



# Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values

---

**First Published: August 18, 2006**  
**Last Updated: August 18, 2006**

This module describes how to use an IP access list to filter IP packets that contain certain IP Options, TCP flags, noncontiguous ports, or time-to-live (TTL) values.

## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values”](#) section on page 22.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required..

## Contents

- [Prerequisites for Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values, page 2](#)
- [Information About Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values, page 2](#)
- [How to Create an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values, page 5](#)
- [Configuration Examples for Filtering IP Options, TCP Flags, Noncontiguous Ports, and TTL Values, page 18](#)
- [Additional References, page 20](#)



---

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

- [Feature Information for Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values, page 22](#)

## Prerequisites for Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values

Before you perform any of the tasks in this module, you should be familiar with the information in the following modules:

- “IP Access List Overview”
- “Creating an IP Access List and Applying It to an Interface”

## Information About Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values

- [IP Options, page 2](#)
- [Benefits of Filtering IP Options, page 3](#)
- [Benefits of Filtering on TCP Flags, page 3](#)
- [TCP Flags, page 3](#)
- [Benefits of Using the ACL—Named ACL Support for Noncontiguous Ports on an Access Control Entry Feature, page 4](#)
- [How Filtering on TTL Works, page 4](#)
- [Benefits of Filtering on TTL, page 5](#)

### IP Options

IP uses four key mechanisms in providing its service: Type of Service, Time to Live, Options, and Header Checksum.

The Options, commonly referred to as IP Options, provide for control functions that are required in some situations but unnecessary for the most common communications. IP Options include provisions for time stamps, security, and special routing.

IP Options may or may not appear in datagrams. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular datagram, not their implementation. In some environments the security option may be required in all datagrams.

The option field is variable in length. There may be zero or more options. IP Options can have one of two formats:

- Format 1: A single octet of option-type.
- Format 2: An option-type octet, an option-length octet, and the actual option-data octets.

The option-length octet counts the option-type octet, the option-length octet, and the option-data octets.

The option-type octet is viewed as having three fields: a 1-bit copied flag, a 2-bit option class, and a 5-bit option number. These fields form an 8-bit value for the option type field. IP Options are commonly referred to by their 8-bit value.

For a complete list and description of IP Options, refer to RFC 791, *Internet Protocol* at the following URL: <http://www.faqs.org/rfcs/rfc791.html>

## Benefits of Filtering IP Options

- Filtering of packets that contain IP Options from the network relieves downstream routers and hosts of the load from options packets.
- This feature also minimizes load to the Route Processor (RP) for packets with IP Options that require RP processing on distributed systems. Previously, the packets were always routed to or processed by the RP CPU. Filtering the packets prevents them from impacting the RP.

## Benefits of Filtering on TCP Flags

The ACL TCP Flags Filtering feature provides a flexible mechanism for filtering on TCP flags. Before Cisco IOS Release 12.3(4)T, an incoming packet was matched as long as any TCP flag in the packet matched a flag specified in the access control entry (ACE). This behavior allows for a security loophole, because packets with all flags set could get past the access control list (ACL). The ACL TCP Flags Filtering feature allows you to select any combination of flags on which to filter. The ability to match on a flag set and on a flag not set gives you a greater degree of control for filtering on TCP flags, thus enhancing security.

Because TCP packets can be sent as false synchronization packets that can be accepted by a listening port, it is recommended that administrators of firewall devices set up some filtering rules to drop false TCP packets.

The ACEs that make up an access list can be configured to detect and drop unauthorized TCP packets by allowing only the packets that have a very specific group of TCP flags set or not set. The ACL TCP Flags Filtering feature gives users a greater degree of packet-filtering control in the following ways:

- Users can select any desired combination of TCP flags on which to filter TCP packets.
- Users can configure ACEs in order to allow matching on a flag that is set, as well as on a flag that is not set.

## TCP Flags

Table 1 lists the TCP flags, which are further described in RFC 793, *Transmission Control Protocol*.

**Table 1** TCP Flags

TCP Flag	Purpose
ACK	Acknowledge flag—Indicates that the acknowledgment field of a segment specifies the next sequence number the sender of this segment is expecting to receive.
FIN	Finish flag—Used to clear connections.
PSH	Push flag—Indicates the data in the call should be immediately pushed through to the receiving user.

**Table 1** TCP Flags (continued)

TCP Flag	Purpose
RST	Reset flag—Indicates that the receiver should delete the connection without further interaction.
SYN	Synchronize flag—Used to establish connections.
URG	Urgent flag—Indicates that the urgent field is meaningful and must be added to the segment sequence number.

## Benefits of Using the ACL—Named ACL Support for Noncontiguous Ports on an Access Control Entry Feature

This feature greatly reduces the number of ACEs required in an access control list to handle multiple entries for the same source address, destination address, and protocol. If you maintain large numbers of ACEs, we recommend that you use this feature to consolidate existing groups of access list entries wherever it is possible and also when you create new access list entries. When you configure access list entries with noncontiguous ports, you will have fewer access list entries to maintain.

## How Filtering on TTL Works

IP extended named and numbered access lists may filter on the TTL value of packets arriving at or leaving an interface. Packets with any possible TTL values 0 through 255 may be permitted or denied (filtered). Like filtering on other fields, such as source or destination address, the **ip access-group** command specifies **in** or **out**, which makes the access list ingress or egress and applies it to incoming or outgoing packets, respectively. The TTL value is checked in conjunction with the specified protocol, application, and any other settings in the access list entry, and all conditions must be met.

### Special Handling for Packets with TTL of 0 or 1 Arriving on Ingress Interface

The software switching paths—distributed Cisco Express Forwarding (dCEF), CEF, fast switching, and process switching—will usually permit or discard the packets based on the access list statements. However, when the TTL value of packets arriving on an ingress interface have a TTL of 0 or 1, special handling is required. The packets with a TTL of 0 or 1 get sent to the process level before the ingress access list is checked in CEF, dCEF, or fast switching paths. The ingress access list is applied to packets with TTL values 2 through 255 and a permit or deny decision is made.

Packets with a TTL value of 0 or 1 are sent to the process level because they will never be forwarded out of the device; the process level must check whether each packet is destined for the router or not and whether an Internet Control Message Protocol (ICMP) TTL Expire message needs to be sent back or not. This means that even if an ACL with TTL value 0 or 1 filtering is configured on the ingress interface with the intention to drop packets with a TTL of 0 or 1, the dropping of the packets will not happen in the faster paths. It will instead happen in the process level when the process applies the ACL. This is also true for hardware switching platforms. Packets with TTL 0 or 1 are sent to the process level of the route processor (RP) or Multilayer Switch Feature Card (MSFC).

On egress interfaces, access list filtering on TTL work just like other access list features. The check will happen in the fastest switching path enabled in the device. This is because the faster switching paths handle all the TTL values (0-255) equally on the egress interface.

### Control Plane Policing for Filtering TTL Values 0 and 1

The special behavior for packets with a TTL of 0 or 1 results in higher CPU usage for the device. If you are filtering on TTL value 0 or 1, you should use control plane policing (CPP) to protect the CPU from being overwhelmed. In order to leverage CPP, you must configure an access list especially for filtering TTL values 0 and 1 and apply the access list through CPP. This access list will be a separate access list from any interface access lists. Because CPP works for the entire system, not just on individual interfaces, you would need to configure only one such special access list for the entire device. This task is described in the section [“Enabling Control Plane Policing to Filter on TTL Values 0 and 1”](#) section on page 15.

## Benefits of Filtering on TTL

- Filtering on TTL provides a way to control which packets are allowed to reach the router or prevented from reaching the router. By looking at your network layout, you can choose whether to accept or deny packets from a certain router based on how many hops away it is. For example, in a small network, you can deny packets from a location more than three hops away. Filtering on TTL allows you to validate if the traffic originated from a neighboring device, as follows. You can accept only packets that reach you in one hop, for example, by accepting only packets with a TTL of one less than the initial TTL value of a particular protocol.
- Many control plane protocols communicate only with their neighbors, but receive packets from everyone. By applying to receiving routers an access list that filters on TTL, you can block unwanted packets.
- The Cisco IOS software sends all packets with a TTL of 0 or 1 to the process level to be processed. The device must then send an ICMP TTL expire message to the source. By filtering packets that have a TTL of 0 through 2, you can reduce the load on the process level.

## How to Create an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values

- [Filtering Packets That Contain IP Options, page 5](#)
- [Filtering Packets That Contain TCP Flags, page 7](#)
- [Configuring an Access Control Entry with Noncontiguous Ports, page 10](#)
- [Consolidating Access List Entries with Noncontiguous Ports into One Access List Entry, page 12](#)
- [Filtering Packets Based on TTL Value, page 14](#)
- [Enabling Control Plane Policing to Filter on TTL Values 0 and 1, page 15](#)

## Filtering Packets That Contain IP Options

The task in this section configures an access list to filter packets that contain IP Options and verifies that the access list has been configured correctly.

### Restrictions

- The ACL Support for Filtering IP Options feature can be used only with named, extended ACLs.

- Resource Reservation Protocol (RSVP) Multiprotocol Label Switching Traffic Engineering (MPLS TE), Internet Group Management Protocol Version 2 (IGMPV2), and other protocols that use IP Options packets may not function in drop or ignore mode if this feature is configured.
- On most Cisco routers, a packet with IP Options is not switched in hardware, but requires control plane software processing (primarily because there is a need to process the options and rewrite the IP header), so all IP packets with IP Options will be filtered and switched in software.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. [*sequence-number*] **deny** *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
5. [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
6. Repeat Step 4 or Step 5 as necessary.
7. **end**
8. **show ip access-lists** *access-list-name*

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<b>ip access-list extended</b> <i>access-list-name</i>  <b>Example:</b> Router(config)# ip access-list extended mylist1	Specifies the IP access list by name and enters named access list configuration mode.  <b>Note</b> The ACL Support for Filtering IP Options feature works only with named, extended ACLs.

Command or Action	Purpose
<p><b>Step 4</b></p> <pre>[sequence-number] deny protocol source source-wildcard destination destination-wildcard [option option-value] [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]</pre> <p><b>Example:</b> Router(config-ext-nacl)# deny ip any any option traceroute</p>	<p>(Optional) Specifies a <b>deny</b> statement in named IP access list mode.</p> <ul style="list-style-type: none"> <li>This access list happens to use a <b>deny</b> statement first, but a <b>permit</b> statement could appear first, depending on the order of statements you need.</li> <li>Use the <b>option</b> keyword and <i>option-value</i> argument to filter packets that contain a particular IP Option.</li> <li>In this example, any packet that contains the traceroute IP Option will be filtered out.</li> <li>Use the <b>no sequence-number</b> form of this command to delete an entry.</li> </ul>
<p><b>Step 5</b></p> <pre>[sequence-number] permit protocol source source-wildcard destination destination-wildcard [option option-value] [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]</pre> <p><b>Example:</b> Router(config-ext-nacl)# permit ip any any option security</p>	<p>Specifies a <b>permit</b> statement in named IP access list mode.</p> <ul style="list-style-type: none"> <li>In this example, any packet (not already filtered) that contains the security IP Option will be permitted.</li> <li>Use the <b>no sequence-number</b> form of this command to delete an entry.</li> </ul>
<p><b>Step 6</b></p> <p>Repeat Step 4 or Step 5 as necessary.</p>	<p>Allows you to revise the access list.</p>
<p><b>Step 7</b></p> <p><b>end</b></p> <p><b>Example:</b> Router(config-ext-nacl)# end</p>	<p>(Optional) Exits the configuration mode and returns to privileged EXEC mode.</p>
<p><b>Step 8</b></p> <p><b>show ip access-lists</b> <i>access-list-name</i></p> <p><b>Example:</b> Router# show ip access-lists mylist1</p>	<p>(Optional) Displays the contents of the IP access list.</p> <ul style="list-style-type: none"> <li>Review the output to verify that the access list includes the new entry.</li> </ul>

## What to Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.



**Note**

To effectively eliminate all packets that contain IP Options, we recommend that you configure the global **ip options drop** command.

## Filtering Packets That Contain TCP Flags

The task in this section configures an access list to filter packets that contain TCP flags and verifies that the access list has been configured correctly.

**Caution**

If a router having ACEs with the new syntax format is reloaded with an older version of Cisco IOS software that does not support the ACL TCP Flags Filtering feature, the ACEs will not be applied, leading to possible security loopholes.

**Restrictions**

- TCP flag filtering can be used only with named, extended ACLs.
- The ACL TCP Flags Filtering feature is supported only for Cisco IOS ACLs.
- Before Cisco IOS Release 12.3(4)T, the following command-line interface (CLI) format could be used to configure a TCP flag-checking mechanism:

**permit tcp any any rst**

The following format that represents the same ACE can be used with Cisco IOS Release 12.3(4)T and later releases:

**permit tcp any any match-any +rst**

Both the CLI formats are accepted; however, if the new keywords **match-all** or **match-any** are chosen, they must be followed by the new flags that are prefixed with “+” or “-”. It is advisable to use only the old format or the new format in a single ACL. You cannot mix and match the old and new CLI formats.

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. [*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*] [**log**] [**time-range** *time-range-name*] [**fragments**]
5. [*sequence-number*] **deny tcp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established** | {**match-any** | **match-all**} {+ | -} *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
6. Repeat Step 4 or Step 5 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.
7. **end**
8. **show ip access-lists** *access-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><b>enable</b></p> <p><b>Example:</b> Router&gt; enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<p><b>configure terminal</b></p> <p><b>Example:</b> Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<p><b>ip access-list extended access-list-name</b></p> <p><b>Example:</b> Router(config)# ip access-list extended kmd1</p>	<p>Specifies the IP access list by name and enters named access list configuration mode.</p> <p><b>Note</b> The ACL TCP Flags Filtering feature works only with named, extended ACLs.</p>
Step 4	<p>[sequence-number] <b>permit tcp</b> source source-wildcard [operator [port]] destination destination-wildcard [operator [port]] [established   {match-any   match-all} {+   -} flag-name] [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]</p> <p><b>Example:</b> Router(config-ext-nacl)# permit tcp any any match-any +rst</p>	<p>Specifies a <b>permit</b> statement in named IP access list mode.</p> <ul style="list-style-type: none"> <li>This access list happens to use a <b>permit</b> statement first, but a <b>deny</b> statement could appear first, depending on the order of statements you need.</li> <li>Use the TCP command syntax of the <b>permit</b> command.</li> <li>Any packet with the RST TCP header flag set will be matched and allowed to pass the named access list kmd1 in Step 3.</li> </ul>
Step 5	<p>[sequence-number] <b>deny tcp</b> source source-wildcard [operator [port]] destination destination-wildcard [operator [port]] [established   {match-any   match-all} {+   -} flag-name] [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]</p> <p><b>Example:</b> Router(config-ext-nacl)# deny tcp any any match-all -ack -fin</p>	<p>(Optional) Specifies a <b>deny</b> statement in named IP access list mode.</p> <ul style="list-style-type: none"> <li>This access list happens to use a <b>permit</b> statement first, but a <b>deny</b> statement could appear first, depending on the order of statements you need.</li> <li>Use the TCP command syntax of the <b>deny</b> command.</li> <li>Any packet that does not have the ACK flag set, and also does not have the FIN flag set, will not be allowed to pass the named access list kmd1 in Step 3.</li> <li>See the <b>deny (IP)</b> command for additional command syntax to permit upper-layer protocols (ICMP, IGMP, TCP, and UDP).</li> </ul>
Step 6	<p>Repeat Step 4 or Step 5 as necessary, adding statements by sequence number where you planned. Use the <b>no sequence-number</b> command to delete an entry.</p>	<p>Allows you to revise the access list.</p>

	Command or Action	Purpose
Step 7	<code>end</code>  <b>Example:</b> <code>Router(config-ext-nacl)# end</code>	(Optional) Exits the configuration mode and returns to privileged EXEC mode.
Step 8	<code>show ip access-lists access-list-name</code>  <b>Example:</b> <code>Router# show ip access-lists kmd1</code>	(Optional) Displays the contents of the IP access list. <ul style="list-style-type: none"> <li>Review the output to confirm that the access list includes the new entry.</li> </ul>

## What to Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.

## Configuring an Access Control Entry with Noncontiguous Ports

Perform this task to create access list entries that use noncontiguous TCP or UDP port numbers. Although this task uses TCP ports, you could use the UDP syntax of the **permit** and **deny** commands to filter noncontiguous UDP ports.

Although this task uses a **permit** command first, use the **permit** and **deny** commands in the order that achieves your filtering goals.

## Restrictions

The ACL—Named ACL Support for Noncontiguous Ports on an Access Control Entry feature can be used only with named, extended ACLs.

## SUMMARY STEPS

- enable**
- configure terminal**
- ip access-list extended access-list-name**
- `[sequence-number] permit tcp source source-wildcard [operator port [port]] destination destination-wildcard [operator [port]] [established | {match-any | match-all} {+ | -} flag-name] [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]`
- `[sequence-number] deny tcp source source-wildcard [operator port [port]] destination destination-wildcard [operator [port]] [established | {match-any | match-all} {+ | -} flag-name] [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]`
- Repeat Step 4 or Step 5 as necessary, adding statements by sequence number where you planned. Use the **no sequence-number** command to delete an entry.
- end**
- show ip access-lists access-list-name**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><b>enable</b></p> <p><b>Example:</b> Router&gt; enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<p><b>configure terminal</b></p> <p><b>Example:</b> Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<p><b>ip access-list extended access-list-name</b></p> <p><b>Example:</b> Router(config)# ip access-list extended kmdl</p>	<p>Specifies the IP access list by name and enters named access list configuration mode.</p>
Step 4	<p>[sequence-number] <b>permit tcp</b> source source-wildcard [operator port [port]] destination destination-wildcard [operator [port]] [<b>established</b>   {<b>match-any</b>   <b>match-all</b>} {+   -} flag-name] [<b>precedence precedence</b>] [<b>tos tos</b>] [<b>log</b>] [<b>time-range time-range-name</b>] [<b>fragments</b>]</p> <p><b>Example:</b> Router(config-ext-nacl)# permit tcp any eq telnet ftp any eq 450 679</p>	<p>Specifies a <b>permit</b> statement in named IP access list configuration mode.</p> <ul style="list-style-type: none"> <li>Operators include <b>lt</b> (less than), <b>gt</b> (greater than), <b>eq</b> (equal), <b>neq</b> (not equal), and <b>range</b> (inclusive range).</li> <li>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.</li> <li>The <b>range</b> operator requires two port numbers. You can configure up to 10 ports after the <b>eq</b> and <b>neq</b> operators. All other operators require one port number.</li> <li>To filter UDP ports, use the UDP syntax of this command.</li> </ul>
Step 5	<p>[sequence-number] <b>deny tcp</b> source source-wildcard [operator port [port]] destination destination-wildcard [operator [port]] [<b>established</b>   {<b>match-any</b>   <b>match-all</b>} {+   -} flag-name] [<b>precedence precedence</b>] [<b>tos tos</b>] [<b>log</b>] [<b>time-range time-range-name</b>] [<b>fragments</b>]</p> <p><b>Example:</b> Router(config-ext-nacl)# deny tcp any neq 45 565 632</p>	<p>(Optional) Specifies a <b>deny</b> statement in named access list configuration mode.</p> <ul style="list-style-type: none"> <li>Operators include <b>lt</b> (less than), <b>gt</b> (greater than), <b>eq</b> (equal), <b>neq</b> (not equal), and <b>range</b> (inclusive range).</li> <li>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.</li> <li>The <b>range</b> operator requires two port numbers. You can configure up to 10 ports after the <b>eq</b> and <b>neq</b> operators. All other operators require one port number.</li> <li>To filter UDP ports, use the UDP syntax of this command.</li> </ul>

	Command or Action	Purpose
Step 6	Repeat Step 4 or Step 5 as necessary, adding statements by sequence number where you planned. Use the <b>no</b> <i>sequence-number</i> command to delete an entry.	Allows you to revise the access list.
Step 7	<b>end</b>  <b>Example:</b> Router(config-ext-nacl)# end	(Optional) Exits named access list configuration mode and returns to privileged EXEC mode.
Step 8	<b>show ip access-lists</b> <i>access-list-name</i>  <b>Example:</b> Router# show ip access-lists kmd1	(Optional) Displays the contents of the access list. <ul style="list-style-type: none"> <li>Review the output to verify that the access list displays the new entries that you created.</li> </ul>

## Consolidating Access List Entries with Noncontiguous Ports into One Access List Entry

Perform this task to consolidate a group of access list entries with noncontiguous ports into one access list entry.

Although this task uses TCP ports, you could use the UDP syntax of the **permit** and **deny** commands to filter noncontiguous UDP ports.

Although this task uses a **permit** command first, use the **permit** and **deny** commands in the order that achieves your filtering goals.

### SUMMARY STEPS

- enable**
- show ip access-lists** *access-list-name*
- configure terminal**
- ip access-list extended** *access-list-name*
- no** [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
- [*sequence-number*] **permit** *protocol source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator port* [*port*]] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
- Repeat Steps 5 and 6 as necessary, adding **permit** or **deny** statements to consolidate access list entries where possible. Use the **no** *sequence-number* command to delete an entry.
- end**
- show ip access-lists** *access-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p><b>enable</b></p> <p><b>Example:</b> Router&gt; enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<p><b>show ip access-lists</b> <i>access-list-name</i></p> <p><b>Example:</b> Router# show ip access-lists mylist1</p>	<p>(Optional) Displays the contents of the IP access list.</p> <ul style="list-style-type: none"> <li>Review the output to see if you can consolidate any access list entries.</li> </ul>
Step 3	<p><b>configure terminal</b></p> <p><b>Example:</b> Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 4	<p><b>ip access-list extended</b> <i>access-list-name</i></p> <p><b>Example:</b> Router(config)# ip access-list extended mylist1</p>	<p>Specifies the IP access list by name and enters named access list configuration mode.</p>
Step 5	<p><b>no</b> [<i>sequence-number</i>] <b>permit</b> <i>protocol source source-wildcard destination destination-wildcard</i> [<b>option</b> <i>option-name</i>] [<b>precedence</b> <i>precedence</i>] [<b>tos</b> <i>tos</i>] [<b>log</b>] [<b>time-range</b> <i>time-range-name</i>] [<b>fragments</b>]</p> <p><b>Example:</b> Router(config-ext-nacl)# no 10</p>	<p>Removes the redundant access list entry that can be consolidated.</p> <ul style="list-style-type: none"> <li>Repeat this step to remove entries to be consolidated because only the port numbers differ.</li> <li>After this step is repeated to remove the access list entries 20, 30, and 40, for example, those entries are removed because they will be consolidated into one <b>permit</b> statement.</li> <li>If a <i>sequence-number</i> is specified, the rest of the command syntax is optional.</li> </ul>
Step 6	<p>[<i>sequence-number</i>] <b>permit</b> <i>protocol source source-wildcard</i> [<i>operator port</i> [<i>port</i>]] <i>destination destination-wildcard</i> [<i>operator port</i> [<i>port</i>]] [<b>option</b> <i>option-name</i>] [<b>precedence</b> <i>precedence</i>] [<b>tos</b> <i>tos</i>] [<b>log</b>] [<b>time-range</b> <i>time-range-name</i>] [<b>fragments</b>]</p> <p><b>Example:</b> Router(config-ext-nacl)# permit tcp any neq 45 565 632 any eq 23 45 34 43</p>	<p>Specifies a <b>permit</b> statement in named access list configuration mode.</p> <ul style="list-style-type: none"> <li>In this instance, a group of access list entries with noncontiguous ports was consolidated into one <b>permit</b> statement.</li> <li>You can configure up to 10 ports after the <b>eq</b> and <b>neq</b> operators.</li> </ul>
Step 7	<p>Repeat Steps 5 and 6 as necessary, adding <b>permit</b> or <b>deny</b> statements to consolidate access list entries where possible. Use the <b>no</b> <i>sequence-number</i> command to delete an entry.</p>	<p>Allows you to revise the access list.</p>

	Command or Action	Purpose
Step 8	<code>end</code>  <b>Example:</b> Router(config-std-nacl)# end	(Optional) Exits named access list configuration mode and returns to privileged EXEC mode.
Step 9	<code>show ip access-lists access-list-name</code>  <b>Example:</b> Router# show ip access-lists mylist1	(Optional) Displays the contents of the access list. <ul style="list-style-type: none"> <li>Review the output to verify that the redundant access list entries have been replaced with your new consolidated entries.</li> </ul>

## What To Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.

## Filtering Packets Based on TTL Value

Because access lists are very flexible, it is not possible to define only one combination of **permit** and **deny** commands to filter packets based on the TTL value. This task illustrates just one example that achieves TTL filtering. Configure the appropriate **permit** and **deny** statements that will accomplish your filtering plan.

## Restrictions

When the access list specifies the operation EQ or NEQ, routers running Cisco IOS Release 12.2S can have that access list specify up to ten TTL values. However, for Release 12.0S, only one TTL value can be specified.

## SUMMARY STEPS

- enable**
- configure terminal**
- ip access-list extended access-list-name**
- `[sequence-number] permit protocol source source-wildcard destination destination-wildcard [option option-name] [precedence precedence] [tos tos] [ttl operator value] [log] [time-range time-range-name] [fragments]`
- Continue to add **permit** or **deny** statements to achieve the filtering you want.
- exit**
- interface type number**
- ip access-group access-list-name {in | out}**

**DETAILED STEPS**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	<p><b>enable</b></p> <p><b>Example:</b> Router&gt; enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<p><b>configure terminal</b></p> <p><b>Example:</b> Router# configure terminal</p>	<p>Enters global configuration mode.</p>
<b>Step 3</b>	<p><b>ip access-list extended access-list-name</b></p> <p><b>Example:</b> Router(config)# ip access-list extended ttlfilter</p>	<p>Defines an IP access list by name.</p> <ul style="list-style-type: none"> <li>An access list that filters on TTL value must be an extended access list.</li> </ul>
<b>Step 4</b>	<p>[sequence-number] <b>permit</b> protocol source source-wildcard destination destination-wildcard [option option-name] [precedence precedence] [tos tos] [ttl operator value] [log] [time-range time-range-name] [fragments]</p> <p><b>Example:</b> Router(config-ext-nacl)# permit ip host 172.16.1.1 any ttl lt 2</p>	<p>Sets conditions to allow a packet to pass a named IP access list.</p> <ul style="list-style-type: none"> <li>Every access list must have at least one <b>permit</b> statement.</li> <li>This example permits packets from source 172.16.1.1 to any destination with a TTL value less than 2.</li> </ul>
<b>Step 5</b>	<p>Continue to add <b>permit</b> or <b>deny</b> statements to achieve the filtering you want.</p>	—
<b>Step 6</b>	<p><b>exit</b></p> <p><b>Example:</b> Router(config-ext-nacl)# exit</p>	<p>Exits any configuration mode to the next highest mode in the CLI mode hierarchy.</p>
<b>Step 7</b>	<p><b>interface type number</b></p> <p><b>Example:</b> Router(config)# interface ethernet 0</p>	<p>Configures an interface type and enters interface configuration mode.</p>
<b>Step 8</b>	<p><b>ip access-group access-list-name {in   out}</b></p> <p><b>Example:</b> Router(config-if)# ip access-group ttlfilter in</p>	<p>Applies the access list to an interface.</p>

## Enabling Control Plane Policing to Filter on TTL Values 0 and 1

Perform this task if you want to filter IP packets based on a TTL value of 0 or 1 and you want to protect the CPU from being overwhelmed. This task configures an access list for classification on TTL 0 and 1, configures Modular QoS CLI (MQC), and applies the policy map to the control plane. Any packets that pass the access list are dropped. This special access list is separate from any interface access lists.

Because access lists are very flexible, it is not possible to define only one combination of **permit** and **deny** commands to filter packets based on the TTL value. This task illustrates just one example that achieves TTL filtering. Configure the appropriate **permit** and **deny** statements that will accomplish your filtering plan.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard ttl operator value*
5. Continue to add **permit** or **deny** statements to achieve the filtering you want.
6. **exit**
7. **class-map** *class-map-name* [**match-all** | **match-any**]
8. **match access-group** {*access-group* | **name** *access-group-name*}
9. **exit**
10. **policy-map** *policy-map-name*
11. **class** {*class-name* | **class-default**}
12. **drop**
13. **exit**
14. **exit**
15. **control-plane**
16. **service-policy** {**input** | **output**} *policy-map-name*

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<b>ip access-list extended</b> <i>access-list-name</i>  <b>Example:</b> Router(config)# ip access-list extended ttlfilter	Defines an IP access list by name. <ul style="list-style-type: none"> <li>• An access list that filters on a TTL value must be an extended access list.</li> </ul>

	Command or Action	Purpose
Step 4	<p>[sequence-number] <b>permit</b> protocol source source-wildcard destination destination-wildcard <b>t</b>tl operator value</p> <p><b>Example:</b> Router(config-ext-nacl)# permit ip host 172.16.1.1 any ttl lt 2</p>	<p>Sets conditions to allow a packet to pass a named IP access list.</p> <ul style="list-style-type: none"> <li>• Every access list must have at least one <b>permit</b> statement.</li> <li>• This example permits packets from source 172.16.1.1 to any destination with a TTL value less than 2.</li> </ul>
Step 5	Continue to add <b>permit</b> or <b>deny</b> statements to achieve the filtering you want.	The packets that pass the access list will be dropped.
Step 6	<p><b>exit</b></p> <p><b>Example:</b> Router(config-ext-nacl)# exit</p>	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.
Step 7	<p><b>class-map</b> class-map-name [<b>match-all</b>   <b>match-any</b>]</p> <p><b>Example:</b> Router(config)# class-map acl-filtering</p>	Creates a class map to be used for matching packets to a specified class.
Step 8	<p><b>match access-group</b> {access-group   <b>name</b> access-group-name}</p> <p><b>Example:</b> Router(config-cmap)# match access-group name ttlfilter</p>	Configures the match criteria for a class map on the basis of the specified access control list.
Step 9	<p><b>exit</b></p> <p><b>Example:</b> Router(config-cmap)# exit</p>	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.
Step 10	<p><b>policy-map</b> policy-map-name</p> <p><b>Example:</b> Router(config)# policy-map acl-filter</p>	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.
Step 11	<p><b>class</b> {class-name   <b>class-default</b>}</p> <p><b>Example:</b> Router(config-pmap)# class acl-filter-class</p>	Specifies the name of the class whose policy you want to create or change or to specify the default class (commonly known as the class-default class) before you configure its policy.
Step 12	<p><b>drop</b></p> <p><b>Example:</b> Router(config-pmap-c)# drop</p>	Configures a traffic class to discard packets belonging to a specific class.
Step 13	<p><b>exit</b></p> <p><b>Example:</b> Router(config-pmap-c)# exit</p>	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.

	Command or Action	Purpose
Step 14	<b>exit</b>  <b>Example:</b> Router(config-pmap)# exit	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.
Step 15	<b>control-plane</b>  <b>Example:</b> Router(config)# control-plane	Associates or modifies attributes or parameters that are associated with the control plane of the device.
Step 16	<b>service-policy</b> {input   output} <i>policy-map-name</i>  <b>Example:</b> Router(config-cp)# service-policy input acl-filter	Attaches a policy map to a control plane for aggregate control plane services.

## Configuration Examples for Filtering IP Options, TCP Flags, Noncontiguous Ports, and TTL Values

- [Example: Filtering Packets That Contain IP Options, page 18](#)
- [Example: Filtering Packets That Contain TCP Flags, page 19](#)
- [Example: Creating an Access List Entry with Noncontiguous Ports, page 19](#)
- [Example: Consolidating Some Existing Access List Entries into One Access List Entry with Noncontiguous Ports, page 19](#)
- [Example: Filtering on TTL Value, page 20](#)
- [Example: Control Plane Policing to Filter on TTL Values 0 and 1, page 20](#)

### Example: Filtering Packets That Contain IP Options

The following example shows an extended access list named mylist2 that contains access list entries (ACEs) that are configured to permit TCP packets only if they contain the IP Options that are specified in the ACEs:

```
ip access-list extended mylist2
 10 permit ip any any option eool
 20 permit ip any any option record-route
 30 permit ip any any option zsu
 40 permit ip any any option mtup
```

The **show access-list** command has been entered to show how many packets were matched and therefore permitted:

```
Router# show ip access-list mylist2

Extended IP access list test
10 permit ip any any option eool (1 match)
20 permit ip any any option record-route (1 match)
30 permit ip any any option zsu (1 match)
40 permit ip any any option mtup (1 match)
```

## Example: Filtering Packets That Contain TCP Flags

The following access list allows TCP packets only if the TCP flags ACK and SYN are set and the FIN flag is not set:

```
ip access-list extended aaa
 permit tcp any any match-all +ack +syn -fin
end
```

The **show access-list** command has been entered to display the ACL:

```
Router# show access-list aaa

Extended IP access list aaa
 10 permit tcp any any match-all +ack +syn -fin
```

## Example: Creating an Access List Entry with Noncontiguous Ports

The following access list entry can be created because up to ten ports can be entered after the **eq** and **neq** operators:

```
ip access-list extended aaa
 permit tcp any eq telnet ftp any eq 23 45 34
end
```

Enter the **show access-lists** command to display the newly created access list entry.

```
Router# show access-lists aaa

Extended IP access list aaa
 10 permit tcp any eq telnet ftp any eq 23 45 34
```

## Example: Consolidating Some Existing Access List Entries into One Access List Entry with Noncontiguous Ports

The **show access-lists** command is used to display a group of access list entries for the access list named abc:

```
Router# show access-lists abc

Extended IP access list abc
 10 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 abc
 no 10
 no 20
 no 30
 no 40
 permit tcp any eq telnet ftp any eq 450 679
end
```

When the **show access-lists** command is reentered, the consolidated access list entry is displayed:

```
Router# show access-lists abc

Extended IP access list abc
 10 permit tcp any eq telnet ftp any eq 450 679
```

## Example: Filtering on TTL Value

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 it sends log messages about such packets to the console. All other packets are denied.

```
ip access-list extended incomingfilter
 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
!
interface ethernet 0
 ip access-group incomingfilter in
```

## Example: Control Plane Policing to Filter on TTL Values 0 and 1

The following example configures a traffic class called `acl-filter-class` for use in a policy map called `acl-filter`. An access list permits IP packets from any source having a TTL of 0 or 1. Any packets matching the access list are dropped. The policy map is attached to the control plane.

```
ip access-list extended ttlfilter
 permit ip any any ttl eq 0 1
class-map acl-filter-class
 match access-group name ttlfilter
policy-map acl-filter
 class acl-filter-class
 drop
control-plane
 service-policy input acl-filter
```

## Additional References

### Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Commands List, All Releases</a>
Security commands	<a href="#">Cisco IOS Security Command Reference</a>
Configuring the router to drop or ignore packets containing IP Options by using the <b>no ip options</b> command.	<a href="#">ACL IP Options Selective Drop</a>
QoS commands	<a href="#">Cisco IOS Quality of Service Solutions Command Reference</a>

## MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a>

## RFCs

RFC	Title
RFC 791	<i>Internet Protocol</i> <a href="http://www.faqs.org/rfcs/rfc791.html">http://www.faqs.org/rfcs/rfc791.html</a>
RFC 793	<i>Transmission Control Protocol</i>
RFC 1393	<i>Traceroute Using an IP Option</i>

## Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

# Feature Information for Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values

Table 2 lists the features in this module and provides links to specific configuration information.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



## Note

Table 2 lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

**Table 2** Feature Information for Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values

Feature Name	Releases	Feature Configuration Information
ACL Support for Filtering IP Options	12.3(4)T 12.2(25)S	This feature allows you to filter packets having IP Options, in order to prevent routers from becoming saturated with spurious packets. See the following sections: <ul style="list-style-type: none"> <li>• <a href="#">Filtering Packets That Contain IP Options, page 5</a></li> <li>• <a href="#">Example: Filtering Packets That Contain IP Options, page 18</a></li> </ul>
ACL TCP Flags Filtering	12.3(4)T 12.2(25)S	This feature provides a flexible mechanism for filtering on TCP flags. Before Cisco IOS Release 12.3(4)T, an incoming packet was matched as long as any TCP flag in the packet matched a flag specified in the access control entry (ACE). This behavior allows for a security loophole, because packets with all flags set could get past the access control list (ACL). The ACL TCP Flags Filtering feature allows you to select any combination of flags on which to filter. The ability to match on a flag set and on a flag not set gives you a greater degree of control for filtering on TCP flags, thus enhancing security. See the following sections: <ul style="list-style-type: none"> <li>• <a href="#">Filtering Packets That Contain TCP Flags, page 7</a></li> <li>• <a href="#">Example: Filtering Packets That Contain TCP Flags, page 19</a></li> </ul>

**Table 2** Feature Information for Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values (continued)

Feature Name	Releases	Feature Configuration Information
ACL—Named ACL Support for Noncontiguous Ports on an Access Control Entry	12.3(7)T 12.2(25)S	<p>This feature allows you to specify noncontiguous ports in a single access control entry, which greatly reduces the number of entries required in an access control list when several entries have the same source address, destination address, and protocol, but differ only in the ports. See the following sections:</p> <ul style="list-style-type: none"> <li>• <a href="#">Configuring an Access Control Entry with Noncontiguous Ports, page 10</a></li> <li>• <a href="#">Consolidating Access List Entries with Noncontiguous Ports into One Access List Entry, page 12</a></li> <li>• <a href="#">Example: Creating an Access List Entry with Noncontiguous Ports, page 19</a></li> <li>• <a href="#">Example: Consolidating Some Existing Access List Entries into One Access List Entry with Noncontiguous Ports, page 19</a></li> </ul>
ACL Support for Filtering on TTL Value	12.4(2)T	<p>Customers may use extended IP access lists (named or numbered) to filter packets based on their time-to-live (TTL) value, from 0 to 255. This filtering enhances a customer’s control over which packets reach a router. See the following sections:</p> <ul style="list-style-type: none"> <li>• <a href="#">Filtering Packets Based on TTL Value, page 14</a></li> <li>• <a href="#">Enabling Control Plane Policing to Filter on TTL Values 0 and 1, page 15</a></li> <li>• <a href="#">Example: Filtering on TTL Value, page 20</a></li> <li>• <a href="#">Example: Control Plane Policing to Filter on TTL Values 0 and 1, page 20</a></li> </ul>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

© 2006–2007 Cisco Systems, Inc. All rights reserved.

