



Configuring Network Security with ACLs

This chapter describes how to configure network security on your switch by using access control lists (ACLs), which are also referred to in commands and tables as access lists. To take advantage of some of the features described in this chapter, you must have the enhanced multilayer software image installed on your switch.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 3550 Multilayer Switch Command Reference* for this release and the “Configuring IP Services” section of the *Cisco IOS IP and IP Routing Configuration Guide* and the *Cisco IOS IP and IP Routing Command Reference for IOS Release 12.1*.

This chapter consists of these sections:

- [Understanding ACLs, page 19-1](#)
- [Configuring Router ACLs, page 19-5](#)
- [Configuring VLAN Maps, page 19-27](#)
- [Using VLAN Maps with Router ACLs, page 19-36](#)



Note

To allocate system resources to maximize the number of security access control entries (ACEs) allowed on the switch, you can use the **sdm prefer access** global configuration command to set the Switch Database Management feature to the access template. For more information on the SDM templates, see the “[Optimizing System Resources for User-Selected Features](#)” section on page 6-57.

Understanding ACLs

Packet filtering can help limit network traffic and restrict network use by certain users or devices. ACLs can filter traffic as it passes through a router and permit or deny packets from crossing specified interfaces. An ACL is a sequential collection of permit and deny conditions that apply to packets. When a packet is received on an interface, the switch compares the fields in the packet against any applied ACLs to verify that the packet has the required permissions to be forwarded, based on the criteria specified in the access lists. It tests packets against the conditions in an access list one by one. The first match determines whether the switch accepts or rejects the packets. Because the switch stops testing conditions after the first match, the order of conditions in the list is critical. If no conditions match, the switch rejects the packets. If there are no restrictions, the switch forwards the packet; otherwise, the switch drops the packet.

Switches traditionally operate at Layer 2 only, switching traffic within a VLAN, whereas routers route traffic between VLANs. The Catalyst 3550 switch with the enhanced multilayer software image installed can accelerate packet routing between VLANs by using Layer 3 switching. The switch bridges the packet, the packet is then routed internally without going to an external router, and then the packet is bridged again to send it to its destination. During this process, the switch can access-control all packets it switches, including packets bridged within a VLAN.

You configure access lists on a router or Layer 3 switch to provide basic security for your network. If you do not configure ACLs, all packets passing through the switch could be allowed onto all parts of the network. You can use ACLs to control which hosts can access different parts of a network or to decide which types of traffic are forwarded or blocked at router interfaces. For example, you can allow e-mail traffic to be forwarded but not Telnet traffic. ACLs can be configured to block inbound traffic, outbound traffic, or both.

An ACL contains an ordered list of access control entries (ACEs). Each ACE specifies *permit* or *deny* and a set of conditions the packet must satisfy in order to match the ACE. The meaning of *permit* or *deny* depends on the context in which the ACL is used.

The switch supports two types of ACLs:

- IP ACLs filter IP traffic, including TCP, User Datagram Protocol (UDP), Internet Group Management Protocol (IGMP), and Internet Control Message Protocol (ICMP).
- Ethernet ACLs filter non-IP traffic.

Supported ACLs

The switch supports two applications of ACLs to filter traffic:

- Router ACLs access-control routed traffic between VLANs. All Catalyst 3550 switches can create router ACLs, but you must have the enhanced multilayer software image on your switch to filter packets routed between VLANs.
- VLAN ACLs or VLAN maps access-control all packets (bridged and routed). You can use VLAN maps to filter traffic between devices in the same VLAN. You do not need the enhanced image to create or apply VLAN maps. VLAN maps are configured to provide access-control based on Layer 3 addresses for IP. Unsupported protocols are access-controlled through MAC addresses using Ethernet ACEs.

After a VLAN map is applied to a VLAN, all packets (routed or bridged) entering the VLAN are checked against the VLAN map. Packets can either enter the VLAN through a switch port or through a routed port after being routed.

This switch also supports Quality of Service (QoS) classification ACLs. For more information, see the [“Classification Based on QoS ACLs” section on page 20-7](#).

Router ACLs

You can apply router ACLs on switch virtual interfaces (SVIs), which are Layer 3 interfaces to VLANs; on physical Layer 3 interfaces; and on Layer 3 EtherChannel interfaces. Router ACLs are applied on interfaces for specific directions (inbound or outbound).

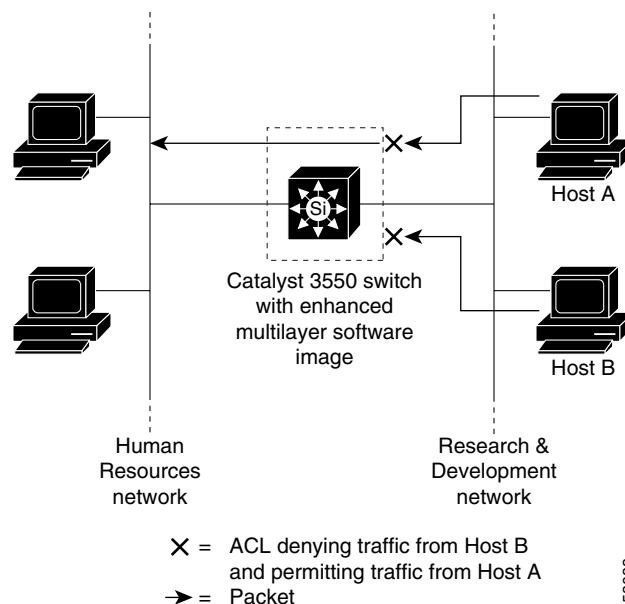
One ACL can be used with multiple features for a given interface, and one feature can use multiple ACLs. When a single router ACL is used by multiple features, it is examined multiple times.

- Standard IP access lists use source addresses for matching operations.
- Extended IP access lists use source and destination addresses and optional protocol type information for matching operations.

The switch examines ACLs associated with features configured on a given interface and a direction. As packets enter the switch on an interface, ACLs associated with all inbound features configured on that interface are examined. After packets are routed and before they are forwarded to the next hop, all ACLs associated with outbound features configured on the egress interface are examined.

ACLs permit or deny packet forwarding based on how the packet matches the entries in the ACL. For example, you can use access lists to allow one host to access a part of a network, but prevent another host from accessing the same part. In [Figure 19-1](#), ACLs applied at the router input allow Host A to access the Human Resources network, but prevent Host B from accessing the same network.

Figure 19-1 Using ACLs to Control Traffic to a Network



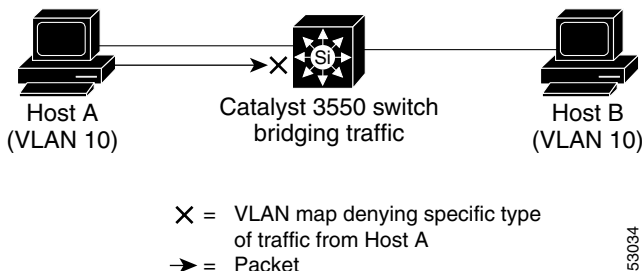
VLAN Maps

VLAN maps can access-control *all* traffic. You can apply VLAN maps on the switch to all packets that are routed into or out of a VLAN or are bridged within a VLAN. VLAN maps are used strictly for security packet filtering. Unlike router ACLs, VLAN maps are not defined by direction (input or output).

You can configure VLAN maps to match Layer 3 addresses for IP traffic. All non-IP protocols are access-controlled through MAC addresses and Ethertype using MAC VLAN maps. (IP traffic *is not* access controlled by MAC VLAN maps.) You can enforce VLAN maps only on packets going through the switch; you cannot enforce VLAN maps on traffic between hosts on a hub or on another switch connected to this switch.

With VLAN maps, forwarding of packets is permitted or denied, based on the action specified in the map. Figure 19-2 illustrates how a VLAN map is applied to deny a specific type of traffic from Host A in VLAN 10 from being forwarded.

Figure 19-2 Using VLAN Maps to Control Traffic



Handling Fragmented and Unfragmented Traffic

IP packets can be fragmented as they cross the network. When this happens, only the fragment containing the beginning of the packet contains the Layer 4 information, such as TCP or UDP port numbers, ICMP type and code, and so on. All other fragments are missing this information.

Some ACEs do not check Layer 4 information and therefore can be applied to all packet fragments. ACEs that do test Layer 4 information cannot be applied in the standard manner to most of the fragments in a fragmented IP packet. When the fragment contains no Layer 4 information and the ACE tests some Layer 4 information, the matching rules are modified:

- Permit ACEs that check the Layer 3 information in the fragment (including protocol type, such as TCP, UDP, and so on) are considered to match the fragment regardless of what the missing Layer 4 information might have been.
- Deny ACEs that check Layer 4 information never match a fragment unless the fragment contains Layer 4 information.

Consider access list 102, configured with these commands, applied to three fragmented packets:

```
Switch (config)# access-list 102 permit tcp any host 10.1.1.1 eq smtp
Switch (config)# access-list 102 deny tcp any host 10.1.1.2 eq telnet
Switch (config)# access-list 102 permit tcp any host 10.1.1.2
Switch (config)# access-list 102 deny tcp any any
```



Note

In the first and second ACEs in the examples, the *eq* keyword after the destination address means to test for the TCP-destination-port well-known numbers equaling Simple Mail Transfer Protocol (SMTP) and Telnet, respectively.

- Packet A is a TCP packet from host 10.2.2.2., port 65000, going to host 10.1.1.1 on the SMTP port. If this packet is fragmented, the first fragment matches the first ACE (a permit) as if it were a complete packet because all Layer 4 information is present. The remaining fragments also match the first ACE, even though they do not contain the SMTP port information, because the first ACE only checks Layer 3 information when applied to fragments. The information in this example is that the packet is TCP and that the destination is 10.1.1.1.

- Packet B is from host 10.2.2.2, port 65001, going to host 10.1.1.2 on the Telnet port. If this packet is fragmented, the first fragment matches the second ACE (a deny) because all Layer 3 and Layer 4 information is present. The remaining fragments in the packet do not match the second ACE because they are missing Layer 4 information. Instead, they match the third ACE (a permit).

Because the first fragment was denied, host 10.1.1.2 cannot reassemble a complete packet, so packet B is effectively denied. However, the later fragments that are permitted will consume bandwidth on the network and resources of host 10.1.1.2 as it tries to reassemble the packet.

- Fragmented packet C is from host 10.2.2.2, port 65001, going to host 10.1.1.3, port ftp. If this packet is fragmented, the first fragment matches the fourth ACE (a deny). All other fragments also match the fourth ACE because that ACE does not check any Layer 4 information and because Layer 3 information in all fragments shows that they are being sent to host 10.1.1.3, and the earlier permit ACEs were checking different hosts.

Configuring Router ACLs

Configuring router ACLs on Layer 3 switch-routed VLAN interfaces is the same as configuring ACLs on other Cisco routers. The process is briefly described here. For more detailed information on configuring router ACLs, refer to the “Configuring IP Services” chapter in the *Cisco IP and IP Routing Configuration Guide for IOS Release 12.1*. For detailed information about the commands, refer to *Cisco IOS IP and IP Routing Command Reference for IOS Release 12.1*. For a list of IOS features not supported on the Catalyst 3550 switch, see the “Unsupported Features” section on page 19-6.



Caution

By default, the router sends Internet Control Message Protocol (ICMP) unreachable messages when a packet is denied by an access group; these access-group denied packets are not dropped in hardware but are bridged to the switch CPU so that it can generate the ICMP-unreachable message. To drop access-group denied packets in hardware, you must disable ICMP unreachables by using the **no ip unreachables** interface configuration command. Note that the **ip unreachables** command is enabled by default.

This section includes the following information:

- [Hardware and Software Handling of Router ACLs, page 19-5](#)
- [Unsupported Features, page 19-6](#)
- [Creating Standard and Extended IP ACLs, page 19-6](#)
- [Applying the ACL to an Interface or Terminal Line, page 19-18](#)
- [Displaying ACLs and Access Groups, page 19-20](#)
- [ACL Configuration Examples, page 19-22](#)

Hardware and Software Handling of Router ACLs

ACL processing is primarily accomplished in hardware, but requires forwarding of some traffic flows to the CPU for software processing. The forwarding rate for software-forwarded traffic is substantially less than for hardware-forwarded traffic. When traffic flows are both logged and forwarded, forwarding is done by hardware, but logging must be done by software. Because of the difference in packet handling capacity between hardware and software, if the sum of all flows being logged (both permitted flows and denied flows) is of great enough bandwidth, not all of the packets that are forwarded can be logged.

These factors can cause packets to be sent to the CPU:

- Using the **log** keyword
- Enabling ICMP unreachable
- Hardware reaching its capacity to store ACL configurations

If ACLs cause large numbers of packets to be sent to the CPU, the switch performance can be negatively affected.

When you enter the **show ip access-lists** privileged EXEC command, the match count displayed does not account for packets that are access controlled in hardware. Use the **show access-lists hardware counters** privileged EXEC command to obtain some basic hardware ACL statistics for switched and routed packets.

Router ACLs function as follows:

- The hardware controls permit and deny actions of standard and extended ACLs (input and output) for security access control.
- If **log** has not been specified, the flows that match a *deny* statement in a security ACL are dropped by the hardware if *ip unreachable* is disabled. The flows matching a *permit* statement are switched in hardware.
- Adding the **log** keyword to an ACE in a router ACL causes a copy of the packet to be sent to the CPU only for logging. If the ACE is a *permit* statement, the packet is still switched and routed in hardware.

Unsupported Features

The Catalyst 3550 switch does not support these IOS router ACL-related features:

- Non-IP protocol ACLs (see [Table 19-1 on page 19-7](#)).
- Bridge-group ACLs.
- IP accounting.
- Inbound and outbound rate limiting (except with QoS ACLs).
- IP packets with a header length of less than five are not access controlled (results in an ICMP parameter error).
- Reflexive ACLs.
- Dynamic ACLs (except for certain specialized dynamic ACLs used by the switch clustering feature).

Creating Standard and Extended IP ACLs

This section summarizes how to create router IP ACLs. An ACL is a sequential collection of permit and deny conditions. The switch tests packets against the conditions in an access list one by one. The first match determines whether the switch accepts or rejects the packet. Because the switch stops testing conditions after the first match, the order of the conditions is critical. If no conditions match, the switch denies the packet.

These are the steps to use ACLs:

-
- Step 1** Create an ACL by specifying an access list number or name and access conditions.
- Step 2** Apply the ACL to interfaces or terminal lines. You can also apply standard and extended IP ACLs to VLAN maps.
-

The software supports these styles of ACLs or access lists for IP:

- Standard IP access lists use source addresses for matching operations.
- Extended IP access lists use source and destination addresses for matching operations and optional protocol-type information for finer granularity of control.

These sections describe access lists and the steps for using them:

- [Access List Numbers, page 19-7](#)
- [Creating a Numbered Standard ACL, page 19-8](#)
- [Creating a Numbered Extended ACL, page 19-9](#)
- [Creating Named Standard and Extended ACLs, page 19-14](#)
- [Applying Time Ranges to ACLs, page 19-15](#)
- [Including Comments About Entries in ACLs, page 19-18](#)

Access List Numbers

The number you use to denote your ACL shows the type of access list that you are creating. [Table 19-1](#) lists the access-list number and corresponding access list type and shows whether or not they are supported in the switch. The Catalyst 3550 switch supports IP standard and IP extended access lists, numbers 1 to 199 and 1300 to 2699.

Table 19-1 Access List Numbers

Access List Number	Type	Supported
1–99	IP standard access list	Yes
100–199	IP extended access list	Yes
200–299	Protocol type-code access list	No
300–399	DECnet access list	No
400–499	XNS standard access list	No
500–599	XNS extended access list	No
600–699	AppleTalk access list	No
700–799	48-bit MAC address access list	No
800–899	IPX standard access list	No
900–999	IPX extended access list	No
1000–1099	IPX SAP access list	No
1100–1199	Extended 48-bit MAC address access list	No
1200–1299	IPX summary address access list	No

Table 19-1 Access List Numbers (continued)

Access List Number	Type	Supported
1300–1999	IP standard access list (expanded range)	Yes
2000–2699	IP extended access list (expanded range)	Yes

**Note**

In addition to numbered standard and extended ACLs, you can also create standard and extended named IP ACLs using the supported numbers. That is, the name of a standard IP ACL can be 1 to 99; the name of an extended IP ACL can be 100 to 199. The advantage of using named ACLs instead of numbered lists is that you can delete individual entries from a named list.

Creating a Numbered Standard ACL

Beginning in privileged EXEC mode, follow these steps to create a numbered standard ACL:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	access-list <i>access-list-number</i> { deny permit } <i>source</i> [<i>source-wildcard</i>] [log]	<p>Define a standard IP access list by using a source address and wildcard.</p> <p>The <i>access-list-number</i> is a decimal number from 1 to 99 or 1300 to 1999.</p> <p>Enter deny or permit to specify whether to deny or permit access if conditions are matched.</p> <p>The <i>source</i> is the source address of the network or host from which the packet is being sent specified as:</p> <ul style="list-style-type: none"> The 32-bit quantity in dotted-decimal format. The keyword any as an abbreviation for <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.255. You do not need to enter a source-wildcard. The keyword host as an abbreviation for <i>source</i> and <i>source-wildcard</i> of <i>source</i> 0.0.0.0. <p>(Optional) The <i>source-wildcard</i> applies wildcard bits to the source.</p> <p>(Optional) Enter log to create an informational logging message about the packet that matches the entry to be sent to the console.</p>
Step 3	end	Return to privileged EXEC mode.
Step 4	show access-lists [<i>number</i> <i>name</i>]	Show the access list configuration.
Step 5	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Use the **no access-list** *access-list-number* global configuration command to delete the entire ACL. You cannot delete individual ACEs from numbered access lists.

**Note**

When creating an ACL, remember that, by default, the end of the ACL contains an implicit deny statement for all packets that it did not find a match for before reaching the end. With standard access lists, if you omit the mask from an associated IP host address ACL specification, 0.0.0.0 is assumed to be the mask.

This example shows how to create a standard ACL to deny access to IP host 171.69.198.102, permit access to any others, and display the results.

```
Switch (config)# access-list 2 deny host 171.69.198.102
Switch (config)# access-list 2 permit any
Switch(config)# end
Switch# show access-lists
Standard IP access list 2
    deny 171.69.198.102
    permit any
```

The switch always rewrites the order of standard access lists so that entries with **host** matches and entries with matches having a *don't care* mask of 0.0.0.0 are moved to the top of the list, above any entries with non-zero *don't care* masks. Therefore, in **show** command output and in the configuration file, the ACEs do not necessarily appear in the order in which they were entered.

The switch software can provide logging messages about packets permitted or denied by a standard IP access list. That is, any packet that matches the ACL causes an informational logging message about the packet to be sent to the console. The level of messages logged to the console is controlled by the logging console commands controlling the syslog messages.

**Note**

Because routing is done in hardware and logging is done in software, if a large number of packets match a *permit* or *deny* ACE containing a **log** keyword, the software might not be able to match the hardware processing rate, and not all packets will be logged.

The first packet that triggers the ACL causes a logging message right away, and subsequent packets are collected over 5-minute intervals before they are displayed or logged. The logging message includes the access list number, whether the packet was permitted or denied, the source IP address of the packet, and the number of packets from that source permitted or denied in the prior 5-minute interval.

**Note**

An output ACL cannot log multicast packets.

After creating an ACL, you must apply it to a line or interface, as described in the [“Applying the ACL to an Interface or Terminal Line”](#) section on page 19-18.

Creating a Numbered Extended ACL

Although standard ACLs use only source addresses for matching, you can use extended ACL source and destination addresses for matching operations and optional protocol type information for finer granularity of control. Some protocols also have specific parameters and keywords that apply to that protocol.

These IP protocols are supported (protocol keywords are in parentheses in bold):

Authentication Header Protocol (**ahp**), Enhanced Interior Gateway Routing Protocol (**eigrp**), Encapsulation Security Payload (**esp**), generic routing encapsulation (**gre**), Internet Control Message Protocol (**icmp**), Internet Group Management Protocol (**igmp**), Interior Gateway Routing

Protocol (**igrp**), any Interior Protocol (**ip**), IP in IP tunneling (**ipinip**), KA9Q NOS-compatible IP over IP tunneling (**nos**), Open Shortest Path First routing (**ospf**), Payload Compression Protocol (**pcp**), Protocol Independent Multicast (**pim**), Transmission Control Protocol (**tcp**), or User Datagram Protocol (**udp**).

Supported parameters can be grouped into these categories: TCP, UDP, ICMP, IGMP, or other IP.

Table 19-2 lists the possible filtering parameters for ACEs for each protocol type.

Table 19-2 Filtering Parameter ACEs Supported by Different IP Protocols

Filtering Parameter ¹	TCP	UDP	ICMP ²	IGMP	Other IP
Layer 3 Parameters:					
IP ToS byte ³	X	X	X	X	X
Differentiated Services Code Point (DSCP)	X	X	X	X	X
IP source address	X	X	X	X	X
IP destination address	X	X	X	X	X
Fragments	X	X	X	X	X
TCP or UDP	X	X	–	–	–
ICMP	–	–	X	–	–
IGMP	–	–	–	X	–
Other IP protocol	–	–	–	–	X
Layer 4 Parameters					
Source port	X	X	–	–	–
Source port operator	X	X	–	–	–
Destination port	X	X	–	–	–
Destination port operator	X	X	–	–	–
TCP flag	X	–	–	–	–
ICMP code	–	–	X	–	–
ICMP type	–	–	X	–	–
IGMP message type	–	–	–	X	–

1. X in a protocol column means support for the filtering parameter.
2. ICMP echo-reply cannot be filtered. All other ICMP codes or types can be filtered.
3. No support for type of service (TOS) minimize monetary cost bit.

For more details on the specific keywords relative to each protocol, refer to *Cisco IP and IP Routing Command Reference for IOS Release 12.1*.



Note

The Catalyst 3550 switch does not support dynamic or reflexive access lists. It also does not support filtering based on the type of service (ToS) minimize-monetary-cost bit.

When you are creating ACEs in numbered extended access lists, remember that after you create the ACL, any additions are placed at the end of the list. You cannot reorder the list or selectively add or remove ACEs from a numbered list.

Beginning in privileged EXEC mode, follow these steps to create an extended ACL:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2a	access-list <i>access-list-number</i> { deny permit } <i>protocol</i> <i>source source-wildcard</i> <i>destination destination-wildcard</i> [precedence <i>precedence</i>] [tos <i>tos</i>] [fragments] [log] [log-input] [time-range <i>time-range-name</i>] [dscp <i>dscp</i>] Note If you enter a dscp value, you cannot enter tos or precedence . You can enter both a tos and a precedence value with no dscp .	Define an extended IP access list and the access conditions. The <i>access-list-number</i> is a decimal number from 100 to 199 or 2000 to 2699. Enter deny or permit to specify whether to deny or permit the packet if conditions are matched. For <i>protocol</i> , enter the name or number of an IP protocol: ahp , eigrp , esp , gre , icmp , igmp , igrp , ip , ipinip , nos , ospf , pcp , pim , tcp , or udp , or an integer in the range 0 to 255 representing an IP protocol number. To match any Internet protocol (including ICMP, TCP, and UDP) use the keyword ip . Note This step includes options for most IP protocols. For additional specific parameters for TCP, UDP, ICMP, and IGMP, see steps 2b through 2e. The <i>source</i> is the number of the network or host from which the packet is sent. The <i>source-wildcard</i> applies wildcard bits to the source. The <i>destination</i> is the network or host number to which the packet is sent. The <i>destination-wildcard</i> applies wildcard bits to the destination. Source, source-wildcard, destination, and destination-wildcard can be specified as: <ul style="list-style-type: none"> • The 32-bit quantity in dotted-decimal format. • The keyword any for 0.0.0.0 255.255.255.255 (any host). • The keyword host for a single host 0.0.0.0. The other keywords are optional and have these meanings: <ul style="list-style-type: none"> • precedence—Enter to match packets with a precedence level specified as a number from 0 to 7 or by name: routine (0), priority (1), immediate (2), flash (3), flash-override (4), critical (5), internet (6), network (7). • fragments—Enter to check non-initial fragments. • tos—Enter to match by type of service level, specified by a number from 0 to 15 or a name: normal (0), max-reliability (2), max-throughput (4), min-delay (8). • log—Enter to create an informational logging message to be sent to the console about the packet that matches the entry or log-input to include the input interface in the log entry. • time-range—For an explanation of this keyword, see the “Applying Time Ranges to ACLs” section on page 19-15. • dscp—Enter to match packets with the DSCP value specified by a number from 0 to 63, or use the question mark (?) to see a list of available values.

	Command	Purpose
or	access-list <i>access-list-number</i> { deny permit } <i>protocol any any</i> [precedence <i>precedence</i>] [tos <i>tos</i>] [fragments] [log] [log-input] [time-range <i>time-range-name</i>] [dscp <i>dscp</i>]	In access-list configuration mode, define an extended IP access list 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. You can use the any keyword in place of source and destination address and wildcard.
or	access-list <i>access-list-number</i> { deny permit } <i>protocol</i> host <i>source host destination</i> [precedence <i>precedence</i>] [tos <i>tos</i>] [fragments] [log] [log-input] [time-range <i>time-range-name</i>] [dscp <i>dscp</i>]	Define an extended IP access list using an abbreviation for a source and source wildcard of <i>source</i> 0.0.0.0 and an abbreviation for a destination and destination wildcard of <i>destination</i> 0.0.0.0. You can use the host keyword in place of source and destination wildcard or mask.
Step 2b	access-list <i>access-list-number</i> { deny permit } tcp <i>source</i> <i>source-wildcard</i> [<i>operator</i> [<i>port</i>]] <i>destination destination-wildcard</i> [<i>operator</i> [<i>port</i>]] [established] [precedence <i>precedence</i>] [tos <i>tos</i>] [fragments] [log] [log-input] [time-range <i>time-range-name</i>] [dscp <i>dscp</i>] [<i>flag</i>]	(Optional) Define an extended TCP access list and the access conditions. Enter tcp for Transmission Control Protocol. The parameters are the same as those described in Step 2a with these exceptions: (Optional) Enter an <i>operator</i> and <i>port</i> to compare source (if positioned after <i>source source-wildcard</i>) or destination (if positioned after <i>destination destination-wildcard</i>) port. Possible operators include eq (equal), gt (greater than), lt (less than), neq (not equal), and range (inclusive range). Operators require a port number (range requires two port numbers separated by a space). Enter the <i>port</i> number as a decimal number (from 0 to 65535) or the name of a TCP port. To see TCP port names, use the ? or refer to “Configuring IP Services” section of <i>Cisco IOS IP and IP Routing Command Reference for IOS Release 12.1</i> . Use only TCP port numbers or names when filtering TCP. The additional optional keywords have these meanings: <ul style="list-style-type: none"> • established—Enter to match an established connection. This has the same function as matching on the ack or rst flag. • <i>flag</i>—Enter one of these flags to match by the specified TCP header bits: ack (acknowledge), fin (finish), psh (push), rst (reset), syn (synchronize), or urg (urgent).
Step 2c	access-list <i>access-list-number</i> { deny permit } udp <i>source source-wildcard</i> [<i>operator</i> [<i>port</i>]] <i>destination destination-wildcard</i> [<i>operator</i> [<i>port</i>]] [precedence <i>precedence</i>] [tos <i>tos</i>] [fragments] [log] [log-input] [time-range <i>time-range-name</i>] [dscp <i>dscp</i>]	(Optional) Define an extended UDP access list and the access conditions. Enter udp for the User Datagram Protocol. The UDP parameters are the same as those described for TCP except that [<i>operator</i> [<i>port</i>]] port number or name must be a UDP port number or name, and the flag and established parameters are not valid for UDP.

	Command	Purpose
Step 2d	access-list <i>access-list-number</i> { deny permit } icmp <i>source</i> <i>source-wildcard</i> <i>destination</i> <i>destination-wildcard</i> [<i>icmp-type</i> [[<i>icmp-type icmp-code</i>] <i>icmp-message</i>]] [precedence <i>precedence</i>] [tos <i>tos</i>] [fragments] [log] [log-input] [time-range <i>time-range-name</i>] [dscp <i>dscp</i>]	(Optional) Define an extended ICMP access list and the access conditions. Enter icmp for Internet Control Message Protocol. The ICMP parameters are the same as those described for most IP protocols in Step 2a, with the addition of the ICMP message type and code parameters. These optional keywords have these meanings: <ul style="list-style-type: none"> <i>icmp-type</i>—Enter to filter by ICMP message type, a number from 0 to 255. <i>icmp-code</i>—Enter to filter ICMP packets that are filtered by ICMP message type by the ICMP message code, a number from 0 to 255. <i>icmp-message</i>—Enter to filter ICMP packets by ICMP message type name or ICMP message type and code name. To see a list of ICMP message type names and ICMP message type and code names, use the ? or refer to the “Configuring IP Services” section of <i>Cisco IOS IP and IP Routing Command Reference for IOS Release 12.1</i>.
Step 2e	access-list <i>access-list-number</i> { deny permit } igmp <i>source</i> <i>source-wildcard</i> <i>destination</i> <i>destination-wildcard</i> [<i>igmp-type</i>] [precedence <i>precedence</i>] [tos <i>tos</i>] [fragments] [log] [log-input] [time-range <i>time-range-name</i>] [dscp <i>dscp</i>]	(Optional) Define an extended IGMP access list and the access conditions. Enter igmp for Internet Group Management Protocol. The IGMP parameters are the same as those described for most IP protocols in Step 2a, with the addition of this optional parameter. <i>igmp-type</i> —To match IGMP message type, enter a number from 0 to 15, or enter the message name (dvmp , host-query , host-report , pim , or trace).
Step 3	show access-lists [<i>number</i> <i>name</i>]	Verify the access list configuration.
Step 4	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Use the **no access-list** *access-list-number* global configuration command to delete the entire access list. You cannot delete individual ACEs from numbered access lists.

This example shows how to create and display an extended access list to deny Telnet access from any host in network 171.69.198.0 to any host in network 172.20.52.0 and permit any others. (The **eq** keyword after the destination address means to test for the TCP destination port number equaling Telnet.)

```
Switch(config)# access-list 102 deny tcp 171.69.198.0 0.0.0.255 172.20.52.0 0.0.0.255 eq telnet
Switch(config)# access-list 102 permit tcp any any
Switch(config)# end
Switch# show access-lists
Extended IP access list 102
  deny tcp 171.69.198.0 0.0.0.255 172.20.52.0 0.0.0.255 eq telnet
  permit tcp any any
```

After an ACL is created, any additions (possibly entered from the terminal) are placed at the end of the list. You cannot selectively add or remove access list entries from a numbered access list.



Note

When you are creating an ACL, remember that, by default, the end of the access list contains an implicit deny statement for all packets if it did not find a match before reaching the end.

After creating an ACL, you must apply it to a line or interface, as described in the [“Applying the ACL to an Interface or Terminal Line”](#) section on page 19-18.

Creating Named Standard and Extended ACLs

You can identify IP ACLs with an alphanumeric string (a name) rather than a number. You can use named ACLs to configure more IP access lists in a router than if you were to use numbered access lists. If you identify your access list with a name rather than a number, the mode and command syntax are slightly different. However, not all commands that use IP access lists accept a named access list.



Note

The name you give to a standard or extended ACL can also be a number in the supported range of access list numbers. That is, the name of a standard IP ACL can be 1 to 99; the name of an extended IP ACL can be 100 to 199. The advantage of using named ACLs instead of numbered lists is that you can delete individual entries from a named list.

Consider these guidelines and limitations before configuring named ACLs:

- Not all commands that accept a numbered ACL accept a named ACL. ACLs for packet filters and route filters on interfaces can use a name. VLAN maps also accept a name.
- A standard ACL and an extended ACL cannot have the same name.
- Numbered ACLs are also available, as described in the [“Creating Standard and Extended IP ACLs” section on page 19-6](#).
- You can apply standard and extended ACLs (named or numbered) to VLAN maps.

Beginning in privileged EXEC mode, follow these steps to create a standard ACL using names:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	ip access-list standard <i>name</i>	Define a standard IP access list using a name, and enter access-list configuration mode. Note The name can be a number from 1 to 99.
Step 3	deny { <i>source</i> [<i>source-wildcard</i>] host <i>source</i> any } [log] or permit { <i>source</i> [<i>source-wildcard</i>] host <i>source</i> any } [log]	In access-list configuration mode, specify one or more conditions denied or permitted to determine if the packet is forwarded or dropped. <ul style="list-style-type: none"> • host <i>source</i>—A source and source wildcard of <i>source</i> 0.0.0.0. • any—A source and source wildcard of 0.0.0.0 255.255.255.255.
Step 4	end	Return to privileged EXEC mode.
Step 5	show access-lists [<i>number</i> <i>name</i>]	Show the access list configuration.
Step 6	copy running-config startup-config	(Optional) Save your entries in the configuration file.

To remove a named standard ACL, use the **no ip access-list standard** *name* global configuration command.

Beginning in privileged EXEC mode, follow these steps to create an extended ACL using names:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	ip access-list extended <i>name</i>	Define an extended IP access list using a name and enter access-list configuration mode. Note The name can be a number from 100 to 199.
Step 3	{deny permit} <i>protocol</i> {source [<i>source-wildcard</i>] host <i>source</i> any {destination [<i>destination-wildcard</i>] host <i>destination</i> any [precedence <i>precedence</i>] [tos <i>tos</i>] [established] [log] [time-range <i>time-range-name</i>]	In access-list configuration mode, specify the conditions allowed or denied. Use the log keyword to get access list logging messages, including violations. See the “ Creating a Numbered Extended ACL ” section on page 19-9 for definitions of protocols and other keywords. <ul style="list-style-type: none"> • host <i>source</i>—A source and source wildcard of <i>source</i> 0.0.0.0. • host <i>destination</i>—A destination and destination wildcard of <i>destination</i> 0.0.0.0. • any—A source and source wildcard or destination and destination wildcard of 0.0.0.0 255.255.255.255.
Step 4	end	Return to privileged EXEC mode.
Step 5	show access-lists [<i>number</i> <i>name</i>]	Show the access list configuration.
Step 6	copy running-config startup-config	(Optional) Save your entries in the configuration file.

To remove a named extended ACL, use the **no ip access-list extended** *name* global configuration command.

When you are creating standard extended ACLs, remember that, 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. For standard ACLs, if you omit the mask from an associated IP host address access list specification, 0.0.0.0 is assumed to be the mask.

After you create an ACL, any additions are placed at the end of the list. You cannot selectively add ACL entries to a specific ACL. However, you can use **no permit** and **no deny** access-list configuration mode commands to remove entries from a named ACL. This example shows how you can delete individual ACEs from the named access list *border-list*:

```
Switch(config)# ip access-list extended border-list
Switch(config-ext-nacl)# no permit ip host 10.1.1.3 any
```

Being able to selectively remove lines from a named ACL is one reason you might use named ACLs instead of numbered ACLs.

After creating an ACL, you must apply it to a line or interface, as described in the “[Applying the ACL to an Interface or Terminal Line](#)” section on page 19-18.

Applying Time Ranges to ACLs

You can implement extended ACLs based on the time of day and week by using the **time-range** global configuration command. First, define the name and times of the day and week of the time range, and then reference the time range by name in an ACL to apply restrictions to the access list. You can use the time range to define when the permit or deny statements in the ACL are in effect. The **time-range** keyword

and argument are referenced in the named and numbered extended ACL task tables in the previous sections, the “[Creating Standard and Extended IP ACLs](#)” section on page 19-6, and the “[Creating Named Standard and Extended ACLs](#)” section on page 19-14.

These are some of the many possible benefits of using time ranges:

- You have more control over permitting or denying a user access to resources, such as an application (identified by an IP address/mask pair and a port number).
- You can control logging messages. ACL entries can log traffic at certain times of the day, but not constantly. Therefore, you can simply deny access without needing to analyze many logs generated during peak hours.



Note

The time range relies on the switch system clock. For this feature to work the way you intend, you need a reliable clock source. We recommend that you use Network Time Protocol (NTP) to synchronize the switch clock. For more information, see the “[Managing the System Time and Date](#)” section on page 6-32.

Beginning in privileged EXEC mode, follow these steps to configure a time-range parameter for an ACL:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	time-range <i>time-range-name</i>	Identify the time-range by a meaningful name, and enter time-range configuration mode. The name cannot contain a space or quotation mark and must begin with a letter.
Step 3	absolute [<i>start time date</i>] [<i>end time date</i>] or periodic <i>day-of-the-week hh:mm to</i> <i>[day-of-the-week] hh:mm</i> or periodic { weekdays weekend daily } <i>hh:mm to hh:mm</i>	Specify when the function it will be applied to is in effect. Use some combination of these commands; multiple periodic statements are allowed; only one absolute statement is allowed. If more than one absolute statement is configured, only the one configured last is executed.
Step 4	end	Return to privileged EXEC mode.
Step 5	show time-range	Verify the time-range configuration.
Step 6	copy running-config startup-config	(Optional) Save your entries in the configuration file.

To remove a configured time-range limitation, use the **no time-range** *time-range-name* global configuration command.

Repeat the steps if you have multiple items that you want in effect at different times. This example shows how to configure time ranges for *workhours* and for company holidays and how to verify your configuration.

```
Switch(config)# time-range workhours
Switch(config-time-range)# periodic weekdays 8:00 to 12:00
Switch(config-time-range)# periodic weekdays 13:00 to 17:00
Switch(config-time-range)# exit
Switch(config)# time-range new_year_day_2000
Switch(config-time-range)# absolute start 00:00 1 Jan 2000 end 23:59 1 Jan 2000
Switch(config-time-range)# exit
```

```

Switch(config)# time-range thanksgiving_2000
Switch(config-time-range)# absolute start 00:00 22 Nov 2000 end 23:59 23 Nov 2000
Switch(config-time-range)# exit
Switch(config)# time-range christmas_2000
Switch(config-time-range)# absolute start 00:00 24 Dec 2000 end 23:50 25 Dec 2000
Switch(config-time-range)# end
Switch# show time-range
time-range entry: christmas_2000 (inactive)
    absolute start 00:00 24 December 2000 end 23:50 25 December 2000
time-range entry: new_year_day_2000 (inactive)
    absolute start 00:00 01 January 2000 end 23:59 01 January 2000
time-range entry: thanksgiving_2000 (inactive)
    absolute start 00:00 22 November 2000 end 23:59 23 November 2000
time-range entry: workhours (inactive)
    periodic weekdays 8:00 to 12:00
    periodic weekdays 13:00 to 17:00

```

For a time range to be applied, you must reference it by name in an extended ACL that can implement time ranges. This example shows how to create and verify extended access list 188 that denies TCP traffic from any source to any destination during the defined holiday time ranges and permits all TCP traffic during work hours.

```

Switch(config)# access-list 188 deny tcp any any time-range new_year_day_2000
Switch(config)# access-list 188 deny tcp any any time-range thanksgiving_2000
Switch(config)# access-list 188 deny tcp any any time-range christmas_2000
Switch(config)# access-list 188 permit tcp any any time-range workhours
Switch(config)# end
Switch# show access-lists
Extended IP access list 188
    deny tcp any any time-range new_year_day_2000 (inactive)
    deny tcp any any time-range thanksgiving_2000 (active)
    deny tcp any any time-range christmas_2000 (inactive)
    permit tcp any any time-range workhours (inactive)

```

This example uses named ACLs to permit and deny the same traffic.

```

Switch(config)# ip access-list extended deny_access
Switch(config-ext-nacl)# deny tcp any any time-range new_year_day_2000
Switch(config-ext-nacl)# deny tcp any any time-range thanksgiving_2000
Switch(config-ext-nacl)# deny tcp any any time-range christmas_2000
Switch(config-ext-nacl)# exit
Switch(config)# ip access-list extended may_access
Switch(config-ext-nacl)# permit tcp any any time-range workhours
Switch(config-ext-nacl)# end
Switch# show ip access-lists
Extended IP access list deny_access
    deny tcp any any time-range new_year_day_2000 (inactive)
    deny tcp any any time-range thanksgiving_2000 (inactive)
    deny tcp any any time-range christmas_2000 (inactive)
Extended IP access list may_access
    permit tcp any any time-range workhours (inactive)

```

Including Comments About Entries in ACLs

You can use the **remark** keyword to include comments (remarks) about entries in any IP standard or extended ACL. The remarks make the ACL easier for you to understand and scan. Each remark line is limited to 100 characters.

The remark can go before or after a permit or deny statement. You should be consistent about where you put the remark so that it is clear which remark describes which permit or deny statement. For example, it would be confusing to have some remarks before the associated permit or deny statements and some remarks after the associated statements.

For IP numbered standard or extended ACLs, use the **access-list** *access-list number* **remark** *remark* global configuration command to include a comment about an access list. To remove the remark, use the **no** form of this command.

In this example, the workstation belonging to Jones is allowed access, and the workstation belonging to Smith is not allowed access:

```
Switch(config)# access-list 1 remark Permit only Jones workstation through
Switch(config)# access-list 1 permit 171.69.2.88
Switch(config)# access-list 1 remark Do not allow Smith workstation through
Switch(config)# access-list 1 deny 171.69.3.13
```

For an entry in a named IP ACL, use the **remark** *access-list* configuration command. To remove the remark, use the **no** form of this command.

In this example, the Jones subnet is not allowed to use outbound Telnet:

```
Switch(config)# ip access-list extended telnetting
Switch(config-ext-nacl)# remark Do not allow Jones subnet to telnet out
Switch(config-ext-nacl)# deny tcp host 171.69.2.88 any eq telnet
```

Applying the ACL to an Interface or Terminal Line

After you create an ACL, you can apply it to one or more interfaces or terminal lines. ACLs can be applied on *either* outbound or inbound interfaces. This section describes how to accomplish this task for both terminal lines and network interfaces. Note these guidelines:

- When controlling access to a line, you must use a number. Only numbered ACLs can be applied to lines.
- When controlling access to an interface, you can use a name or number.
- Set identical restrictions on all the virtual terminal lines because a user can attempt to connect to any of them.
- If you apply an ACL to a Layer-3 interface and the enhanced multilayer software image is not installed on your switch, the ACL only filters packets that are intended for the CPU, such as SNMP, Telnet, or Web traffic.

Beginning in privileged EXEC mode, follow these steps to restrict incoming and outgoing connections between a virtual terminal line and the addresses in an ACL:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	line [console vty] <i>line-number</i>	Identify a specific line for configuration, and enter in-line configuration mode. <ul style="list-style-type: none"> console—Enter to specify the console terminal line. The console port is DCE. vty—Enter to specify a virtual terminal for remote console access. The <i>line-number</i> is the first line number in a contiguous group that you want to configure when the line type is specified. The range is from 0 to 16.
Step 3	access-class <i>access-list-number</i> {in out}	Restrict incoming and outgoing connections between a particular virtual terminal line (into a device) and the addresses in an access list.
Step 4	end	Return to privileged EXEC mode.
Step 5	show running-config	Display the access list configuration.
Step 6	copy running-config startup-config	(Optional) Save your entries in the configuration file.

To remove access restrictions on a terminal line, use the **no access-class** *access-list-number* {in | out} line configuration command.

Beginning in privileged EXEC mode, follow these steps to control access to a Layer 3 interface:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	interface <i>interface-id</i>	Identify a specific interface for configuration, and enter interface configuration mode. The interface must be a Layer 3 interface, either a routed port or an SVI VLAN ID.
Step 3	ip access-group { <i>access-list-number</i> <i>name</i> } {in out}	Control access to the specified interface.
Step 4	end	Return to privileged EXEC mode.
Step 5	show running-config	Display the access list configuration.
Step 6	copy running-config startup-config	(Optional) Save your entries in the configuration file.

To remove the specified access group, use the **no ip access-group** {*access-list-number* | *name*} {in | out} interface configuration command.

This example shows how to apply access list 2 on Gigabit Ethernet interface 0/3 to filter packets entering the interface:

```
Switch(config)# interface gigabitethernet0/3
Router(config-if)# ip access-group 2 in
```

**Note**

The **ip access-group** interface configuration command is only valid when applied to a Layer 3 interface: an SVI, a Layer 3 EtherChannel, or a routed port. The interface must have been configured with an IP address. Layer 3 access groups filter packets that are routed or are received by Layer 3 processes on the CPU. They do not affect packets bridged within a VLAN.

For inbound ACLs, after receiving a packet, the switch checks the packet against the ACL. If the ACL permits the packet, the switch continues to process the packet. If the ACL rejects the packet, the switch discards the packet.

For outbound ACLs, after receiving and routing a packet to a controlled interface, the switch checks the packet against the ACL. If the ACL permits the packet, the switch sends the packet. If the ACL rejects the packet, the switch discards the packet.

If the input interface is configured to send ICMP Unreachable messages, these messages are sent whenever a packet is discarded, regardless of whether the packet was discarded because of an ACL on the input interface or because of an ACL on the output interface. ICMP Unreachables are normally limited to no more than one every one-half second per input interface, but this can be changed by using the **ip icmp rate-limit unreachable** global configuration command.

When you apply an undefined ACL to an interface, the switch acts as if the ACL has not been applied to the interface and permits all packets. Remember this behavior if you use undefined ACLs for network security.

Displaying ACLs and Access Groups

You can display existing ACLs and when you use the **ip access-group** interface configuration command to apply ACLs to a Layer 3 interface, you can display the access groups on the interface. You can use the privileged EXEC commands as described in [Table 19-3](#) to display this information.

Table 19-3 Commands for Displaying Access Lists and Access Groups

Command	Purpose
show access-lists [<i>number</i> <i>name</i>]	Display the contents of one or all current IP and MAC address access lists or a specific access list (numbered or named).
show ip access-lists [<i>number</i> <i>name</i>]	Display the contents of all current IP access lists or a specific IP access list (numbered or named).
show ip interface <i>interface-id</i>	Display detailed configuration and status of an interface. If IP is enabled on the interface and ACLs have been applied by using the ip access-group interface configuration command, the access groups are included in the display.
show running-config [interface <i>interface-id</i>]	Displays the contents of the configuration file for the switch or the specified interface, including all configured access groups, whether or not IP is enabled on an interface.

This is an example of output from the **show access-lists** privileged EXEC command, displaying all standard and extended ACLs:

```
Switch# show access-lists
Standard IP access list 1
  permit 172.20.10.10
Standard IP access list 10
  permit 12.12.12.12
Standard IP access list 12
  deny 1.3.3.2
Standard IP access list 32
  permit 172.20.20.20
Standard IP access list 34
  permit 10.24.35.56
  permit 23.45.56.34
Extended IP access list 120
  permit eigrp host 12.3.6.5 host 25.36.1.24
Extended MAC access list mac1
```

This is an example of output from the **show ip access-lists** privileged EXEC command. It displays only IP standard and extended ACLs. Note that the named MAC extended ACL displayed in the previous example is not included in this display.

```
Switch# show ip access-lists
Standard IP access list 1
  permit 172.20.10.10
Standard IP access list 10
  permit 12.12.12.12
Standard IP access list 12
  deny 1.3.3.2
Standard IP access list 32
  permit 172.20.20.20
Standard IP access list 34
  permit 10.24.35.56
  permit 23.45.56.34
Extended IP access list 120
  permit eigrp host 12.3.6.5 host 25.36.1.24
```

This example shows how to view all access groups configured for Gigabit Ethernet interface 0/2, which has IP enabled:

```
Switch# show ip interface gigabitethernet0/2
GigabitEthernet0/2 is up, line protocol is down
  Internet address is 10.20.30.1/16
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is permit Any
  Inbound access list is 13
```

<output truncated>

This example shows how to use the **show running-config interface** privileged EXEC command to display the ACL configuration of Gigabit Ethernet interface 0/2:

```
Switch# show running-config interface gigabitethernet0/2
Building configuration...

Building configuration...

Current configuration : 85 bytes
!
interface GigabitEthernet0/2
 ip address 10.20.30.1 255.255.0.0
 ip access-group 13 in
 ip access-group permit Any out
 no switchport
!
<output truncated>
!
access-list 13 permit any log
access-list 101 permit icmp any any conversion-error
access-list 101 permit 234 host 172.30.40.1 host 123.23.23.2
access-list 103 permit icmp any any 123 23 tos max-throughput
access-list 103 permit igmp any any 12
<information truncated>
!
```

ACL Configuration Examples

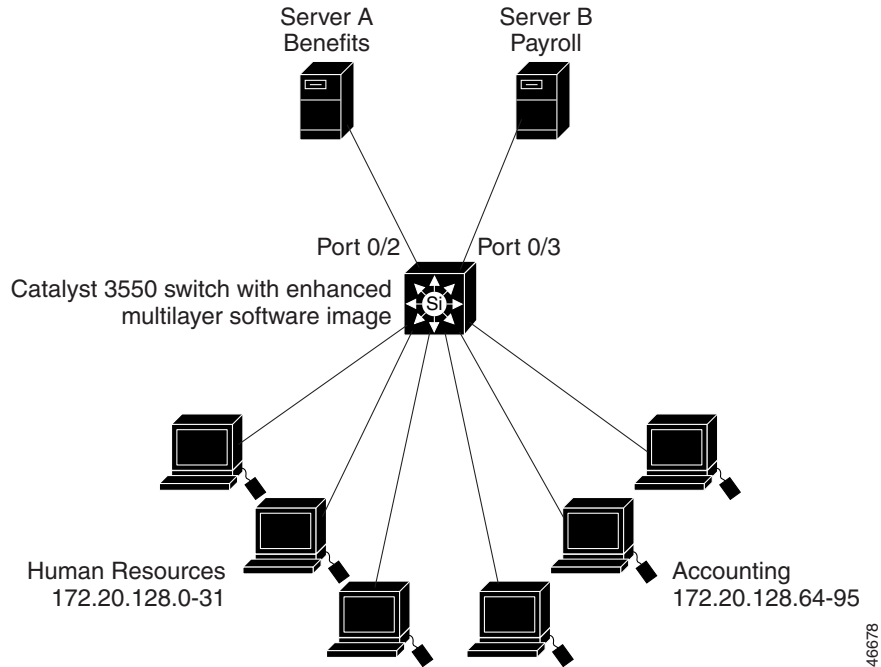
This section provides examples of configuring ACLs. For detailed information about compiling ACLs, refer to the *Security Configuration Guide* and the “IP Services” chapter of the *Cisco IOS IP and IP Routing Configuration Guide for IOS Release 12.1*.

Figure 19-3 shows a small networked office environment with the routed port 0/2 connected to Server A, containing benefits and other information that all employees can access, and routed port 0/3 connected to Server B, containing confidential payroll data. All users can access Server A, but Server B has restricted access.

Use router ACLs to do this in one of two ways:

- Create a standard ACL, and filter traffic coming to the server from port 0/3.
- Create an extended ACL, and filter traffic coming from the server into port 0/3.

Figure 19-3 Using Router ACLs to Control Traffic



This example uses a standard ACL to filter traffic coming into Server B from port 0/3, permitting traffic only from Accounting's source addresses 172.20.128.64 to 172.20.128.95.

```
Switch(config)# access-list 6 permit 172.20.128.64 0.0.0.31
Switch(config)# end
Switch# show access-lists
Standard IP access list 6
    permit 172.20.128.64, wildcard bits 0.0.0.31
Switch(config)# interface gigabitethernet0/3
Switch(config-if)# ip access-group 6 out
```

The ACL is applied to traffic coming out of routed port 0/3 from the specified source address.

This example uses an extended ACL to filter traffic coming from Server B into port 0/3, permitting traffic from any source address (in this case Server B) to only the Accounting destination addresses 172.20.128.64 to 172.20.128.95.

```
Switch(config)# access-list 106 permit ip any 172.20.128.64 0.0.0.31
Switch(config)# end
Switch# show access-lists
Extended IP access list 106
    permit ip any 172.20.128.64 0.0.0.31
Switch(config)# interface gigabitethernet0/3
Switch(config-if)# ip access-group 106 in
```

The ACL is then applied to traffic going into routed port 0/3, permitting it to go only to the specified destination addresses. Note that with extended ACLs, you must enter the protocol (IP) before the source and destination information.

Numbered ACLs

In this example, network 36.0.0.0 is a Class A network whose second octet specifies a subnet; that is, its subnet mask is 255.255.0.0. The third and fourth octets of a network 36.0.0.0 address specify a particular host. Using access list 2, the switch accepts one address on subnet 48 and reject all others on that subnet. The last line of the list shows that the switch accepts addresses on all other network 36.0.0.0 subnets. The ACL is then applied to packets entering Gigabit Ethernet interface 0/1.

```
Switch(config)# access-list 2 permit 36.48.0.3
Switch(config)# access-list 2 deny 36.48.0.0 0.0.255.255
Switch(config)# access-list 2 permit 36.0.0.0 0.255.255.255
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# ip access-group 2 in
```

Extended ACLs

In this example, the first line permits any incoming TCP connections with destination ports greater than 1023. The second line permits incoming TCP connections to the Simple Mail Transfer Protocol (SMTP) port of host 128.88.1.2. The third line permits incoming ICMP messages for error feedback.

```
Switch(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 gt 1023
Switch(config)# access-list 102 permit tcp any host 128.88.1.2 eq 25
Switch(config)# access-list 102 permit icmp any any
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# ip access-group 102 in
```

For another example of using an extended ACL, suppose that you have a network connected to the Internet, and you want any host on the network to be able to form TCP connections to any host on the Internet. However, you do not want IP hosts to be able to form TCP connections to hosts on your network, except to the mail (SMTP) port of a dedicated mail host.

SMTP uses TCP port 25 on one end of the connection and a random port number on the other end. The same port numbers are used throughout the life of the connection. Mail packets coming in from the Internet have a destination port of 25. Outbound packets have the port numbers reversed. Because the secure system behind the router always accepts mail connections on port 25, the incoming and outgoing services are separately controlled. The ACL must be configured as an input ACL on the outbound interface and an output ACL on the inbound interface.

In this example, the network is a Class B network with the address 128.88.0.0, and the mail host address is 128.88.1.2. The **established** keyword is used only for the TCP to show an established connection. A match occurs if the TCP datagram has the ACK or RST bits set, which show that the packet belongs to an existing connection. Gigabit Ethernet interface 0/1 is the interface that connects the router to the Internet.

```
Switch(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 established
Switch(config)# access-list 102 permit tcp any host 128.88.1.2 eq 25
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# ip access-group 102 in
```

Named ACLs

The following configuration creates a standard ACL named *internet_filter* and an extended ACL named *marketing_group*. The *internet_filter* ACL allows all traffic from the source address 1.2.3.4.

```
Switch(config)# ip access-list standard Internet_filter
Switch(config-ext-nacl)# permit 1.2.3.4
Switch(config-ext-nacl)# exit
```

The *marketing_group* ACL allows any TCP Telnet traffic to the destination address and wildcard 171.69.0.0 0.0.255.255 and denies any other TCP traffic. It permits any ICMP traffic, denies UDP traffic from any source to the destination address range 171.69.0.0 through 179.69.255.255 with a destination port less than 1024, denies any other IP traffic, and provides a log of the result.

```
Switch(config)# ip access-list extended marketing_group
Switch(config-ext-nacl)# permit tcp any 171.69.0.0 0.0.255.255 eq telnet
Switch(config-ext-nacl)# deny tcp any any
Switch(config-ext-nacl)# permit icmp any any
Switch(config-ext-nacl)# deny udp any 171.69.0.0 0.0.255.255 lt 1024
Switch(config-ext-nacl)# deny ip any any log
Switch(config-ext-nacl)# exit
```

The ACLs are applied to Gigabit Ethernet port 0/5, which is configured as a Layer 3 port, with the *Internet_filter* ACL applied to incoming traffic and the *marketing_group* ACL applied to outgoing traffic.

```
Switch(config)# interface gigabitethernet0/5
Switch(config-if)# no switchport
Switch(config-if)# ip address 2.0.5.1 255.255.255.0
Switch(config-if)# ip access-group Internet_filter out
Switch(config-if)# ip access-group marketing_group in
...
```

Time Range Applied to an IP ACL

This example denies Hypertext Transfer Protocol (HTTP) traffic on IP on Monday through Friday between the hours of 8:00 a.m. and 6:00 p.m. The example allows UDP traffic only on Saturday and Sunday from noon to 8:00 p.m.

```
Switch(config)# time-range no-http
Switch(config)# periodic weekdays 8:00 to 18:00
!
Switch(config)# time-range udp-yes
Switch(config)# periodic weekend 12:00 to 20:00
!
Switch(config)# ip access-list extended strict
Switch(config-ext-nacl)# deny tcp any any eq www time-range no-http
Switch(config-ext-nacl)# permit udp any any time-range udp-yes
!
Switch(config-ext-nacl)# exit
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# ip access-group strict in
```

Commented IP ACL Entries

In this example of a numbered ACL, the workstation belonging to Jones is allowed access, and the workstation belonging to Smith is not allowed access:

```
Switch(config)# access-list 1 remark Permit only Jones workstation through
Switch(config)# access-list 1 permit 171.69.2.88
Switch(config)# access-list 1 remark Do not allow Smith workstation through
Switch(config)# access-list 1 deny 171.69.3.13
```

In this example of a numbered ACL, the Winter and Smith workstations are not allowed to browse the Web:

```
Switch(config)# access-list 100 remark Do not allow Winter to browse the web
Switch(config)# access-list 100 deny host 171.69.3.85 any eq www
Switch(config)# access-list 100 remark Do not allow Smith to browse the web
Switch(config)# access-list 100 deny host 171.69.3.13 any eq www
```

In this example of a named ACL, the Jones subnet is not allowed access:

```
Switch(config)# ip access-list standard prevention
Switch(config-std-nacl)# remark Do not allow Jones subnet through
Switch(config-std-nacl)# deny 171.69.0.0 0.0.255.255
```

In this example of a named ACL, the Jones subnet is not allowed to use outbound Telnet:

```
Switch(config)# ip access-list extended telnetting
Switch(config-ext-nacl)# remark Do not allow Jones subnet to telnet out
Switch(config-ext-nacl)# deny tcp 171.69.0.0 0.0.255.255 any eq telnet
```

ACL Logging

Two variations of logging are supported on router ACLs. The **log** keyword sends an informational logging message to the console about the packet that matches the entry; the **log-input** keyword includes the input interface in the log entry.

In this example, standard named access list *stan1* denies traffic from 10.1.1.0 0.0.0.255, allows traffic from all other sources, and includes the **log** keyword.

```
Switch(config)# ip access-list standard stan1
Switch(config-std-nacl)# deny 10.1.1.0 0.0.0.255 log
Switch(config-std-nacl)# permit any log
Switch(config-std-nacl)# exit
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# ip access-group stan1 in
Switch(config-if)# end

Switch# show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Console logging: level debugging, 37 messages logged
  Monitor logging: level debugging, 0 messages logged
  Buffer logging: level debugging, 37 messages logged
  File logging: disabled
  Trap logging: level debugging, 39 message lines logged

Log Buffer (4096 bytes):

00:00:48: NTP: authentication delay calculation problems

<output truncated>

00:09:34:%SEC-6-IPACCESSLOGS:list stan1 permitted 0.0.0.0 1 packet
00:09:59:%SEC-6-IPACCESSLOGS:list stan1 denied 10.1.1.15 1 packet
00:10:11:%SEC-6-IPACCESSLOGS:list stan1 permitted 0.0.0.0 1 packet
00:15:33:%SEC-6-IPACCESSLOGS:list stan1 denied 10.1.1.15 2009 packets
```

This example is a named extended access list *ext1* that permits ICMP packets from any source to 10.1.1.0 0.0.0.255 and denies all UDP packets.

```
Switch(config)# ip access-list extended ext1
Switch(config-ext-nacl)# permit icmp any 10.1.1.0 0.0.0.255 log
Switch(config-ext-nacl)# deny udp any any log
Switch(config-std-nacl)# exit
Switch(config)# interface gigabitethernet0/3
Switch(config-if)# ip access-group ext1 in
```

This is an example of a log for an extended ACL:

```
01:24:23:%SEC-6-IPACCESSLOGDP:list ext1 permitted icmp 10.1.1.15 -> 10.1.1.61 (0/0), 1
packet
01:25:14:%SEC-6-IPACCESSLOGDP:list ext1 permitted icmp 10.1.1.15 -> 10.1.1.61 (0/0), 7
packets
01:26:12:%SEC-6-IPACCESSLOGP:list ext1 denied udp 0.0.0.0(0) -> 255.255.255.255(0), 1
packet
01:31:33:%SEC-6-IPACCESSLOGP:list ext1 denied udp 0.0.0.0(0) -> 255.255.255.255(0), 8
packets
```

Note that all logging entries for IP ACLs start with %SEC-6-IPACCESSLOG with minor variations in format depending on the kind of ACL and the access entry that has been matched.

This is an example of an output message when the **log-input** keyword is entered:

```
00:04:21:%SEC-6-IPACCESSLOGDP:list inputlog permitted icmp 10.1.1.10 (Vlan1
0001.42ef.a400) -> 10.1.1.61 (0/0), 1 packet
```

A log message for the same sort of packet using the **log** keyword does not include the input interface information:

```
00:05:47:%SEC-6-IPACCESSLOGDP:list inputlog permitted icmp 10.1.1.10 -> 10.1.1.61 (0/0), 1
packet
```

Configuring VLAN Maps

This section describes how to configure VLAN maps, which is the only way to control filtering within a VLAN. VLAN maps have no direction. To filter traffic in a specific direction by using a VLAN map, you need to include an ACL with specific source or destination addresses. If there is a match clause for that type of packet (IP or MAC) in the VLAN map, the default action is to drop the packet if the packet does not match any of the entries within the map. If there is no match clause for that type of packet, the default is to forward the packet.



Note

For complete syntax and usage information for the commands used in this section, refer to the *Catalyst 3550 Multilayer Switch Command Reference* for this release.

To create a VLAN map and apply it to one or more VLANs, perform these steps:

- Step 1** Create the standard or extended IP ACLs or named MAC extended ACLs that you want to apply to the VLAN. See the “[Creating Standard and Extended IP ACLs](#)” section on page 19-6 and the “[Creating Named MAC Extended ACLs](#)” section on page 19-28.
- Step 2** Enter the **vlan access-map** global configuration command to create a VLAN ACL map entry.
- Step 3** In access map configuration mode, optionally enter an **action—forward** (the default) or **drop**—and enter the **match** command to specify an IP packet or a non-IP packet (with only a known MAC address) and to match the packet against one or more ACLs (standard or extended).



Note

If the VLAN map has a match clause for the type of packet (IP or MAC) and the packet does not match the type, the default is to drop the packet. If there is no match clause in the VLAN map for that type of packet, and no action specified, the packet is forwarded.

Step 4 Use the **vlan filter** global configuration command to apply a VLAN map to one or more VLANs.

This section contains these topics:

- [VLAN Map Configuration Guidelines, page 19-28](#)
- [Creating Named MAC Extended ACLs, page 19-28](#)
- [Creating a VLAN Map, page 19-30](#)
- [Applying a VLAN Map to a VLAN, page 19-32](#)
- [Displaying VLAN Map Information, page 19-33](#)

VLAN Map Configuration Guidelines

Follow these guidelines when configuring VLAN maps:

- If there is no router ACL configured to deny traffic on a routed VLAN interface (input or output), and *no* VLAN map configured, all traffic is permitted.
- Each VLAN map consists of a series of entries. The order of entries in a VLAN map is important. A packet that comes into the switch is tested against the first entry in the VLAN map. If it matches, the action specified for that part of the VLAN map is taken. If there is no match, the packet is tested against the next entry in the map.
- If the VLAN map has at least one match clause for the type of packet (IP or MAC) and the packet does not match any of these match clauses, the default is to drop the packet. If there is no match clause for that type of packet in the VLAN map, the default is to forward the packet.
- The system might take longer to boot if you have configured a very large number of ACLs.
- For information about using both router ACLs and VLAN maps, see the [“Guidelines” section on page 19-36](#).
- See the [“Using VLAN Maps in Your Network” section on page 19-33](#) for configuration examples.

Creating Named MAC Extended ACLs

You can filter non-IP traffic on a VLAN by using MAC addresses and named MAC extended ACLs. The procedure is similar to that of configuring other extended named ACLs.



Note

Named MAC extended ACLs can only be applied to VLAN maps.

For more information about the supported non-IP protocols in the **mac access-list extended** command, refer to the *Catalyst 3550 Multilayer Switch Command Reference* for this release.



Note

Though visible in the command-line help strings, **appletalk** is not supported as a matching condition for the **deny** and **permit** MAC access-list configuration mode commands, nor is matching on any SNAP-encapsulated packet with a non-zero Organizational Unique Identifier (OUI).

Beginning in privileged EXEC mode, follow these steps to create a named MAC extended ACL:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	mac access-list extended <i>name</i>	Define an extended MAC access list using a name.
Step 3	{deny permit} {any host source MAC address source MAC address mask} {any host destination MAC address destination MAC address mask} [type mask lsap lsap mask aarp amber dec-spanning decnet-iv diagnostic dsm etype-6000 etype-8042 lat lave-sca mop-console mop-dump msdos mumps netbios vines-echo vines-ip xns-idp 0-65535] [cos cos]	<p>In extended MAC access-list configuration mode, specify to permit or deny any source MAC address, a source MAC address with a mask, or a specific host source MAC address and any destination MAC address, destination MAC address with a mask, or a specific destination MAC address.</p> <p>(Optional) You can also enter these options:</p> <ul style="list-style-type: none"> <i>type mask</i>—An arbitrary EtherType number of a packet with Ethernet II or SNAP encapsulation in decimal, hex, or octal with optional mask of <i>don't care</i> bits applied to the EtherType before testing for a match. lsap lsap mask—An LSAP number of a packet with 802.2 encapsulation in decimal, hex, or octal with optional mask of <i>don't care</i> bits. aarp amber dec-spanning decnet-iv diagnostic dsm etype-6000 etype-8042 lat lave-sca mop-console mop-dump msdos mumps netbios vines-echo vines-ip xns-idp—A non-IP protocol. cos cos—An IEEE 802.1Q cost of service number from 0 to 7 used to set priority.
Step 4	end	Return to privileged EXEC mode.
Step 5	show access-lists [<i>number name</i>]	Show the access list configuration.
Step 6	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Use the **no mac access-list extended** *name* global configuration command to delete the entire ACL. You can also delete individual ACEs from named MAC extended ACLs.

This example shows how to create and display an access list named *mac1*, denying only EtherType DECnet Phase IV traffic, but permitting all other types of traffic.

```
Switch(config)# mac access-list extended mac1
Switch(config-ext-macl)# deny any any decnet-iv
Switch(config-ext-macl)# permit any any
Switch(config-ext-macl)# end
Switch# show access-lists
Extended MAC access list mac1
    deny any any decnet-iv
    permit any any
```

Creating a VLAN Map

Each VLAN map consists of an ordered series of entries. Beginning in privileged EXEC mode, follow these steps to create, add to, or delete a VLAN map entry:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	vlan access-map <i>name</i> [<i>number</i>]	Create a VLAN map, and give it a name and (optionally) a number. The number is the sequence number of the entry within the map. When you create VLAN maps with the same name, numbers are assigned sequentially in increments of 10. When modifying or deleting maps, you can enter the number of the map entry that you want to modify or delete. Entering this command changes to access-map configuration mode.
Step 3	action { drop forward }	(Optional) Set the action for the map entry. The default is to forward.
Step 4	match { ip mac } address { <i>name</i> <i>number</i> } [<i>name</i> <i>number</i>]	Match the packet (using either the IP or MAC address) against one or more standard or extended access lists. Note that packets are only matched against access lists of the correct protocol type. IP packets are matched against standard or extended IP access lists. Non-IP packets are only matched against named MAC extended access lists.
Step 5	end	Return to global configuration mode.
Step 6	show running-config	Display the access list configuration.
Step 7	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Use the **no vlan access-map** *name* global configuration command to delete a map.

Use the **no vlan access-map** *name number* global configuration command to delete a single sequence entry from within the map.

Use the **no action** access-map configuration command to enforce the default action, which is to forward.

VLAN maps do not use the specific permit or deny keywords. To deny a packet by using VLAN maps, create an ACL that would match the packet, and set the action to drop. A permit in the ACL counts as a match. A deny in the ACL means no match.

Examples of ACLs and VLAN Maps

These examples show how to create ACLs and VLAN maps that for specific purposes.

Example 1

This example shows how to create an ACL and a VLAN map to deny a packet. In the first map, any packets that match the *ip1* ACL (TCP packets) would be dropped. You first create the *ip1* ACL to permit any TCP packet and no other packets. Because there is a match clause for IP packets in the VLAN map, the default action is to drop any IP packet that does not match any of the match clauses.

```
Switch(config)# ip access-list extended ip1
Switch(config-ext-nacl)# permit tcp any any
Switch(config-ext-nacl)# exit
Switch(config)# vlan access-map map_1 10
Switch(config-access-map)# match ip address ip1
Switch(config-access-map)# action drop
```

This example shows how to create a VLAN map to permit a packet. ACL *ip2* permits UDP packets and any packets that match the *ip2* ACL are forwarded.

```
Switch(config)# ip access-list extended ip2
Switch(config-ext-nacl)# permit udp any any
Switch(config-ext-nacl)# exit
Switch(config)# vlan access-map map_1 20
Switch(config-access-map)# match ip address ip2
Switch(config-access-map)# action forward
```

In this map, any IP packets that did not match any of the previous ACLs (that is, packets that are not TCP packets or UDP packets) would get dropped.

Example 2

In this example, the VLAN map has a default action of drop for IP packets and a default action of forward for MAC packets. Used with standard ACL 101 and extended named access lists **igmp-match** and **tcp-match**, the map will have the following results:

- Forward all UDP packets
- Drop all IGMP packets
- Forward all TCP packets
- Drop all other IP packets
- Forward all non-IP packets

```
Switch(config)# access-list 101 permit udp any any
Switch(config)# ip access-list extended igmp-match
Switch(config-ext-nacl)# permit igmp any any
Switch(config)# ip access-list extended tcp-match
Switch(config-ext-nacl)# permit tcp any any
Switch(config-ext-nacl)# exit
Switch(config)# vlan access-map drop-ip-default 10
Switch(config-access-map)# match ip address 101
Switch(config-access-map)# action forward
Switch(config-access-map)# exit
Switch(config)# vlan access-map drop-ip-default 20
Switch(config-access-map)# match ip address igmp-match
Switch(config-access-map)# action drop
Switch(config-access-map)# exit
Switch(config)# vlan access-map drop-ip-default 30
Switch(config-access-map)# match ip address tcp-match
Switch(config-access-map)# action forward
```

Example 3

In this example, the VLAN map has a default action of drop for MAC packets and a default action of forward for IP packets. Used with MAC extended access lists **good-hosts** and **good-protocols**, the map will have the following results:

- Forward MAC packets from hosts 0000.0c00.0111 and 0000.0c00.0211
- Forward MAC packets with decnet-iv or vines-ip protocols
- Drop all other non-IP packets
- Forward all IP packets

```

Switch(config)# mac access-list extended good-hosts
Switch(config-ext-macl)# permit host 000.0c00.0111 any
Switch(config-ext-macl)# permit host 000.0c00.0211 any
Switch(config-ext-nacl)# exit
Switch(config)# mac access-list extended good-protocols
Switch(config-ext-macl)# permit any any decnet-ip
Switch(config-ext-macl)# permit any any vines-ip
Switch(config-ext-nacl)# exit
Switch(config)# vlan access-map drop-mac-default 10
Switch(config-access-map)# match mac address good-hosts
Switch(config-access-map)# action forward
Switch(config-access-map)# exit
Switch(config)# vlan access-map drop-mac-default 20
Switch(config-access-map)# match mac address good-protocols
Switch(config-access-map)# action forward

```

Example 4

In this example, the VLAN map has a default action of drop for all packets (IP and non-IP). Used with access lists **tcp-match** and **good-hosts** from Examples 2 and 3, the map will have the following results:

- Forward all TCP packets
- Forward MAC packets from hosts 0000.0c00.0111 and 0000.0c00.0211
- Drop all other IP packets
- Drop all other MAC packets

```

Switch(config)# vlan access-map drop-all-default 10
Switch(config-access-map)# match ip address tcp-match
Switch(config-access-map)# action forward
Switch(config-access-map)# exit
Switch(config)# vlan access-map drop-all-default 20
Switch(config-access-map)# match mac address good-hosts
Switch(config-access-map)# action forward

```

Applying a VLAN Map to a VLAN

Beginning in privileged EXEC mode, follow these steps to apply a VLAN map to one or more VLANs:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	vlan filter <i>mapname</i> <i>vlan-list list</i>	Apply the VLAN map to one or more VLAN IDs. The list can be a single VLAN ID (22), a consecutive list (10-22), or a string of VLAN IDs (12, 22, 30). Spaces around the comma and hyphen are optional.
Step 3	show running-config	Display the access list configuration.
Step 4	copy running-config startup-config	(Optional) Save your entries in the configuration file.

To remove the VLAN map, use the **no vlan filter *mapname* *vlan-list list*** global configuration command.

This example shows how to apply VLAN map 1 to VLANs 20 through 22:

```
Switch(config)# vlan filter map 1 vlan-list 20-22
```

Displaying VLAN Map Information

You can display information about VLAN access maps or VLAN filters. Use the privileged EXEC commands in [Table 19-4](#) to display VLAN map information.

Table 19-4 Commands for Displaying VLAN Map Information

Command	Purpose
<code>show vlan access-map [mapname]</code>	Show information about all VLAN access-maps or the specified access map.
<code>show vlan filter [access-map name vlan vlan-id]</code>	Show information about all VLAN filters or about a specified VLAN or VLAN access map.

This is an example of output from the `show vlan access-map` privileged EXEC command:

```
Switch# show vlan access-map
Vlan access-map "map_1" 10
  Match clauses:
    ip address: ip1
  Action:
    drop
Vlan access-map "map_1" 20
  Match clauses:
    mac address: mac1
  Action:
    forward
Vlan access-map "map_1" 30
  Match clauses:
  Action:
    drop
```

This is an example of output from the `show vlan filter` privileged EXEC command:

```
Switch# show vlan filter
VLAN Map map_1 is filtering VLANs:
  20-22
```

Using VLAN Maps in Your Network

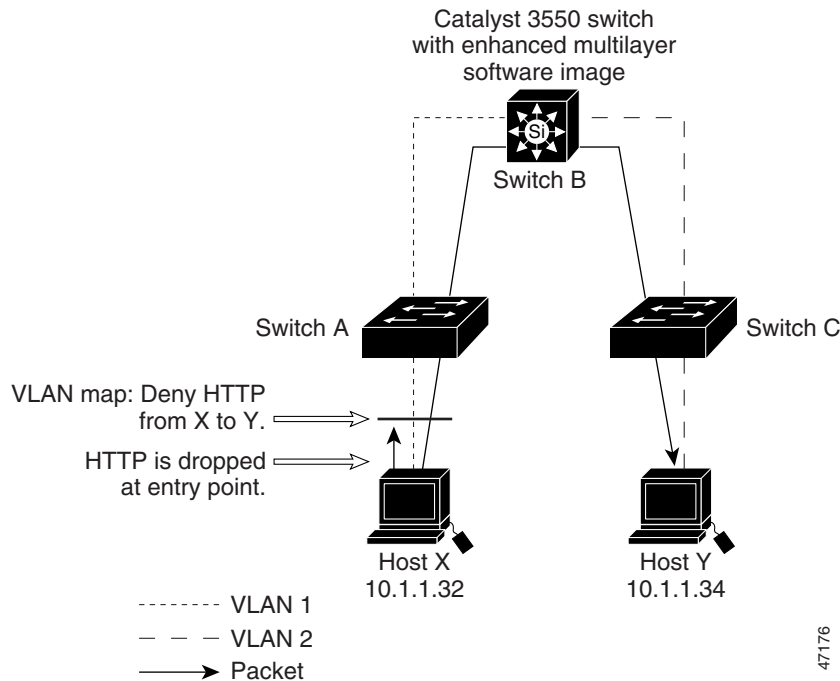
This section describes some typical uses for VLAN maps and includes these topics:

- [Wiring Closet Configuration, page 19-34](#)
- [Denying Access to a Server on Another VLAN, page 19-35](#)

Wiring Closet Configuration

In a wiring closet configuration, the Catalyst 3550 switch might not be running the enhanced multilayer software image. In this configuration, the switch can still support a VLAN map and a QoS classification ACL. In Figure 19-4, assume that Host X and Host Y are in different VLANs and are connected to wiring closet switches A and C. Traffic from Host X to Host Y is eventually being routed by Switch B, which is running the enhanced multilayer software image. Traffic from Host X to Host Y can be access-controlled at the traffic entry point, Switch A.

Figure 19-4 Wiring Closet Configuration



If you do not want HTTP traffic switched from Host X to Host Y, you can configure a VLAN map on Switch A to drop all HTTP traffic from Host X (IP address 10.1.1.32) to Host Y (IP address 10.1.1.34) at Switch A and not bridge it to Switch B.

First, define the IP access list *http* that permits (matches) any TCP traffic on the HTTP port.

```
Switch(config)# ip access-list extended http
Switch(config-ext-nacl)# permit tcp host 10.1.1.32 host 10.1.1.34 eq www
Switch(config-ext-nacl)# exit
```

Next, create VLAN access map *map2* so that traffic that matches the *http* access list is dropped and all other IP traffic is forwarded.

```
Switch(config)# vlan access-map map2 10
Switch(config-access-map)# match ip address http
Switch(config-access-map)# action drop
Switch(config-access-map)# exit
Switch(config)# ip access-list extended match_all
Switch(config-ext-nacl)# permit ip any any
Switch(config-ext-nacl)# exit
Switch(config)# vlan access-map map2 20
Switch(config-access-map)# match ip address match_all
Switch(config-access-map)# action forward
```

Then, apply VLAN access map *map2* to VLAN 1.

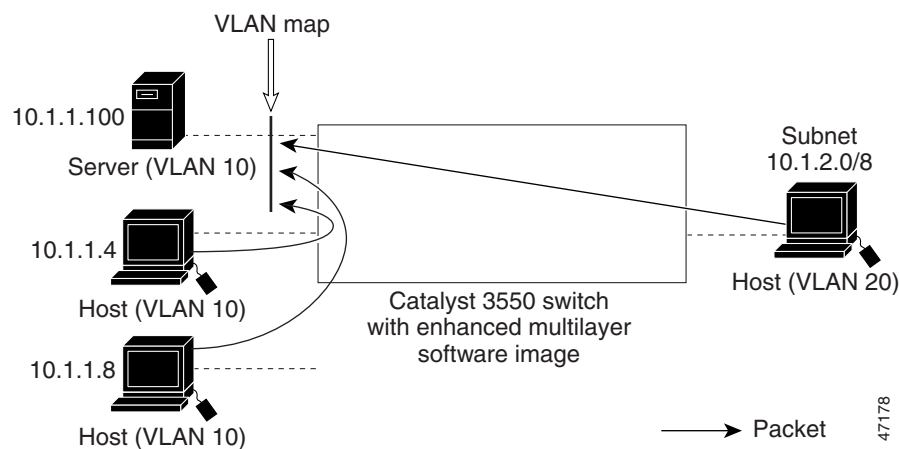
```
Switch(config)# vlan filter map2 vlan 1
```

Denying Access to a Server on Another VLAN

You can restrict access to a server on another VLAN. For example, server 10.1.1.100 in VLAN 10 needs to have access restricted as follows (see [Figure 19-5](#)):

- Hosts in subnet 10.1.2.0/8 in VLAN 20 should not have access.
- Hosts 10.1.1.4 and 10.1.1.8 in VLAN 10 should not have access.

Figure 19-5 Deny Access to a Server on Another VLAN



This example shows how to deny access to a server on another VLAN by creating the VLAN map SERVER1 that denies access to hosts in subnet 10.1.2.0/8, host 10.1.1.4, and host 10.1.1.8 and permits other IP traffic. The final step is to apply the map SERVER1 to VLAN 10.

Step 1 Define the IP ACL that will match the correct packets.

```
Switch(config)# ip access-list extended SERVER1_ACL
Switch(config-ext-nacl)# permit ip 10.1.2.0 0.0.0.255 host 10.1.1.100
Switch(config-ext-nacl)# permit ip host 10.1.1.4 host 10.1.1.100
Switch(config-ext-nacl)# permit ip host 10.1.1.8 host 10.1.1.100
Switch(config-ext-nacl)# exit
```

Step 2 Define a VLAN map using this ACL that will drop IP packets that match SERVER1_ACL and forward IP packets that do not match the ACL.

```
Switch(config)# vlan access-map SERVER1_MAP
Switch(config-access-map)# match ip address SERVER1_ACL
Switch(config-access-map)# action drop
Switch(config)# vlan access-map SERVER1_MAP 20
Switch(config-access-map)# action forward
Switch(config-access-map)# exit
```

Step 3 Apply the VLAN map to VLAN 10.

```
Switch(config)# vlan filter SERVER1_MAP vlan-list 10.
```

Using VLAN Maps with Router ACLs

To access control both bridged and routed traffic, you can use VLAN maps only or a combination of router ACLs and VLAN maps. You can define router ACLs on both input and output routed VLAN interfaces, and you can define a VLAN map to access control the bridged traffic.

If a packet flow matches a VLAN-map deny clause in the ACL, regardless of the router ACL configuration, the packet flow is denied.



Note

When you use router ACLs with VLAN maps, packets that require logging on the router ACLs are not logged if they are denied by a VLAN map.

If the VLAN map has a match clause for the type of packet (IP or MAC) and the packet does not match the type, the default is to drop the packet. If there is no match clause in the VLAN map, and no action specified, the packet is forwarded if it does not match any VLAN map entry.

This section includes this information about using VLAN maps with router ACLs:

- [Guidelines, page 19-36](#)
- [Determining if the ACL Configuration Fits in Hardware, page 19-37](#)
- [Examples of Router ACLs and VLAN Maps Applied to VLANs, page 19-39](#)

Guidelines

These guidelines are for configurations where you need to have an router ACL *and* a VLAN map on the same VLAN. These guidelines do not apply to configurations where you are mapping router ACLs and VLAN maps on different VLANs.

The switch hardware provides one lookup for security ACLs for each direction (input and output); therefore, you must merge a router ACL and a VLAN map when they are configured on the same VLAN. Merging the router ACL with the VLAN map might significantly increase the number of ACEs.

If you must configure a router ACL and a VLAN map on the same VLAN, use these guidelines for both router ACL and VLAN map configuration:

- Whenever possible, try to write the ACL with all entries having a single action except for the final, default action of the other type. That is, write the ACL using one of these two forms:

```
permit...
permit...
permit...
deny ip any any
```

or

```
deny...
deny...
deny...
permit ip any any
```

- To define multiple actions in an ACL (permit, deny), group each action type together to reduce the number of entries.

- Avoid including Layer 4 information in an ACL; adding this information complicates the merging process. The best merge results are obtained if the ACLs are filtered based on IP addresses (source and destination) and not on the full flow (source IP address, destination IP address, protocol, and protocol ports). It is also helpful to use *don't care* bits in the IP address, whenever possible.

If you need to specify the full-flow mode and the ACL contains both IP ACEs and TCP/UDP/ICMP ACEs with Layer 4 information, put the Layer 4 ACEs at the end of the list. This gives priority to the filtering of traffic based on IP addresses.

Determining if the ACL Configuration Fits in Hardware

As previously stated, ACL processing in the Catalyst 3550 switch is mostly accomplished in hardware. However, if the hardware reaches its capacity to store ACL configurations, the switch software attempts to fit a simpler configuration into the hardware. This simpler configuration does not do all the filtering that has been configured, but instead sends some or all packets to the CPU to be filtered by software. In this way, all configured filtering will be accomplished, but performance is greatly decreased when the filtering is done in software.

For example, if the combination of an input router ACL applied to a VLAN interface and a VLAN map applied to the same VLAN does not fit into the hardware, these results might occur:

- If the VLAN map alone fits in hardware, the software sets up the hardware to send to the CPU all packets that need to be routed for filtering and possible routing (if the packet passes the filter). Packets that only require bridging within the input VLAN are still handled entirely by hardware and not sent to the CPU.
- If the VLAN map does not fit in the hardware, all packets on that VLAN must be both filtered and forwarded by software.

Any problem in fitting the configuration into hardware is logged, but it is possible that not everyone who configures the switch can see the log messages as they occur. You can use the **show fm** privileged EXEC commands to determine if any interface configuration or VLAN configuration did not fit into hardware.

Beginning in privileged EXEC mode, follow these steps to see if a configuration fits into hardware:

	Command	Purpose
Step 1	show fm vlan <i>vlan-id</i>	Display feature manager information for the interface or the VLAN.
	or show fm interface <i>interface-id</i>	Determine what label was used in the hardware for the interface or VLAN configuration.
Step 2	show fm label <i>name</i>	Display which of the configured ACL features fit into hardware.

This example shows how to display detailed feature manager information on a specified interface:

```
Switch# show fm interface gigabitethernet0/12
Input Label: 0 (default)
Output Label: 0 (default)
Priority: normal
```

This output from the **show fm label** privileged EXEC command shows a merge failure on an input access group:

```
Switch# show fm label 1
Unloaded due to merge failure or lack of space:
  InputAccessGroup
  Merge Fail:input
Input Features:
  Interfaces or VLANs: V11
  Priority:normal
  Vlan Map:(none)
  Access Group:131, 6788 VMRs
  Multicast Boundary:(none), 0 VMRs
Output Features:
  Interfaces or VLANs:
  Priority:low
  Bridge Group Member:no
  Vlan Map:(none)
  Access Group:(none), 0 VMRs
```

This output from the **show fm label** privileged EXEC command shows insufficient room for an input access group in the hardware:

```
Switch# show fm label 1
Unloaded due to merge failure or lack of space:
  InputAccessGroup
Input Features:
  Interfaces or VLANs: V11
  Priority:normal
  Vlan Map:(none)
  Access Group:bigone, 11 VMRs
  Multicast Boundary:(none), 0 VMRs
Output Features:
  Interfaces or VLANs:
  Priority:low
  Bridge Group Member:no
  Vlan Map:(none)
  Access Group:(none), 0 VMRs
```

This output from the **show fm label** privileged EXEC command shows not enough room for the input access group or the output access group on the label. (Note that the access groups were configured on two different interfaces. Labels are assigned independently for input and output.)

```
Switch# show fm label 1
Unloaded due to merge failure or lack of space:
  InputAccessGroup OutputAccessGroup
Input Features:
  Interfaces or VLANs: V11
  Priority:normal
  Vlan Map:(none)
  Access Group:bigone, 11 VMRs
  Multicast Boundary:(none), 0 VMRs
Output Features:
  Interfaces or VLANs: V12
  Priority:normal
  Bridge Group Member:no
  Vlan Map:(none)
  Access Group:bigtwo, 11 VMRs
```

**Note**

When configuring ACLs on the switch, to allocate maximum hardware resources for ACLs, you can use the **sdm prefer access** global configuration command to set the Switch Database Management feature to the access template. For more information on the SDM templates, see the “[Optimizing System Resources for User-Selected Features](#)” section on page 6-57.

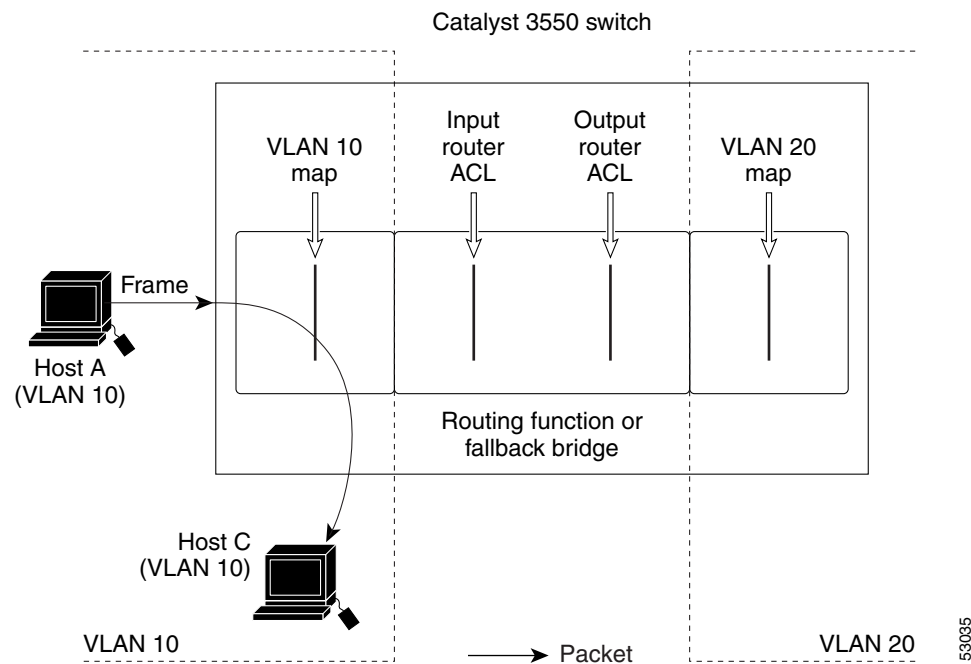
Examples of Router ACLs and VLAN Maps Applied to VLANs

This section gives examples of applying router ACLs and VLAN maps to a VLAN for switched, bridged, routed, and multicast packets. Although the following illustrations show packets being forwarded to their destination, each time the packet’s path crosses a line indicating a VLAN map or an ACL, it is also possible that the packet might be dropped, rather than forwarded.

ACLs and Switched Packets

Figure 19-6 shows how an ACL is applied on packets that are switched within a VLAN. Packets switched within the VLAN without being routed or forwarded by fallback bridging are only subject to the VLAN map of the input VLAN.

Figure 19-6 Applying ACLs on Switched Packets

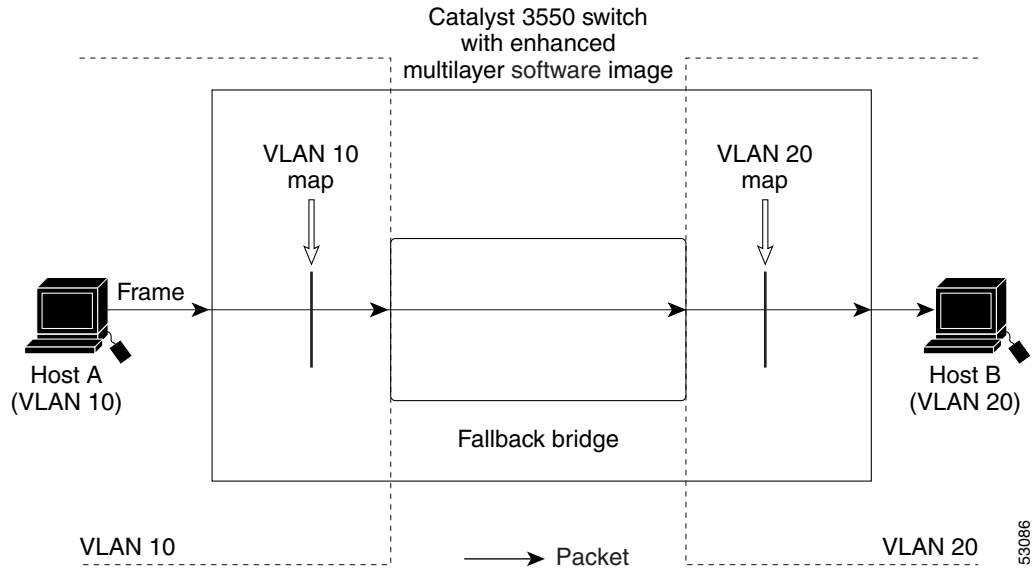


53035

ACLs and Bridged Packets

Figure 19-7 shows how an ACL is applied on fallback-bridged packets. For bridged packets, only Layer 2 ACLs are applied to the input VLAN. Only non-IP, non-ARP packets can be fallback-bridged.

Figure 19-7 Applying ACLs on Bridged Packets



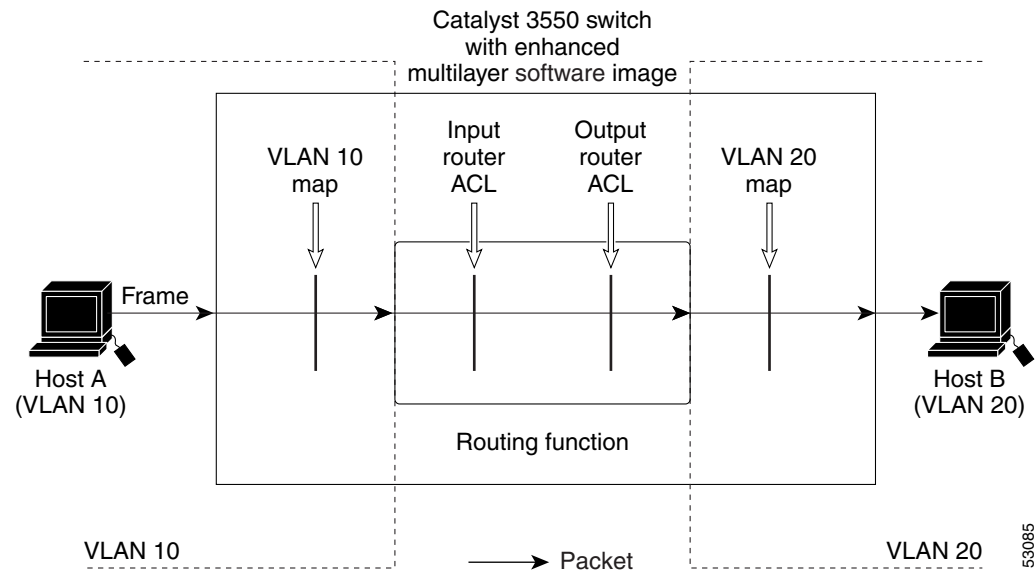
53086

ACLs and Routed Packets

Figure 19-8 shows how ACLs are applied on routed packets. For routed packets, the ACLs are applied in this order:

1. VLAN map for input VLAN
2. Input router ACL
3. Output router ACL
4. VLAN map for output VLAN

Figure 19-8 Applying ACLs on Routed Packets



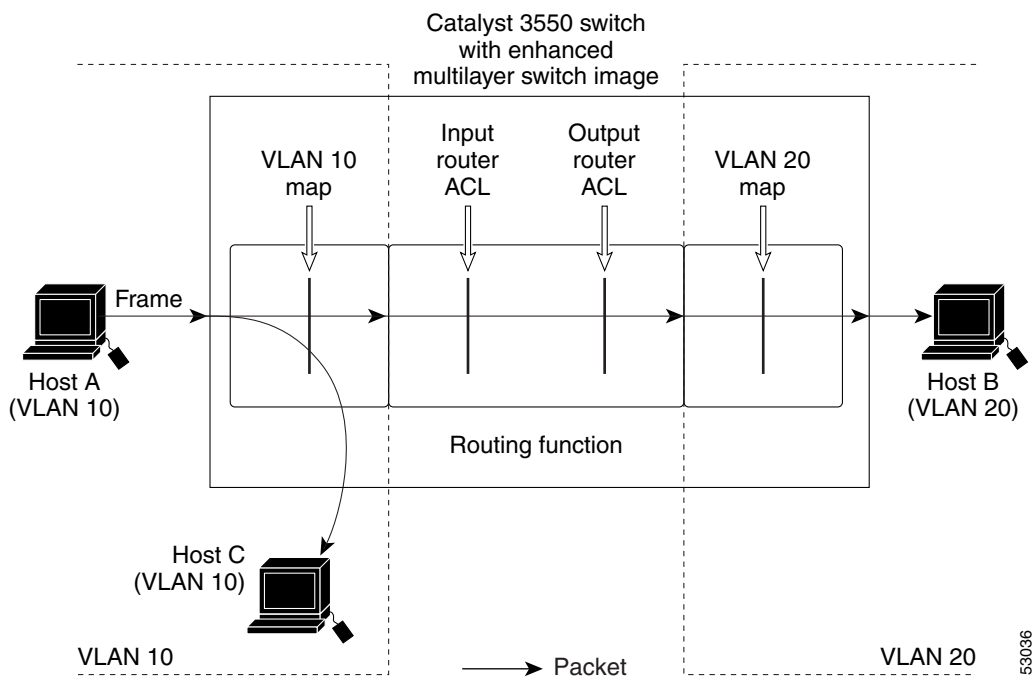
53085

ACLs and Multicast Packets

Figure 19-9 shows how ACLs are applied on packets that are replicated for IP multicasting. A multicast packet being routed has two different kinds of filters applied: one for destinations that are other ports in the input VLAN and another for each of the destinations that are in other VLANs to which the packet has been routed. The packet might be routed to more than one output VLAN, in which case a different router output ACL and VLAN map would apply for each destination VLAN.

The final result is that the packet might be permitted in some of the output VLANs and not in others. A copy of the packet is forwarded to those destinations where it is permitted. However, if the input VLAN map (VLAN 10 map in Figure 19-9) drops the packet, no destination receives a copy of the packet.

Figure 19-9 Applying ACLs on Multicast Packets



53036