pac key through port-misuse

permit

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

permit *protocol* [*source-addr source-wildcard*] {**any**| **host** {*address*| *name*}| **object-group** *object-group-name*} {*destination-addr destination-wildcard*| **any**| **host** {*address*| *name*}| **object-group** *object-group-name*} [**dscp** *dscp-value*| **precendence** *precedence-value*| **fragments** *fragment-value*| **option** *option-value*| **reflect** *access-list-name*| **time-range** *time-range-value*| **ttl** *match-value ttl-value* **[ttl-value]**| **tos** *tos-value*| **timeout** *max-time*| **log** [*log-value*]| **log-input** [*log-input-value*]]

no permit *protocol* [*source-addr source-wildcard*] {**any**| **host** {*address*| *name*}| **object-group** *object-group-name*} {*destination-addr destination-wildcard*| **any**| **host** {*address*| *name*}| **object-group** *object-group-name*}

permit {**tcp**| **udp**} {*source-addr source-wildcard*| **any**| **host** *source-addr*| **object-group** *source-obj-group*} {*destination-addr destination-wildcard*| **any**| **host** *dest-addr*| **object-group** *dest-obj-group*| *port-match-criteria* {*destination-addr destination-wildcard*| **any**| **host** *dest-addr*| **object-group** *dest-obj-group*}} [*port-match-criteria port-number*| **fragments**| **ack**| **established**| **fin**| **psh**| **rst**| **syn**| **urg**| **match-all** *match-value*| **match-any** *match-value*| **dscp** *dscp-value*| **precendence** *precedence-value*| **option** *option-value*| **time-range** *time-range-value*| **ttl** *match-value ttl-value* [*ttl-value*]| **tos** *tos-value*| **log** [*log-value*]| **log-input** [*log-input-value*]]

no permit {**tcp**| **udp**} {*source-addr source-wildcard*| **any**| **host** *source-addr*| **object-group** *source-obj-group*} {*destination-addr destination-wild-card*| **any**| **host** *dest-addr*| **object-group** *dest-obj-group*| *port-match-criteria* {*destination-addr destination-wild-card*| **any**| **host** *dest-addr*| **object-group** *dest-obj-group*}}

Syntax Description

protocol	Name or number of a protocol; valid values are; valid values are **ahp**, **eigrp**, **esp**, **gre**, **icmp**, **igmp**, **igrp**, **ip**, **ipinip**, **nos**, **ospf**, **object-group**, **tcp**, **pcp**, **pim**, **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.
source-addr	(Optional) Number of the network or host from which the packet is being sent in a 32-bit quantity in four-part, dotted-decimal format.
source-wildcard	(Optional) Wildcard bits to be applied to the source in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore.

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 *address name*	Specifies the source or destination address and name of a single host.
object-group *object-group-name*	Specifies the source or destination name of the object group.
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.
object-group *dest-addr-group-name*	Specifies the destination address group name.
dscp *dscp-value*	(Optional) Matches the packets with the given Differentiated Services Code Point (DSCP) value; see the “Usage Guidelines” section for valid values.
precedence *precedence-value*	(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.
fragments *fragment-value*	(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 or OGACL Processing of Fragments" and “Fragments and Policy Routing” sections in the “Usage Guidelines” section.
option *option-value*	(Optional) Matches the packets with the given IP options value number; see the “Usage Guidelines” section for valid values.
reflect *access-list-name*	(Optional) Create reflexive access list entry.
time-range *time-range-value*	(Optional) Specifies a time-range entry name.
ttl *match-value ttl-value*	(Optional) Specifies the match packets with given TTL value; see the “Usage Guidelines” section for valid values.

tos *tos-value*	(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.
timeout *max-time*	Specifies the maximum time for a reflexive ACL to live; the valid values are from 1 to 2147483 seconds.
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 port numbers and the user-defined cookie or router-generated hash value. 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. After you specify the **log** keyword (and the associated *word* argument), you cannot specify any other keywords or settings for this command.

log-value	(Optional) User-defined cookie appended to the log message. The cookie: • cannot be more than characters • cannot start with hexadecimal notation (such as 0x) • cannot be the same as, or a subset of, the following keywords: **reflect**, **fragment**, **time-range** • must contain alphanumeric characters only The user-defined cookie is appended to the access control entry (ACE) syslog entry and uniquely identifies the ACE, within the access control list, that generated the syslog entry.
log-input *log-input-value*	(Optional) Matches the log against this entry, including the input interface. After you specify the **log-input** keyword (and the associated *log-input-value* argument), you cannot specify any other keywords or settings for this command.
tcp	Specifies the TCP protocol.
udp	Specifies the UDP protocol.
object-group *source-obj-group*	Specifies the source address group name.
port-match-criteria port-number	Matches only packets on a given port number; see the “Usage Guidelines” section for valid values.

Command Default

There are no specific conditions under which a packet passes 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.
12.4(22)T	The *word* argument was added to the **log** and **log-input** keywords.

Usage Guidelines

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

In Cisco IOS 15.0(1)M and later Releases, to remove the log entry from the **permit ip any any log** command, use the **permit ip any any** command.

In releases earlier than Cisco IOS Release15.0(1)M, to remove the **log** option from the **permit ip any any log** command, use the **no permit ip any any log** and the **permit ip any any** commands.

In Cisco IOS 15.0(1)M and later releases, to remove the log entry and the user-defined cookie, use the **permit ip any any** [*log-value*] command.

In releases earlier than Cisco IOS Release 15.0(1)M, to remove the log entry and user-defined cookies, use the **no permit ip any any log** [*log-value*] and **permit ip any any** commands.

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.

Ensure that you do 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 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 *source-addr* and *destination-addr*arguments allow you to create an object group based on a source or destination group. The following keywords and arguments are available:

- **dscp** *dscp-value* --(Optional) Matches the packets with the given DSCP value; the valid values are as follows:
  - **0** to **63**--Differentiated services codepoint value
  - **af11**--Matches the packets with AF11 dscp (001010)
  - **af12**--Matches the packets with AF12 dscp (001100)
  - **af13**--Matches the packets with AF13 dscp (001110)
  - **af21**--Matches the packets with AF21 dscp (010010)
  - **af22**--Matches the packets with AF22 dscp (010100)
  - **af23**--Matches the packets with AF23 dscp (010110)
  - **af31**--Matches the packets with AF31 dscp (011010)
  - **af32**--Matches the packets with AF32 dscp (011100)
  - **af33**--Matches the packets with AF33 dscp (011110)
  - **af41**--Matches the packets with AF41 dscp (100010)
  - **af42**--Matches the packets with AF42 dscp (100100)
  - **af43**--Matches the packets with AF43 dscp (100110)
  - **cs1**--Matches the packets with CS1 (precedence 1) dscp (001000)
  - **cs2**--Matches the packets with CS2 (precedence 2) dscp (010000)
  - **cs3**--Matches the packets with CS3 (precedence 3) dscp (011000)
  - **cs4**--Matches the packets with CS4 (precedence 4) dscp (100000)
  - **cs5**--Matches the packets with CS5 (precedence 5) dscp (101000)

- **cs6**--Matches the packets with CS6 (precedence 6) dscp (110000)
- **cs7**--Matches the packets with CS7 (precedence 7) dscp (111000)
- **default**--Matches the packets with default dscp (000000)
- **ef**--Matches the packets with EF dscp (101110)

- **fragments** --(Optional) Checks for noninitial fragments. See the table above.
- **log** --(Optional) Logs the matches against this entry.
- **log-input** --(Optional) Logs the matches against this entry, including the input interface.
- **option** *option-value* --(Optional) Matches the packets with given IP Options value. The valid values 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 Extended IP Option (145).
  - **ext-security**--Matches the packets with Extended Security Option (133).
  - **finn**--Matches the packets with 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**--Matches the packets on the SYN bit.
- **timestamp**--Matches the packets with Time Stamp Option (68).
- **traceroute**--Matches the packets with Trace Route Option (82).
- **ump**--Matches the packets with Upstream Multicast Packet Option (152).
- **visa**--Matches the packets with Experimental Access Control Option (142).
- **zsu**--Matches the packets with Experimental Measurement Option (10).

- **precedence** *precedence-value* --(Optional) Matches the packets with given precedence value; the valid values are as follows:
  - 0 to 7--Precedence value.
  - **critical**--Matches the packets with critical precedence (5).
  - **flash**--Matches the packets with flash precedence (3).
  - **flash-override**--Matches the packets with flash override precedence (4).
  - **immediate**--Matches the packets with immediate precedence (2).
  - **internet**--Matches the packets with internetwork control precedence (6).
  - **network**--Matches the packets with network control precedence (7).
  - **priority**--Matches the packets with priority precedence (1).
  - **routine**--Matches the packets with routine precedence (0).
- **reflect acl-name** -- (Optional) Creates reflexive access list entry.
- **ttl** *match-value ttl-value* -- (Optional) Specifies the match packets with given TTL value; the valid values are as follows:
  - **eq**--Matches packets on a given TTL number.
  - **gt**--Matches packets with a greater TTL number.
  - **lt**--Matches packets with a lower TTL number.
  - **neq**--Matches packets not on a given TTL number.
  - **range**--Matches packets in the range of TTLs.
- **time-range** *time-range-value* --(Optional) Specifies a time-range entry name.
- **tos** --(Optional) Matches the packets with given ToS value; the valid values 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 the packets with the minimum monetary cost ToS (1).
- **normal**--Matches the packets with the normal ToS (0).

- **timeout** *max-time* -- (Optional) Specifies the maximum time for a reflexive ACL to live; the valid values are from 1 to 2147483 seconds.

Examples

The following example shows how to create an access list that permits packets from the users in my_network_object_group if the protocol ports match the ports specified in my_network_object_group:

```
Router> enable
Router# configure terminal
Router(config)# ip access-list extended my_ogacl_policy
Router(config-ext-nacl)# permit tcp object-group my_network_object_group portgroup
my_service_object_group any
```

The following example shows how to create an access list that permits packets from the users in my_network_object_group if the protocol ports match the ports specified in my_network_object_group. In addition, logging is enabled for the access list, and all syslog entries for this ACE include the word MyServiceCookieValue:

```
Router> enable
Router# configure terminal
Router(config)# ip access-list extended my_ogacl_policy
Router(config-ext-nacl)# permit tcp object-group my_network_object_group portgroup
my_service_object_group any log MyServiceCookieValue
```

Related Commands

Command	Description
deny	Sets conditions in a named IP access list or OGACL that will deny packets.
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.
ip access-list logging hash-generation	Enables hash value generation for ACE syslog entries.
object-group network	Defines network object groups for use in OGACLs.
object-group service	Defines service object groups for use in OGACLs.
show ip access-list	Displays the contents of IP access lists or OGACLs.
show object-group	Displays information about object groups that are configured.

permit (IP)

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

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

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

no *sequence-number*

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

no permit *protocol source source-wildcard destination destination-wildcard* [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] [**time-range** *time-range-name*] **[fragments]** [**log** [*user-defined-cookie*]]

Internet Control Message Protocol (ICMP)

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

Internet Group Management Protocol (IGMP)

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

Transmission Control Protocol (TCP)

[sequence-number] permit tcp *source source-wildcard* [*operator* [*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**| **log** | [*user-defined-cookie*]]

User Datagram Protocol (UDP)

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

Syntax Description

sequence-number	(Optional) Sequence number assigned to the permit 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	(Optional) 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 **permit** command. **Note** To configure a packet filter to allow BGP traffic, use protocol **tcp** and specify the port number as 179 or **bgp**

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 **permit** 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.
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.
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.
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.) After you specify the **log** keyword (and the associated *word* argument), you cannot specify any other keywords or settings for this command.

user-defined-cookie	(Optional) User-defined cookie appended to the log message. The cookie: • Cannot be more than 64 characters. • Cannot start with hexadecimal notation (such as 0x). • Cannot be the same as, or a subset of, the following keywords: **fragment, reflect**, **time-range.** • Must contain alphanumeric characters only. The user-defined cookie is appended to the Allegro Crypto Engine (ACE) syslog entry and uniquely identifies the ACE, within the access control list, that generated the syslog entry.
icmp	Permits only ICMP packets. When you enter the **icmp** keyword, you must use the specific command syntax shown for the ICMP form of the **permit** command.
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	Permits only IGMP packets. When you enter the **igmp** keyword, you must use the specific command syntax shown for the IGMP form of the **permit** 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.
tcp	Permits only TCP packets. When you enter the **tcp** keyword, you must use the specific command syntax shown for the TCP form of the **permit** command.

operator	(Optional) Compares source or destination ports. Operators are **eq** (equal) , **gt** (greater than),**lt** (less than), **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.
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 matches IP packets if their TCP headers contain the TCP flags that are specified by the *flag-name* argument. The **-** keyword matches 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: **ack**, **fin**, **psh**, **rst**, **syn**, and **urg**.

udp	Permits only UDP packets. When you enter the **udp** keyword, you must use the specific command syntax shown for the UDP form of the **permit** command.

Command Default

There are no specific conditions under which a packet passes the named access list.

Command Modes

Access list configuration (config-ext-nacl)

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 was 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.
12.4	The **drip** keyword was added to specify the TCP port number used for Optimized Edge Routing (OER) communication.
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.
12.4(22)T	The *word* argument was added to the **log** keyword.
Cisco IOS XE Release 3.2	This command was implemented on Cisco ASR 1000 Series Aggregation Services Routers.

Usage Guidelines

Use the **permit** command following the **ip access-list** command to define the conditions under which a packet passes the named access list.

Note In Cisco IOS XE, an inclusive port range for users to access a network cannot be matched in the extended ACL using the **permit** command.

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

log Keyword

A log message includes the access list number or access list name, and 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 port numbers, and the user-defined cookie or router-generated hash value. 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 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.

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

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.
add-ext	Match packets with Address Extension Option (147).
any-options	Match packets with any IP Option.

IP Option Value or Name	Description
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	Match the packets on the PSH bit.
record-route	Match packets with Router Record Route Option (7).
reflect	Create reflexive access list entry.
router-alert	Match packets with Router Alert Option (148).
rst	Matche the packets on the RST bit.
sdb	Match packets with Selective Directed Broadcast Option (149).
security	Match packets with Base Security Option (130).
ssr	Match packets with Strict Source Routing Option (137).
stream-id	Match packets with Stream ID Option (136).

IP Option Value or Name	Description
syn	Matches the packets on the SYN bit.
timestamp	Match packets with Time Stamp Option (68).
traceroute	Match packets with Trace Route Option (82).
ump	Match packets with Upstream Multicast Packet Option (152).
visa	Match packets with Experimental Access Control Option (142).
zsu	Match packets with Experimental Measurement Option (10).

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.

Permitting Optimized Edge Routing (OER) Communication

The **drip** keyword was introduced under the **tcp** keyword to support packet filtering in a network where OER is configured. The **drip** keyword specifies port 3949 that OER uses for internal communication. This option allows you to build a packet filter that permits communication between an OER master controller and border routers. The **drip** keyword is entered following the TCP source, destination addresses, and the **eq** operator. See the example in the “Examples” section.

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.

If you specify the **fragments** keyword in access list entries, 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 shows how to set 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 shows how to permit 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 ethernet0
 ip access-group legal in
```

The following example shows how to set a permit condition for an extended access list named filter2. The access list entry specifies that a packet may pass the named access list only if it contains the NSAP Addresses IP Option, which is represented by the IP Option value nsapa.

```
ip access-list extended filter2
 permit ip any any option nsapa
```

The following example shows how to set a permit condition for an extended access list named kmdfilter1. The access list entry specifies that a packet can pass the named access list only if the RST IP flag has been set for that packet:

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

The following example shows how to set a permit condition for an extended access list named kmdfilter1. The access list entry specifies that a packet can pass the named access list if the RST TCP flag or the FIN TCP flag has been set for that packet:

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

The following example shows how to verify the access list by using the **show access-lists** command and then to add an entry to an existing access list:

```
Router# show access-lists
Standard IP access list 1
2 permit 10.0.0.0, 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
ip access-list standard 1
 15 permit 10.0.0.0 0.0.255.255
```

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

```
ip access-list standard 1
 no 20
!Verify that the list has been removed.
Router# show access-lists
Standard IP access list 1
10 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
```

The following example 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-lists 101
Extended IP access list 101
    10 permit ip host 10.0.0.0 host 10.5.5.34
    20 permit icmp any any
    30 permit ip host 10.0.0.0 host 10.2.54.2
    40 permit ip host 10.0.0.0 host 10.3.32.3 log
ip access-list extended 101
 100 permit icmp any any
Router# show access-lists 101
Extended IP access list 101
    10 permit ip host 10.3.3.3 host 10.5.5.34
    20 permit icmp any any
    30 permit ip host 10.34.2.2 host 10.2.54.2
    40 permit ip host 10.3.4.31 host 10.3.32.3 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-lists 101
Extended IP access lists 101
    10 permit ip host 10.3.3.3 host 10.5.5.34
    20 permit icmp any any
    30 permit ip host 10.34.2.2 host 10.2.54.2
    40 permit ip host 10.3.4.31 host 10.3.32.3 log
ip access-lists extended 101
 20 permit udp host 10.1.1.1 host 10.2.2.2
%Duplicate sequence number.
Router# show access-lists 101
Extended IP access lists 101
    10 permit ip host 10.3.3.3 host 10.5.5.34
    20 permit icmp any any
    30 permit ip host 10.34.2.2 host 10.2.54.2
    40 permit ip host 10.3.4.31 host 10.3.32.3 log
```

The following example shows several **permit** 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 aaa.

```
Router# show access-lists aaa
Extended IP access lists aaa
 10 permit tcp any eq telnet any eq 450
```

```
 20 permit tcp any eq telnet any eq 679
 30 permit tcp any eq ftp any eq 450
 40 permit tcp any eq ftp any eq 679
```

Because the entries are all for the same **permit** 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 aaa
 no 10
 no 20
 no 30
 no 40
 permit tcp any eq telnet ftp any eq 450 679
```

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

```
Router# show access-lists aaa
Extended IP access list aaa
 10 permit 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
```

The following example shows how to configure a packet filter, for any TCP source and destination, that permits communication between an OER master controller and border router:

```
ip access-list extended 100
 permit any any tcp eq drip
 exit
```

The following example shows how to set a permit condition for an extended access list named filter_logging. The access list entry specifies that a packet may pass the named access list only if it is of TCP protocol type and destined to host 10.5.5.5, all other packets are denied. In addition, the logging mechanism is enabled and one of the user defined cookies (Permit_tcp_to_10.5.5.5 or Deny_all) is appended to the appropriate syslog entry.

```
ip access-list extended filter_logging
 permit tcp any host 10.5.5.5 log Permit_tcp_to_10.5.5.5
 deny ip any any log Deny_all
```

The following example shows how to configure a packet filter for any TCP source and destination that permits inbound and outbound BGP traffic:

```
ip access-list extended 100
 permit tcp any eq bgp any eq bgp
```

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.

Command	Description
access-list (IP standard)	Defines a standard IP access list.
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 log-update	Sets the threshold number of packets that cause a logging message.
ip access-list logging hash-generation	Enables hash value generation for ACE syslog entries.
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.
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.

port

To specify the port on which a device listens for RADIUS requests from configured RADIUS clients, use the **port** command in dynamic authorization local server configuration mode. To restore the default, use the **no** form of this command.

port *port-number*

no port *port-number*

Syntax Description

port-number	Port number. The default value is port 1700.

Command Default

The device listens for RADIUS requests on the default port (port 1700).

Command Modes

Dynamic authorization local server configuration (config-locsvr-da-radius)

Command History

Release	Modification
12.2(28)SB	This command was introduced.
Cisco IOS XE Release 2.6	This command was integrated into Cisco IOS XE Release 2.6.

Usage Guidelines

A device (such as a router) can be configured to allow an external policy server to dynamically send updates to the router. This functionality is facilitated by the CoA RADIUS extension. CoA introduced peer-to-peer capability to RADIUS, enabling a router and external policy server each to act as a RADIUS client and server. Use the **port** command to specify the ports on which the router will listen for requests from RADIUS clients.

Examples

The following example specifies port 1650 as the port on which the device listens for RADIUS requests:

```
aaa server radius dynamic-author
 client 10.0.0.1
 port 1650
```

Related Commands

Command	Description
aaa server radius dynamic-author	Configures a device as a AAA server to facilitate interaction with an external policy server.

port (TACACS+)

To specify the TCP port to be used for TACACS+ connections, use the **port**command in TACACS+ server configuration mode. To remove the TCP port, use the **no** form of this command.

port [*number*]

no port [*number*]

Syntax Description

number	(Optional) Specifies the port where the TACACS+ server receives access-request packets. The range is from 1 to 65535.

Command Default

If no port is configured, port 49 is used.

Command Modes

TACACS+ server configuration (config-server-tacacs)

Command History

Release	Modification
Cisco IOS XE Release 3.2S	This command was introduced.

Usage Guidelines

TCP port 49 is used if the *number* argument is not used when using the **port** command.

Examples

The following example shows how to specify TCP port 12:

```
Router (config)# tacacs server server1
Router(config-server-tacacs)# port 12
```

Related Commands

Command	Description
tacacs server	Configures the TACACS+ server for IPv6 or IPv4 and enters TACACS+ server configuration mode.