Configuring Access Control Lists

This chapter describes the access control list (ACL) features built into the ML-Series card.

This chapter contains the following major sections:

Understanding ACLs

ACLs provide network control and security, allowing you to filter packet flow into or out of ML-Series interfaces. ACLs, which are sometimes called filters, allow you to restrict network use by certain users or devices. ACLs are created for each protocol and are applied on the interface for either inbound or outbound traffic. ACLs do not apply to outbound control plane traffic. Only one ACL filter can be applied per direction per subinterface.

When creating ACLs, you define criteria to apply to each packet processed by the ML-Series card; the ML-Series card decides whether to forward or block the packet based on whether or not the packet matches the criteria in your list. Packets that do not match any criteria in your list are automatically blocked by the implicit “deny all traffic” criteria statement at the end of every ACL.

ML-Series ACL Support

Both control-plane and data-plane ACLs are supported on the ML-Series card:

- Control-plane ACLs: ACLs used to filter control data that is processed by the CPU of the ML-Series card (for example, distribution of routing information, Internet Group Membership Protocol (IGMP) joins, and so on).
- Data-plane ACLs: ACLs used to filter user data being routed or bridged through the ML Series in hardware (for example, denying access to a host, and so on). These ACLs are applied to an interface in the input or output direction using the **ip access-group** command.

The following apply when using data-plane ACLs on the ML-Series card:

- ACLs are supported on all interface types, including bridged interfaces.
- Reflexive and dynamic ACLs are not supported on the ML-Series card.
- Access violations accounting is not supported on the ML-Series card.

- ACL logging is supported only for packets going to the CPU, not for switched packets.
- IP standard ACLs applied to bridged egress interfaces are not supported in the data-plane. When bridging, ACLs are only supported on ingress.

IP ACLs

The following ACL styles for IP are supported:

- Standard IP ACLs: These use source addresses for matching operations.
- Extended IP ACLs (control plane only): These use source and destination addresses for matching operations and optional protocol type and port numbers for finer granularity of control.
- Named ACLs: These use source addresses for matching operations.

Note By default, the end of the ACL contains an implicit deny statement for everything if it did not find a match before reaching the end. With standard ACLs, if you omit the mask from an associated IP host address ACL specification, 0.0.0.0 is assumed to be the mask.

After creating an ACL, you must apply it to an interface, as shown in the "Applying the ACL to an Interface" section on page 15-4.

Named IP ACLs

You can identify IP ACLs with a name, but it must be an alphanumeric string. Named IP ACLs allow you to configure more IP ACLs in a router than if you used numbered ACLs. If you identify your ACL with an alphabetic rather than a numeric string, the mode and command syntax are slightly different.

Consider the following before configuring named ACLs:

- A standard ACL and an extended ACL cannot have the same name.
- Numbered ACLs are also available, as described in the "Creating Numbered Standard and Extended IP ACLs" section on page 15-3.

User Guidelines

Keep the following in mind when you configure IP network access control:

- You can program ACL entries into Ternary Content Addressable Memory (TCAM).
- You do not have to enter a deny everything statement at the end of your ACL; it is implicit.
- You can enter ACL entries in any order without any performance impact.
- For every eight TCAM entries, the ML-Series card uses one entry for TCAM management purposes.
- Do not set up conditions that result in packets getting lost. This situation can happen when a device or interface is configured to advertise services on a network that has ACLs that deny these packets.
- IP ACLs are not supported for double-tagged (QinQ) packets. They will however by applied to IP packets entering on a QinQ access port.

Creating IP ACLs

The following sections describe how to create numbered standard, extended, and named standard IP ACLs:

Creating Numbered Standard and Extended IP ACLs

Table 15-1 lists the global configuration commands used to create numbered standard and extended IP ACLs.

Table 15-1 Commands for Numbered Standard and Extended IP ACLs

Command	Purpose
Router(config)# **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]	Defines a standard IP ACL using a source address and wildcard.
Router(config)# **access-list** access-list-number {**deny** | **permit**} **any**	Defines a standard IP ACL using an abbreviation for the source and source mask of 0.0.0.0 255.255.255.255.
Router(config)# **access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*]	Defines an extended IP ACL number and the access conditions.
Router(config)# **access-list** *access-list-number* {**deny** | **permit**} *protocol* **any any**	Defines an extended IP ACL using an abbreviation for a source and source wildcard of 0.0.0.0 255.255.255.255, and an abbreviation for a destination and destination wildcard of 0.0.0.0 255.255.255.255.
Router(config)# **access-list** *access-list-number* {**deny** | **permit**} *protocol* **host** *source* **host** *destination*	Defines an extended IP ACL using an abbreviation for a source and source wildcard of source 0.0.0.0, and an abbreviation for a destination and destination wildcard of destination 0.0.0.0.

Creating Named Standard IP ACLs

To create a named standard IP ACL, perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip access-list standard` *name*	Defines a standard IP ACL using an alphabetic name.
Step 2	`Router(config-std-nac1)# deny` {*source* [*source-wildcard*] | **any**} or **permit** {*source* [*source-wildcard*] | **any**}	In access-list configuration mode, specifies one or more conditions as permitted or denied. This determines whether the packet is passed or dropped.
Step 3	`Router(config)# exit`	Exits access-list configuration mode.

Creating Named Extended IP ACLs (Control Plane Only)

To create a named extended IP ACL, perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip access-list extended` *name*	Defines an extended IP ACL using an alphabetic name.
Step 2	`Router(config-ext-nacl)#` {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] or {**deny** | **permit**} *protocol* **any any** or {**deny** | **permit**} *protocol* **host** *source* **host** *destination*	In access-list configuration mode, specifies the conditions allowed or denied. Or: Defines an extended IP ACL using an abbreviation for a source and source wildcard of 0.0.0.0 255.255.255.255, and an abbreviation for a destination and destination wildcard of 0.0.0.0 255.255.255.255. Or: Defines an extended IP ACL using an abbreviation for a source and source wildcard of *source* 0.0.0.0, and an abbreviation for a destination and destination wildcard of *destination* 0.0.0.0.

Applying the ACL to an Interface

After you create an ACL, you can apply it to one or more interfaces. ACLs can be applied on either the inbound or the outbound direction of an interface. When controlling access to an interface, you can use a name or number. If a standard ACL is applied, the ML-Series card compares the source IP address with the ACL. To apply an ACL to one or more interfaces, use the command in Table 15-2.

Note IP standard ACLs applied to the ingress of a Bridge Group Virtual Interface (BVI) will be applied to all bridged IP traffic in the associated bridge-group, in addition to the BVI ingress traffic.

Table 15-2 Applying ACL to Interface

Command	Purpose
ip access-group {*access-list-number* | *name*} {**in** | **out**}	Controls access to an interface.

Modifying ACL TCAM Size

You can change the TCAM size by entering the **sdm access-list** command. For more information on ACL TCAM sizes, see the "Configuring Access Control List Size in TCAM" section on page 14-3. Example 15-1 provides an example of modifying and verifying ACLs.

Note To increase the ACL TCAM size, you must decrease another region's TCAM size, such as IP, IP multicast, or L2 switching.

Caution You will need to increase the TCAM size if you see the following error message:

```
Warning:Programming TCAM entries failed
Please remove last ACL command to re-activate ACL operation.
!<ACL number or name> <IP or IPX> <INPUT_ACL or OUTPUT_ACL> from TCAM group for !<interface>
Please see the documentation to see if TCAM space can be
increased on this platform to alleviate the problem.
```

Example 15-1 Monitor and Verify ACLs

```
Router# show ip access-lists 1
Standard IP access list 1
    permit 192.168.1.1
    permit 192.168.1.2
```