CHAPTER 29

Configuring IP Access Control Lists

Cisco MDS 9000 Family switches can route IP traffic between Ethernet and Fibre Channel interfaces. The IP static routing feature is used to route traffic between VSANs. To do so, each VSAN must be in a different IP subnetwork. Each Cisco MDS 9000 Family switch provides the following services for network management systems (NMS):

- IP forwarding on the out-of-band Ethernet interface (mgmt0) on the front panel of the supervisor modules.
- IP forwarding or in-band Fibre Channel interface using the IP over Fibre Channel (IPFC) function—IPFC specifies how IP frames can be transported over Fibre Channel using encapsulation techniques. IP frames are encapsulated into Fibre Channel frames so NMS information can cross the Fibre Channel network without using an overlay Ethernet network.
- IP routing (default routing and static routing)—If your configuration does not need an external router, you can configure a default route using static routing.

Switches are compliant with RFC 2338 standards for Virtual Router Redundancy Protocol (VRRP) features. VRRP is a restartable application that provides a redundant, alternate path to the gateway switch.

This chapter includes the following sections:

IP Access Control Lists

IP Access Control Lists (IP-ACLs) provide basic network security to all switches in the Cisco MDS 9000 Family. IP-ACLs restrict IP-related traffic based on the configured IP filters. A filter contains the rules to match an IP packet, and if the packet matches, the rule also stipulates if the packet should be permitted or denied.

Each switch in the Cisco MDS 9000 Family can have a maximum total of 128 IP-ACLs, each IP-ACL can have a maximum of 256 filters.

IP-ACL Configuration Guidelines

Follow these guidelines when configuring IP-ACLs in any switch or director in the Cisco MDS 9000 Family:

- You could apply IP-ACLs to VSAN interfaces, the management interface, Gigabit Ethernet interfaces on IPS modules, and Ethernet PortChannel interfaces.

Tip If IP-ACLs are already configured in a Gigabit Ethernet interface, you cannot add this interface to a Ethernet PortChannel group. Refer to the "Gigabit Ethernet IP-ACL Guidelines" section on page 37-10 for guidelines on configuring IP ACLs.

Caution Do not apply IP-ACLs to only one member of a PortChannel group. Apply IP-ACLs to the entire channel group.

- Configure the order of conditions accurately. As the IP-ACL filters are sequentially applied to the IP flows, only the first match determines the action taken. Subsequent matches are not considered. Be sure to configure the most important condition first. If no conditions match, the software drops the packet.

Filter Contents

An IP filter contains rules for matching an IP packet based on the protocol, address, port, ICMP type, and type of service (TOS).

Protocol Information

The protocol information is required in each filter. It identifies the name or number of an IP protocol. You can specify the IP protocol in one of two ways:

- Specify an integer ranging from 0 to 255. This number represents the IP protocol.
- Specify the name of a protocol including, but not restricted to, Internet Protocol (IP), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Internet Control Message Protocol (ICMP).

Note When configuring IP-ACLs on Gigabit Ethernet interfaces, only use the TCP or ICMP options.

Address Information

The address information is required in each filter. It identifies the following details:

- Source: the address of the network or host from which the packet is being sent.
- Source-wildcard: the wildcard bits applied to the source.
- Destination: the number of the network or host to which the packet is being sent.
- Destination-wildcard: the wildcard bits applied to the destination.

Specify the source and source-wildcard or the destination and destination-wildcard in one of two ways:

- Using the 32-bit quantity in four-part, dotted decimal format (10.1.1.2/0.0.0.0 is the same as host 10.1.1.2).
  - Each wildcard bit set to zero indicates that the corresponding bit position in the packet's ip address must exactly match the bit value in the corresponding bit position in the source.
  - Each wildcard bit set to one indicates that both a zero bit and a one bit in the corresponding position of the packet's ip address will be considered a match to this access list entry. Place ones in the bit positions you want to ignore. For example, 0.0.255.255 to require an exact match of only the first 16 bits of the source. Wildcard bits set to one do not need to be contiguous in the source-wildcard. For example, a source-wildcard of 0.255.0.64 would be valid.
- Using the **any** option as an abbreviation for a source and source-wildcard or destination and destination-wildcard (0.0.0.0/255.255.255.255)

Port Information

The port information is optional. To compare the source and destination ports, use the **eq** (equal) option, the **gt** (greater than) option, the **lt** (less than) option, or the **range** (range of ports) option. You can specify the port information in one of two ways:

- Specify the number of the port. Port numbers range from 0 to 65535. Table 29-1 displays the port numbers recognized by the Cisco SAN-OS software for associated TCP and UDP ports.
- Specify the name of a TCP or UDP port as follows:
  - TCP port names can only be used when filtering TCP.
  - UDP port names can only be used when filtering UDP.

Table 29-1 TCP and UDP Port Numbers

Protocol	Port	Number
UDP	dns	53
	tftp	69
	ntp	123
	radius accounting	1646 or 1813
	radius authentication	1645 or 1812
	snmp	161
	snmp-trap	162
	syslog	514

Table 29-1 TCP and UDP Port Numbers (continued)

Protocol	Port	Number
TCP[1]	ftp	20
	ftp-data	21
	ssh	22
	telnet	23
	smtp	25
	tasacs-ds	65
	www	80
	sftp	115
	http	143
	wbem-http	5988
	wbem-https	5989

1. If the TCP connection is already **established**, use the established option to find matches. A match occurs if the TCP datagram has the ACK, FIN, PSH, RST, or URG control bit set.

ICMP Information

IP packets can be filtered based on the following optional ICMP conditions:

- The icmp-type: ICMP message type. The type is a number from 0 to 255.
- The icmp-code: ICMP message code. The code is a number from 0 to 255.

Table 29-2 displays the value for each ICMP type.

Table 29-2 ICMP Type Value

ICMP Type[1]	Code
echo	8
echo-reply	0
destination unreachable	3
traceroute	30
time exceeded	11

1. ICMP redirect packets are always rejected.

TOS Information

IP packets can be filtered based on the following optional TOS conditions:

- The TOS level, as specified by a number from 0 to 15
- The TOS name: max-reliability, max-throughput, min-delay, min-monetary-cost, and normal

IP-ACL Creation

Traffic coming into the switch is compared to IP-ACL filters based on the order that the filters occur in the switch. New filters are added to the end of the IP-ACL. The switch keeps looking until it has a match. If no matches are found when the switch reaches the end of the filter, the traffic is denied. For this reason, you should have the frequently hit filters at the top of the filter. There is an *implied deny* for traffic that is not permitted. A single-entry IP-ACL with only one deny entry has the effect of denying all traffic.

To configure an IP-ACL, you must complete the following tasks:

1. Create an IP-ACL by specifying a filter name and one or more access condition(s). Filters require the source and destination address to match a condition. Use optional keywords to configure finer granularity.
2. Apply the access filter to specified interfaces.

To create an IP-ACL, follow these steps:

	Command	Purpose
Step 1	`switch# config t`	Enters configuration mode.
Step 2	`switch(config)# ip access-list List1 permit ip any any`	Configures an IP-ACL called List1 and permits IP traffic from any source address to any destination address.
	`switch(config)# no ip access-list List1 permit ip any any`	Removes the IP-ACL called List1.
Step 3	`switch(config)# ip access-list List1 deny tcp any any`	Updates List1 to deny TCP traffic from any source address to any destination address.

To define an IP-ACL that restricts management access, follow these steps:

	Command	Purpose
Step 1	`switch# config t`	Enters configuration mode.
Step 2	`switch(config)# ip access-list restrict_mgmt permit ip 10.67.16.0 0.0.0.255 any`	Defines an entry in IP-ACL named restrict_mgmt allowing all addresses in the 10.67.16.0/24 subnet.
Step 3	`switch(config)# ip access-list restrict_mgmt permit icmp any any eq 8`	Adds an entry to IP-ACL named restrict_mgmt to allow any device to ping the MDS (icmp type 8).
Step 4	`switch(config)# ip access-list restrict_mgmt deny ip any any`	Explicitly blocks all other access for access-list named restrict_mgmt.

To use the operand and port options, follow these steps:

	Command	Purpose
Step 1	`switch# config t`	Enters configuration mode.
Step 2	`switch(config)# ip access-list List2 deny tcp 1.2.3.0 0.0.0.255 eq port 5 any`	Denies TCP traffic from 1.2.3.0 through source port 5 to any destination.

Adding filters to an Existing IP-ACL

After you create an IP-ACL, you place subsequent additions at the end of the IP-ACL. You cannot insert filters in the middle of an IP-ACL. Each configured entry is automatically added to the end of a IP-ACL.

To add entries to an existing IP-ACL, follow these steps:

	Command	Purpose
Step 1	`switch# config t`	Enters configuration mode.

Step 2	`switch(config)# ip access-list List1 permit tcp 10.1.1.2 0.0.0.0 172.16.1.1 0.0.0.0 eq port telnet`
	`switch(config)# ip access-list List1 permit tcp 10.1.1.2 0.0.0.0 172.16.1.1 0.0.0.0 eq port http`
	`switch(config)# ip access-list List1 permit udp 10.1.1.2 0.0.0.0 172.16.1.1 0.0.0.0`
		Permits TCP for Telnet traffic
		Permits TCP for HTTP traffic.
		Permits UDP for all traffic.

Removing Entries from an Existing IP-ACL

To remove configured entries from an IP-ACL, follow these steps:

	Command	Purpose
Step 1	`switch# config t`	Enters configuration mode.
Step 2	`switch(config)# no ip access-list List2 deny tcp 1.2.3.0 0.0.0.255 eq port 5 any`	Removes this entry from the IP-ACL.
	`switch(config)# no ip access-list x3 deny ip any any`	Removes this entry from the IP-ACL.
	`switch(config)# no ip access-list x3 permit ip any any`	Removes this entry from the IP-ACL.

Reading the IP-ACL Log Dump

Use the **log-deny** option at the end of an filter condition to log information about packets that match dropped entries. The log output displays the IP-ACL number, permit or deny status, and port information.

Note To capture these messages in a logging destination, you must configure severity level 7 for the kernel and ipacl facilities (see the "Facility Severity Level" section on page 44-5) and severity level 7 for the logging destination: logfile (see the "Log Files" section on page 44-6), monitor (see the "Monitor Severity Level" section on page 44-5) or console (see the "Console Severity Level" section on page 44-4). For example:

```
switch# config t
switch(config)# logging level kernel 7
switch(config)# logging level ipacl 7
switch(config)# logging logfile message 7
```

For the input IP-ACL, the log displays the raw MAC information. The keyword "MAC=" does not refer to showing an Ethernet MAC frame with MAC address information. It refers to the Layer 2 MAC-layer information dumped to the log. For the output ACL, the raw Layer 2 information is not logged.

The following is example of an input IP-ACL log dump.

```
Jul 17 20:38:44 excal-2
%KERN-7-SYSTEM_MSG:
%IPACL-7-DENY:IN=vsan1 OUT=
MAC=10:00:00:05:30:00:47:df:10:00:00:05:30:00:8a:1f:aa:aa:03:00:00:00:08:00:45:00:00:54:00
:00:40:00:40:01:0e:86:0b:0b:0b:0c:0b:0b:0b:02:08:00:ff:9c:01:15:05:00:6f:09:17:3f:80:02:01
:00:08:09:0a:0b:0c:0d:0e:0f:10:11:12:13:14:15:16:17:18:19:1a:1b:1c:1d:1e:1f:20:21:22:23:24
:25:26:27:28:29:2a:2b SRC=11.11.11.12 DST=11.11.11.2 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=0
DF PROTO=ICMP TYPE=8 CODE=0 ID=277 SEQ=1280
```

The following example is an output ACL log dump.

```
Jul 17 20:38:44 excal-2
%KERN-7-SYSTEM_MSG:
%IPACL-7-DENY:IN= OUT=vsan1 SRC=11.11.11.2 DST=11.11.11.12 LEN=84 TOS=0x00 PREC=0x00
TTL=255 ID=38095 PROTO=ICMP TYPE=0 CODE=0 ID=277 SEQ=1280
```

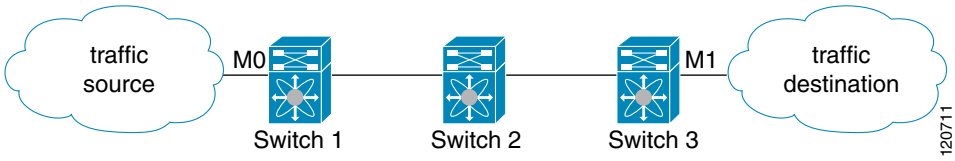
Applying an IP-ACL to an Interface

You can define IP-ACLs without applying them. However, the IP-ACLs will have no effect until they are applied to an interface on the switch.

Tip Apply the IP-ACL on the interface closest to the source of the traffic.

When you are trying to block traffic from source to destination, you can apply an inbound IP-ACL to M0 on Switch 1 instead of an outbound filter to M1 on Switch 3 (see Figure 29-1).

Figure 29-1 Denying Traffic on the Inbound Interface



The **access-group** option controls access to an interface. Each interface can only be associated with one IP-ACL per direction. The ingress direction can have a different IP-ACL than the egress direction. The IP-ACL becomes active on when applied to the interface.

Tip Create all conditions in an IP-ACL before applying it to the interface.

Caution If you apply an IP-ACL to an interface before creating it, all packets in that interface are dropped because the IP-ACL is empty.

The terms *in, out, source*, and *destination* are used as referenced by the switch.

- In—Traffic that arrives at the interface and which will go through the switch; the source is where it transmitted from and the destination is where it is transmitted to (on the other side of the router).

Tip The IP-ACL applied to the interface for the ingress traffic affects both local and remote traffic.

- Out—Traffic that has already been through the switch and is leaving the interface; the source is where it transmitted from and the destination is where it is transmitted to.

Tip The IP-ACL applied to the interface for the egress traffic only affects local traffic.

To apply an IP-ACL to an interface, follow these steps:

	Command	Purpose
Step 1	`switch# config t`	Enters configuration mode.
Step 2	`switch(config)# interface mgmt0` `switch(config-if)#`	Configures a management interface (mgmt0).
Step 3	`switch(config-if)# ip access-group restrict_mgmt`	Applies an IP-ACL called restrict_mgmt for both the ingress and egress traffic (default).
	`switch(config-if)# no ip access-group NotRequired`	Removes the IP-ACL called NotRequired.
Step 4	`switch(config-if)# ip access-group restrict_mgmt in`	Applies an IP-ACL called restrict_mgmt (if it does not already exist) for ingress traffic.
	`switch(config-if)# no ip access-group restrict_mgmt in`	Removes the IP-ACL called restrict_mgmt for ingress traffic.
	`switch(config-if)# ip access-group SampleName2 out`	Applies an IP-ACL called SampleName (if it does not already exist) for local egress traffic.
	`switch(config-if)# no ip access-group SampleName2 out`	Remove the IP-ACL called SampleName for local egress traffic.

IP-ACL Configuration Verification

Use the **show ip access-list** command to view the contents of configured access filters. Each access filter can have several conditions.

Example 29-1 Displays Configured IP-ACLs

```
switch# show ip access-list usage
Access List Name/Number          Filters IF   Status    Creation Time
-------------------------------- ------- ---- --------- -------------
abc                              3       7    active    Tue Jun 24 17:51:40 2003
x1                               3       1    active    Tue Jun 24 18:32:25 2003
x3                               0       1    not-ready Tue Jun 24 18:32:28 2003
```

Example 29-2 Displays a Summary of the Specified IP-ACL

```
switch# show ip access-list abc
ip access-list abc permit tcp any any (0 matches)
ip access-list abc permit udp any any (0 matches)
ip access-list abc permit icmp any any (0 matches)
ip access-list abc permit ip 10.1.1.0 0.0.0.255 (2 matches)
ip access-list abc permit ip 10.3.70.0 0.0.0.255 (7 matches)
```

IP-ACL Counter Cleanup

Use the **clear** command to clear the counters for a specified IP-ACL entry.

Note You cannot use this command to clear the counters for each individual filter.

```
switch# clear ip access-list counters abc
```