Creating an IP Access List for Filtering

Last Updated: January 18, 2012

This module describes how to use an IP access list to filter IP packets that contain certain IP options, TCP flags, or noncontiguous ports.

- Finding Feature Information, page 1
- Information About IP Access List for Filtering, page 1
- How to Create an IP Access List for Filtering, page 3
- Configuration Examples for IP Access Lists for Filtering, page 13
- Additional References, page 14
- Feature Information for Using an IP Access List for Filtering, page 16

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 Table at the end of this document.

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

Information About IP Access List for Filtering

- IP Options, page 2
- Benefits of Filtering IP Options, page 2
- Benefits of Filtering on TCP Flags, page 2
- TCP Flags, page 3
- Benefits of Using the ACL-Named ACL Support for Noncontiguous Ports, page 3
**IP Options**

IP uses four key mechanisms in providing its service: Type of Service (ToS), Time to Live (TTL), 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](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. Without this feature, when multiple flags are specified on the access control entry (ACE), the packet will be allowed if one of the flags is a match. 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**

The table below lists the TCP flags, which are further described in RFC 793, *Transmission Control Protocol*.

<table>
<thead>
<tr>
<th>TCP Flag</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK</td>
<td>Acknowledge flag—Indicates that the acknowledgment field of a segment specifies the next sequence number the sender of this segment is expecting to receive.</td>
</tr>
<tr>
<td>FIN</td>
<td>Finish flag—Used to clear connections.</td>
</tr>
<tr>
<td>PSH</td>
<td>Push flag—Indicates the data in the call should be immediately pushed through to the receiving user.</td>
</tr>
<tr>
<td>RST</td>
<td>Reset flag—Indicates that the receiver should delete the connection without further interaction.</td>
</tr>
<tr>
<td>SYN</td>
<td>Synchronize flag—Used to establish connections.</td>
</tr>
<tr>
<td>URG</td>
<td>Urgent flag—Indicates that the urgent field is meaningful and must be added to the segment sequence number.</td>
</tr>
</tbody>
</table>

**Benefits of Using the ACL-Named ACL Support for Noncontiguous Ports**

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 to Create an IP Access List for Filtering**

- Filtering Packets That Contain IP Options, page 3
- Filtering Packets That Contain TCP Flags, page 6
- Configuring an Access Control Entry with Noncontiguous Ports, page 8
- Consolidating Access List Entries with Noncontiguous Ports into One ACL, page 11

**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.
Note

- 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 1 enable                      | Enables privileged EXEC mode.  
  - Enter your password if prompted. |
| Example:                           |         |
| Router> enable                     |         |
| Step 2 configure terminal          | Enters global configuration mode. |
| Example:                           |         |
| Router# configure terminal         |         |
| Step 3 ip access-list extended access-list-name | Specifies the IP access list by name and enters named access list configuration mode. |
| Example:                           |         |
| Router(config)# ip access-list extended mylist1 |         |

Note The ACL Support for Filtering IP Options feature works only with named, extended ACLs.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td>(Optional) Specifies a <em>deny</em> statement in named IP access list mode.</td>
</tr>
</tbody>
</table>
| [sequence-number] deny protocol source source-wildcard destination destination-wildcard [option option-value] [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments] | - This access list happens to use a *deny* statement first, but a *permit* statement could appear first, depending on the order of statements you need.  
- Use the *option* keyword and *option-value* argument to filter packets that contain a particular IP Option.  
- In this example, any packet that contains the traceroute IP option will be filtered out.  
- Use the no *sequence-number* form of this command to delete an entry. |
| **Example:** | Router(config-ext-nacl)# deny ip any any option traceroute |
| **Step 5** | Specifies a *permit* statement in named IP access list mode. |
| [sequence-number] permit protocol source source-wildcard destination destination-wildcard [option option-value] [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments] | - In this example, any packet (not already filtered) that contains the security IP option will be permitted.  
- Use the no *sequence-number* form of this command to delete an entry. |
| **Example:** | Router(config-ext-nacl)# permit ip any any option security |
| **Step 6** | Allows you to revise the access list. |
| Repeat Step 4 or Step 5 as necessary. | |
| **Step 7** | (Optional) Exits named access list configuration mode and returns to privileged EXEC mode. |
| end | |
| **Example:** | Router(config-ext-nacl)# end |
| **Step 8** | (Optional) Displays the contents of the IP access list. |
| show ip access-lists access-list-name | - Review the output to verify that the access list includes the new entry. |
| **Example:** | Router# show ip access-lists mylist1 |
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.

**Note**
- TCP flag filtering can be used only with named, extended ACLs.
- The ACL TCP Flags Filtering feature is supported only for Cisco IOS XE ACLs.
- In releases prior to Cisco IOS XE Release 2.1, the following CLI format could be used to configure a TCP flag-checking mechanism:
  
  ```
  permit tcp any any rst
  ```

- In Cisco IOS XE Release 2.1 and later releases, the following CLI format that represents the same ACE can be used:
  
  ```
  permit tcp any any match-any +rst
  ```

  Both the CLI formats are accepted; however, if new keywords `match-all` or `match-any` are chosen, they must be followed by 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.

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

**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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ip access-list extended <em>access-list-name</em></td>
<td>Specifies the IP access list by name and enters named access list configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# ip access-list extended acl-extd-1</td>
<td><strong>Note</strong> The ACL TCP Flags Filtering feature works only with named, extended ACLs.</td>
</tr>
<tr>
<td><strong>Step 4</strong> [sequence-number] permit tcp source source-wildcard [operator [port]] destination destination-wildcard [operator [port]] [established {match-any</td>
<td>match-all} {+</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-ext-nacl)# permit tcp any any match-any +rst</td>
<td>• This access list happens to use a <strong>permit</strong> statement first, but a <strong>deny</strong> statement could appear first, depending on the order of statements you need.</td>
</tr>
<tr>
<td><strong>Step 5</strong> [sequence-number] deny tcp source source-wildcard [operator [port]] destination destination-wildcard [operator [port]] [established {match-any</td>
<td>match-all} {+</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-ext-nacl)# deny tcp any any match-all -ack -fin</td>
<td>• This access list happens to use a <strong>permit</strong> statement first, but a <strong>deny</strong> statement could appear first, depending on the order of statements you need.</td>
</tr>
<tr>
<td><strong>Step 6</strong> 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.</td>
<td>Allows you to revise the access list.</td>
</tr>
</tbody>
</table>
### Command or Action | Purpose
--- | ---
**Step 7** end | (Optional) Exits the configuration mode and returns to privileged EXEC mode.

**Example:**

```
Router(config-ext-nacl)# end
```

**Step 8** show ip access-lists access-list-name | (Optional) Displays the contents of the IP access list.

**Example:**

```
Router# show ip access-lists kmd1
```

- Review the output to confirm that the access list includes the new entry.

---

**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.

---

**Note**

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** `enable` | Enables privileged EXEC mode.  
  • Enter your password if prompted. |
| **Example:**  
  `Router> enable` | |
| **Step 2** `configure terminal` | Enters global configuration mode. |
| **Example:**  
  `Router# configure terminal` | |
| **Step 3** `ip access-list extended access-list-name` | Specifies the IP access list by name and enters named access list configuration mode. |
| **Example:**  
  `Router(config)# ip access-list extended acl-extd-1` | |
### Command or Action

| Step 4 | `[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] |

**Example:**

```
Router(config-ext-nacl)# permit tcp any eq telnet ftp any eq 450 679
```

| Step 5 | `[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] |

**Example:**

```
Router(config-ext-nacl)# deny tcp any neq 45 565 632
```

| Step 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. |

| Step 7 | `end` |

**Example:**

```
Router(config-ext-nacl)# end
```

| Step 8 | `show ip access-lists access-list-name` |

**Example:**

```
Router# show ip access-lists kmd1
```
Consolidating Access List Entries with Noncontiguous Ports into One ACL

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**

1. `enable`
2. `show ip access-lists access-list-name`
3. `configure terminal`
4. `ip access-list extended access-list-name`
5. `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]`
6. `[sequence-number] permit protocol source source-wildcard [operator port [port]] destination destination-wildcard [option option-name] [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]`
7. 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.
8. `end`
9. `show ip access-lists access-list-name`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> show ip access-lists access-list-name</td>
<td>(Optional) Displays the contents of the IP access list.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Review the output to see if you can consolidate any access list entries.</td>
</tr>
<tr>
<td>Router# show ip access-lists mylist1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command/Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>ip access-list extended <strong>access-list-name</strong></td>
<td>Specifies the IP access list by name and enters named access list configuration mode.</td>
</tr>
</tbody>
</table>
| 5    | **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**] | Removes the redundant access list entry that can be consolidated.  
  - Repeat this step to remove entries to be consolidated because only the port numbers differ.  
  - 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 **permit** statement.  
  - If a **sequence-number** is specified, the rest of the command syntax is optional. |
| 6    | `[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**] | Specifies a **permit** statement in named access list configuration mode.  
  - In this instance, a group of access list entries with noncontiguous ports was consolidated into one **permit** statement.  
  - You can configure up to 10 ports after the `eq` and `neq` operators. |
| 7    | 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. | Allows you to revise the access list. |
| 8    | **end** | (Optional) Exits named access list configuration mode and returns to privileged EXEC mode. |
| 9    | **show ip access-lists** **access-list-name** | (Optional) Displays the contents of the access list.  
  - Review the output to verify that the redundant access list entries have been replaced with your new consolidated entries. |

### What To Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.
Configuration Examples for IP Access Lists for Filtering

- Example: Filtering Packets That Contain IP Options, page 13
- Example: Filtering Packets That Contain TCP Flags, page 13
- Example: Creating an Access List Entry with Noncontiguous Ports, page 13
- Example: Consolidating Existing Access List Entries into One Access List Entry with Noncontiguous Ports, page 14

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 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 `aaa`:

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

**Additional References**

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td><em>Cisco IOS Master Commands List, All Releases</em></td>
</tr>
</tbody>
</table>
## Related Topic

<table>
<thead>
<tr>
<th>Security commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Cisco IOS Security Command Reference: Commands A to C</td>
</tr>
<tr>
<td>- Cisco IOS Security Command Reference: Commands D to L</td>
</tr>
<tr>
<td>- Cisco IOS Security Command Reference: Commands M to R</td>
</tr>
<tr>
<td>- Cisco IOS Security Command Reference: Commands S to Z</td>
</tr>
</tbody>
</table>

## Configuring the router to drop or ignore packets containing IP Options by using the `no ip options` command.

“ACL IP Options Selective Drop” module

## QoS commands

Cisco IOS Quality of Service Solutions Command Reference

## Standards and RFCs

<table>
<thead>
<tr>
<th>Standard &amp; RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 791</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>RFC 793</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>RFC 1393</td>
<td>Traceroute Using an IP Option</td>
</tr>
</tbody>
</table>

## MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS XE 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></td>
</tr>
</tbody>
</table>
Feature Information for Using an IP Access List for Filtering

The following table provides release information about the feature or features described in this module. This table 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.

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

Table 2  Feature Information for Creating an IP Access List for Filtering

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACL—DHCP Matching</td>
<td>Cisco IOS XE Release 3.5S</td>
<td>In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.</td>
</tr>
<tr>
<td>ACL--Named ACL Support for Noncontiguous Ports on an Access Control Entry</td>
<td>Cisco IOS XE Release 2.1</td>
<td>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. No commands were introduced or modified for this feature.</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Configuration Information</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>---------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ACL Support for Filtering IP</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature allows you to filter packets having IP options, in order to prevent routers from becoming saturated with spurious packets. No commands were introduced or modified for this feature.</td>
</tr>
<tr>
<td>Options</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACL TCP Flags Filtering</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature provides a flexible mechanism for filtering on TCP flags. It 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. No commands were introduced or modified for this feature.</td>
</tr>
</tbody>
</table>

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [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. (1110R)

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.

© 2012 Cisco Systems, Inc. All rights reserved.