CHAPTER 24

Managing Network Security with ACLs

This chapter describes how to manage network security on your switch.

For complete syntax and usage information for the commands used in this chapter, see the command reference for this release, see the "Configuring IP Services" section in the "IP Addressing and Services" chapter of the *Cisco IOS IP Configuration Guide, Release 12.2*, and the *Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services, Release 12.2*.

This chapter consists of these sections:

Understanding ACLs

Note On the Catalyst 2360 switch, security ACLs are applied to only the management VLAN.

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 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 interfaces. For example, you can allow e-mail traffic to be forwarded but not Telnet traffic.

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

The switch supports IP ACLs, including TCP, User Datagram Protocol (UDP), Internet Group Management Protocol (IGMP), and Internet Control Message Protocol (ICMP).

Handling Fragmented and Unfragmented Traffic

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

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

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

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

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

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

- Packet A is a TCP packet from host 10.2.2.2., port 65000, going to host 10.1.1.1 on the SMTP port. If this packet is fragmented, the first fragment matches the first ACE (a permit) as if it were a complete packet because all Layer 4 information is present. The remaining fragments also match the first ACE, even though they do not contain the SMTP port information, because the first ACE only checks Layer 3 information when applied to fragments. The information in this example is that the packet is TCP and that the destination is 10.1.1.1.
- Packet B is from host 10.2.2.2, port 65001, going to host 10.1.1.2 on the Telnet port. If this packet is fragmented, the first fragment matches the second ACE (a deny) because all Layer 3 and Layer 4 information is present. The remaining fragments in the packet do not match the second ACE because they are missing Layer 4 information. Instead, they match the third ACE (a permit).

  Because the first fragment was denied, host 10.1.1.2 cannot reassemble a complete packet, so packet B is effectively denied. However, the later fragments that are permitted will consume bandwidth on the network and resources of host 10.1.1.2 as it tries to reassemble the packet.
- Fragmented packet C is from host 10.2.2.2, port 65001, going to host 10.1.1.3, port ftp. If this packet is fragmented, the first fragment matches the fourth ACE (a deny). All other fragments also match the fourth ACE because that ACE does not check any Layer 4 information and because Layer 3 information in all fragments shows that they are being sent to host 10.1.1.3, and the earlier permit ACEs were checking different hosts.

Configuring IPv4 ACLs

Configuring IP v4ACLs on the switch is the same as configuring IPv4 ACLs on other Cisco switches and routers. The process is briefly described here. For more detailed information on configuring ACLs, see the "Configuring IP Services" section in the "IP Addressing and Services" chapter of the *Cisco IOS IP Configuration Guide, Release 12.2*. For detailed information about the commands, see the *Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services, Release 12.2*.

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

- Non-IP protocol ACLs (see Table 24-1 on page 24-4) or bridge-group ACLs
- IP accounting
- Inbound and outbound rate limiting (except with QoS ACLs)
- Reflexive ACLs or dynamic ACLs (except for some specialized dynamic ACLs used by the switch clustering feature)
- ACL logging

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

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

Step 2 Apply the ACL to the management VLAN. You can also apply standard and extended IP ACLs to VLAN maps.

These sections contain this configuration information:

Creating 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.

These sections describe access lists and how to create them:

Access List Numbers

The number you use to denote your ACL shows the type of access list that you are creating. Table 24-1 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 24-1 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
2000–2699	IP extended access list (expanded range)	Yes

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

Creating a Numbered Standard ACL

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

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]	Define 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. 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. **Note** Although visible in the command-line help, the **log** keyword is not supported.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show access-lists** [*number* | *name*]	Show the access list configuration.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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

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

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

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

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

After creating a numbered standard IPv4 ACL, you can apply it to terminal lines (see the "Applying an IPv4 ACL to a Terminal Line" section on page 24-12) or to management VLANs (see the "Applying an IPv4 ACL to a Management VLAN" section on page 24-13).

Creating a Numbered Extended ACL

Although standard ACLs use only source addresses for matching, you can use extended ACL source and destination addresses for matching operations and optional protocol type information for finer granularity of control. 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.

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

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

Authentication Header Protocol (**ahp**), Enhanced Interior Gateway Routing Protocol (**eigrp**), Encapsulation Security Payload (**esp**), generic routing encapsulation (**gre**), Internet Control Message Protocol (**icmp**), Internet Group Management Protocol (**igmp**), any Interior Protocol (**ip**), IP in IP tunneling (**ipinip**), KA9Q NOS-compatible IP over IP tunneling (**nos**), Open Shortest Path First routing (**ospf**), Payload Compression Protocol (**pcp**), Protocol-Independent Multicast (**pim**), Transmission Control Protocol (**tcp**), or User Datagram Protocol (**udp**).

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

For more details on the specific keywords for each protocol, see these command references:

- *Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services, Release 12.2*
- *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.2*
- *Cisco IOS IP Command Reference, Volume 3 of 3: Multicast, Release 12.2*

Note 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.

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

Note Although visible in the command-line help, the **log** keyword is not supported for the **access-list** command.

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

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**access-list** *access-list-number* {**deny** | **permit** | **remark**}*source-source-wildcard* {**any** | **host**} *destination destination-wildcard* [**log**]	Define 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. Enter **remark** to speecify a comment string. 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. 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. **Note** Although visible in the command-line help, the **log** keyword is not supported.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show access-lists** [*number* | *name*]	Verify the access list configuration.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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

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

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

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

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

After creating a numbered extended ACL, you can apply it to a management VLAN (see the "Applying an IPv4 ACL to a Terminal Line" section on page 24-12) or to interfaces (see the "Applying an IPv4 ACL to a Management VLAN" section on page 24-13).

Resequencing ACEs in an ACL

Sequence numbers for the entries in an access list are automatically generated when you create a new ACL. You can use the **ip access-list resequence** global configuration command to edit the sequence numbers in an ACL and change the order in which ACEs are applied. For example, if you add a new ACE to an ACL, it is placed at the bottom of the list. By changing the sequence number, you can move the ACE to a different position in the ACL.

Creating Named Standard and Extended 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 than if you were to use numbered access lists. If you identify your access list with a name rather than a number, the mode and command syntax are slightly different. However, not all commands that use IP access lists accept a named access list.

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

Consider these guidelines and limitations before configuring named ACLs:

- Not all commands that accept a numbered ACL accept a named ACL. ACLs for packet filters on interfaces can use a name.
- A standard ACL and an extended ACL cannot have the same name.
- Numbered ACLs are also available, as described in the "Creating Standard and Extended IPv4 ACLs" section on page 24-3.

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

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip access-list standard** *name*	Define a standard IPv4 access list using a name, and enter access-list configuration mode. The name can be a number from 1 to 99.

	Command	Purpose
Step 3	**deny** {*source* [*source-wildcard*] | **host** *source* | **any**} or **permit** {*source* [*source-wildcard*] | **host** *source* | **any**}	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. **Note** Although visible in the command-line help, the **log** keyword is not supported.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show access-lists** [*number* | *name*]	Show the access list configuration.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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

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

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip access-list extended** *name*	Define 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 3	{**deny** | **permit**} *protocol* {*source* [*source-wildcard*] | **host** *source* | **any**} {*destination* [*destination-wildcard*] | **host** *destination* | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**time-range** *time-range-name*]	In access-list configuration mode, specify the conditions allowed or denied. See the "Creating a Numbered Extended ACL" section on page 24-6 for definitions of protocols and other keywords. • **host** *source*—A source and source wildcard of *source* 0.0.0.0. • ***host*** *destination*—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. **Note** Although visible in the command-line help, the **log** keyword is not supported.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show access-lists** [*number* | *name*]	Show the access list configuration.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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

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

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

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

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

After creating a named ACL, you can apply it to management VLANs(see the "Applying an IPv4 ACL to a Management VLAN" section on page 24-13).

Using Time Ranges with 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 in the previous sections, the "Creating Standard and Extended IPv4 ACLs" section on page 24-3, and the "Creating Named Standard and Extended ACLs" section on page 24-8.

Using time ranges allows you 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).

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. For more information, see the "Managing the System Time and Date" section on page 5-1.

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

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**time-range** *time-range-name*	Assign 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.

	Command	Purpose
Step 3	**absolute** [**start** *time date*] [**end** *time date*] or **periodic** *day-of-the-week hh:mm to* [*day-of-the-week*] *hh:mm* or **periodic** {**weekdays** | **weekend** | **daily**} *hh:mm to hh:mm*	Specify 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 4	**end**	Return to privileged EXEC mode.
Step 5	**show time-range**	Verify the time-range configuration.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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

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

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

```
Switch(config)# time-range workhours
Switch(config-time-range)# periodic weekdays 8:00 to 12:00
Switch(config-time-range)# periodic weekdays 13:00 to 17:00
Switch(config-time-range)# exit
Switch(config)# time-range new_year_day_2006
Switch(config-time-range)# absolute start 00:00 1 Jan 2006 end 23:59 1 Jan 2006
Switch(config-time-range)# end
Switch# 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.

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

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

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:

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

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

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

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

Applying 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.

For procedures for applying ACLs to interfaces, see the "Applying an IPv4 ACL to a Management VLAN" section on page 24-13.

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

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**line** [**console** | **vty**] *line-number*	Identify a specific line to configure, and enter in-line configuration mode. • **console**—Specify the console terminal line. The console port is DCE. • **vty**—Specify 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 3	**access-class** *access-list-number* {**in** | **out**}	Restrict incoming and outgoing connections between a particular virtual terminal line (into a device) and the addresses in an access list.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show running-config**	Display the access list configuration.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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

Applying an IPv4 ACL to a Management VLAN

Note This section describes how to apply IPv4 ACLs to a management VLAN. By default, the router sends Internet Control Message Protocol (ICMP) unreachable messages when a packet is denied by an access group. These access-group denied packets are not dropped in hardware but are bridged to the switch CPU so that it can generate the ICMP-unreachable message.

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

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface vlan 1**	Configure management VLAN.
Step 3	**ip access-group** {*access-list-number* | *name*} **in**	Control access to the specified interface.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show running-config**	Display the access list configuration.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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

This example shows how to apply access list 2 to a port to filter packets entering the port:

```
Switch(config)# interface vlan 1
Switch(config-if)# ip access-group 2 in
```

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 12.2* and to the Configuring IP Services" section in the "IP Addressing and Services" chapter of the *Cisco IOS IP Configuration Guide, Release 12.2*.

This example uses a standard ACL to allow a port access to a specific Internet host with the address 172.20.128.64.

```
Switch(config)# access-list 6 permit 172.20.128.64 0.0.0
Switch(config)# end
Switch# show access-lists
Standard IP access list 6
    10 permit 172.20.128.64, wildcard bits 0.0.0.0
Switch(config)# interface vlan 1
Switch(config-if)# ip access-group 6 in
```

This example uses an extended ACL to deny to a port traffic coming from port 80 (HTTP). It permits all other types of traffic.

```
Switch(config)# access-list 106 deny tcp any any eq 80
Switch(config)# access-list 106 permit ip any any
Switch(config)# end
Switch(config)# interface vlan 1
Switch(config-if)# ip access-group 106 in
```

Numbered ACLs

This ACL accepts addresses on network 36.0.0.0 subnets and denies all packets coming from 56.0.0.0 subnets. The ACL is applied to packets entering a port.

```
Switch(config)# access-list 2 permit 36.0.0.0 255.255.255.255
Switch(config)# access-list 2 deny 56.0.0 0.0 255.255.255.255
Switch(config)# interface vlan 1
Switch(config-if)# ip access-group 2 in
```

Extended ACLs

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. Because the secure system of the network al

ways accepts mail connections on port 25, the incoming services are controlled.

```
Switch(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 eq 23
Switch(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 eq 25
Switch(config)# interface vlan 1
Switch(config-if)# ip access-group 102 in
```

Named ACL

This example creates an extended ACL named *marketing_group*. The *marketing_group* ACL allows any TCP Telnet traffic to the destination address and wildcard 171.69.0.0 0.0.255.255 and denies any other TCP traffic. It permits any other IP traffic.

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

The *marketing_group* ACL is applied to incoming traffic on a port.

```
Switch(config)# interface vlan 1
Switch(config-if)# ip access-group marketing_group in
```

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).

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

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:

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

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

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

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

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

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

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

Displaying IPv4 ACL Configuration

When you use the **ip access-group** interface configuration command to apply ACLs, you can display the access groups applied on the management VLAN. You can use the privileged EXEC commands as described in Table 24-2 to display this information.

Table 24-2 Commands for Displaying Access Lists and Access Groups

Command	Purpose
show access-lists [*number* | *name*]	Display the contents of one or all current IP and MAC address access lists or a specific access list (numbered or named).
show access-lists [*number* | *name*]	Display the contents of all current IP access lists or a specific IP access list (numbered or named).
show interface vlan *vlan-number*	Display detailed configuration and status of the VLANs.
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.