default wildcard mask of 0.0.0.0.

Unlike subnet masks, which require contiguous bits indicating network and subnet to be ones, wildcard masks allow noncontiguous bits in the mask.

Transport Layer Information

You can filter packets based on transport layer information, such as whether the packet is a TCP, UDP, ICMP or IGMP packet.

IP Access List Entry Sequence Numbering

Benefits

The ability to apply sequence numbers to IP access list entries simplifies access list changes. Prior to the IP Access List Entry Sequence Numbering feature, there was no way to specify the position of an entry within an access list. If a user wanted to insert an entry (statement) in the middle of an existing list, all of the entries after the desired position had to be removed, then the new entry was added, and then all the removed entries had to be reentered. This method was cumbersome and error prone.

This feature allows users to add sequence numbers to access list entries and resequence them. When a user adds a new entry, the user chooses the sequence number so that it is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced to create room to insert the new entry.

Sequence Numbering Behavior

- For backward compatibility with previous releases, if entries with no sequence numbers are applied, the first entry is assigned a sequence number of 10, and successive entries are incremented by 10. The maximum sequence number is 2147483647. If the generated sequence number exceeds this maximum number, the following message is displayed:

```
Exceeded maximum sequence number.
```

- If the user enters an entry without a sequence number, it is assigned a sequence number that is 10 greater than the last sequence number in that access list and is placed at the end of the list.

- If the user enters an entry that matches an already existing entry (except for the sequence number), then no changes are made.

- If the user enters a sequence number that is already present, the following error message is generated:

```
Duplicate sequence number.
```

- If a new access list is entered from global configuration mode, then sequence numbers for that access list are generated automatically.

- Distributed support is provided so that the sequence numbers of entries in the Route Processor (RP) and line card (LC) are in synchronization at all times.

- Sequence numbers are not nvgened. That is, the sequence numbers themselves are not saved. In the event that the system is reloaded, the configured sequence numbers revert to the default sequence starting number and increment. The function is provided for backward compatibility with software releases that do not support sequence numbering.

- This feature works with named standard and extended IP access lists. Because the name of an access list can be designated as a number, numbers are acceptable.

How to Use Sequence Numbers in an IP Access List

This section describes how to use sequence numbers in an IP access list.

-

Sequencing Access-List Entries and Revising the Access List

This task shows how to assign sequence numbers to entries in a named IP access list and how to add or delete an entry to or from an access list. It is assumed a user wants to revise an access list. The context of this task is the following:

- A user need not resequence access lists for no reason; resequencing in general is optional. The resequencing step in this task is shown as required because that is one purpose of this feature and this task demonstrates the feature.

- Step 5 happens to be a **permit** statement and Step 6 happens to be a **deny** statement, but they need not be in that order.

SUMMARY STEPS

1. **enable**

2. **configure terminal**

3. **ip access-list resequence** *access-list-name starting-sequence-number increment*

4. **ip access-list** {**standard** | **extended**} *access-list-name*

5. *sequence-number* **permit** *source source-wildcard*

or

sequence-number **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

6. *sequence-number* **deny** *source source-wildcard*

    or

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

7. Repeat Step 5 and/or Step 6 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.

8. **end**

9. **show ip access-lists** *access-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list resequence** *access-list-name* *starting-sequence-number increment* **Example:** `Router(config)# ip access-list resequence kmd1 100 15`	Resequences the specified IP access list using the starting sequence number and the increment of sequence numbers. • This example resequences an access list named kmd1. The starting sequence number is 100 and the increment is 15.
Step 4	**ip access-list** {**standard** | **extended**} *access-list-name* **Example:** `Router(config)# ip access-list standard kmd1`	Specifies the IP access list by name and enters named access list configuration mode. • If you specify **standard**, make sure you subsequently specify **permit** and/or **deny** statements using the standard access list syntax. • If you specify **extended**, make sure you subsequently specify **permit** and/or **deny** statements using the extended access list syntax.

	Command or Action	Purpose
Step 5	*sequence-number* **permit** *source source-wildcard* or *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-std-nacl)# 105 permit 10.5.5.5 0.0.0 255`	Specifies a permit statement in named IP access list mode. • This access list happens to use a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • See the permit (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP). • Use the **no** *sequence-number* command to delete an entry. • As the prompt indicates, this access list was a standard access list. If you had specified **extended** in Step 4, the prompt for this step would be `Router(config-ext-nacl)` and you would use the extended **permit** command syntax.
Step 6	*sequence-number* **deny** *source source-wildcard* or *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-std-nacl)# 105 deny 10.6.6.7 0.0.0 255`	(Optional) Specifies a deny statement in named IP access list mode. • This access list happens to use a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • See the deny (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP). • Use the **no** *sequence-number* command to delete an entry. • As the prompt indicates, this access list was a standard access list. If you had specified **extended** in Step 4, the prompt for this step would be `Router(config-ext-nacl)` and you would use the extended **deny** command syntax.
Step 7	Repeat Step 5 and/or Step 6 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.	Allows you to revise the access list.

	Command or Action	Purpose
Step 8	**end** **Example:** `Router(config-std-nacl)# end`	(Optional) Exits the configuration mode and returns to privileged EXEC mode.

Step 9	**show ip access-lists** *access-list-name*
	
	Example:
	`Router# show ip access-lists kmd1`
		(Optional) Displays the contents of the IP access list.
		
		• Review the output to see that the access list includes the new entry.
		
		`Router# `**`show ip access-lists kmd1`**
		
		`Standard IP access list kmd1`
		`100 permit 10.4.4.0, wildcard bits 0.0.0.255`
		`105 permit 10.5.5.0, wildcard bits 0.0.0.255`
		`115 permit 10.0.0.0, wildcard bits 0.0.0.255`
		`130 permit 10.5.5.0, wildcard bits 0.0.0.255`
		`145 permit 10.0.0.0, wildcard bits 0.0.0.255`

What to Do Next

If your access list is not already applied to an interface or line or otherwise referenced, apply the access list. Refer to the "Configuring IP Services" chapter of the *Cisco IOS IP Configuration Guide* for information about how to apply an IP access list.

Configuration Examples for IP Access List Entry Sequence Numbering

This section provides the following examples related to sequence numbering of entries in an IP access list:

Resequencing Entries in an Access List: Example

The following example shows access list resequencing. The starting value is 1, and increment value is 2. The subsequent entries are ordered based on the increment values that users provide, and the range is from 1 to 2147483647.

When an entry with no sequence number is entered, by default it has a sequence number of 10 more than the last entry in the access list.

```
Router# show access-list 150

Extended IP access list 150
    10 permit ip host 10.3.3.3 host 172.16.5.34
    20 permit icmp any any
    30 permit tcp any host 10.3.3.3
    40 permit ip host 10.4.4.4 any
    50 Dynamic test permit ip any any
```

```
    60 permit ip host 172.16.2.2 host 10.3.3.12
    70 permit ip host 10.3.3.3 any log
    80 permit tcp host 10.3.3.3 host 10.1.2.2
    90 permit ip host 10.3.3.3 any
    100 permit ip any any

Router(config)# ip access-list extended 150
Router(config)# ip access-list resequence 150 1 2
Router(config)# end

Router# show access-list 150

Extended IP access list 150
    1 permit ip host 10.3.3.3 host 172.16.5.34
    3 permit icmp any any
    5 permit tcp any host 10.3.3.3
    7 permit ip host 10.4.4.4 any
    9 Dynamic test permit ip any any
    11 permit ip host 172.16.2.2 host 10.3.3.12
    13 permit ip host 10.3.3.3 any log
    15 permit tcp host 10.3.3.3 host 10.1.2.2
    17 permit ip host 10.3.3.3 any
    19 permit ip any any
```

Adding Entries with Sequence Numbers: Example

In the following example, an new entry is added to a specified access list:

```
Router# show ip access-list

Standard IP access list tryon
2 permit 10.4.4.2, wildcard bits 0.0.255.255
5 permit 10.0.0.44, wildcard bits 0.0.0.255
10 permit 10.0.0.1, wildcard bits 0.0.0.255
20 permit 10.0.0.2, wildcard bits 0.0.0.255

Router(config)# ip access-list standard tryon

Router(config-std-nacl)# 15 permit 10.5.5.5 0.0.0.255

Router# show ip access-list

Standard IP access list tryon
2 permit 10.4.0.0, wildcard bits 0.0.255.255
5 permit 10.0.0.0, wildcard bits 0.0.0.255
10 permit 10.0.0.0, wildcard bits 0.0.0.255
15 permit 10.5.5.0, wildcard bits 0.0.0.255
20 permit 10.0.0.0, wildcard bits 0.0.0.255
```

Entry without Sequence Number: Example

The following example shows how an entry with no specified sequence number is added to the end of an access list. When an entry is added without a sequence number, it is automatically given a sequence number that puts it at the end of the access list. Because the default increment is 10, the entry will have a sequence number 10 higher than the last entry in the existing access list.

```
Router(config)# ip access-list standard 1

Router(config-std-nacl)# permit 1.1.1.1 0.0.0.255
```

```
Router(config-std-nacl)# permit 2.2.2.2 0.0.0.255
Router(config-std-nacl)# permit 3.3.3.3 0.0.0.255

Router# show access-list
Standard IP access list 1
10 permit 0.0.0.0, wildcard bits 0.0.0.255
20 permit 0.0.0.0, wildcard bits 0.0.0.255
30 permit 0.0.0.0, wildcard bits 0.0.0.255

Router(config)# ip access-list standard 1
Router(config-std-nacl)# permit 4.4.4.4 0.0.0.255
Router(config-std-nacl)# end

Router# show access-list

Standard IP access list 1
10 permit 0.0.0.0, wildcard bits 0.0.0.255
20 permit 0.0.0.0, wildcard bits 0.0.0.255
30 permit 0.0.0.0, wildcard bits 0.0.0.255
40 permit 0.4.0.0, wildcard bits 0.0.0.255
```

Additional References

The following sections provide references related to IP access lists.

Related Documents

Related Topic	Document Title
Configuring IP access lists	"Configuring IP Services" chapter in the *Cisco IOS IP Configuration Guide*, Release 12.2
IP access list commands	"IP Services Commands" chapter in the *Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services*, Release 12.2

Standards

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

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

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

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, tools, and lots more. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This section documents new and revised commands. All other commands used with this feature are documented in the Cisco IOS Release 12.2T command reference publications.

New Command

- **ip access-list resequence**

Revised Commands

- **deny (IP)**
- **permit (IP)**

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* [**precedence** *precedence*] [**tos** *tos*] [**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)

For ICMP, you can also use the following syntax:

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

Internet Group Management Protocol (IGMP)

For IGMP, you can also use the following syntax:

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

Transmission Control Protocol (TCP)

For TCP, you can also use the following syntax:

[*sequence-number*] **deny tcp** *source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established**] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

User Datagram Protocol (UDP)

For UDP, you can also use the following syntax:

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

Syntax Description

sequence-number	(Optional) Sequence number assigned to the deny statement, causing 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 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. It can be one of the keywords **eigrp**, **gre**, **icmp**, **igmp**, **igrp**, **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. Some protocols allow further qualifiers described later.
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 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.
precedence *precedence*	(Optional) Packets can be filtered by precedence level, as specified by a number from 0 to 7 or by name as listed in the section "Usage Guidelines."
tos *tos*	(Optional) Packets can be filtered by type of service (ToS) level, as specified by a number from 0 to 15, or by name as listed in the section "Usage Guidelines" of the **access-list** (IP extended) command.

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, 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 due to 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 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.
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.
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 ICMP message type and code name. The possible names are listed in the section "Usage Guidelines" 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 section "Usage Guidelines" of the **access-list** (IP extended) command.
operator	(Optional) Compares source or destination ports. Possible operands 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*, it must match the source port.
	If the operator is positioned after the *destination* and *destination-wildcard*, it must match the destination port.
	The **range** operator requires two port numbers. 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 section "Usage Guidelines" of the **access-list** (IP extended) command.
	TCP port names can only be used when filtering TCP. UDP port names can only be used 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 bits set. The nonmatching case is that of the initial TCP datagram to form a connection.
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 "Access List Processing of Fragments" and "Fragments and Policy Routing" sections in the "Usage Guidelines" section.

Defaults   There is no specific condition 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) and 12.1(2)	The **fragments** keyword was added.
12.2(14)S	The *sequence-number* argument was added.

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

Access List Processing of Fragments

The behavior of access-list entries regarding the use or lack 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 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 addres**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.

Examples The following example sets a deny condition for a standard access list named Internetfilter:

```
ip access-list standard Internetfilter
 deny 192.5.34.0  0.0.0.255
 permit 128.88.0.0  0.0.255.255
 permit 36.0.0.0  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:

```
Router(config)# ip access-list extended 150
Router(config-std-nacl)# 25 deny ip host 3.3.3.3 host 45.5.5.34
```

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

```
Router(config-std-nacl)# no 25
```

Related Commands

Command	Description
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.
ip access-list log-update	Sets the threshold number of packets that cause a logging message.

Command	Description
ip access-list resequence	Applies sequence numbers to the access list entries in an access list.
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 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.

ip access-list resequence

To apply sequence numbers to the access list entries in an access list, use the **ip access-list resequence** command in global configuration mode.This command does not have a **no** version.

ip access-list resequence *access-list-name starting-sequence-number increment*

Syntax Description

access-list-name	Name of the access list. Names cannot contain a space or quotation mark.
starting-sequence-number	Access list entries will be resequenced using this initial value. The default value is 10. The range of possible sequence numbers is 1 through 2147483647.
increment	The number by which the sequence numbers change. The default value is 10. For example, if the increment value is 5 and the beginning sequence number is 20, the subsequent sequence numbers are 25, 30, 25, 40, and so on.

Defaults   Disabled

Command Modes   Global configuration

Command History

Release	Modification
12.2(14)S	This command was introduced.

Usage Guidelines   This feature allows the **permit** and **deny** entries of a specified access list to be resequenced with an initial sequence number value determined by the *starting-sequence-number* argument, and continuing in increments determined by the *increment* argument. If the highest sequence number exceeds the maximum possible sequence number, then no sequencing occurs.

For backward compatibility with previous releases, if entries with no sequence numbers are applied, the first entry is assigned a sequence number of 10, and successive entries are incremented by 10. The maximum sequence number is 2147483647. If the generated sequence number exceeds this maximum number, the following message is displayed:

```
Exceeded maximum sequence number.
```

If the user enters an entry without a sequence number, it is assigned a sequence number that is 10 greater than the last sequence number in that access list and is placed at the end of the list.

If the user enters an entry that matches an already existing entry (except for the sequence number), then no changes are made.

If the user enters a sequence number that is already present, the following error message is generated:

```
Duplicate sequence number.
```

If a new access list is entered from global configuration mode, then sequence numbers for that access list are generated automatically.

Distributed support is provided so that the sequence numbers of entries in the Route Processor (RP) and line card (LC) are in synchronization at all times.

Sequence numbers are not nvgened. That is, the sequence numbers themselves are not saved. In the event that the system is reloaded, the configured sequence numbers revert to the default sequence starting number and increment.

This feature works with named standard and extended IP access lists. Because the name of an access list can be designated as a number, numbers are acceptable as names as long as they are entered in named access list configuration mode.

Examples

The following example resequences an access list named kmd1. The starting sequence number is 100, and the increment value is 5:

```
Router(config)# ip access-list resequence kmd1 100 5
```

Related Commands

Command	Description
deny (IP)	Sets conditions under which a packet does not pass a named IP access list.
permit (IP)	Sets conditions under which a packet passes a named IP access list.

permit (IP)

To set conditions to allow a packet to pass a named IP access list, use the **permit** access-list configuration command. To remove a permit condition from an access list, use a **no** form of this command.

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

[*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard*
    [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

no *sequence-number*

no permit *source* [*source-wildcard*]

no permit *protocol source source-wildcard destination destination-wildcard*
    [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

Internet Control Message Protocol (ICMP)

For ICMP, you can also use the following syntax:

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

Internet Group Management Protocol (IGMP)

For IGMP, you can also use the following syntax:

[*sequence-number*] **permit igmp** *source source-wildcard destination destination-wildcard*
    [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*]
    [**fragments**]

Transmission Control Protocol (TCP)

For TCP, you can also use the following syntax:

[*sequence-number*] **permit tcp** *source source-wildcard* [*operator* [*port*]] *destination*
    destination-wildcard [*operator* [*port*]] [**established**] [**precedence** *precedence*] [**tos** *tos*] [**log**]
    [**time-range** *time-range-name*] [**fragments**]

User Datagram Protocol UDP)

For UDP, you can also use the following syntax:

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

■ **permit (IP)**

Syntax Description		
	sequence-number	(Optional) Sequence number assigned to the permit statement, causing 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 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 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. It can be one of the keywords **eigrp**, **gre**, **icmp**, **igmp**, **igrp**, **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. Some protocols allow further qualifiers described later.
	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 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.
	precedence *precedence*	(Optional) Packets can be filtered by precedence level, as specified by a number from 0 to 7 or by name as listed in the section "Usage Guidelines."
	tos *tos*	(Optional) Packets can be filtered by type of service (ToS) level, as specified by a number from 0 to 15, or by name as listed in the section "Usage Guidelines" of the **access-list** (IP extended) command.

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, 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 due to 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 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.
time-range *time-range-name*	(Optional) Name of the time range that applies to this **permit** statement. The name of the time range and its restrictions are specified by the **time-range** and **absolute** or **periodic** commands, respectively.
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 ICMP message type and code name. The possible names are found in the section "Usage Guidelines" 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 section "Usage Guidelines" of the **access-list** (IP extended) command.
operator	(Optional) Compares source or destination ports. Possible operands 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*, it must match the source port.
	If the operator is positioned after the *destination* and *destination-wildcard*, it must match the destination port.
	The **range** operator requires two port numbers. 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 section "Usage Guidelines" of the **access-list** (IP extended) command.
	TCP port names can only be used when filtering TCP. UDP port names can only be used 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 bits set. The nonmatching case is that of the initial TCP datagram to form a connection.
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 "Access List Processing of Fragments" and "Fragments and Policy Routing" sections in the "Usage Guidelines" section.

Defaults   There are no specific conditions under which a packet passes 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) and 12.1(2)	The **fragments** keyword was added.
12.2(14)S	The *sequence-number* argument was added.

Usage Guidelines   Use this command following the **ip access-list** command to define the conditions under which a packet passes the access list.

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

Access List Processing of Fragments

The behavior of access-list entries regarding the use or lack 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 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,	The access-list entry is applied only to noninitial fragments. **Note** 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 addres**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.

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

```
ip access-list standard Internetfilter
 deny 192.5.34.0  0.0.0.255
 permit 128.88.0.0  0.0.255.255
 permit 36.0.0.0  0.255.255.255
! (Note: all other access implicitly denied)
```

The following example permits Telnet traffic on Mondays, Tuesdays, and Fridays from 9:00 a.m. to 5:00 p.m.:

```
time-range testing
 periodic Monday Tuesday Friday 9:00 to 17:00
!
ip access-list extended legal
 permit tcp any any eq telnet time-range testing
!
interface ethernet 0
 ip access-group legal in
```

The following example shows how to add an entry to an existing access list:

```
Router# show access-list

Standard IP access list 1
2 permit 10.4.0.0, wildcard bits 0.0.255.255
5 permit 10.0.0.0, wildcard bits 0.0.255.255
10 permit 10.0.0.0, wildcard bits 0.0.255.255
20 permit 10.0.0.0, wildcard bits 0.0.255.255

Router(config)# ip access-list standard 1
Router(config-std-nacl)# 15 permit 5.5.5.5 0.0.255.255
```

The following examples shows how the entry with the sequence number of 20 is removed from the access list:

```
Router(config)# ip access-list standard 1
Router(config-std-nacl)# no 20
```

```
Router# show access-list

Standard IP access list 1
10 permit 0.0.0.1, wildcard bits 0.0.0.255
30 permit 0.0.0.3, wildcard bits 0.0.0.255
40 permit 0.4.0.4, wildcard bits 0.0.0.255
```

The following examples shows how, if a user tries to enter an entry that is a duplicate of an entry already on the list, no changes occur. The entry that the user is trying to add is a duplicate of the entry already in the access list with a sequence number of 20.

```
Router# show access-list 101

Extended IP access list 101
    10 permit ip host 3.3.3.3 host 45.5.5.34
    20 permit icmp any any
    30 permit ip host 65.34.2.2 host 43.2.54.2
    40 permit ip host 45.3.4.31 host 34.3.32.3 log

Router(config)# ip access-list extended 101
Router(config-ext-nacl)# 100 permit icmp any any
Router(config-ext-nacl)# end

Router# show access-list 101

Extended IP access list 101
    10 permit ip host 3.3.3.3 host 45.5.5.34
    20 permit icmp any any
    30 permit ip host 65.34.2.2 host 43.2.54.2
    40 permit ip host 45.3.4.31 host 34.3.32.3 log
```

The following example shows what occurs if a user tries to enter a new entry with a sequence number of 20 when an entry with a sequence number of 20 is already in the list. An error message appears, and no change is made to the access list.

```
Router# show access-list 101

Extended IP access list 101
    10 permit ip host 3.3.3.3 host 45.5.5.34
    20 permit icmp any any
    30 permit ip host 65.34.2.2 host 43.2.54.2
    40 permit ip host 45.3.4.31 host 34.3.32.3 log

Router(config)# ip access-list extended 101
Router(config-ext-nacl)# 20 permit udp host 1.1.1.1 host 2.2.2.2

Duplicate sequence number.

Router(config-ext-nacl)# end

Router# show access-list 101

Extended IP access list 101
    10 permit ip host 3.3.3.3 host 45.5.5.34
    20 permit icmp any any
    30 permit ip host 65.34.2.2 host 43.2.54.2
    40 permit ip host 45.3.4.31 host 34.3.32.3 log
```

■ **permit (IP)**

	Command	Description
Related Commands	**deny (IP)**	Sets conditions under which a packet does not pass a named IP access list.
	ip access-group	Controls access to an interface.
	ip access-list	Defines an IP access list by name.
	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.
	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.