Adding an EtherType Access List

This chapter describes how to configure EtherType access lists and includes the following topics:

- Information About EtherType Access Lists, page 12-1
- Licensing Requirements for EtherType Access Lists, page 12-2
- Guidelines and Limitations, page 12-2
- Default Settings, page 12-3
- Configuring EtherType Access Lists, page 12-4
- Monitoring EtherType Access Lists, page 12-6
- What to Do Next, page 12-6
- Configuration Examples for EtherType Access Lists, page 12-7
- Feature History for EtherType Access Lists, page 12-7

Information About EtherType Access Lists

An EtherType access list is made up of one or more Access List Entries (ACEs) that specify an EtherType. This section includes the following topics:

- Supported EtherTypes, page 12-1
- Implicit Permit of IP and ARPs Only, page 12-2
- Implicit and Explicit Deny ACE at the End of an Access List, page 12-2
- Allowing MPLS, page 12-2

Supported EtherTypes

An EtherType ACE controls any EtherType identified by a 16-bit hexadecimal number. You can apply only one access list of each type (extended and EtherType) to each direction of an interface. You can also apply the same access lists on multiple interfaces.
Implicit Permit of IP and ARPs Only

IPv4 traffic is allowed through the transparent firewall automatically from a higher security interface to a lower security interface, without an access list. ARPs are allowed through the transparent firewall in both directions without an access list. ARP traffic can be controlled by ARP inspection.

However, to allow any traffic with EtherTypes other than IPv4 and ARP, you need to apply an EtherType access list, even from a high security to a low security interface.

Because EtherTypes are connectionless, you need to apply the access list to both interfaces if you want traffic to pass in both directions.

Implicit and Explicit Deny ACE at the End of an Access List

For EtherType access lists, the implicit deny at the end of the access list does not affect IP traffic or ARPs; for example, if you allow EtherType 8037, the implicit deny at the end of the access list does not now block any IP traffic that you previously allowed with an extended access list (or implicitly allowed from a high security interface to a low security interface). However, if you explicitly deny all traffic with an EtherType ACE, then IP and ARP traffic is denied.

Allowing MPLS

If you allow MPLS, ensure that Label Distribution Protocol and Tag Distribution Protocol TCP connections are established through the ASA by configuring both MPLS routers connected to the ASA to use the IP address on the ASA interface as the router-id for LDP or TDP sessions. (LDP and TDP allow MPLS routers to negotiate the labels [addresses] used to forward packets.)

On Cisco IOS routers, enter the appropriate command for your protocol, either LDP or TDP. The interface is the interface connected to the ASA.

hostname(config)# mpls ldp router-id interface force

Or

hostname(config)# tag-switching tdp router-id interface force

Licensing Requirements for EtherType Access Lists

The following table shows the licensing requirements for this feature:

<table>
<thead>
<tr>
<th>Model</th>
<th>License Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>All models</td>
<td>Base License.</td>
</tr>
</tbody>
</table>

Guidelines and Limitations

This section includes the guidelines and limitations for this feature:

- Context Mode Guidelines, page 12-3
- Firewall Mode Guidelines, page 12-3
- Additional Guidelines and Limitations, page 12-3
Context Mode Guidelines
Available in single and multiple context modes.

Firewall Mode Guidelines
Supported in transparent firewall mode only.

Additional Guidelines and Limitations
The following guidelines and limitations apply to EtherType access lists:
- When you enter the `access-list` command for a given access list name, the ACE is added to the end of the access list.
- EtherType access lists support Ethernet V2 frames.
- 802.3-formatted frames are not handled by the access list because they use a length field as opposed to a type field. Bridge protocol data units, which are allowed by default, are the only exception; they are SNAP-encapsulated, and the adaptive security appliance is designed to specifically handle BPDUs.
- Because EtherTypes are connectionless, you need to apply the ACL to both interfaces if you want traffic to pass in both directions.
- If you allow MPLS, ensure that LDP and TDP TCP connections are established through the adaptive security appliance by configuring both MPLD routers connected to the adaptive security appliance to use the IP address on the adaptive security appliance interface as the router-ID for LDP or TDP sessions. (LDP and TDP allow MPLS routers to negotiate the labels, or addresses, used to forward packets.)
- For EtherType access lists, the implicit deny at the end of the access list does not affect IP traffic or ARPs; for example, if you allow EtherType 8037, the implicit deny at the end of the access list does not now block any IP traffic that you previously allowed with an extended access list (or implicitly allowed from a high security interface to a low security interface). However, if you explicitly deny all traffic with an EtherType ACE, then IP and ARP traffic is denied.
- You can apply only one access list of each type (extended and Ethertype) to each direction of an interface. You can also apply the same access lists on multiple interfaces.

Default Settings

Table 12-1 lists the default settings for EtherType access lists parameters.

**Table 12-1  Default EtherType Access Lists Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>bpdu</td>
<td>By default, BPDUs are permitted.</td>
</tr>
<tr>
<td>deny</td>
<td>The adaptive security appliance denies all packets on the originating interface unless you specifically permit access.</td>
</tr>
</tbody>
</table>
Configuring EtherType Access Lists

This section includes the following topics:

- Task Flow for Configuring EtherType Access Lists, page 12-4
- Adding EtherType Access Lists, page 12-5
- Adding Remarks to Access Lists, page 12-6

**Task Flow for Configuring EtherType Access Lists**

Use the following guidelines to create and implement an access list:

- Create an access list by adding an ACE and applying an access list name, as shown in the “Adding EtherType Access Lists” section on page 12-5.
- Apply the access list to an interface. (See the “Applying an Access List to an Interface” section on page 35-4 for more information.)

---

**Table 12-1 Default EtherType Access Lists Parameters (continued)**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>deny</td>
<td>Access list logging generates system log message 106023 for denied packets. Deny packets must be present to log denied packets.</td>
</tr>
<tr>
<td>log</td>
<td>When the log optional keyword is specified, the default severity level for system log message 106100 is 6 (informational).</td>
</tr>
</tbody>
</table>
Adding EtherType Access Lists

To configure an access list that controls traffic based upon its EtherType, enter the following command:

```
access-list access_list_name ethertype
  (deny | permit) (ipx | bpdu | mpls-unicast
  | mpls-multicast | any | hex_number)
```

Example:
```
hostname(config)#
hostname(config)#
access-list ETHER ethertype permit ipx
```

### Command Purpose

- **access-list** `<access_list_name>` `<ethertype>`
  - Adds an EtherType ACE.
  - The `access_list_name` argument lists the name or number of an access list. When you specify an access list name, the ACE is added to the end of the access list. Enter the `access_list_name` in upper case letters so that the name is easy to see in the configuration. You might want to name the access list for the interface (for example, INSIDE) or for the purpose (for example, MPLS or PIX).
  - The `any` keyword specifies access to anyone.
  - The `bpdu` keyword specifies access to bridge protocol data units, which are permitted by default.
  - The `deny` keyword denies access if the conditions are matched. If an EtherType access list is configured to deny all, all ethernet frames are discarded. Only physical protocol traffic, such as auto-negotiation, is still allowed.
  - The `hex_number` argument indicates any Ethertype that can be identified by a 16-bit hexadecimal number greater than or equal to 0x600. (See RFC 1700, “Assigned Numbers,” at http://www.ietf.org/rfc/rfc1700.txt for a list of EtherTypes.)
  - The `ipx` keyword specifies access to IPX.
  - The `mpls-multicast` keyword specifies access to MPLS multicast.
  - The `mpls-unicast` keyword specifies access to MPLS unicast.
  - The `permit` keyword permits access if the conditions are matched.

**Note**

To remove an EtherType ACE, enter the `no access-list` command with the entire command syntax string as it appears in the configuration.

### Example

The following sample access list allows common Ethertypes originating on the inside interface:
```
hostname(config)# access-list ETHER ethertype permit ipx
hostname(config)# access-list ETHER ethertype permit mpls-unicast
hostname(config)# access-group ETHER in interface inside
```
Adding Remarks to Access Lists

You can include remarks about entries in any access list, including extended, EtherType, IPv6, standard, and Webtype access lists. The remarks make an access list easier to understand.

To add a remark after the last `access-list` command you entered, enter the following command:

```
access-list access_list_name remark text
```

**Example:**

```
hostname(config)# access-list OUT remark - this is the inside admin address
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>access-list access_list_name remark text</code></td>
<td>Adds a remark after the last access-list command you entered. The text can be up to 100 characters in length. You can enter leading spaces at the beginning of the text. Trailing spaces are ignored. If you enter the remark before any <code>access-list</code> command, then the remark is the first line in the access list. If you delete an access list using the <code>no access-list access_list_name</code> command, then all remarks are also removed.</td>
</tr>
</tbody>
</table>

**Example**

You can add remarks before each ACE, and the remarks appear in the access list in these locations. Entering a dash (-) at the beginning of a remark helps to set it apart from the ACE.

```
hostname(config)# access-list OUT remark - this is the inside admin address
hostname(config)# access-list OUT extended permit ip host 209.168.200.3 any
hostname(config)# access-list OUT remark - this is the hr admin address
hostname(config)# access-list OUT extended permit ip host 209.168.200.4 any
```

What to Do Next

Apply the access list to an interface. (See the “Applying an Access List to an Interface” section on page 35-4 for more information.)

Monitoring EtherType Access Lists

To monitor EtherType access lists, enter one of the following commands:

```
show access-list
show running-config access-list
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show access-list</code></td>
<td>Displays the access list entries by number.</td>
</tr>
<tr>
<td><code>show running-config access-list</code></td>
<td>Displays the current running access-list configuration.</td>
</tr>
</tbody>
</table>
Configuration Examples for EtherType Access Lists

The following example shows how to configure EtherType access lists:

The following access list allows some EtherTypes through the ASA, but it denies IPX:

```
hostname(config)# access-list ETHER ethertype deny ipx
hostname(config)# access-list ETHER ethertype permit 0x1234
hostname(config)# access-list ETHER ethertype permit mpls-unicast
hostname(config)# access-group ETHER in interface inside
hostname(config)# access-group ETHER in interface outside
```

The following access list denies traffic with EtherType 0x1256, but it allows all others on both interfaces:

```
hostname(config)# access-list nonIP ethertype deny 1256
hostname(config)# access-list nonIP ethertype permit any
hostname(config)# access-group ETHER in interface inside
hostname(config)# access-group ETHER in interface outside
```

Feature History for EtherType Access Lists

Table 12-2 lists the release history for this feature.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>EtherType access lists</td>
<td>7.0</td>
<td>EtherType access lists control traffic based upon its EtherType.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The feature and the following command were introduced: access-list ethertype.</td>
</tr>
</tbody>
</table>