Configuring IPv4 ACLs

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Prerequisites for Configuring IPv4 Access Control Lists

This section lists the prerequisites for configuring network security with access control lists (ACLs).

- On switches running the LAN base feature set, VLAN maps are not supported.

Restrictions for Configuring IPv4 Access Control Lists

General Network Security

The following are restrictions for configuring network security with 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.

- 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.

IPv4 ACL Network Interfaces

The following restrictions apply to IPv4 ACLs to network interfaces:

- When controlling access to an interface, you can use a named or numbered ACL.

- If you apply an ACL to a Layer 2 interface that is a member of a VLAN, the Layer 2 (port) ACL takes precedence over an input Layer 3 ACL applied to the VLAN interface or a VLAN map applied to the VLAN.

- If you apply an ACL to a Layer 3 interface and routing is not enabled on the switch, the ACL only filters packets that are intended for the CPU, such as SNMP, Telnet, or web traffic.

- You do not have to enable routing to apply ACLs to Layer 2 interfaces.

Note By default, the router sends Internet Control Message Protocol (ICMP) unreachable messages when a packet is denied by an access group on a Layer 3 interface. 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. They do not generate ICMP unreachable messages. ICMP unreachable messages can be disabled on router ACLs with the **no ip unreachables** interface command.

MAC ACLs on a Layer 2 Interface

After you create a MAC ACL, you can apply it to a Layer 2 interface to filter non-IP traffic coming in that interface. When you apply the MAC ACL, consider these guidelines:

- You can apply no more than one IP access list and one MAC access list to the same Layer 2 interface. The IP access list filters only IP packets, and the MAC access list filters non-IP packets.

- A Layer 2 interface can have only one MAC access list. If you apply a MAC access list to a Layer 2 interface that has a MAC ACL configured, the new ACL replaces the previously configured one.

Note The **mac access-group** interface configuration command is only valid when applied to a physical Layer 2 interface. You cannot use the command on EtherChannel port channels.

IP Access List Entry Sequence Numbering

- This feature does not support dynamic, reflexive, or firewall access lists.

Related Topics

Information about Network Security with ACLs

This chapter describes how to configure network security on the switch by using access control lists (ACLs), which in commands and tables are also referred to as access lists.

ACL Overview

Packet filtering can help limit network traffic and restrict network use by certain users or devices. ACLs filter traffic as it passes through a router or switch and permit or deny packets crossing specified interfaces or VLANs. 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. One by one, it tests packets against the conditions in an access list. The first match decides whether the switch accepts or rejects the packets. Because the switch stops testing after the first match, the order of conditions in the list is critical. If no conditions match, the switch rejects the packet. If there are no restrictions, the switch forwards the packet; otherwise, the switch drops the packet. The switch can use ACLs on all packets it forwards, 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.

Access Control Entries

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.

ACL Supported Types

The switch supports IP ACLs and Ethernet (MAC) ACLs:

- IP ACLs filter IPv4 traffic, including TCP, User Datagram Protocol (UDP), Internet Group Management Protocol (IGMP), and Internet Control Message Protocol (ICMP).

- Ethernet ACLs filter non-IP traffic.

This switch also supports quality of service (QoS) classification ACLs.

Supported ACLs

The switch supports three types of ACLs to filter traffic:

- Port ACLs access-control traffic entering a Layer 2 interface. You can apply only one IP access list and one MAC access list to a Layer 2 interface.

- Router ACLs access-control routed traffic between VLANs and are applied to Layer 3 interfaces in a specific direction (inbound or outbound).

- 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. VLAN maps are configured to provide access control based on Layer 3 addresses for IPv4. 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.

ACL Precedence

When VLAN maps, Port ACLs, and router ACLs are configured on the same switch, the filtering precedence, from greatest to least, is port ACL, router ACL, then VLAN map. The following examples describe simple use cases:

- When both an input port ACL and a VLAN map are applied, incoming packets received on ports with a port ACL applied are filtered by the port ACL. Other packets are filtered by the VLAN map

- When an input router ACL and input port ACL exist in a switch virtual interface (SVI), incoming packets received on ports to which a port ACL is applied are filtered by the port ACL. Incoming routed IP packets received on other ports are filtered by the router ACL. Other packets are not filtered.

- When an output router ACL and input port ACL exist in an SVI, incoming packets received on the ports to which a port ACL is applied are filtered by the port ACL. Outgoing routed IP packets are filtered by the router ACL. Other packets are not filtered.

- When a VLAN map, input router ACL, and input port ACL exist in an SVI, incoming packets received on the ports to which a port ACL is applied are only filtered by the port ACL. Incoming routed IP packets received on other ports are filtered by both the VLAN map and the router ACL. Other packets are filtered only by the VLAN map.

- When a VLAN map, output router ACL, and input port ACL exist in an SVI, incoming packets received on the ports to which a port ACL is applied are only filtered by the port ACL. Outgoing routed IP packets are filtered by both the VLAN map and the router ACL. Other packets are filtered only by the VLAN map.

**Related Topics**

Port ACLs

Port ACLs are ACLs that are applied to Layer 2 interfaces on a switch. Port ACLs are supported only on physical interfaces and not on EtherChannel interfaces. Port ACLs can be applied on outbound and inbound interfaces. The following access lists are supported:

- Standard IP access lists using source addresses

- Extended IP access lists using source and destination addresses and optional protocol type information

- MAC extended access lists using source and destination MAC addresses and optional protocol type information

The switch examines ACLs on an interface and permits or denies packet forwarding based on how the packet matches the entries in the ACL. In this way, ACLs control access to a network or to part of a network.

Figure 1: Using ACLs to Control Traffic in a Network

This is an example of using port ACLs to control access to a network when all workstations are in the same VLAN. ACLs applied at the Layer 2 input would allow Host A to access the Human Resources network, but prevent Host B from accessing the same network. Port ACLs can only be applied to Layer 2 interfaces in the
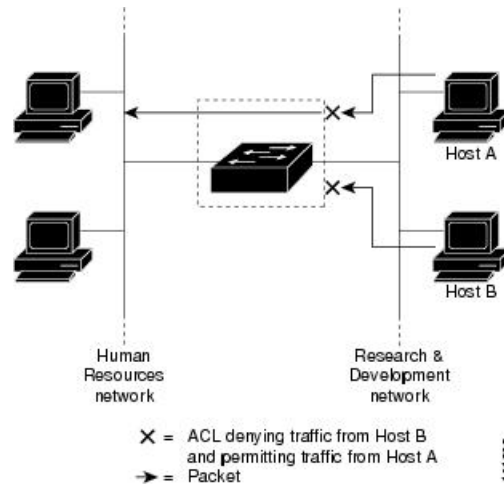
inbound direction.

When you apply a port ACL to a trunk port, the ACL filters traffic on all VLANs present on the trunk port. When you apply a port ACL to a port with voice VLAN, the ACL filters traffic on both data and voice VLANs.

With port ACLs, you can filter IP traffic by using IP access lists and non-IP traffic by using MAC addresses. You can filter both IP and non-IP traffic on the same Layer 2 interface by applying both an IP access list and a MAC access list to the interface.

Note　You cannot apply more than one IP access list and one MAC access list to a Layer 2 interface. If an IP access list or MAC access list is already configured on a Layer 2 interface and you apply a new IP access list or MAC access list to the interface, the new ACL replaces the previously configured one.

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. You apply router ACLs on interfaces for specific directions (inbound or outbound). You can apply one router ACL in each direction on an interface.

The switch supports these access lists for IPv4 traffic:

- • 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.

As with port ACLs, the switch examines ACLs associated with features configured on a given interface. 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, and can be used to control access to a network or to part of a network.

VLAN Maps

Use VLAN ACLs or VLAN maps to access-control all traffic. You can apply VLAN maps to all packets that are routed into or out of a VLAN or are bridged within a VLAN in the switch or switch stack.

Use VLAN maps for security packet filtering. VLAN maps are not defined by direction (input or output).

You can configure VLAN maps to match Layer 3 addresses for IPv4 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 2: Using VLAN Maps to Control Traffic

This shows how a VLAN map is applied to prevent a specific type of traffic from Host A in VLAN 10 from being forwarded. You can apply only one VLAN map to a
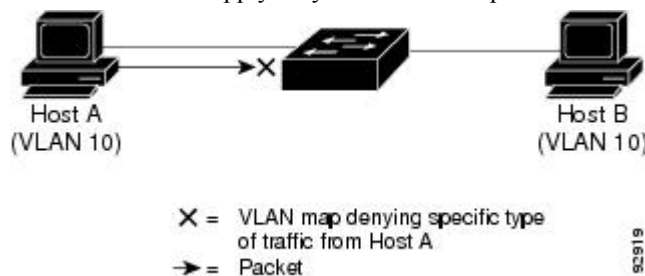
VLAN.

ACEs and 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 access control entries (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.

ACEs and Fragmented and Unfragmented Traffic Examples

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

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

Standard and Extended IPv4 ACLs

This section describes IP ACLs.

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

The software supports these types of ACLs or access lists for IPv4:

- 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.

IPv4 ACL Switch Unsupported Features

Configuring IPv4 ACLs on the switch is the same as configuring IPv4 ACLs on other Cisco switches and routers.

The following ACL-related features are not supported:

- Non-IP protocol ACLs

- IP accounting

- Reflexive ACLs and dynamic ACLs are not supported.

- ACL logging for port ACLs and VLAN maps

Access List Numbers

The number you use to denote your ACL shows the type of access list that you are creating.

This lists the access-list number and corresponding access list type and shows whether or not they are supported in the switch. The switch supports IPv4 standard and extended access lists, numbers 1 to 199 and 1300 to 2699.

Table 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
1300–1999	IP standard access list (expanded range)	Yes

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

In addition to numbered standard and extended ACLs, you can also create standard and extended named IP ACLs by 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.

Numbered Standard IPv4 ACLs

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.

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.

After creating a numbered standard IPv4 ACL, you can apply it to VLANs, to terminal lines, or to interfaces.

Numbered Extended IPv4 ACLs

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. 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.

The 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.

Some protocols also have specific parameters and keywords that apply to that protocol.

You can define an extended TCP, UDP, ICMP, IGMP, or other IP ACL. The switch also supports these IP protocols:

Note ICMP echo-reply cannot be filtered. All other ICMP codes or types can be filtered.

These IP protocols are supported:

- Authentication Header Protocol (**ahp**)

- Encapsulation Security Payload (**esp**)

- Enhanced Interior Gateway Routing Protocol (**eigrp**)

- generic routing encapsulation (**gre**)

- Internet Control Message Protocol (**icmp**)

- Internet Group Management Protocol (**igmp**)

> • 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**)
>
> • User Datagram Protocol (**udp**)

Named IPv4 ACLs

You can identify IPv4 ACLs with an alphanumeric string (a name) rather than a number. You can use named ACLs to configure more IPv4 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 and . The advantage of using named ACLs instead of numbered lists is that you can delete individual entries from a named list.

Consider these guidelines before configuring named ACLs:

> • Numbered ACLs are also available.
>
> • A standard ACL and an extended ACL cannot have the same name.
>
> • You can use standard or extended ACLs (named or numbered) in VLAN maps.

ACL Logging

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 appear 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	The logging facility might drop some logging message packets if there are too many to be handled or if there is more than one logging message to be handled in 1 second. This behavior prevents the router from crashing due to too many logging packets. Therefore, the logging facility should not be used as a billing tool or an accurate source of the number of matches to an access list.

Hardware and Software Treatment of IP ACLs

ACL processing is performed in hardware. If the hardware reaches its capacity to store ACL configurations, all packets on that interface are dropped.

Note	If an ACL configuration cannot be implemented in hardware due to an out-of-resource condition on a switch or stack member, then only the traffic in that VLAN arriving on that switch is affected.

For router ACLs, other factors can cause packets to be sent to the CPU:

- Using the **log** keyword

- Generating ICMP unreachable messages

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.

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 platform acl counters hardware** 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 unreachables* 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 for logging only. If the ACE is a *permit* statement, the packet is still switched and routed in hardware.

VLAN Map Configuration Guidelines

VLAN maps are 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.

The following are the VLAN map configuration guidelines:

- If there is no ACL configured to deny traffic on an interface and no VLAN map is configured, all traffic is permitted.

- Each VLAN map consists of a series of entries. The order of entries in an 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.

- Logging is not supported for VLAN maps.

- When a switch has an IP access list or MAC access list applied to a Layer 2 interface, and you apply a VLAN map to a VLAN that the port belongs to, the port ACL takes precedence over the VLAN map.

- If a VLAN map configuration cannot be applied in hardware, all packets in that VLAN are dropped.

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.

VLAN Maps and Router ACL Configuration 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.

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:

- You can configure only one VLAN map and one router ACL in each direction (input/output) on a VLAN interface.

- 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.

VACL Logging

When you configure VACL logging, syslog messages are generated for denied IP packets under these circumstances:

- When the first matching packet is received.

- For any matching packets received within the last 5 minutes.

- If the threshold is reached before the 5-minute interval.

Log messages are generated on a per-flow basis. A flow is defined as packets with the same IP addresses and Layer 4 (UDP or TCP) port numbers. If a flow does not receive any packets in the 5-minute interval, that flow is removed from the cache. When a syslog message is generated, the timer and packet counter are reset.

VACL logging restrictions:

- Only denied IP packets are logged.

- Packets that require logging on the outbound port ACLs are not logged if they are denied by a VACL.

Time Ranges for ACLs

You can selectively apply extended ACLs based on the time of day and the week by using the **time-range** global configuration command. First, define a time-range name and set the times and the dates or the days of the week in the time range. Then enter the time-range name when applying an ACL to set 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, for example, during a specified time period or on specified days of the week. The **time-range** keyword and argument are referenced in the named and numbered extended ACL task tables.

These are some 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 be set to log traffic only at certain times of the day. Therefore, you can simply deny access without needing to analyze many logs generated during peak hours.

Time-based access lists trigger CPU activity because the new configuration of the access list must be merged with other features and the combined configuration loaded into the hardware memory. For this reason, you should be careful not to have several access lists configured to take affect in close succession (within a small number of minutes of each other.)

Note The time range relies on the switch system clock; therefore, you need a reliable clock source. We recommend that you use Network Time Protocol (NTP) to synchronize the switch clock.

Related Topics

IPv4 ACL Interface Considerations

When you apply the **ip access-group** interface configuration command 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.

By default, the input interface sends ICMP Unreachable messages 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.

Related Topics

How to Configure ACLs

Configuring IPv4 ACLs

These are the steps to use IP ACLs on the switch:

SUMMARY STEPS

1. Create an ACL by specifying an access list number or name and the access conditions.

2. Apply the ACL to interfaces or terminal lines. You can also apply standard and extended IP ACLs to VLAN maps.

DETAILED STEPS

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

Creating a Numbered Standard ACL

Follow these steps to create a numbered standard ACL:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} *source source-wildcard* [**log**]
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** SwitchDevice> **enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** SwitchDevice# **configure terminal**	Enters the global configuration mode.
Step 3	**access-list** *access-list-number* {**deny**	**permit**} *source source-wildcard* [**log**] **Example:** SwitchDevice(config)# **access-list 2 deny your_host**	Defines a standard IPv4 access list by using a source address and wildcard. The *access-list-number* is a decimal number from 1 to 99 or 1300 to 1999. Enter **deny** or **permit** to specify whether to deny or permit access if conditions are matched.

Command or Action	Purpose
	The *source* is the source address of the network or host from which the packet is being sent specified as:
	• The 32-bit quantity in dotted-decimal format.
	• The keyword **any** as an abbreviation for *source* and *source-wildcard* 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 source and *source-wildcard* of *source* 0.0.0.0.
	(Optional) The *source-wildcard* applies wildcard bits to the source.
	(Optional) Enter **log** to cause an informational logging message about the packet that matches the entry to be sent to the console.
	Note Logging is supported only on ACLs attached to Layer 3 interfaces.
Step 4 **end** **Example:** SwitchDevice(config)# **end**	Returns to privileged EXEC mode.
Step 5 **show running-config** **Example:** SwitchDevice# **show running-config**	Verifies your entries.
Step 6 **copy running-config startup-config** **Example:** SwitchDevice# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Related Topics

Creating a Numbered Extended ACL

Follow these steps to create a numbered extended ACL:

SUMMARY STEPS

1. **configure terminal**

2. **access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** tos] [**fragments**] [**log** [**log-input**] [**time-range** *time-range-name*] [**dscp** *dscp*]

3. **access-list** *access-list-number* {**deny** | **permit**} **tcp** *source source-wildcard* [*operator port*] *destination destination-wildcard* [*operator port*] [**established**] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**log** [**log-input**] [**time-range** *time-range-name*] [**dscp** *dscp*] [*flag*]

4. **access-list** *access-list-number* {**deny** | **permit**} **udp** *source source-wildcard* [*operator port*] *destination destination-wildcard* [*operator port*] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**log** [**log-input**] [**time-range** *time-range-name*] [**dscp** *dscp*]

5. **access-list** *access-list-number* {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard* [*icmp-type* | [[*icmp-type icmp-code*] | [*icmp-message*]] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**time-range** *time-range-name*] [**dscp** *dscp*]

6. **access-list** *access-list-number* {**deny** | **permit**} **igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**log** [**log-input**] [**time-range** *time-range-name*] [**dscp** *dscp*]

7. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** SwitchDevice# **configure terminal**	Enters the global configuration mode.
Step 2	**access-list** *access-list-number* {**deny**	**permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** tos] [**fragments**] [**log** [**log-input**] [**time-range** *time-range-name*] [**dscp** *dscp*] **Example:** SwitchDevice(config)# **access-list 101 permit ip host 10.1.1.2 any precedence 0 tos 0 log**	Defines an extended IPv4 access list and the access conditions. The *access-list-number* 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 *protocol*, enter the name or number of an P 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 the following steps. The *source* is the number of the network or host from which the packet is sent. The *source-wildcard* applies wildcard bits to the source. The *destination* is the network or host number to which the packet is sent.

Command or Action	Purpose
	The *destination-wildcard* applies wildcard bits to the destination.
	Source, source-wildcard, destination, and destination-wildcard can be specified as:
	• 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:
	• **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**—Specify the time-range name.
	• **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.
	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**.
Step 3 **access-list** *access-list-number* {**deny** | **permit**} **tcp** *source source-wildcard* [*operator port*] *destination destination-wildcard* [*operator port*] [**established**] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**log** [**log-input**] [**time-range** *time-range-name*] [**dscp** *dscp*] [*flag*] **Example:** SwitchDevice(config)# **access-list 101 permit tcp**	Defines an extended TCP access list and the access conditions. The parameters are the same as those described for an extended IPv4 ACL, with these exceptions: (Optional) Enter an *operator* and *port* to compare source (if positioned after *source source-wildcard*) or destination (if positioned after *destination destination-wildcard*) port. Possible operators include **eq** (equal), **gt** (greater than), **lt** (less than), **neq** (not equal), and **range** (inclusive range).

	Command or Action	**Purpose**
	`any any eq 500`	Operators require a port number (range requires two port numbers separated by a space).
		Enter the *port* number as a decimal number (from 0 to 65535) or the name of a TCP port. Use only TCP port numbers or names when filtering TCP.
		The other optional keywords have these meanings:
		• **established**—Enter to match an established connection. This has the same function as matching on the **ack** or **rst** flag.
		• *flag*—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 4	**access-list** *access-list-number* {**deny** | **permit**} **udp** *source source-wildcard* [*operator port*] *destination destination-wildcard* [*operator port*] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**log** [**log-input**] [**time-range** *time-range-name*] [**dscp** *dscp*] **Example:** SwitchDevice(config)# **access-list 101 permit udp any any eq 100**	(Optional) Defines an extended UDP access list and the access conditions. The UDP parameters are the same as those described for TCP except that the [operator [port]] port number or name must be a UDP port number or name, and the **flag** and **established** keywords are not valid for UDP.
Step 5	**access-list** *access-list-number* {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard* [*icmp-type* | [[*icmp-type icmp-code*] | [*icmp-message*]] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**time-range** *time-range-name*] [**dscp** *dscp*] **Example:** SwitchDevice(config)# **access-list 101 permit icmp any any 200**	Defines an extended ICMP access list and the access conditions. The ICMP parameters are the same as those described for most IP protocols in an extended IPv4 ACL, with the addition of the ICMP message type and code parameters. These optional keywords have these meanings: • *icmp-type*—Enter to filter by ICMP message type, a number from 0 to 255. • *icmp-code*—Enter to filter ICMP packets that are filtered by the ICMP message code type, a number from 0 to 255. • *icmp-message*—Enter to filter ICMP packets by the ICMP message type name or the ICMP message type and code name.
Step 6	**access-list** *access-list-number* {**deny** | **permit**} **igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**fragments**] [**log** [**log-input**] [**time-range** *time-range-name*] [**dscp** *dscp*] **Example:**	(Optional) Defines an extended IGMP access list and the access conditions. The IGMP parameters are the same as those described for most IP protocols in an extended IPv4 ACL, with this optional parameter.

	Command or Action	Purpose
	SwitchDevice(config)# **access-list 101 permit igmp any any 14**	*igmp-type*—To match IGMP message type, enter a number from 0 to 15, or enter the message name: **dvmrp**, **host-query**, **host-report**, **pim**, or **trace**.
Step 7	**end** **Example:** SwitchDevice(config)# **end**	Returns to privileged EXEC mode.

Related Topics

Creating Named Standard ACLs

Follow these steps to create a standard ACL using names:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list standard** *name*
4. Use one of the following:

   - **deny** {*source* [*source-wildcard*] | **host** *source* | **any**} [**log**]
   - **permit** {*source* [*source-wildcard*] | **host** *source* | **any**} [**log**]

5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** SwitchDevice> **enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** SwitchDevice# **configure terminal**	Enters the global configuration mode.

	Command or Action	Purpose
Step 3	**ip access-list standard** *name* **Example:** SwitchDevice(config)# **ip access-list standard 20**	Defines a standard IPv4 access list using a name, and enter access-list configuration mode. The name can be a number from 1 to 99.

Step 4	Use one of the following:
	
	  • **deny** {*source* [*source-wildcard*] | **host** *source* | **any**} [**log**]
	  • **permit** {*source* [*source-wildcard*] | **host** *source* | **any**} [**log**]
	
	Example:
	
	SwitchDevice(config-std-nacl)# **deny 192.168.0.0 0.0.255.255 255.255.0.0 0.0.255.255**
	
	or
	
	SwitchDevice(config-std-nacl)# **permit 10.108.0.0 0.0.0.0 255.255.255.0 0.0.0.0**
		In access-list configuration mode, specify one or more conditions denied or permitted to decide if the packet is forwarded or dropped.
		
		  • **host** *source*—A source and source wildcard of *source* 0.0.0.0.
		
		  • **any**—A source and source wildcard of 0.0.0.0 255.255.255.255.

Step 5	**end** **Example:** SwitchDevice(config-std-nacl)# **end**	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** SwitchDevice# **show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** SwitchDevice# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Creating Extended Named ACLs

Follow these steps to create an extended ACL using names:

SUMMARY STEPS

    1. **enable**

2. **configure terminal**
3. **ip access-list extended** *name*
4. {**deny** | **permit**} *protocol* {*source* [*source-wildcard*] | **host** *source* | **any**} {*destination* [*destination-wildcard*] | host *destination* | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**] [**time-range** *time-range-name*]
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** SwitchDevice> **enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** SwitchDevice# **configure terminal**	Enters the global configuration mode.
Step 3	**ip access-list extended** *name* **Example:** SwitchDevice(config)# **ip access-list extended 150**	Defines an extended IPv4 access list using a name, and enter access-list configuration mode. The name can be a number from 100 to 199.
Step 4	{**deny**	**permit**} *protocol* {*source* [*source-wildcard*]	**host** *source*	**any**} {*destination* [*destination-wildcard*]	host *destination*	**any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**] [**time-range** *time-range-name*] **Example:** SwitchDevice(config-ext-nacl)# **permit 0 any any**	In access-list configuration mode, specify the conditions allowed or denied. Use the **log** keyword to get access list logging messages, including violations. • **host** *source*—A source and source wildcard of *source* 0.0.0.0. • **host** *destintation*—A destination and destination wildcard of *destination* 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 5	**end** **Example:** SwitchDevice(config-ext-nacl)# **end**	Returns to privileged EXEC mode.

	Command or Action	**Purpose**
Step 6	**show running-config** **Example:** SwitchDevice# **show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** SwitchDevice# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

When you are creating 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.

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

What to do next

After creating a named ACL, you can apply it to interfaces or to VLANs .

Configuring Time Ranges for ACLs

Follow these steps to configure a time-range parameter for an ACL:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **time-range** *time-range-name*
4. Use one of the following:

   - **absolute** [**start** *time date*] [**end** *time date*]
   - **periodic** *day-of-the-week hh:mm to* [*day-of-the-week*] *hh:mm*
   - **periodic** {**weekdays** | **weekend** | **daily**} *hh:mm to hh:mm*

5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** SwitchDevice(config)# **enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** SwitchDevice# **configure terminal**	Enters the global configuration mode.
Step 3	**time-range** *time-range-name* **Example:** SwitchDevice(config)# **time-range workhours**	Assigns a meaningful name (for example, *workhours*) to the time range to be created, and enter time-range configuration mode. The name cannot contain a space or quotation mark and must begin with a letter.

Step 4	Use one of the following:
	
	   • **absolute** [**start** *time date*] [**end** *time date*]
	   • **periodic** *day-of-the-week hh:mm to* [*day-of-the-week*] *hh:mm*
	   • **periodic** {**weekdays** | **weekend** | **daily**} *hh:mm to hh:mm*
	
	Example:
	
	SwitchDevice(config-time-range)# **absolute start 00:00 1 Jan 2006 end 23:59 1 Jan 2006**
	
	or
	
	SwitchDevice(config-time-range)# **periodic weekdays 8:00 to 12:00**
		Specifies when the function it will be applied to is operational.
		
		   • You can use only one **absolute** statement in the time range. If you configure more than one absolute statement, only the one configured last is executed.
		
		   • You can enter multiple **periodic** statements. For example, you could configure different hours for weekdays and weekends.
		
		See the example configurations.

Step 5	**end** **Example:** SwitchDevice(config)# **end**	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** SwitchDevice# **show running-config**	Verifies your entries.

	Command or Action	Purpose
Step 7	**copy running-config startup-config** **Example:** SwitchDevice# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

What to do next

Repeat the steps if you have multiple items that you want in effect at different times.

Related Topics

Applying an IPv4 ACL to a Terminal Line

You can use numbered ACLs to control access to one or more terminal lines. You cannot apply named ACLs to lines. You must set identical restrictions on all the virtual terminal lines because a user can attempt to connect to any of them.

Follow these steps to restrict incoming and outgoing connections between a virtual terminal line and the addresses in an ACL:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** [**console** | **vty**] *line-number*
4. **access-class** *access-list-number* {**in** | **out**}
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** SwitchDevice(config)# **enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** SwitchDevice# **configure terminal**	Enters the global configuration mode.

	Command or Action	Purpose
Step 3	**line** [**console** | **vty**] *line-number* **Example:** SwitchDevice(config)# **line console 0**	Identifies a specific line to configure, and enter in-line configuration mode. • **console**—Specifies the console terminal line. The console port is DCE. • **vty**—Specifies a virtual terminal for remote console access. The *line-number* 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 4	**access-class** *access-list-number* {**in** | **out**} **Example:** SwitchDevice(config-line)# **access-class 10 in**	Restricts incoming and outgoing connections between a particular virtual terminal line (into a device) and the addresses in an access list.
Step 5	**end** **Example:** SwitchDevice(config-line)# **end**	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** SwitchDevice# **show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** SwitchDevice# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Applying an IPv4 ACL to an Interface

This section describes how to apply IPv4 ACLs to network interfaces.

Beginning in privileged EXEC mode, follow these steps to control access to an interface:

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-id*
3. **ip access-group** {*access-list-number* \| *name*} {**in** \| **out**}
4. **end**

5. **show running-config**

6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** SwitchDevice# **configure terminal**	Enters the global configuration mode.
Step 2	**interface** *interface-id* **Example:** SwitchDevice(config)# **interface gigabitethernet1/0/1**	Identifies a specific interface for configuration, and enter interface configuration mode. The interface can be a Layer 2 interface (port ACL), or a Layer 3 interface (router ACL).
Step 3	**ip access-group** {*access-list-number* | *name*} {**in** | **out**} **Example:** SwitchDevice(config-if)# **ip access-group 2 in**	Controls access to the specified interface. The **out** keyword is not supported for Layer 2 interfaces (port ACLs).
Step 4	**end** **Example:** SwitchDevice(config-if)# **end**	Returns to privileged EXEC mode.
Step 5	**show running-config** **Example:** SwitchDevice# **show running-config**	Displays the access list configuration.
Step 6	**copy running-config startup-config** **Example:** SwitchDevice# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Related Topics

IPv4 ACL Interface Considerations, on page 14

Restrictions for Configuring IPv4 Access Control Lists, on page 1

Creating Named MAC Extended ACLs

You can filter non-IPv4 traffic on a VLAN or on a Layer 2 interface by using MAC addresses and named MAC extended ACLs. The procedure is similar to that of configuring other extended named ACLs.

Follow these steps to create a named MAC extended ACL:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mac access-list extended** *name*
4. {**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** | **lavc-sca** | **mop-console** | **mop-dump** | **msdos** | **mumps** | **netbios** | **vines-echo** | **vines-ip** | **xns-idp** | 0-*65535*] [**cos** *cos*]
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** SwitchDevice> **enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** SwitchDevice# **configure terminal**	Enters the global configuration mode.
Step 3	**mac access-list extended** *name* **Example:** SwitchDevice(config)# **mac access-list extended mac1**	Defines an extended MAC access list using a name.
Step 4	{**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**	**lavc-sca**	**mop-console**	**mop-dump**	**msdos**	**mumps**	**netbios**	**vines-echo**	**vines-ip**	**xns-idp**	0-*65535*] [**cos** *cos*]	In extended MAC access-list configuration mode, specifies 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. (Optional) You can also enter these options:

	Command or Action	Purpose
	Example: SwitchDevice(config-ext-macl)# **deny any any decnet-iv** or SwitchDevice(config-ext-macl)# **permit any any**	• *type mask*—An arbitrary EtherType number of a packet with Ethernet II or SNAP encapsulation in decimal, hexadecimal, or octal with optional mask of *don't care* bits applied to the EtherType before testing for a match. • **lsap** *lsap mask*—An LSAP number of a packet with IEEE 802.2 encapsulation in decimal, hexadecimal, or octal with optional mask of *don't care* bits. • **aarp** | **amber** | **dec-spanning** | **decnet-iv** | **diagnostic** | **dsm** | **etype-6000** | **etype-8042** | **lat** | **lavc-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 5	**end** **Example:** SwitchDevice(config-ext-macl)# **end**	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** SwitchDevice# **show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** SwitchDevice# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Related Topics

Applying a MAC ACL to a Layer 2 Interface

Follow these steps to apply a MAC access list to control access to a Layer 2 interface:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*

4. **mac access-group** {*name*} {**in** | **out** }
5. **end**
6. **show mac access-group** [**interface** *interface-id*]
7. **show running-config**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** SwitchDevice> **enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** SwitchDevice# **configure terminal**	Enters the global configuration mode.
Step 3	**interface** *interface-id* **Example:** SwitchDevice(config)# **interface gigabitethernet1/0/2**	Identifies a specific interface, and enter interface configuration mode. The interface must be a physical Layer 2 interface (port ACL).
Step 4	**mac access-group** {*name*} {**in**	**out** } **Example:** SwitchDevice(config-if)# **mac access-group mac1 in**	Controls access to the specified interface by using the MAC access list. Port ACLs are supported in the outbound and inbound directions .
Step 5	**end** **Example:** SwitchDevice(config-if)# **end**	Returns to privileged EXEC mode.
Step 6	**show mac access-group** [**interface** *interface-id*] **Example:** SwitchDevice# **show mac access-group interface gigabitethernet1/0/2**	Displays the MAC access list applied to the interface or all Layer 2 interfaces.
Step 7	**show running-config** **Example:**	Verifies your entries.

	Command or Action	Purpose
	SwitchDevice# **show running-config**	
Step 8	**copy running-config startup-config** **Example:** SwitchDevice# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

After receiving a packet, the switch checks it against the inbound ACL. If the ACL permits it, the switch continues to process the packet. If the ACL rejects the packet, the switch discards it. When you apply an undefined ACL to an interface, the switch acts as if the ACL has not been applied and permits all packets. Remember this behavior if you use undefined ACLs for network security.

Related Topics

Restrictions for Configuring IPv4 Access Control Lists, on page 1

Configuring VLAN Maps

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

Before you begin

Create the standard or extended IPv4 ACLs or named MAC extended ACLs that you want to apply to the VLAN.

SUMMARY STEPS

1. **vlan access-map** *name* [**number**]
2. **match** {**ip** | **mac**} **address** {*name* | *number*} [*name* | *number*]
3. Enter one of the following commands 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):

   • **action** { **forward**}

   SwitchDevice(config-access-map)# **action forward**

   • **action** { **drop**}

   SwitchDevice(config-access-map)# **action drop**

4. **vlan filter** *mapname* **vlan-list** *list*

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**vlan access-map** *name* [**number**] **Example:** SwitchDevice(config)# **vlan access-map map_1 20**	Creates 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. 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. Entering this command changes to access-map configuration mode.
Step 2	**match** {**ip** | **mac**} **address** {*name* | *number*} [*name* | *number*] **Example:** SwitchDevice(config-access-map)# **match ip address ip2**	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. **Note** If the VLAN map is configured with a match clause for a type of packet (IP or MAC) and the map action is drop, all packets that match the type are dropped. If the VLAN map has no match clause, and the configured action is drop, all IP and Layer 2 packets are dropped.

Step 3	Enter one of the following commands 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):
	
	 • **action** { **forward**}
	
	   SwitchDevice(config-access-map)# **action forward**
	
	 • **action** { **drop**}
	
	   SwitchDevice(config-access-map)# **action drop**
		Sets the action for the map entry.

Step 4	**vlan filter** *mapname* **vlan-list** *list* **Example:**	Applies the VLAN map to one or more VLAN IDs.

Command or Action	Purpose
`SwitchDevice(config)# ` **`vlan filter map 1 vlan-list 20-22`**	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.

Related Topics

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:

SUMMARY STEPS

1. **configure terminal**
2. **vlan access-map** *name* [**number**]
3. **match** {**ip** | **mac**} **address** {*name* | *number*} [*name* | *number*]
4. **action** {**drop** | **forward**}
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `SwitchDevice# ` **`configure terminal`**	Enters global configuration mode.
Step 2	**vlan access-map** *name* [**number**] **Example:** `SwitchDevice(config)# ` **`vlan access-map map_1 20`**	Creates 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. 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

	Command or Action	Purpose
		drop. A permit in the ACL counts as a match. A deny in the ACL means no match.
		Entering this command changes to access-map configuration mode.
Step 3	**match** {**ip**	**mac**} **address** {*name*	*number*} [*name*	*number*] **Example:** SwitchDevice(config-access-map)# **match ip address ip2**	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 4	**action** {**drop**	**forward**} **Example:** SwitchDevice(config-access-map)# **action forward**	(Optional) Sets the action for the map entry. The default is to forward.
Step 5	**end** **Example:** SwitchDevice(config-access-map)# **end**	Returns to global configuration mode.
Step 6	**show running-config** **Example:** SwitchDevice# **show running-config**	Displays the access list configuration.
Step 7	**copy running-config startup-config** **Example:** SwitchDevice# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Related Topics

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:

SUMMARY STEPS

1. **configure terminal**
2. **vlan filter** *mapname* **vlan-list** *list*

3. **end**

4. **show running-config**

5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** SwitchDevice# **configure terminal**	Enters the global configuration mode.
Step 2	**vlan filter** *mapname* **vlan-list** *list* **Example:** SwitchDevice(config)# **vlan filter map 1 vlan-list 20-22**	Applies 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	**end** **Example:** SwitchDevice(config)# **end**	Returns to privileged EXEC mode.
Step 4	**show running-config** **Example:** SwitchDevice# **show running-config**	Displays the access list configuration.
Step 5	**copy running-config startup-config** **Example:** SwitchDevice# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Related Topics

Monitoring IPv4 ACLs

You can monitor IPv4 ACLs by displaying the ACLs that are configured on the switch, and displaying the ACLs that have been applied to interfaces and VLANs.

When you use the **ip access-group** interface configuration command to apply ACLs to a Layer 2 or 3 interface, you can display the access groups on the interface. You can also display the MAC ACLs applied to a Layer 2 interface. You can use the privileged EXEC commands as described in this table to display this information.

Table 2: Commands for Displaying Access Lists and Access Groups

Command	Purpose
show access-lists [*number* | *name*]	Displays 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 [*number* | *name*]	Displays the contents of all current IP access lists or a specific IP access list (numbered or named).
show ip interface *interface-id*	Displays 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** *interface-id*]	Displays the contents of the configuration file for the switch or the specified interface, including all configured MAC and IP access lists and which access groups are applied to an interface.
show mac access-group [**interface** *interface-id*]	Displays MAC access lists applied to all Layer 2 interfaces or the specified Layer 2 interface.

Configuration Examples for ACLs

Examples: Using Time Ranges with ACLs

This example shows how to verify after you configure time ranges for *workhours* and to configure January 1, 2006, as a company holiday.

```
SwitchDevice# show time-range
time-range entry: new_year_day_2003 (inactive)
   absolute start 00:00 01 January 2006 end 23:59 01 January 2006
time-range entry: workhours (inactive)
   periodic weekdays 8:00 to 12:00
   periodic weekdays 13:00 to 17:00
```

To apply a time range, enter the time-range 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 times and permits all TCP traffic during work hours.

```
SwitchDevice(config)# access-list 188 deny tcp any any time-range new_year_day_2006
SwitchDevice(config)# access-list 188 permit tcp any any time-range workhours
```

```
SwitchDevice(config)# end
SwitchDevice# show access-lists
Extended IP access list 188
   10 deny tcp any any time-range new_year_day_2006 (inactive)
   20 permit tcp any any time-range workhours (inactive)
```

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

```
SwitchDevice(config)# ip access-list extended deny_access
SwitchDevice(config-ext-nacl)# deny tcp any any time-range new_year_day_2006
SwitchDevice(config-ext-nacl)# exit
SwitchDevice(config)# ip access-list extended may_access
SwitchDevice(config-ext-nacl)# permit tcp any any time-range workhours
SwitchDevice(config-ext-nacl)# end
SwitchDevice# show ip access-lists
Extended IP access list lpip_default
   10 permit ip any any
Extended IP access list deny_access
   10 deny tcp any any time-range new_year_day_2006 (inactive)
Extended IP access list may_access
   10 permit tcp any any time-range workhours (inactive)
```

Examples: Including Comments 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.

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

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

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

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

IPv4 ACL Configuration Examples

This section provides examples of configuring and applying IPv4 ACLs. For detailed information about compiling ACLs, see the *Cisco IOS Security Configuration Guide, Release 12.4* and to the Configuring IP Services" section in the "IP Addressing and Services" chapter of the *Cisco IOS IP Configuration Guide, Release 12.4.*

ACLs in a Small Networked Office

Figure 3: Using Router ACLs to Control Traffic

This shows a small networked office environment with routed Port 2 connected to Server A, containing benefits and other information that all employees can access, and routed Port 1 connected to Server B, containing confidential payroll data. All users can access Server A, but Server B has restricted
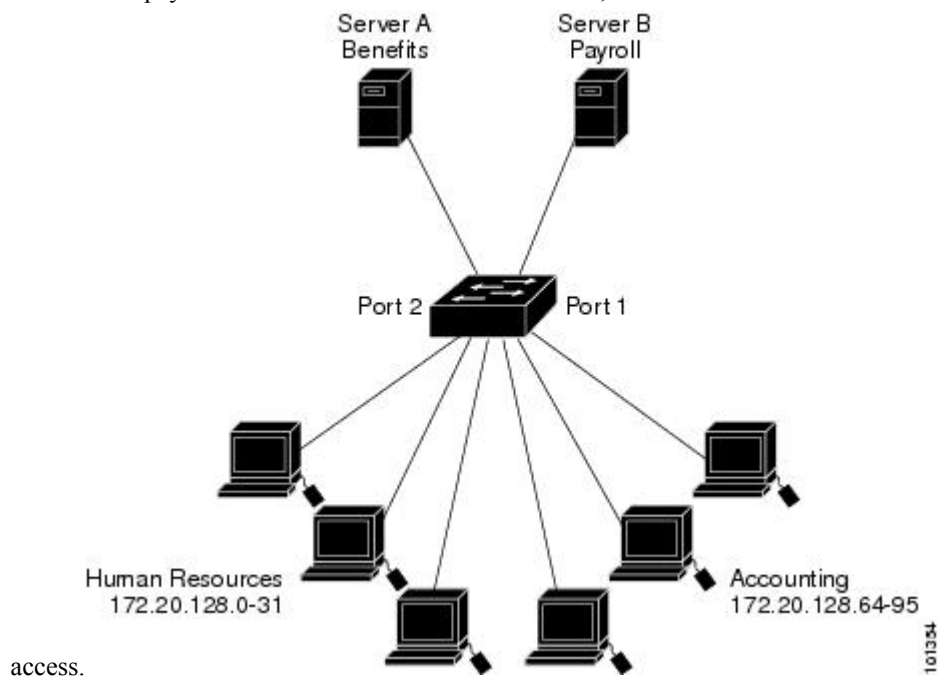
access.

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

- Create a standard ACL, and filter traffic coming to the server from Port 1.

- Create an extended ACL, and filter traffic coming from the server into Port 1.

Examples: ACLs in a Small Networked Office

This example uses a standard ACL to filter traffic coming into Server B from a port, permitting traffic only from Accounting's source addresses 172.20.128.64 to 172.20.128.95. The ACL is applied to traffic coming out of routed Port 1 from the specified source address.

```
SwitchDevice(config)# access-list 6 permit 172.20.128.64 0.0.0.31
SwitchDevice(config)# end
SwitchDevice# how access-lists
Standard IP access list 6
    10 permit 172.20.128.64, wildcard bits 0.0.0.31
```

```
SwitchDevice(config)# interface gigabitethernet1/0/1
SwitchDevice(config-if)# ip access-group 6 out
```

This example uses an extended ACL to filter traffic coming from Server B into a port, 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. The ACL is applied to traffic going into routed Port 1, 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.

```
SwitchDevice(config)# access-list 106 permit ip any 172.20.128.64 0.0.0.31
SwitchDevice(config)# end
SwitchDevice# show access-lists
Extended IP access list 106
    10 permit ip any 172.20.128.64 0.0.0.31
SwitchDevice(config)# interface gigabitethernet1/0/1
SwitchDevice(config-if)# ip access-group 106 in
```

Example: 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 applied to packets entering a port.

```
SwitchDevice(config)# access-list 2 permit 36.48.0.3
SwitchDevice(config)# access-list 2 deny 36.48.0.0 0.0.255.255
SwitchDevice(config)# access-list 2 permit 36.0.0.0 0.255.255.255
SwitchDevice(config)# interface gigabitethernet2/0/1
SwitchDevice(config-if)# ip access-group 2 in
```

Examples: 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.

```
SwitchDevice(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 gt 1023
SwitchDevice(config)# access-list 102 permit tcp any host 128.88.1.2 eq 25
SwitchDevice(config)# access-list 102 permit icmp any any
SwitchDevice(config)# interface gigabitethernet2/0/1
SwitchDevice(config-if)# ip access-group 102 in
```

In this example, 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 of the network 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.

```
SwitchDevice(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 eq 23
SwitchDevice(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 eq 25
SwitchDevice(config)# interface gigabitethernet1/0/1
SwitchDevice(config-if)# ip access-group 102 in
```

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 1 on stack member 1 is the interface that connects the router to the Internet.

```
SwitchDevice(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 established
SwitchDevice(config)# access-list 102 permit tcp any host 128.88.1.2 eq 25
SwitchDevice(config)# interface gigabitethernet1/0/1
SwitchDevice(config-if)# ip access-group 102 in
```

Examples: Named ACLs

Creating named standard and extended ACLs

This example 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.

```
SwitchDevice(config)# ip access-list standard Internet_filter
SwitchDevice(config-ext-nacl)# permit 1.2.3.4
SwitchDevice(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 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.

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

The *Internet_filter* ACL is applied to outgoing traffic and the *marketing_group* ACL is applied to incoming traffic on a Layer 3 port.

```
SwitchDevice(config)# interface gigabitethernet3/0/2
SwitchDevice(config-if)# no switchport
SwitchDevice(config-if)# ip address 2.0.5.1 255.255.255.0
SwitchDevice(config-if)# ip access-group Internet_filter out
SwitchDevice(config-if)# ip access-group marketing_group in
```

Deleting individual ACEs from named ACLs

This example shows how you can delete individual ACEs from the named access list *border-list*:

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

Examples: Time Range Applied to an IP ACL

This example denies HTTP traffic on IP on Monday through Friday between the hours of 8:00 a.m. and 6:00 p.m (18:00). The example allows UDP traffic only on Saturday and Sunday from noon to 8:00 p.m. (20:00).

```
SwitchDevice(config)# time-range no-http
SwitchDevice(config)# periodic weekdays 8:00 to 18:00
!
SwitchDevice(config)# time-range udp-yes
SwitchDevice(config)# periodic weekend 12:00 to 20:00
!
SwitchDevice(config)# ip access-list extended strict
SwitchDevice(config-ext-nacl)# deny tcp any any eq www time-range no-http
SwitchDevice(config-ext-nacl)# permit udp any any time-range udp-yes
!
SwitchDevice(config-ext-nacl)# exit
SwitchDevice(config)# interface gigabitethernet2/0/1
SwitchDevice(config-if)# ip access-group strict in
```

Examples: Configuring Commented IP ACL Entries

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

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

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

```
SwitchDevice(config)# ip access-list standard prevention
SwitchDevice(config-std-nacl)# remark Do not allow Jones subnet through
SwitchDevice(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:

```
SwitchDevice(config)# ip access-list extended telnetting
SwitchDevice(config-ext-nacl)# remark Do not allow Jones subnet to telnet out
```

```
SwitchDevice(config-ext-nacl)# deny tcp 171.69.0.0 0.0.255.255 any eq telnet
```

Examples: 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.

```
SwitchDevice(config)# ip access-list standard stan1
SwitchDevice(config-std-nacl)# deny 10.1.1.0 0.0.0.255 log
SwitchDevice(config-std-nacl)# permit any log
SwitchDevice(config-std-nacl)# exit
SwitchDevice(config)# interface gigabitethernet1/0/1
SwitchDevice(config-if)# ip access-group stan1 in
SwitchDevice(config-if)# end
SwitchDevice# 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
```

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.

```
SwitchDevice(config)# ip access-list extended ext1
SwitchDevice(config-ext-nacl)# permit icmp any 10.1.1.0 0.0.0.255 log
SwitchDevice(config-ext-nacl)# deny udp any any log
SwitchDevice(config-std-nacl)# exit
SwitchDevice(config)# interface gigabitethernet1/0/2
SwitchDevice(config-if)# ip access-group ext1 in
```

This is a 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
```

Configuration Examples for ACLs and VLAN Maps

Example: Creating an ACL and a VLAN Map to Deny a Packet

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.

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

Example: Creating an ACL and a VLAN Map to Permit a Packet

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. 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.

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

Example: Default Action of Dropping IP Packets and Forwarding MAC Packets

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

```
SwitchDevice(config)# access-list 101 permit udp any any
SwitchDevice(config)# ip access-list extended igmp-match
SwitchDevice(config-ext-nacl)# permit igmp any any


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

Example: Default Action of Dropping MAC Packets and Forwarding IP Packets

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

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

Example: Default Action of Dropping All Packets

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
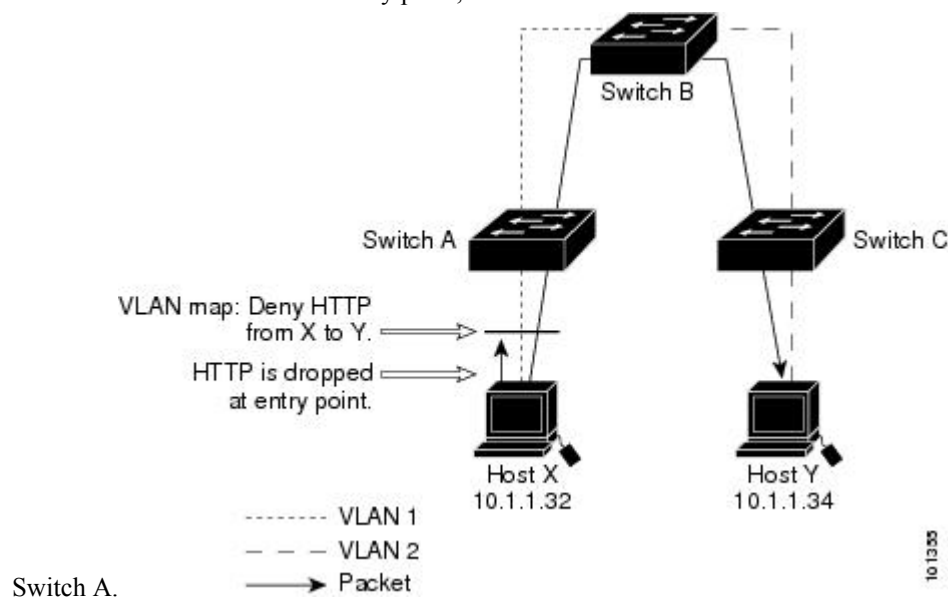
- Drop all other MAC packets

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

Configuration Examples for Using VLAN Maps in Your Network

Example: Wiring Closet Configuration

Figure 4: Wiring Closet Configuration

In a wiring closet configuration, routing might not be enabled on the switch. In this configuration, the switch can still support a VLAN map and a QoS classification ACL. 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, a Layer 3 switch with routing enabled. Traffic from Host X to Host Y can be access-controlled at the traffic entry point,

Switch A.

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.

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

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

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

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

Example: Restricting Access to a Server on Another VLAN

Figure 5: Restricting 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 denied to these hosts:

- 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.
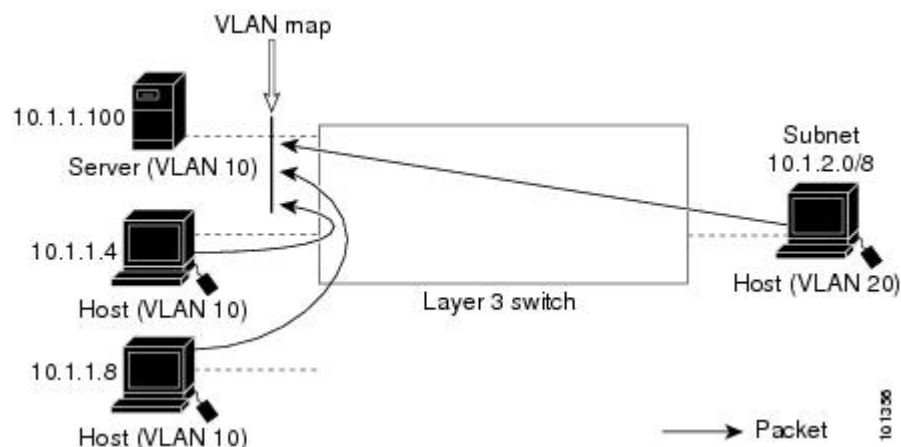
Example: Denying Access to a Server on Another VLAN

This example shows how to deny access to a server on another VLAN by creating the VLAN map SERVER 1 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.

Define the IP ACL that will match the correct packets.

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

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.

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

Apply the VLAN map to VLAN 10.

```
SwitchDevice(config)# vlan filter SERVER1_MAP vlan-list 10
```

Configuration 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.

Example: ACLs and Switched Packets

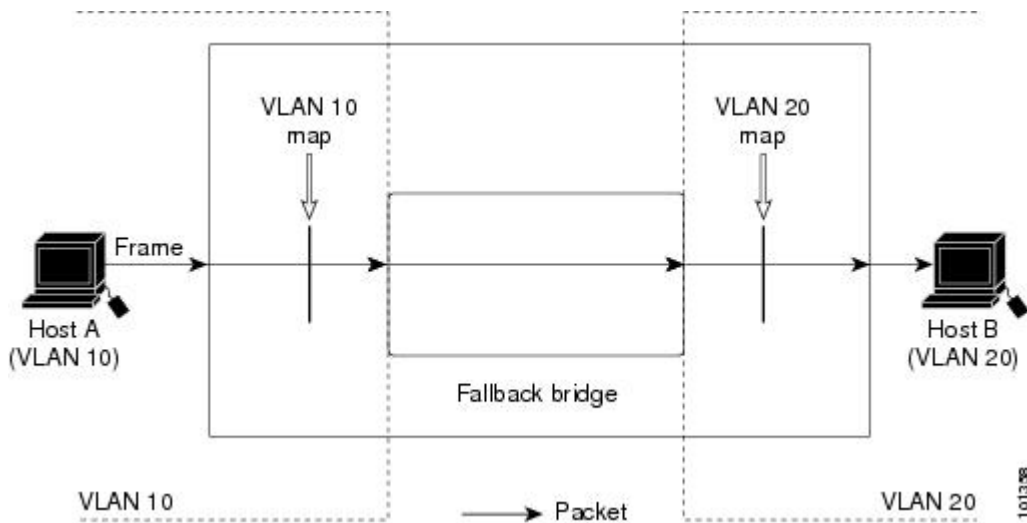### Figure 6: Applying ACLs on Switched Packets

This example 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.
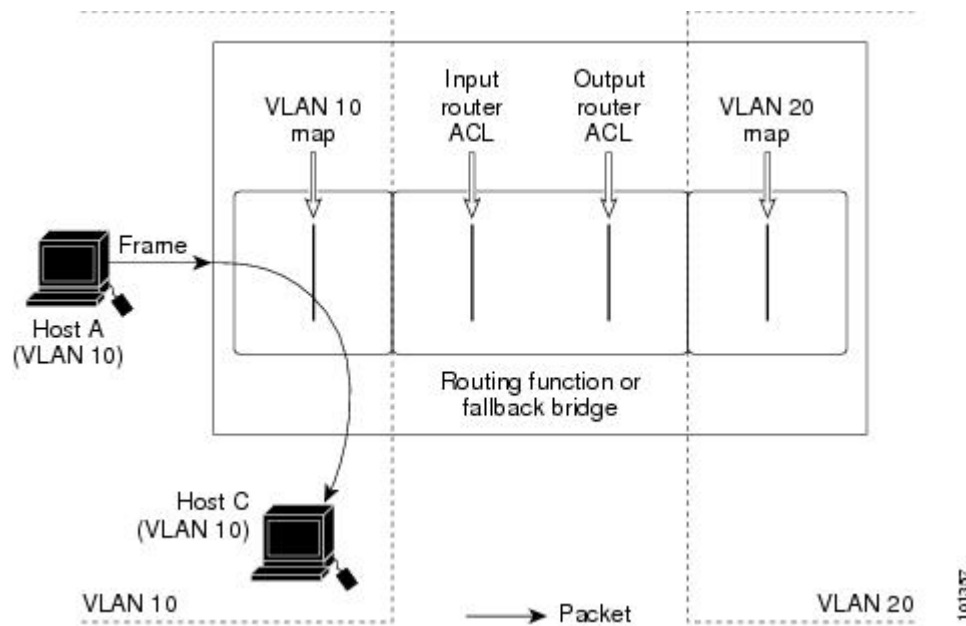
Example: ACLs and Bridged Packets

Figure 7: Applying ACLs on Bridged Packets

This example 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.
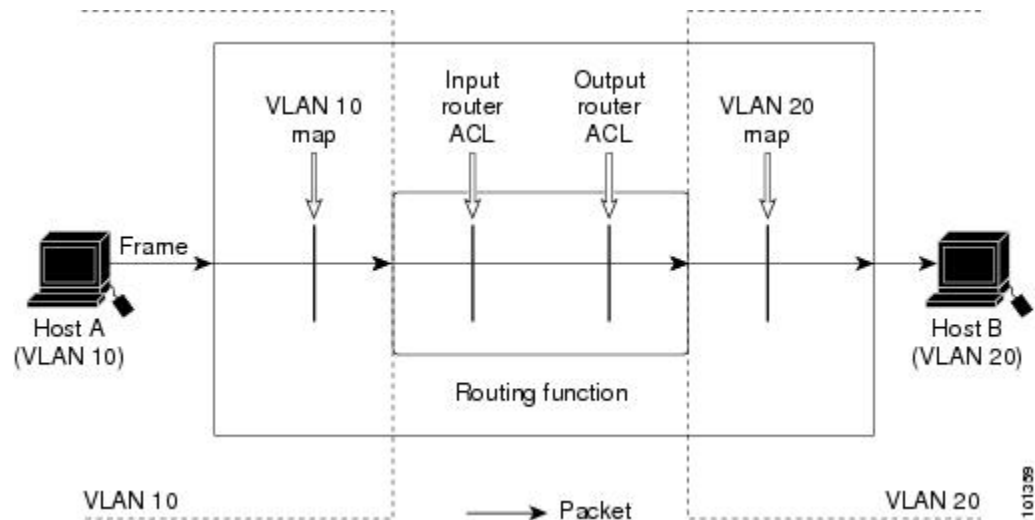

Example: ACLs and Routed Packets

Figure 8: Applying ACLs on Routed Packets

This example shows how ACLs are applied on 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

Example: ACLs and Multicast Packets

Figure 9: Applying ACLs on Multicast Packets

This example 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 drops the packet, no destination receives a copy of the packet.