9

Access Control Lists (ACLs) for the Series 5300xl Switches

Contents

Introduction 9-3

Terminology 9-5

Overview 9-8

Types of IP ACLs 9-8

ACL Inbound and Outbound Application Points 9-8

Features Common to All per-VLAN ACLs 9-10

General Steps for Planning and Configuring ACLs 9-10

ACL Operation 9-12

Introduction 9-12

The Packet-Filtering Process 9-13

Planning an ACL Application 9-16

Traffic Management and Improved Network Performance 9-16

Security 9-17

Guidelines for Planning the Structure of an ACL 9-18

ACL Configuration and Operating Rules 9-18

How an ACE Uses a Mask To Screen Packets for Matches 9-20

What Is the Difference Between Network (or Subnet) Masks and the Masks Used with ACLs? 9-20

Rules for Defining a Match Between a Packet and an Access Control Entry (ACE) 9-21

Configuring and Assigning an ACL 9-25

Overview 9-25

General Steps for Implementing ACLs 9-25

Types of ACLs 9-26

ACL Configuration Structure 9-26

Standard ACL Structure 9-27

Extended ACL Configuration Structure 9-28
ACL Configuration Factors 9-29
The Sequence of Entries in an ACL Is Significant 9-29
In Any ACL, There Will Always Be a Match 9-31
A Configured ACL Has No Effect Until You Apply It to an Interface 9-31
You Can Assign an ACL Name or Number to a VLAN Even if the ACL Does Not Yet Exist in the Switch's Configuration . . 9-31
Using the CLI To Create an ACL 9-31
General ACE Rules 9-32
Using CIDR Notation To Enter the ACL Mask 9-32
Configuring and Assigning a Numbered, Standard ACL 9-33
Configuring and Assigning a Numbered, Extended ACL 9-38
Configuring a Named ACL 9-44
Enabling or Disabling ACL Filtering on a VLAN 9-46

Deleting an ACL from the Switch 9-47

Displaying ACL Data 9-48
Display an ACL Summary 9-48
Display the Content of All ACLs on the Switch 9-49
Display the ACL Assignments for a VLAN 9-50
Displaying the Content of a Specific ACL 9-50
Display All ACLs and Their Assignments in the Switch Startup-Config File and Running-Config File 9-52

Editing ACLs and Creating an ACL Offline 9-53
Using the CLI To Edit ACLs 9-53
General Editing Rules 9-53
Deleting Any ACE from an ACL 9-54
Working Offline To Create or Edit an ACL 9-55
Creating an ACL Offline 9-56

Enable ACL "Deny" Logging 9-59
Requirements for Using ACL Logging 9-59
ACL Logging Operation 9-59
Enabling ACL Logging on the Switch 9-60
Operating Notes for ACL Logging 9-62

General ACL Operating Notes 9-63

Introduction

This chapter applies only to the Series 5300xl Switches. For ACL operation on Series 3400cl and Series 6400clswitches, refer to the chapter 10, "Access Control Lists (ACLs) for the Series 3400cl and Series 6400cl Switches".

Feature	Default	Menu	CLI	Web
Numbered ACLs				
Standard ACLs	None	—	9-33	—
Extended ACLs	None	—	9-38	—
Named ACLs		—	9-44	—
Enable or Disable an ACL		—	9-46	—
Display ACL Data	n/a	—	9-48	—
Delete an ACL	n/a	—	9-47	—
Configure an ACL from a TFTP Server	n/a	—	9-55	—
Enable ACL Logging	n/a	—	9-60	—

Layer 3 IP filtering with ACLs on the Series 5300XL switches can help improve network performance and restrict network use by creating policies for:

- **Switch Management Access:** Permits or denies in-band management access. This includes preventing the use of certain TCP or UDP applications (such as Telnet, SSH, web browser, and SNMP) for transactions between specific source and destination IP addresses.)
- **Application Access Security:** Eliminates unwanted IP, TCP, or UDP traffic in a path by filtering packets where they enter or leave the switch on specific VLAN interfaces.

ACLs on the 5300xl switches can filter traffic to or from a host, a group of hosts, or entire subnets.

This chapter describes how to configure, apply, and edit ACLs in a network populated with ProCurve Series 5300XL switches (with IP routing support enabled) and how to monitor the results of ACL actions.

Notes

ACLs can enhance network security by blocking selected IP traffic, and can serve as part of your network security program. *However, because ACLs do not provide user or device authentication, or protection from malicious manipulation of data carried in IP packet transmissions, they should not be relied upon for a complete security solution.*

Series 5300XL ACLs do not screen non-IP traffic such as AppleTalk and IPX.

For ACL filtering to take effect, configure an ACL and then assign it to either the inbound or outbound traffic on a statically configured VLAN on the switch. (Except for ACEs that screen traffic to an IP address on the switch itself, ACLs assigned to VLANs can operate only while IP routing is enabled. Refer to "Notes on IP Routing" on page 9-11.)

Table 9-1. Comprehensive Command Summary

Action	Command	Page
Configuring Standard (Numbered) ACLs	ProCurve(config)# [no] access-list < 1-99 > < deny | permit > < any | host <*src-ip-addr* > | *src-ip-address/mask* > [1] [log][2]	9-33
Configuring Extended (Numbered) ACLs	ProCurve(config)# [no] access-list <100-199> < deny | permit > ip < any | host <*src-ip-addr* > | *src-ip-address*/mask > [1] [log][2] ProCurve(config)# [no] access-list < 100-199 > < deny | permit > < tcp | udp > < any | host <*src-ip-addr* > | *src-ip-address*/mask > [1] [operator < *src-port tcp/udp-id* >] < any | host <*dest-ip-addr* > | *dest-ip-address*/mask > [1] [operator < *dest-port tcp/udp-id* >] [log][2]	9-38
Configuring Standard (Named) ACLs	ProCurve(config)# [no] ip access-list standard < *name-str* | 1-99 > ProCurve(config-std-nacl)# < deny | permit > < any | host <*src-ip-addr* > | *src-ip-address/mask* > [1] [log] [2]	9-44
Configuring Extended (Named) ACLs	ProCurve(config)# [no] ip access-list extended < *name-str* | 100-199 > ProCurve(config-std-nacl)# < deny | permit > ip < any | host <*src-ip-addr* > | *src-ip-address*/mask >[1] < any | host <*dest-ip-addr* > | *dest-ip-address*/mask >[1] [log][2] ProCurve(config-std-nacl)# < deny | permit > < tcp | udp > < any | host <*src-ip-addr* > | *src-ip-address*/mask > [1] [operator < *src-port tcp/udp-id* >] < any | host <*dest-ip-addr* > | *dest-ip-address*/mask > [1] [operator < *dest-port tcp/udp-id* >] [log][2]	
Enabling or Disabling an ACL	ProCurve(config)# [no] vlan < *vid* > ip access-group < *name-str* | 1-99 | 100-199 > < in/out >	9-46

[1] The mask can be in either dotted-decimal notation (such as 0.0.15.255) or CIDR notation (such as /20).
[2] The [log] function applies only to "deny" ACLs, and generates a message only when there is a "deny" match.

Action	Command	Page
Deleting an ACL from the Switch	ProCurve(config)# no ip access-list < standard | extended > < *name-str* | 1-99 | 100 -199 > < in | out >	9-47
Displaying ACL Data	ProCurve(config)# show access-list ProCurve(config)# show access-list config ProCurve(config)# show access-list vlan < *vid* > ProCurve(config)# show config ProCurve(config)# show running	9-48

Terminology

Access Control Entry (ACE): An ACE is a policy consisting of criteria and an action to take (permit or deny) on a packet if it meets the criteria. The elements composing the criteria include:

- Source IP address and mask (standard and extended ACLs)
- Destination IP address and mask (extended ACLs only)
- TCP or UDP application port numbers (optional, extended ACLs only)

Access Control List (ACL): A list (or set) consisting of one or more explicitly configured Access Control Entries (ACEs) and terminating with an implicit "deny" default which drops any packets that do not have a match with any explicit ACE in the named ACL. The two classes of ACLs are "standard" and "extended". See "Standard ACL" and "Extended ACL".

ACE: See "Access Control Entry".

ACL: See "Access Control List".

ACL ID: A number or alphanumeric string used to identify an ACL. A *standard* ACL ID can have either a number from 1 to 99 or an alphanumeric string. An *extended* ACL ID can have either a number from 100 to 199 or an alphanumeric string.

ACL Mask: Follows any IP address (source or destination) listed in an ACE. Defines which bits in a packet's corresponding IP addressing must exactly match the IP addressing in the ACE, and which bits need not match (wildcards). See also "How an ACE Uses a Mask To Screen Packets for Matches" on page 9-20.)

Connection-Rate ACL: An optional feature used with Connection-Rate filtering based on virus-throttling technology, and available in 5300xl switches running software release E.09.*xx* or greater. For more information, refer to the chapter titled "Virus Throttling" in the Access Security Guide for your 5300xl switch.

DA: The acronym for *Destination IP Address*. In an IP packet, this is the destination IP address carried in the header, and identifies the destination intended by the packet's originator. In an extended ACE, this is the second of two IP addresses required by the ACE to determine whether there is a match between a packet and the ACE. See also "SA".

Deny: An ACE configured with this action causes the switch to drop a packet for which there is a match within an applicable ACL.

Extended ACL: This type of Access Control List uses layer-3 IP criteria composed of source and destination IP addresses and (optionally) TCP or UDP port criteria to determine whether there is a match with an IP packet. You can apply extended ACLs to either inbound or outbound routed traffic and to any inbound switched or routed traffic with a DA belonging to the switch itself. Extended ACLs require an identification number (ID) in the range of 100 - 199 or an alphanumeric name.

Implicit Deny: If the switch finds no matches between a routed packet and the configured criteria in an applicable ACL, then the switch denies (drops) the packet with an implicit "deny IP any" operation. You can preempt the implicit "deny IP any" in a given ACL by configuring **permit IP any** (standard) or **permit IP any any** (extended) as the last explicit ACE in the ACL. Doing so permits any routed packet that is not explicitly permitted or denied by other ACEs configured sequentially earlier in the ACL. Unless otherwise noted, "implicit deny IP any" refers to the "deny" action enforced by both standard and extended ACLs.

Inbound Traffic: For the purpose of defining where the switch applies ACLs to filter traffic, inbound traffic is any IP packet that:

- *Enters the switch* on a given subnet.
- Has a destination IP address (DA) that meets either of these criteria:
  – The packet's DA is for an external device on a different subnet.

- The packet's DA is for an IP address configured on the switch itself. (This increases your options for protecting the switch from unauthorized management access.)

Because ACLs are assigned to VLANs, an ACL that filters inbound traffic on a particular VLAN examines packets meeting the above criteria that have entered the switch through any port on that VLAN.

Outbound Traffic: For defining the points where the switch applies ACLs to filter traffic, outbound traffic is routed traffic *leaving the switch* through a physical port; that is, traffic received on a port in one VLAN (subnet) and sent through a port on another VLAN to another device. This requires that you enable IP routing on the switch. The switch does not apply ACLs internally where routed traffic moves between VLANs. Note that for ACL purposes, "outbound traffic" does not include traffic received on one port and switched to the outbound queue of another port on the same VLAN (subnet); that is, traffic arriving on and leaving the switch on the same VLAN. (Refer also to "ACL Inbound and Outbound Application Points" on page 9-8.)

Permit: An ACE configured with this action allows the switch to forward a routed packet for which there is a match within an applicable ACL.

SA: The acronym for *Source IP Address*. In an IP packet, this is the source IP address carried in the IP header, and identifies the packet's sender. In an extended ACE, this is the first of two IP addresses used by the ACE to determine whether there is a match between a packet and the ACE. See also "DA".

Standard ACL: This type of Access Control List uses layer-3 IP criteria of source IP address to determine whether there is a match with an IP packet. You can apply standard ACLs to either inbound or outbound routed traffic and to any inbound switched or routed traffic with a DA belonging to the switch itself. Standard ACLs require an identification number (ID) in the range of 100 - 199 or an alphanumeric name.

Wildcard: The part of a mask that indicates the bits in a packet's IP addressing that do not need to match the corresponding bits specified in an ACL. See also **ACL Mask** on page 9-6.

Overview

Types of IP ACLs

Standard ACL: Use a standard ACL when you need to permit or deny traffic based on source IP address only. Standard ACLs are also useful when you need to quickly control a performance problem by limiting traffic from a subnet, group of devices, or a single device. (This can block all IP traffic from the configured source, but does not hamper traffic from other sources within the network.) This ACL type uses a numeric ID of 1 through 99 or an alphanumeric ID string. You can specify a single host, a finite group of hosts, or any host.

Extended ACL: Extended ACLs are useful whenever simple IP source address restrictions do not provide the breadth of traffic selection criteria you want to exercise on a VLAN interface. Extended ACLs allow use of the following criteria:

- Source and destination IP addresses
- TCP application criteria
- UDP application criteria

Connection-Rate ACL. An optional feature used with Connection-Rate filtering based on virus-throttling technology, and available in 5300xl switches running software release E.09.*xx* or greater. For more information, refer to the chapter titled “Virus Throttling” in the Access Security Guide for your 5300xl switch.

ACL Inbound and Outbound Application Points

You can apply ACL filtering to the following types of traffic:

- IP traffic routed between different subnets. (IP routing *must* be enabled.)
- IP traffic carrying a destination address (DA) on the switch itself. In figure 9-1, below, this is any of the IP addresses shown in VLANs “A”, “B”, and “C” on the switch. (IP routing need not be enabled.)

The switch can apply ACL filtering to traffic *entering or leaving the switch* on VLANs configured to apply ACL filters. (When you assign an ACL to a VLAN, you must specify whether the ACL will filter inbound or outbound traffic.) For example, in figure 9-1:

- You would assign either an inbound ACL on VLAN "A" or an outbound ACL on VLAN "B" to filter a packet routed between subnets; that is, from the workstation 18.28.10.5 on VLAN "A" to the server at 18.28.20.99 on VLAN "B". (An outbound ACL on VLAN "A" or an inbound ACL on VLAN "B" would not filter the packet.)
- Where multiple subnets are configured on the same VLAN, *if*:
  - Traffic you want to filter moves between subnets on the same VLAN.
  - The traffic source and destination IP addresses are on devices external to the switch.

  Then you can use either inbound or outbound ACLs to filter the traffic on the VLAN (because the traffic moves between subnets but enters and leaves the switch in the same VLAN.)

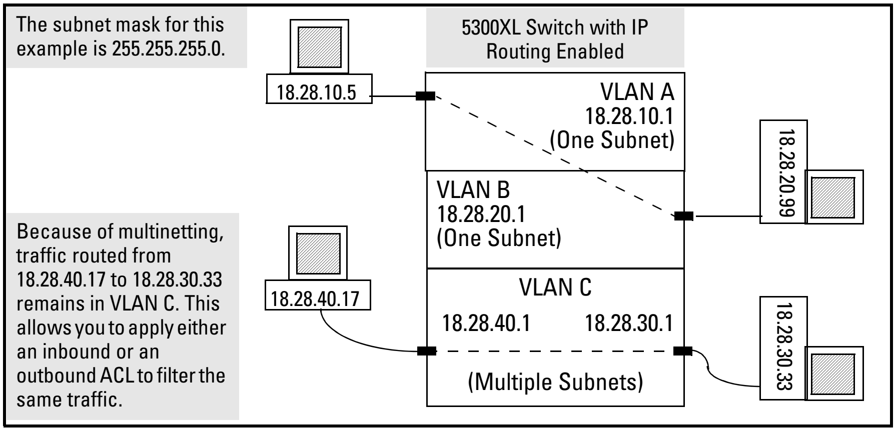

The subnet mask for this example is 255.255.255.0.
5300XL Switch with IP Routing Enabled
18.28.10.5
VLAN A
18.28.10.1
(One Subnet)
18.28.20.99
VLAN B
18.28.20.1
(One Subnet)
Because of multinetting, traffic routed from 18.28.40.17 to 18.28.30.33 remains in VLAN C. This allows you to apply either an inbound or an outbound ACL to filter the same traffic.
18.28.40.17
VLAN C
18.28.40.1
18.28.30.1
(Multiple Subnets)
18.28.30.33

Figure 9-1. Example of Filter Applications

Note

ACLs do not filter traffic that remains in the same subnet from source to destination (switched traffic) unless the destination IP address (DA) is on the switch itself.

Features Common to All per-VLAN ACLs

- On any VLAN you can apply one ACL to inbound traffic and one ACL to outbound traffic. You can use the same ACL or different ACLs for the inbound and outbound traffic.
- Any ACL can have multiple entries (ACEs).
- You can apply any one ACL to multiple VLANs.
- A source or destination IP address and a mask, together, can define a single host, a range of hosts, or all hosts.
- The IP address(es) assigned to a VLAN must not be configured from a DHCP server.
- Every standard ACL includes an implied "**deny IP any**" as the last entry, and every extended ACL includes an implied "**deny IP any any**" as the last entry. The switch applies this action to any packets that do not match other criteria in the ACL.
- In any ACL, you can apply an ACL log function to ACEs that have a "deny" action. The logging occurs when there is a match on a "deny" ACE. (The switch sends ACL logging output to Syslog and, optionally, to a console session.)

You can configure ACLs using either the CLI or a text editor. The text-editor method is recommended when you plan to create or modify an ACL that has more entries than you can easily enter or edit using the CLI alone. Refer to "Editing ACLs and Creating an ACL Offline" on page 9-53.

General Steps for Planning and Configuring ACLs

1. Identify the traffic type to filter. Options include:
   - Any routed IP traffic
   - Routed TCP traffic only
   - Routed UDP traffic only
2. The SA and/or the DA of routed traffic you want to permit or deny.
3. Determine the best points at which to apply specific ACL controls. For example, you can improve network performance by filtering unwanted traffic at the edge of the network instead of in the core. Also, on the switch itself, you can improve performance by filtering unwanted traffic where it is inbound to the switch instead of outbound.

4. Design the ACLs for the control points you have selected. Where you are using explicit “deny” ACEs, you can optionally use the ACL logging feature to help verify that the switch is denying unwanted packets where intended. Remember that excessive ACL logging activity can degrade the switch's performance. (Refer to “Enable ACL “Deny” Logging” on page 9-59.)
5. Create the ACLs in the selected switches.
6. Assign the ACLs to filter the inbound and/or outbound traffic on static VLAN interfaces configured on the switch.
7. Enable IP routing on the switch. (Except for an ACL configured to filter traffic having the switch itself as the destination IP address, IP routing must be enabled before ACLs will operate.)
8. Test for desired results.

For more details on ACL planning considerations, refer to “Planning an ACL Application” on page 9-16.

Notes on IP Routing

To activate an ACL to screen inbound traffic for routing between subnets, assign the ACL to the statically configured VLAN on which the traffic enters the switch. Also, ensure that IP routing is enabled. Similarly, to activate an ACL to screen routed, outbound traffic, assign the ACL to the statically configured VLAN on which the traffic exits from the switch. The only exception to these rules is for an ACL configured to screen inbound traffic with a destination IP address on the switch. In this case, an ACL assigned to a VLAN screens traffic addressed to an IP address on the switch, regardless of whether IP routing is also enabled. (ACLs do not screen outbound traffic generated by the switch, itself. Refer to “ACL Screening of Traffic Generated by the Switch” on page 9-63.)

Caution Regarding the Use of Source Routing

Source routing is enabled by default on the switch and can be used to override ACLs. For this reason, if you are using ACLs to enhance network security, the recommended action is to use the **no ip source-route** command to disable source routing on the switch. (If source routing is disabled in the running-config file, the **show running** command includes “**no ip source-route**” in the running-config file listing.)

ACL Operation

Introduction

An ACL is a list of one or more Access Control Entries (ACEs), where each ACE consists of a matching criteria and an action (permit or deny). An ACL applies only to the switch in which it is configured. ACLs operate on assigned static VLANs, and filter these traffic types:

- Routed traffic entering or leaving the switch on a VLAN. (Note that ACLs do not screen traffic at the internal point where traffic moves between VLANs or subnets within the switch. Refer to "ACL Inbound and Outbound Application Points" on page 9-8.)
- Switched or routed traffic entering the switch on a VLAN and having an IP address on the switch as the destination

You can apply one inbound ACL and one outbound ACL to each static VLAN configured on the switch. The complete range of options per VLAN includes:

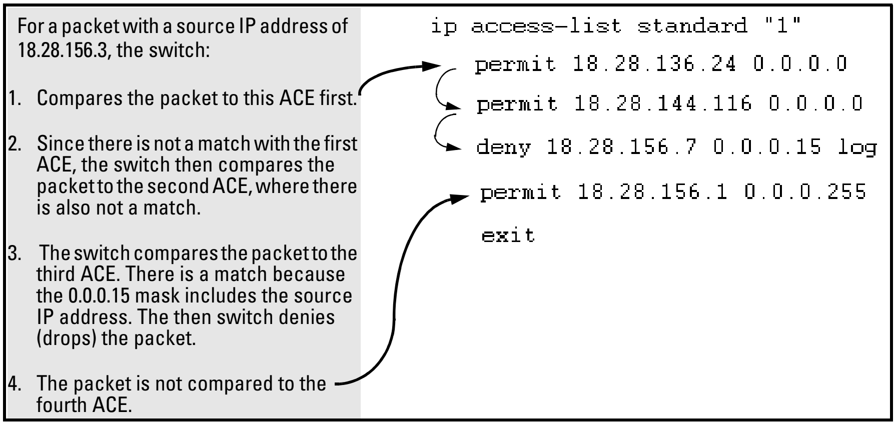
- **No ACL** assigned to a static VLAN. (In this case, all traffic entering or leaving the switch on the VLAN does so without any ACL filtering, which is the default.)
- **One ACL** assigned to filter *either* the inbound or the outbound traffic entering or leaving the switch on a static VLAN.
- **One ACL** assigned to filter *both* the inbound and the outbound traffic entering or leaving the switch on a static VLAN.
- **Two different ACLs** assigned to a static VLAN; one for filtering traffic entering the switch and one for filtering traffic leaving the switch.

Note

On a given switch, after you assign an ACL to a static VLAN, the default action for all physical ports belonging to the VLAN is to deny any IP traffic that is not specifically permitted by the ACL. (This applies only in the direction of traffic flow filtered by the ACL.)

The Packet-Filtering Process

Sequential Comparison and Action. When the switch uses an ACL to filter a packet, it sequentially compares each ACE's filtering criteria to the corresponding data in the packet until it finds a match.



For a packet with a source IP address of 18.28.156.3, the switch:

1. Compares the packet to this ACE first.
2. Since there is not a match with the first ACE, the switch then compares the packet to the second ACE, where there is also not a match.
3. The switch compares the packet to the third ACE. There is a match because the 0.0.0.15 mask includes the source IP address. The then switch denies (drops) the packet.
4. The packet is not compared to the fourth ACE.

```
ip access-list standard "1"
   permit 18.28.136.24 0.0.0.0
   permit 18.28.144.116 0.0.0.0
   deny 18.28.156.7 0.0.0.15 log
   permit 18.28.156.1 0.0.0.255
   exit
```

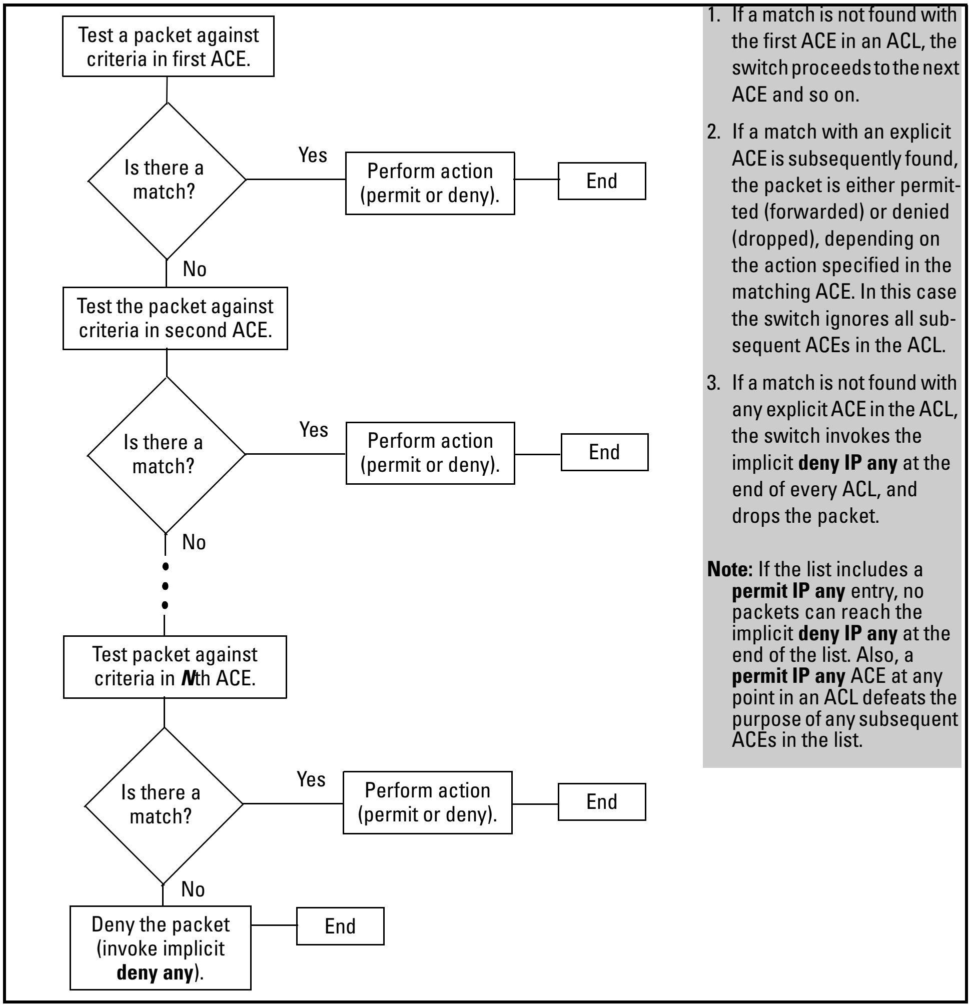
Figure 9-2. Example of Sequential Comparison

That is, the switch tries the first ACE in the list. If there is not a match, it tries the second ACE, and so on. When a match is found, the switch invokes the configured action for that entry (permit or drop the packet) and no further comparisons of the packet are made with the remaining ACEs in the ACL. This means that when the switch finds an ACE whose criteria matches a packet, it invokes the action configured for that ACE, and any remaining ACEs in the ACL are ignored. *Because of this sequential processing, successfully implementing an ACL depends in part on configuring ACEs in the correct order for the overall policy you want the ACL to enforce.*

Implicit Deny. If a packet does not have a match with the criteria in any of the ACEs in the ACL, the switch denies (drops) the packet. (This is termed *implicit deny.*) If you need to override the implicit deny so that any packet that does not have a match will be permitted, then you can enter **permit any** as the last ACE in the ACL. This directs the switch to permit (forward) any packets that do not have a match with any earlier ACE listed in the ACL, and prevents these packets from being filtered by the implicit deny.

Note on Implicit Deny

For ACLs configured to filter inbound packets on a VLAN, remember that Implicit Deny filters routed packets *and any bridged packets with a DA specifying the switch itself*. This operation helps to prevent management access from unauthorized IP sources.


Test a packet against criteria in first ACE.
Is there a match?
Yes
Perform action (permit or deny).
End
No
Test the packet against criteria in second ACE.
Is there a match?
Yes
Perform action (permit or deny).
End
No
Test packet against criteria in *N*th ACE.
Is there a match?
Yes
Perform action (permit or deny).
End
No
Deny the packet (invoke implicit **deny any**).
End

1. If a match is not found with the first ACE in an ACL, the switch proceeds to the next ACE and so on.
2. If a match with an explicit ACE is subsequently found, the packet is either permitted (forwarded) or denied (dropped), depending on the action specified in the matching ACE. In this case the switch ignores all subsequent ACEs in the ACL.
3. If a match is not found with any explicit ACE in the ACL, the switch invokes the implicit **deny IP any** at the end of every ACL, and drops the packet.

Note: If the list includes a **permit IP any** entry, no packets can reach the implicit **deny IP any** at the end of the list. Also, a **permit IP any** ACE at any point in an ACL defeats the purpose of any subsequent ACEs in the list.

Figure 9-3. The Packet-Filtering Process in an ACL with *N* Entries (ACEs)

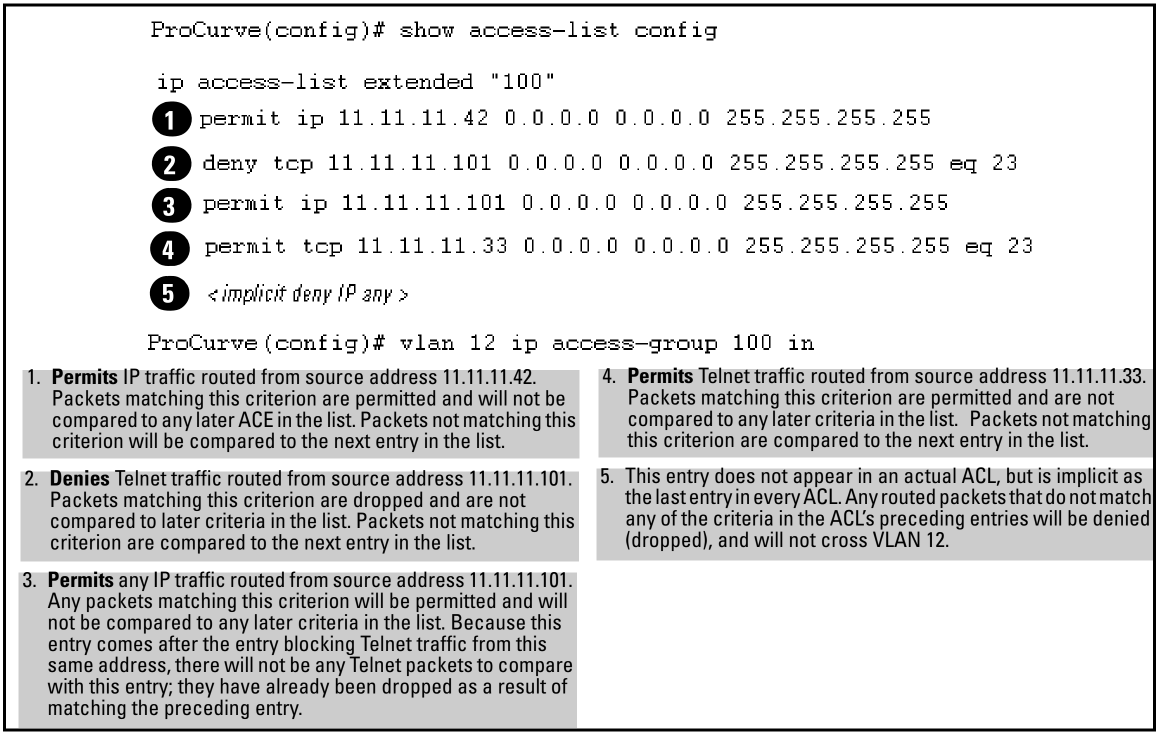
Note

The order in which an ACE occurs in an ACL is significant. For example, if an ACL contains six ACEs, but the first ACE is a “permit IP any”, then the ACL permits all IP traffic, and the remaining ACEs in the list do not apply, even if they specify criteria that would make a match with any of the traffic permitted by the first ACE.

For example, suppose you want to configure an ACL on the switch (with an ID of “100”) to invoke these policies:

1. Permit all inbound traffic on VLAN 12 routed from IP address 11.11.11.42.
2. Deny *only* the inbound Telnet traffic routed from address 11.11.11.101.
3. Permit *only* inbound Telnet traffic routed from IP address 11.11.11.33.
4. Deny *all other* inbound routed traffic on VLAN 12.

The following ACL model, when assigned to inbound filtering on VLAN 12, supports the above case:



```
ProCurve(config)# show access-list config

 ip access-list extended "100"
1 permit ip 11.11.11.42 0.0.0.0 0.0.0.0 255.255.255.255
2 deny tcp 11.11.11.101 0.0.0.0 0.0.0.0 255.255.255.255 eq 23
3 permit ip 11.11.11.101 0.0.0.0 0.0.0.0 255.255.255.255
4 permit tcp 11.11.11.33 0.0.0.0 0.0.0.0 255.255.255.255 eq 23
5 < implicit deny IP any >

ProCurve(config)# vlan 12 ip access-group 100 in
```

1. **Permits** IP traffic routed from source address 11.11.11.42. Packets matching this criterion are permitted and will not be compared to any later ACE in the list. Packets not matching this criterion will be compared to the next entry in the list.
2. **Denies** Telnet traffic routed from source address 11.11.11.101. Packets matching this criterion are dropped and are not compared to later criteria in the list. Packets not matching this criterion are compared to the next entry in the list.
3. **Permits** any IP traffic routed from source address 11.11.11.101. Any packets matching this criterion will be permitted and will not be compared to any later criteria in the list. Because this entry comes after the entry blocking Telnet traffic from this same address, there will not be any Telnet packets to compare with this entry; they have already been dropped as a result of matching the preceding entry.
4. **Permits** Telnet traffic routed from source address 11.11.11.33. Packets matching this criterion are permitted and are not compared to any later criteria in the list. Packets not matching this criterion are compared to the next entry in the list.
5. This entry does not appear in an actual ACL, but is implicit as the last entry in every ACL. Any routed packets that do not match any of the criteria in the ACL’s preceding entries will be denied (dropped), and will not cross VLAN 12.

Figure 9-4. Example of How an ACL Filters Packets

It is important to remember that this ACL (and all ACLs) include an implicit "deny IP any". That is, routed IP packets (and switched packets having the switch as the destination IP address) that the ACL does not *explicitly* permit or deny will be *implicitly* denied, and therefore dropped instead of forwarded on the VLAN. You can preempt the implicit deny by inserting a "permit IP any" at the end of an ACL, but this solution does not apply in the preceding example, where the intention is for the switch to forward only explicitly permitted packets routed on VLAN 12.

Overriding the Implicit "deny IP any". If you want an ACL to permit any routed packets that are not explicitly denied by other entries in the ACL, you can do so by configuring a **permit any** entry as the last entry in the ACL. Doing so permits any packet not explicitly denied by earlier entries.

Planning an ACL Application

Before creating and implementing ACLs, you need to define the policies you want your ACLs to enforce, and understand how your ACLs will impact your network users.

Traffic Management and Improved Network Performance

You can use ACLs to block unnecessary traffic caused by individual hosts, workgroups, or subnets, and to block user access to subnets, devices, and services. Answering the following questions can help you to design and properly position ACLs for optimum network usage.

- What are the logical points for minimizing unwanted traffic? In many cases it makes sense to prevent unwanted traffic from reaching the core of your network by configuring ACLs to drop unwanted traffic at or close to the edge of the network. (The earlier in the network path you can block unwanted traffic, the greater the benefit for network performance.)
- What traffic should you explicitly block? Depending on your network size and the access requirements of individual hosts, this can involve creating a large number of ACEs in a given ACL (or a large number of ACLs), which increases the complexity of your solution.

- What traffic can you implicitly block by taking advantage of the implicit **deny IP any** to deny traffic that you have not explicitly permitted? This can reduce the number of entries needed in an ACL.

- What traffic should you permit? In some cases you will need to explicitly identify permitted traffic. In other cases, depending on your policies, you can insert a **permit any** entry at the end of an ACL. This means that all IP traffic not specifically matched by earlier entries in the list will be permitted.

Security

ACLs can enhance security by blocking routed IP traffic carrying an unauthorized source IP address (SA). This can include:

- Blocking access to or from subnets in your network

- Blocking access to or from the internet

- Blocking access to sensitive data storage or restricted equipment

- Preventing the use of specific TCP or UDP functions (such as Telnet, SSH, web browser) for unauthorized access

You can also enhance switch management security by using ACLs to block bridged IP traffic that has the switch itself as the destination address (DA).

Caution

ACLs can enhance network security by blocking selected IP traffic, and can serve as one aspect of maintaining network security. *However, because ACLs do not provide user or device authentication, or protection from malicious manipulation of data carried in IP packet transmissions, they should not be relied upon for a complete security solution.*

Note

ACLs in the Series 5300XL switches do not screen non-IP traffic such as AppleTalk and IPX.

Guidelines for Planning the Structure of an ACL

The first step in planning a specific ACL is to determine where you will apply it. (Refer to "ACL Inbound and Outbound Application Points" on page 9-8.) You must then determine the order in which you want the individual ACEs in the ACL to filter traffic.

- The first match dictates the action on a packet. Subsequent matches are ignored.
- On any ACL, the switch implicitly denies packets that are not explicitly permitted or denied by the ACEs configured in the ACL. If you want the switch to forward packets for which there is not a match in an ACL, add the "permit IP any" function as the last ACE in an ACL. This ensures that no packets reach the implicit "deny IP any" case.
- Generally, you should list ACEs from the most specific (individual hosts) to the most general (subnets or groups of subnets) unless doing so permits traffic that you want dropped. For example, an ACE allowing a small group of workstations to use a specialized printer should occur earlier in an ACL than an entry used to block widespread access to the same printer.

ACL Configuration and Operating Rules

- **Routing.** Except for any IP traffic with a DA on the switch itself, ACLs filter only routed traffic. Thus, if routing is not enabled on the switch, there is no routed traffic for ACLs to filter. (To enable routing, execute **ip routing** at the global configuration level.) For more on routing, refer to the chapter titled "IP Routing Features" in this manual.
- **Per-Switch ACL Limits.** At a minimum an ACL must have one, explicit "permit" or "deny" Access Control Entry. You can configure up to 255 ACL assignments to VLANs, as follows:
  - Standard ACLs: Up to 99; numeric range: 1 - 99
  - Extended ACLs: Up to 100; numeric range: 100 - 199
  - Named (Extended or Standard) ACLs: Up to 255 (minus any numeric ACL assignments)
  - Total ACEs in all ACLs: 1024
- **Implicit "deny any":** In any ACL, the switch automatically applies an implicit "deny IP any" that does not appear in **show** listings. This means that the ACL denies any packet it encounters that does not have a match with an entry in the ACL. Thus, if you want an ACL to

permit any packets that you have not expressly denied, you must enter a **permit any** or **permit ip any any** as the last ACE in an ACL. Because, for a given packet the switch sequentially applies the ACEs in an ACL until it finds a match, any packet that reaches the **permit any** or **permit ip any any** entry will be permitted, and will not encounter the "deny ip any" ACE the switch automatically includes at the end of the ACL. For an example, refer to figure 9-4 on page 9-15.

- **Explicitly Permitting Any IP Traffic:** Entering a **permit any** or a **permit ip any any** ACE in an ACL permits all IP traffic not previously permitted or denied by that ACL. Any ACEs listed after that point do not have any effect.

- **Explicitly Denying Any IP Traffic:** Entering a **deny any** or a **deny ip any any** ACE in an ACL denies all IP traffic not previously permitted or denied by that ACL. Any ACEs listed after that point have no effect.

- **Replacing One ACL with Another:** The last ACL assigned for inbound ("in") or outbound ("out") packet filtering on an interface replaces any other ACL previously configured for the same purpose. For example, if you configured ACL 100 to filter inbound traffic on VLAN 20, but later, you configured ACL 112 to filter inbound traffic on this same VLAN, ACL 112 replaces ACL 100 as the ACL to use for filtering inbound traffic on VLAN 20.

- **ACLs Operate On Static VLANs:** You can assign an ACL to any VLAN that is statically configured on the switch. ACLs do not operate with dynamic VLANs.

- **An ACL Affects All Physical Ports in a Static VLAN:** An ACL assigned to a VLAN applies to all physical ports on the switch that belong to that VLAN, including ports that have dynamically joined the VLAN.

- **ACLs Screen Traffic Entering or Leaving the Switch on a VLAN:** On a given VLAN, ACLs can screen inbound or outbound traffic at the point where it enters or leaves the switch. ACLs do not screen traffic moving between VLANs within the switch or between subnets in a multinetted VLAN. (See figure 9-1.)

- **ACLs Do Not Filter Switched Traffic Unless the Switch Itself is the DA:** ACLs do not filter:

  - Traffic moving between ports belonging to the same subnet
  - Traffic leaving the switch with an SA on the switch itself

  ACLs *do* filter switched or routed traffic having a DA on the switch.

How an ACE Uses a Mask To Screen Packets for Matches

When the switch applies an ACL to inbound or outbound traffic in a VLAN, each ACE in the ACL uses an IP address and *ACL mask* to enforce a selection policy on the packets being screened. That is, the mask determines the range of IP addresses (SA only or SA/DA) that constitute a match between the policy and a packet being screened.

What Is the Difference Between Network (or Subnet) Masks and the Masks Used with ACLs?

In common IP addressing, a network (or subnet) mask defines which part of the IP address to use for the network number and which part to use for the hosts on the network. For example:

IP Address	Mask	Network Address	Host Address
18.38.252.195	255.255.255.0	first three octets	The fourth octet.
18.38.252.195	255.255.248.0	first two octets and the left-most five bits of the third octet	The right most three bits of the third octet and all bits in the fourth octet.

Thus, the bits set to 1 in a network mask define the part of an IP address to use for the network number, and the bits set to 0 in the mask define the part of the address to use for the host number.

In an ACL, IP addresses and masks provide the criteria for determining whether to deny or permit a packet, or to pass it to the next ACE in the list. If there is a match, the deny or permit action occurs. If there is not a match, the packet is compared with the next ACE in the ACL. Thus, where a standard network mask defines how to identify the network and host numbers in an IP address, the mask used with ACEs defines which bits in a packet's IP address must match the corresponding bits in the IP address listed in an ACE, and which bits can be *wildcards*.

Rules for Defining a Match Between a Packet and an Access Control Entry (ACE)

- For a given ACE, when the switch compares an IP address and corresponding mask in the ACE to an IP address carried in a packet:
  - **A mask-bit setting of 0 ("off")** requires that the corresponding bit in the packet's IP address and in the ACE's IP address must be the same. That is, if a bit in the ACE's IP address is set to 1 ("on"), the same bit in the packet's IP address must also be 1.
  - **A mask-bit setting of 1 ("on")** means the corresponding bit in the packet's IP address and in the ACE's IP address do not have to be the same. That is, if a bit in the ACE's IP address is set to 1, the same bit in the packet's IP address can be either 1 or 0 ("on" or "off").

  For an example, refer to "Example of How the Mask Bit Settings Define a Match" on page 9-23.

- In any ACE, a mask of all ones means *any* IP address is a match. Conversely, a mask of all zeros means the *only* match is an IP address identical to the host IP address specified in the ACL.

- Depending on your network, a single ACE that allows a match with more than one source or destination IP address may allow a match with multiple subnets For example, in a network with a prefix of 31.30.240 and a subnet mask of 255.255.240.0 (the left most 20 bits), applying an ACL mask of 0.0.31.255 causes the subnet mask and the ACL mask to overlap one bit, which allows matches with hosts in two subnets: 31.30.224.0 and 31.30.240.0.

Bit Position in the Third Octet of Subnet Mask 255.255.240.0								
Bit Values	128	64	32	16	8	4	2	1
Subnet Mask Bits	1	1	1	1	n/a	n/a	n/a	n/a
Mask Bit Settings Affecting Subnet Addresses	0	0	0	**1 or 0**	n/a	n/a	n/a	n/a

This ACL supernetting technique can help to reduce the number of ACLs you need. You can apply it to a multinetted VLAN and to multiple VLANs. However, ensure that you exclude subnets that do not belong in the policy. If this creates a problem for your network, you can eliminate the unwanted match by making the ACEs in your ACL as specific as possible, and using multiple ACEs carefully ordered to eliminate unwanted matches.

- Every IP address and mask pair (source or destination) used in an ACE creates one of the following policies:
  - **Any IP address fits the matching criteria.** In this case, the switch automatically enters the IP address and mask in the ACE. For example:

    ```
    access-list 1 deny any
    ```

    produces this policy in an ACL listing:

IP Address	Mask
0.0.0.0	255.255.255.255

    This policy states that every bit in every octet of a packet's SA is a wildcard, which covers any IP address.
  - **One IP address fits the matching criteria.** In this case, you provide the IP address and the switch provides the mask. For example:

    ```
    access-list 1 permit host 18.28.100.15
    ```

    produces this policy in an ACL listing:

IP Address	Mask
18.28.100.15	0.0.0.0

    This policy states that every bit in every octet of a packet's SA must be the same as the corresponding bit in the SA defined in the ACE.
  - **A group of IP addresses fits the matching criteria.** In this case you provide both the IP address and the mask. For example:

    ```
    access-list 1 permit 18.28.32.1 0.0.0.31
    ```

IP Address	Mask
18.28.32.1	0.0.0.31

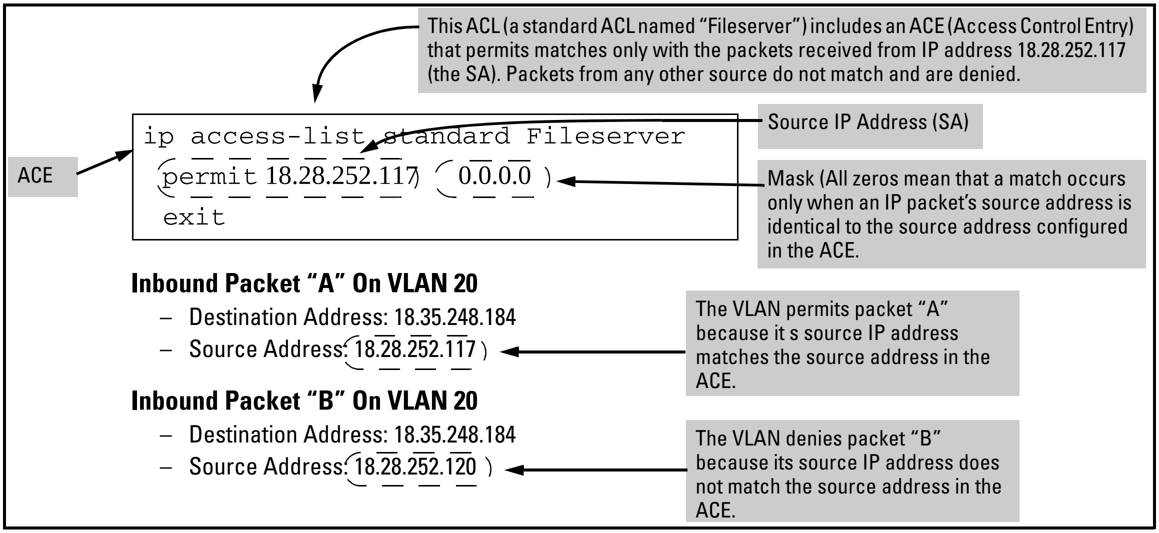
    This policy states that:
    - In the first three octets of a packet's SA, every bit must be set the same as the corresponding bit in the SA defined in the ACE.
    - In the last octet of a packet's SA, the first three bits must be the same as in the ACE, but the last five bits are wildcards and can be any value.
- Unlike subnet masks, the wildcard bits in an ACL mask need not be contiguous. For example, 0.0.7.31 is a valid ACL mask. However, a subnet mask of 255.255.248.224 is not a valid subnet mask.

Example of How the Mask Bit Settings Define a Match . Assume an ACE where the second octet of the mask for an SA is 7 (the rightmost three bits are "on", or "1") and the second octet of the corresponding SA in the ACE is 31 (the rightmost five bits). In this case, a match occurs when the second octet of the SA in a packet being filtered has a value in the range of 24 to 31. Refer to table 9-1, below.

Table 9-1. Example of How the Mask Defines a Match

Location of Octet	**Bit Position in the Octet**							
	128	64	32	16	8	4	2	1
SA in ACE	0	0	0	1	1	1	1	1
Mask for SA	0	0	0	0	0	1	1	1
Corresponding Octet of a Packet's SA	0	0	0	1	1	0/1	0/1	0/1
The shaded area indicates bits in the packet that must exactly match the bits in the source IP in the ACE. Wherever the mask bits are ones (wildcards), the IP bits in the packet can be any value, and where the mask bits are zeros, the IP bits in the packet must be the same as the IP bits in the ACE. **Note:** This example covers only one octet of an IP address. An actual ACE applies this method to all four octets of an IP address.								

Example of Allowing Only One IP Address ("Host" Option). Suppose, for example, that you have configured the ACL in figure 9-5 to filter inbound packets on VLAN 20. Because the mask is all zeros, the ACE policy dictates that a match occurs only when the source IP address on such packets is identical to the IP address configured in the ACE.



```
ip access-list standard Fileserver
 permit 18.28.252.117 0.0.0.0
 exit
```

This ACL (a standard ACL named "Fileserver") includes an ACE (Access Control Entry) that permits matches only with the packets received from IP address 18.28.252.117 (the SA). Packets from any other source do not match and are denied.

Source IP Address (SA)

Mask (All zeros mean that a match occurs only when an IP packet's source address is identical to the source address configured in the ACE.

Inbound Packet "A" On VLAN 20

- Destination Address: 18.35.248.184
- Source Address: 18.28.252.117

The VLAN permits packet "A" because it s source IP address matches the source address in the ACE.

Inbound Packet "B" On VLAN 20

- Destination Address: 18.35.248.184
- Source Address: 18.28.252.120

The VLAN denies packet "B" because its source IP address does not match the source address in the ACE.

Figure 9-5. Example of an ACL with an Access Control Entry (ACE) that Allows Only One Source IP Address

Examples Allowing Multiple IP Addresses. Table 9-2 provides examples of how to apply masks to meet various filtering requirements.

Table 9-2. Example of Using an IP Address and Mask in an Access Control Entry

IP Address in the ACE	Mask	Policy for a Match Between a Packet and the ACE	Allowed IP Addresses
A: 18.38.252.195	0.0.0.255	Exact match in first three octets only.	18.38.252.< 0-255 > (See row A in table 9-3, below.)
B: 18.38.252.195	0.0.7.255	Exact match in the first two octets and the leftmost five bits (248) of the third octet.	18.38.< 248-255 >.< 0-255 > (In the third octet, only the rightmost three bits are wildcard bits. The leftmost five bits must be a match, and in the ACE, these bits are all set to 1. See row B in table 9-3, below.)
C: 18.38.252.195	0.0.0.0	Exact match in all octets.	18.38.252.195 (There are no wildcard bits in any of the octets. See row C in table 9-3, below.)
D: 18.38.252.195	0.15.255.255	Exact match in the first octet and the leftmost four bits of the second octet.	18.< 32-47 >.< 0-255 >.<0-255> (In the second octet, the rightmost four bits are wildcard bits. See row D in table 9-3, below.)

Table 9-3. Mask Effect on Selected Octets of the IP Addresses in Table 9-2

IP Addr	Octet	Mask	Octet Range	128	64	32	16	8	4	2	1
A	3	**0** all bits	**252**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**
B	3	**7** last 3 bits	**248-255**	**1**	**1**	**1**	**1**	**1**	**0 or 1**	**0 or 1**	**0 or 1**
C	4	**0** all bits	**195**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**
D	2	**15** last 4 bits	**32-47**	**0**	**0**	**1**	**0**	**0 or 1**	**0 or 1**	**0 or 1**	**0 or 1**

Shaded areas indicate bit settings that must be an exact match.

If there is a match between the policy in the ACE and the IP address in a packet, then the packet is either permitted or denied, according to how the ACE is configured. If there is not a match, the next ACE in the ACL is then applied to the packet. The same operation applies to a destination IP address (DA) used in an extended ACE. (Where an ACE includes both source and destination IP addresses, there is one IP-address/ACL-mask pair for the source address, and another IP-address/ACL-mask pair for the destination address. See "Configuring and Assigning an ACL" on page 9-25.)

CIDR Notation. For information on using CIDR notation to specify ACL masks, refer to "Using CIDR Notation To Enter the ACL Mask" on page 9-32.

Configuring and Assigning an ACL

ACL Feature	Page
Configuring and Assigning a Numbered, Standard ACL	9-33
Configuring and Assigning a Numbered, Extended ACL	9-38
Configuring a Named ACL	9-44
Enabling or Disabling ACL Filtering	9-46

Overview

General Steps for Implementing ACLs

1. Configure at least one ACL. This creates and stores the ACL(s) in the switch configuration.
2. Assign an ACL. This applies the ACL to either the inbound or outbound traffic on a designated VLAN.
3. Enable IP routing. Except for instances where the switch is the destination, assigned ACLs screen IP traffic only when routing is enabled on the switch.

Caution Regarding the Use of Source Routing

Source routing is enabled by default on the switch and can be used to override ACLs. For this reason, if you are using ACLs to enhance network security, the recommended action is to disable source routing on the switch. To do so, execute **no ip source-route**.

Types of ACLs

- **Standard ACL:** Uses only a packet's source IP address as a criterion for permitting or denying the packet. For a standard ACL ID, use either a unique numeric string in the range of 1-99 or a unique name string of up to 64 alphanumeric characters.

- **Extended ACL:** Offers the following criteria as options for permitting or denying a packet:

  - Source IP address
  - Destination IP address
  - TCP or UDP criteria

  For an extended ACL ID, use either a unique number in the range of 100-199 or a unique name string of up to 64 alphanumeric characters.

You should carefully plan your ACL application before configuring specific ACLs. For more on this topic, refer to “Planning an ACL Application” on page 9-16.

ACL Configuration Structure

After you enter an ACL command, you may want to inspect the resulting configuration. This is especially true where you are entering multiple ACEs into an ACL. Also, it will be helpful to understand the configuration structure when using later sections in this chapter.

The basic ACL structure includes three elements:

1. List type and name: This identifies the ACL as **standard** or **extended** and shows the ACL name.
2. One or more deny/permit list entries (ACEs): One entry per line.

Element	Stnd	Ext	Notes
ID Range	1 - 99	100 - 199	You can also use an alphanumeric name of up to 64 characters, including spaces.
Minimum ACEs per ACL	1		
Maximum ACEs Per ACL and per Switch	1024		The switch allows a total of 1024 ACEs across all ACLs.

3. Implicit **deny any**: Where an ACL is in use, the switch denies any packets that do not have a match with the ACEs explicitly configured in the ACL. The implicit **deny any** does not appear in ACL configuration listings, but

always functions when the switch uses an ACL to filter packets. (You cannot delete the implicit "deny any", but you can supersede it with a "permit any" statement.)

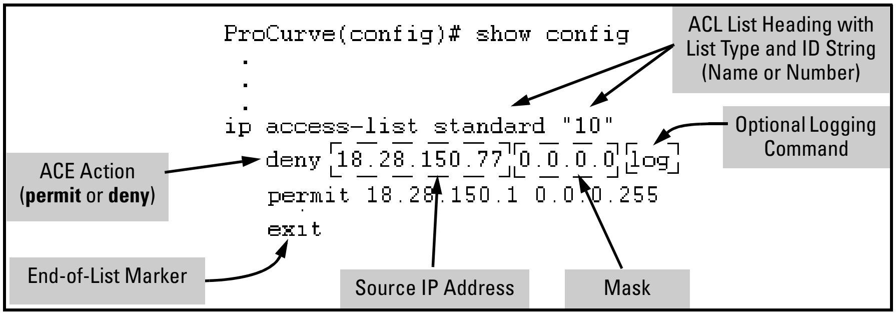
Standard ACL Structure

Individual ACEs in a standard ACL include only a permit/deny "type" statement, the source IP addressing, and an optional **log** command (available with "deny" statements).

```
ip access-list < type > "< id-string >"
   permit host < source-ip-address >
   deny < source-ip-address > < acl-mask > [log]
   .
   .
   .
   permit any
   exit
```

Figure 9-6. Example of the General Structure for a Standard ACL

For example, figure 9-7 shows how to interpret the entries in a standard ACL.



```
ProCurve(config)# show config
 .
 .
 .
ip access-list standard "10"
   deny 18.28.150.77 0.0.0.0 log
   permit 18.28.150.1 0.0.0.255
   exit
```

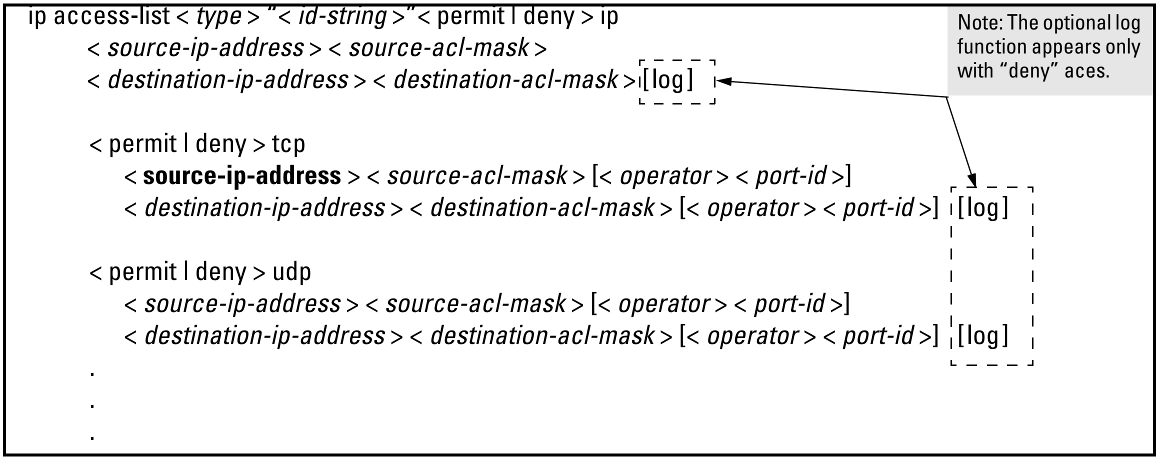
ACL List Heading with List Type and ID String (Name or Number)
Optional Logging Command
ACE Action (**permit** or **deny**)
End-of-List Marker
Source IP Address
Mask

Figure 9-7. Example of a Displayed Standard ACL Configuration with Two ACEs

Extended ACL Configuration Structure

Individual ACEs in an extended ACL include:

- A permit/deny “type” statement
- Source IP addressing
- Optional TCP or UDP port type with optional source port ID and operator and/or optional destination port ID and operator
- Destination IP addressing
- Optional ACL **log** command

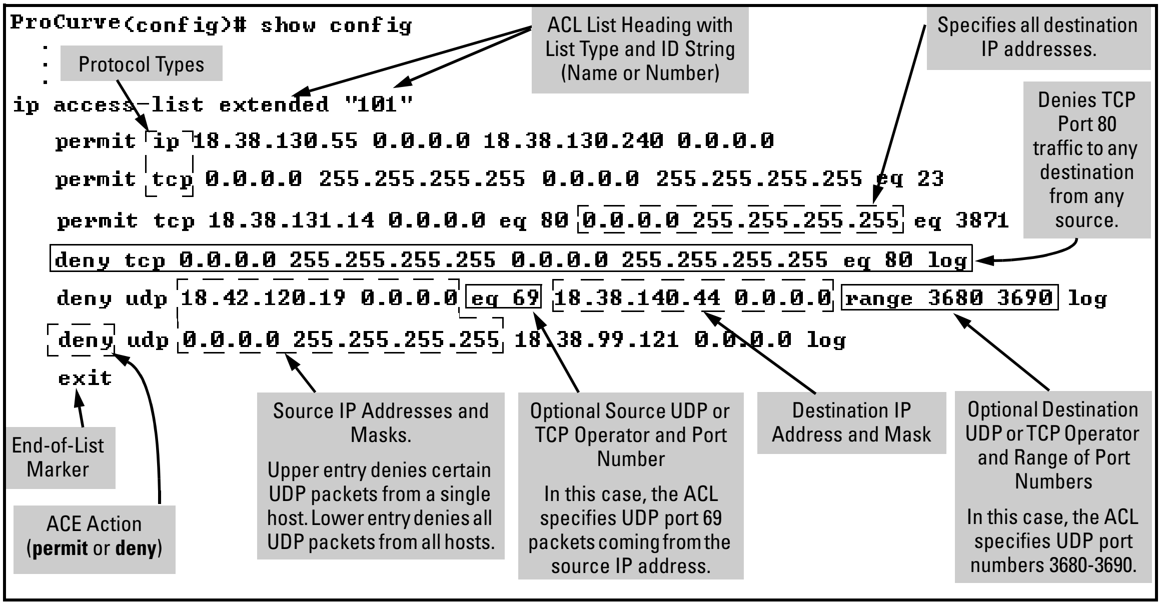


```
ip access-list < type > "< id-string >"< permit | deny > ip
     < source-ip-address > < source-acl-mask >
     < destination-ip-address > < destination-acl-mask > [log]

     < permit | deny > tcp
        < source-ip-address > < source-acl-mask > [< operator > < port-id >]
        < destination-ip-address > < destination-acl-mask > [< operator > < port-id >] [log]

     < permit | deny > udp
        < source-ip-address > < source-acl-mask > [< operator > < port-id >]
        < destination-ip-address > < destination-acl-mask > [< operator > < port-id >] [log]
     .
     .
     .
```

Note: The optional log function appears only with “deny” aces.

Figure 9-8. General Structure for an Extended ACL

For example, figure 9-9 shows how to interpret the entries in an extended ACL.



```
ProCurve(config)# show config
  .
  .
  .
ip access-list extended "101"
   permit ip 18.38.130.55 0.0.0.0 18.38.130.240 0.0.0.0
   permit tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 23
   permit tcp 18.38.131.14 0.0.0.0 eq 80 0.0.0.0 255.255.255.255 eq 3871
   deny tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 80 log
   deny udp 18.42.120.19 0.0.0.0 eq 69 18.38.140.44 0.0.0.0 range 3680 3690 log
   deny udp 0.0.0.0 255.255.255.255 18.38.99.121 0.0.0.0 log
   exit
```

Protocol Types

ACL List Heading with List Type and ID String (Name or Number)

Specifies all destination IP addresses.

Denies TCP Port 80 traffic to any destination from any source.

End-of-List Marker

ACE Action (**permit** or **deny**)

Source IP Addresses and Masks.

Upper entry denies certain UDP packets from a single host. Lower entry denies all UDP packets from all hosts.

Optional Source UDP or TCP Operator and Port Number

In this case, the ACL specifies UDP port 69 packets coming from the source IP address.

Destination IP Address and Mask

Optional Destination UDP or TCP Operator and Range of Port Numbers

In this case, the ACL specifies UDP port numbers 3680-3690.

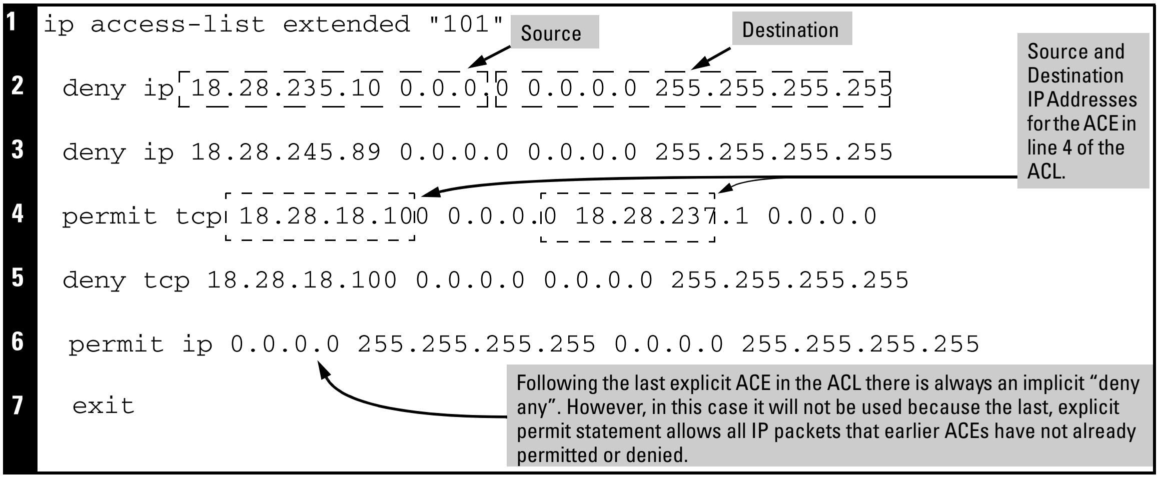
Figure 9-9. Example of a Displayed Extended ACL Configuration

ACL Configuration Factors

The Sequence of Entries in an ACL Is Significant

When the switch uses an ACL to determine whether to permit or deny a packet on a particular VLAN, it compares the packet to the criteria specified in the individual Access Control Entries (ACEs) in the ACL, beginning with the first ACE in the list and proceeding sequentially until a match is found. When a match is found, the switch applies the indicated action (permit or deny) to the packet. This is significant because, once a match is found for a packet, subsequent ACEs in the same ACL will not be used for that packet, regardless of whether they match the packet.

For example, suppose that you have applied the ACL shown in figure 9-10 to inbound traffic on VLAN 1 (the default VLAN):



```
1  ip access-list extended "101"
2   deny ip 18.28.235.10 0.0.0.0 0.0.0.0 255.255.255.255
3   deny ip 18.28.245.89 0.0.0.0 0.0.0.0 255.255.255.255
4   permit tcp 18.28.18.100 0.0.0.0 18.28.237.1 0.0.0.0
5   deny tcp 18.28.18.100 0.0.0.0 0.0.0.0 255.255.255.255
6   permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
7   exit
```

Source and Destination IP Addresses for the ACE in line 4 of the ACL.

Following the last explicit ACE in the ACL there is always an implicit "deny any". However, in this case it will not be used because the last, explicit permit statement allows all IP packets that earlier ACEs have not already permitted or denied.

Figure 9-10. Example of a Standard ACL that Permits All Traffic Not Implicitly Denied

Table 9-4. Effect of the Above ACL on Inbound Traffic in the Assigned VLAN

Line #	Action
1	Shows list type (extended) and ID (101).
2	A packet from IP source address 18.28.235.10 will be denied (dropped). This line filters out all packets received from 18.28.235.10. As a result, IP traffic from that device will not be routed and packets from that device will not be compared against any later entries in the list.
3	A packet from IP source 18.28.245.89 will be denied (dropped). This line filters out all packets received from 18.28.245.89. As the result, IP traffic from that device will not be routed and packets from that device will not be compared against any later entries in the list.
4	A packet from TCP source address 18.28.18.100 with a destination address of 18.28.237.1 will be permitted (forwarded). Since no earlier lines in the list have filtered TCP packets from 18.28.18.100 and destined for 18.28.237.1, the switch will use this line to evaluate such packets. Any packets that meet this criteria will be forwarded. (Any packets that do not meet this TCP source-destination criteria are not affected by this line.)
5	A packet from TCP source address 18.28.18.100 to ***any*** destination address will be denied (dropped). Since, in this example, the intent is to block TCP traffic from 18.28.18.100 to any destination ***except*** the destination stated in line 4, this line must follow line 4. (If their relative positions were exchanged, all TCP traffic from 18.28.18.100 would be dropped, including the traffic for the 18.28.18.1 destination.)
6	Any packet from any IP source address to any destination address will be permitted (forwarded). The only traffic to reach this line will be IP packets not specifically permitted or denied in the earlier lines.
n/a	The "implicit deny any any" is a function automatically added as the last action in all ACLs. It denies (drops) any IP traffic from any source to any destination that has not found a match with earlier entries in the list. In this example, line 6 permits (forwards) any IP traffic not already permitted or denied by the earlier entries in the list, so there is no traffic remaining for action by the "implicit deny any any" function.
7	Indicates the end of the ACL.

In Any ACL, There Will Always Be a Match

As indicated in figure 9-10, the switch automatically uses an implicit "deny IP any" (Standard ACL) or "deny IP any any" (Extended ACL) as the last ACE in any ACL. This means that if you configure the switch to use an ACL for filtering either inbound or outbound traffic on a VLAN, any packets not specifically permitted or denied by the explicit entries you create will be denied by the implicit "deny" action. Note that if you want to preempt the implicit "deny" action, insert an explicit **permit any** or **permit ip any any** as the last line of the ACL.

A Configured ACL Has No Effect Until You Apply It to an Interface

The switch stores ACLs in the configuration file. Thus, until you actually assign an ACL to a VLAN interface, it is present in the configuration, but not used.

You Can Assign an ACL Name or Number to a VLAN Even if the ACL Does Not Yet Exist in the Switch's Configuration

In this case, if you subsequently create an ACL with that name or number, the switch automatically applies each ACE as soon as you enter it in the running-config file. Similarly, if you modify an existing ACE in an ACL you already applied to a VLAN, the switch automatically implements the new ACE as soon as you enter it. (See "General ACL Operating Notes" on page 9-63.) The switch allows a maximum of 255 ACLs in any combination of numeric and alphanumeric names, and determines the total from the number of unique ACL names in the configuration. For example, if you configure two ACLs, but assign only one of them to a VLAN, the ACL total is two, for the two unique ACL names. If you then assign the name of a nonexistent ACL to a VLAN, the new ACL total is three, because the switch now has three unique ACL names in its configuration.

Using the CLI To Create an ACL

Command	Page
access-list (standard ACLs)	9-33
access-list (extended ACLs)	9-38
ip access-list (named ACLs)	9-44

You can use either the switch CLI or an offline text editor to create an ACL. This section describes the CLI method, which is recommended for creating short ACLs. (To use the offline method, refer to "Editing ACLs and Creating an ACL Offline" on page 9-53.)

General ACE Rules

These rules apply to all ACEs you create or edit using the CLI:

- ACEs are placed in an ACL according to the sequence in which you enter them (last entered, last listed).
- You can use the CLI to delete an ACE from anywhere in a given ACL by using the "no" form of the command to enter that ACE. However, when you use the CLI to add an ACE, the new entry is always placed *at the end of the ACL*.
- Duplicate ACEs are allowed in an ACL. However, multiple instances of an ACE have no effect on filtering because the first instance preempts any subsequent duplicates.

For more information, refer to "Editing ACLs and Creating an ACL Offline" on page 9-53.

Using CIDR Notation To Enter the ACL Mask

You can use CIDR (Classless Inter-Domain Routing) notation to enter ACL masks. The switch interprets the bits specified with CIDR notation as the IP address bits in an ACL and the corresponding IP address bits in a packet. The switch then converts the mask to inverse notation for ACL use.

Table 9-5. Examples of CIDR Notation for Masks

IP Address Used In an ACL with CIDR Notation	Resulting ACL Mask	Meaning
18.38.240.125/15	0.1.255.255	The leftmost 15 bits must match; the remaining bits are wildcards.
18.38.240.125/20	0.0.15.255	The leftmost 20 bits must match; the remaining bits are wildcards.
18.38.240.125/21	0.0.7.255	The leftmost 21 bits must match; the remaining bits are wildcards.
18.38.240.125/24	0.0.0.255	The leftmost 24 bits must match; the remaining bits are wildcards.
18.38.240.125/32	0.0.0.0	All bits must match.

Configuring and Assigning a Numbered, Standard ACL

This section describes how to configure numbered, standard ACLs.

- To configure named ACLs, refer to "Configuring a Named ACL" on page 9-44.
- To configure extended, numbered ACLs, refer to "Configuring and Assigning a Numbered, Extended ACL" on page 9-38.

A standard ACL uses only source IP addresses in its ACEs. This type of ACE is useful when you need to:

- Permit or deny traffic based on source IP address only.
- Quickly control the IP traffic from a specific address. This allows you to isolate traffic problems generated by a specific device, group of contiguous devices, or a subnet threatening to degrade network performance. This gives you an opportunity to troubleshoot without sacrificing performance for users outside of the problem area.

You can configure up to 255 standard ACL assignments, depending on how many extended ACL assignments are already configured. (The switch allows a maximum of 255 unique ACL identities; standard and extended combined.) You can identify each standard ACL with a number in the range of 1 - 99, or an alphanumeric string of up to 64 characters. The CLI command process for using an alphanumeric string to name an ACL differs from the command process for a numeric name. For a description of naming an ACL with an alphanumeric character string, refer to "Configuring a Named ACL" on page 9-44. To view the command differences, refer to table 9-1, "Comprehensive Command Summary" on page 9-4.

Note

For a summary of ACL commands, refer to table 9-1, "Comprehensive Command Summary", on page 9-4.

Syntax: [no] access-list

Creates an ACE in the specified (1-99) access list and indicates the action (deny or permit) to take on a packet if there is a match between the packet and the criterion in the entry. If the ACL does not already exist, this command creates the specified ACL and its first ACE. To create a named ACL, refer to "Configuring a Named ACL" on page 9-44.

< 1-99 >

Specifies the ACL ID number. The switch interprets an ACL with a value in this range as a standard ACL.

Note: *To create an access list with an alphanumeric name (**name-str**) instead of a number, refer to "Configuring a Named ACL" on page 9-44.*

< deny | permit >

Specifies whether to deny (drop) or permit (forward) a packet that matches the ACE criteria.

< any | host < *src-ip-addr* > | *ip-addr* / *mask-length* >

- **any** *—Performs the specified action on any IP packet. Use this criterion to designate packets from any IP address.*
- **host** < *host ip-address* > *—Performs the specified action on any IP packet having the < host ip-address > as the source. Use this criterion to designate packets from a single IP address.*
- *IP-addr / mask-length —Performs the specified action on any IP packet having a source address within the range defined by either*

  < *src-ip-addr / cidr-mask-bits* **>**
  or
  < *src-ip-addr < mask* **>>**

  Use this criterion to filter packets received from either a subnet or a group of IP addresses. The mask can be in either dotted-decimal format or CIDR format with the number of significant bits. Refer to "Using CIDR Notation To Enter the ACL Mask" on page 9-32.

The mask is applied to the IP address in the ACL to define which bits in a packet's source IP address must exactly match the IP address configured in the ACL and which bits need not match. Note that specifying a group of contiguous IP addresses may require more than one ACE. For more on how masks operate in ACLs, refer to "How an ACE Uses a Mask To Screen Packets for Matches" on page 9-20.

[log]

Optionally generates an ACL log message if:

- *The action is* **deny.**
- *There is a match.*
- *ACL logging is enabled on the switch. (Refer to "Enable ACL "Deny" Logging" on page 9-59.)*

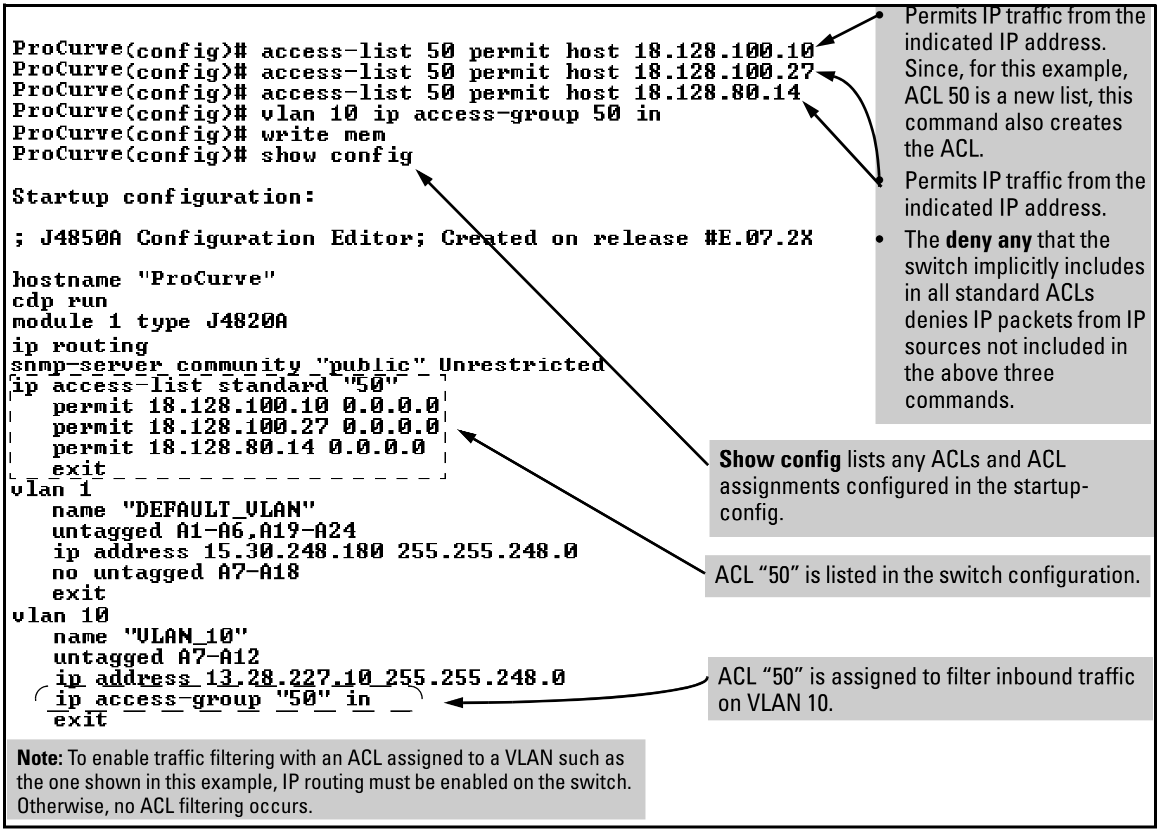
(Use the debug command to direct ACL logging output to the current console session and/or to a Syslog server. Note that you must also use the **logging < i*p-addr* >** *command to specify the IP addresses of Syslog servers to which you want log messages sent. See also "Enable ACL "Deny" Logging" on page 9-59.)*

***Syntax*:** vlan < *vid* > ip access-group < *ASCII-STR* > < in | out >

Assigns an ACL, designated by an ACL ID (< ***ASCII-STR*** *>), to a VLAN.*

Example of a Standard ACL. Suppose you wanted to configure a standard ACL and assign it to filter inbound traffic on VLAN 10 in a particular switch:

- The ID you selected for this ACL is "50".
- You want the ACL to deny IP traffic from all hosts except these three:
  - 18.128.100.10
  - 18.128.100.27
  - 18.128.100.14



```
ProCurve(config)# access-list 50 permit host 18.128.100.10
ProCurve(config)# access-list 50 permit host 18.128.100.27
ProCurve(config)# access-list 50 permit host 18.128.80.14
ProCurve(config)# vlan 10 ip access-group 50 in
ProCurve(config)# write mem
ProCurve(config)# show config

Startup configuration:

; J4850A Configuration Editor; Created on release #E.07.2X

hostname "ProCurve"
cdp run
module 1 type J4820A
ip routing
snmp-server community "public" Unrestricted
ip access-list standard "50"
   permit 18.128.100.10 0.0.0.0
   permit 18.128.100.27 0.0.0.0
   permit 18.128.80.14 0.0.0.0
   exit
vlan 1
   name "DEFAULT_VLAN"
   untagged A1-A6,A19-A24
   ip address 15.30.248.180 255.255.248.0
   no untagged A7-A18
   exit
vlan 10
   name "VLAN_10"
   untagged A7-A12
   ip address 13.28.227.10 255.255.248.0
   ip access-group "50" in
   exit
```

- Permits IP traffic from the indicated IP address. Since, for this example, ACL 50 is a new list, this command also creates the ACL.
- Permits IP traffic from the indicated IP address.
- The **deny any** that the switch implicitly includes in all standard ACLs denies IP packets from IP sources not included in the above three commands.

Show config lists any ACLs and ACL assignments configured in the startup-config.

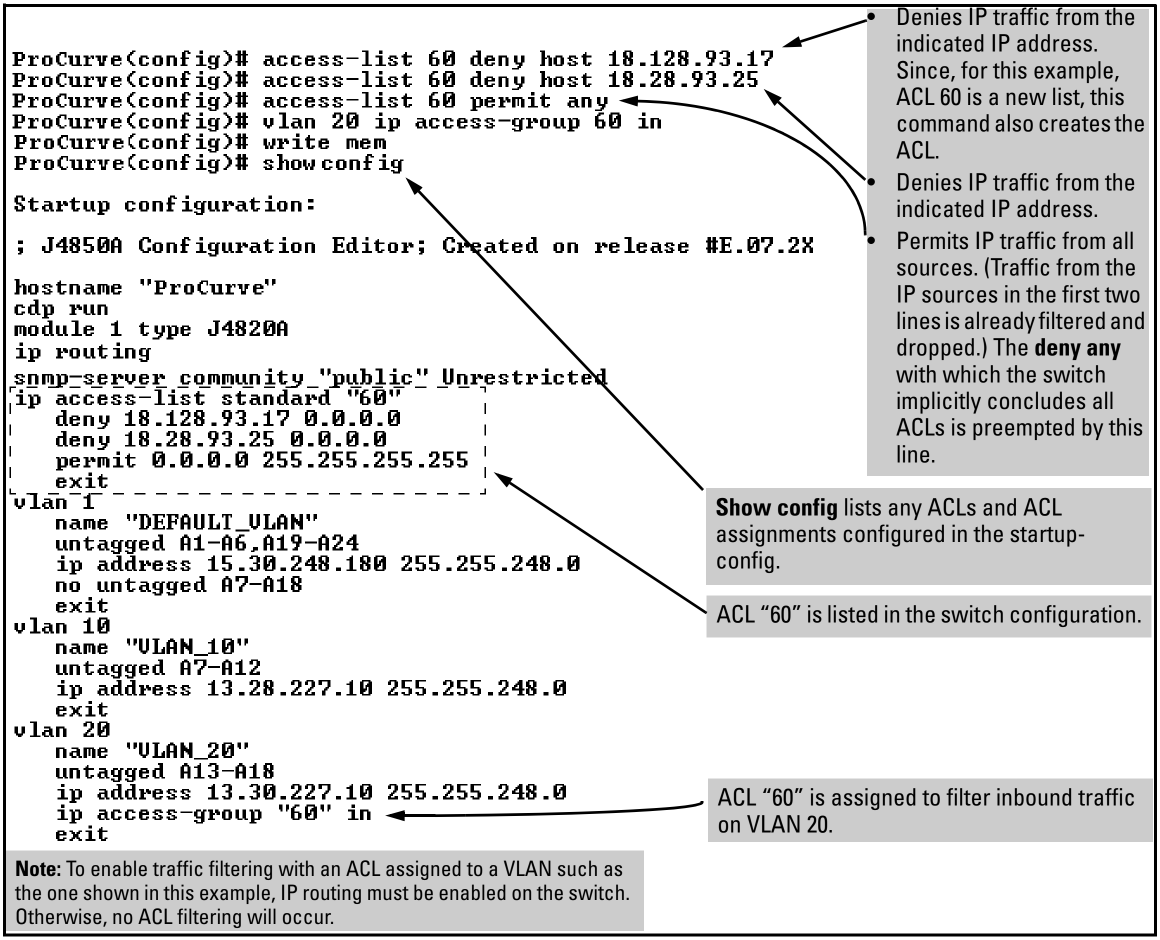
ACL "50" is listed in the switch configuration.

ACL "50" is assigned to filter inbound traffic on VLAN 10.

Note: To enable traffic filtering with an ACL assigned to a VLAN such as the one shown in this example, IP routing must be enabled on the switch. Otherwise, no ACL filtering occurs.

Figure 9-11. Example of Configuring a Standard ACL To Permit Only Traffic from Specific IP Addresses

In a situation opposite to the above, suppose that you wanted to deny inbound IP traffic received on VLAN 20 from 18. 128.93.17 and 18.130.93.25, but permit all other IP traffic on this VLAN. The next ACL achieves this:



```
ProCurve(config)# access-list 60 deny host 18.128.93.17
ProCurve(config)# access-list 60 deny host 18.28.93.25
ProCurve(config)# access-list 60 permit any
ProCurve(config)# vlan 20 ip access-group 60 in
ProCurve(config)# write mem
ProCurve(config)# show config

Startup configuration:

; J4850A Configuration Editor; Created on release #E.07.2X

hostname "ProCurve"
cdp run
module 1 type J4820A
ip routing
snmp-server community "public" Unrestricted
ip access-list standard "60"
   deny 18.128.93.17 0.0.0.0
   deny 18.28.93.25 0.0.0.0
   permit 0.0.0.0 255.255.255.255
   exit
vlan 1
   name "DEFAULT_VLAN"
   untagged A1-A6,A19-A24
   ip address 15.30.248.180 255.255.248.0
   no untagged A7-A18
   exit
vlan 10
   name "VLAN_10"
   untagged A7-A12
   ip address 13.28.227.10 255.255.248.0
   exit
vlan 20
   name "VLAN_20"
   untagged A13-A18
   ip address 13.30.227.10 255.255.248.0
   ip access-group "60" in
   exit
```

- Denies IP traffic from the indicated IP address. Since, for this example, ACL 60 is a new list, this command also creates the ACL.
- Denies IP traffic from the indicated IP address.
- Permits IP traffic from all sources. (Traffic from the IP sources in the first two lines is already filtered and dropped.) The **deny any** with which the switch implicitly concludes all ACLs is preempted by this line.

Show config lists any ACLs and ACL assignments configured in the startup-config.

ACL "60" is listed in the switch configuration.

ACL "60" is assigned to filter inbound traffic on VLAN 20.

Note: To enable traffic filtering with an ACL assigned to a VLAN such as the one shown in this example, IP routing must be enabled on the switch. Otherwise, no ACL filtering will occur.

Figure 9-12. Example of Configuring a Standard ACL To Deny Inbound Traffic from Specific IP Addresses

Configuring and Assigning a Numbered, Extended ACL

This section describes how to configure numbered, extended ACLs.

- To configure named ACLs, refer to "Configuring a Named ACL" on page 9-44.
- To configure standard, numbered ACL, refer to "Configuring and Assigning a Numbered, Standard ACL" on page 9-33.

While standard ACLs use only source IP addresses for filtering criteria, extended ACLs allow multiple ACE criteria. This enables you to more closely define your IP packet-filtering criteria. These criteria include:

- Source and destination IP addresses (required), in one of the following options:
  - Specific host IP
  - Subnet or group of IP addresses
  - Any IP address
- IP protocol (IP, TCP, or UDP)
- Source TCP or UDP port (if the IP protocol is TCP or UDP)
- Destination TCP or UDP port (if the IP protocol is TCP or UDP)
- TCP or UDP comparison operator (if the IP protocol is TCP or UDP)

You can configure up to 100 extended ACLs with a numeric name in the range of 100 -199. You can also configure extended ACLs with alphanumeric names. (Refer to "Configuring a Named ACL" on page 9-44.) The switch allows a maximum of 255 ACLs in any combination of numeric and alphanumeric names, and determines the total from the number of unique ACL names in the configuration. For example, if you configure two ACLs, but assign only one of them to a VLAN, the ACL total is two, for the two unique ACL names. If you then assign the name of a nonexistent ACL to a VLAN, the new ACL total is three, because the switch now has three unique ACL names in its configuration. (The switch allows up to 1024 ACEs total in all ACLs.)

Note

For a summary of ACL commands, refer to table 9-1, "Comprehensive Command Summary", on page 9-4.

Syntax: [no] access-list

Creates an ACE in the specified (100-199) access list and:

- *Indicates the action (deny or permit) to take on a packet if there is a match between the packet and the criteria in the complete ACE.*
- *Specifies the packet protocol type (IP, TCP, or UDP).*
- *Specifies the source and destination addressing options described in the remainder of this section.*
- *Allows optional ACL logging where a packet has a match with a* **deny** *ACE.*

If the ACL does not already exist, this command creates the specified ACL and its first ACE. If the ACL already exists, this command adds a new, explicit ACE to the end of the ACL. For a match to occur, the packet must have the source and destination IP addressing criteria specified by this command, as well as any protocol-specific (TCP or UDP port number) criteria specified by the command. To create a named ACL, refer to "Configuring a Named ACL" on page 9-44.

< 100-199 >

Specifies the ACL ID number. The switch interprets an ACL with a value in this range as an extended ACL.

Note: *To create an access list with an alphanumeric name instead of a number, refer to "Configuring a Named ACL" on page 9-44.*

< deny | permit >

Specifies whether to deny (drop) or permit (forward) a packet that matches the ACE criteria.

< ip | tcp | udp >

Specifies the packet protocol type required for a match:

- **ip** *— any IP packet*
- **tcp** *— only tcp packets*
- **udp** *— only udp packets*

< any | host < *src-ip-addr* > | *ip-addr/mask -length* >

In an extended ACL, this parameter defines the source IP address (SA) that a packet must carry in order to have a match with the ACE.

- **any** — *Specifies all inbound IP packets.*
- **host** < ***src-ip-addr*** > — *Specifies only inbound packets from a single IP address. Use this option when you want to match only the IP packets from one source IP address.*
- ***src-ip-addr*/*mask-length*** — *Performs the specified action on any IP packet having a source address within the range defined by either*

  < *src-ip-addr / cidr-mask-bits* >
  or
  < *src-ip-addr* < *mask* >>

  Use this criterion to filter packets received from either a subnet or a group of IP addresses. The mask can be in either dotted-decimal format or CIDR format with the number of significant bits. Refer to "Using CIDR Notation To Enter the ACL Mask" on page 9-32.

  The mask is applied to the IP address in the ACL to define which bits in a packet's source IP address must exactly match the IP address configured in the ACL and which bits need not match. Note that specifying a group of contiguous IP addresses may require more than one ACE. For more on how masks operate in ACLs, refer to "How an ACE Uses a Mask To Screen Packets for Matches" on page 9-20.

[*operator* < *src-port tcp/udp-id* >]

In an extended ACL where you have selected either **tcp** *or* **udp** *as the packet protocol type (see above), you can optionally use a TCP or UDP source port number or range of numbers to further define the criteria for a match. To specify a TCP or UDP port number, (1) select a comparison operator from the following list and (2) enter the port number or a well-known port name.*

Comparison Operators:

- **eq** < ***tcp/udp-port-nbr*** > — *"Equal To"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must be equal to* < ***tcp/udp-port-nbr*** >.
- **gt** < ***tcp/udp-port-nbr*** > — *"Greater Than"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must be greater than < tcp/udp-port-nbr >.*
- **lt** < ***tcp/udp-port-nbr*** > — *"Less Than"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must be less than* < ***tcp/udp-port-nbr*** >.
- **neq** < ***tcp/udp-port-nbr***> — *"Not Equal"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must not be equal to* ***< tcp/udp-port-nbr >***.
- **range** ***< start-port-nbr > < end-port-nbr >*** — *To have a match with the ACE entry, the TCP or UDP source port number in a packet must be in the range* < ***start-port-nbr > < end-port-nbr*** >.

Port Number or Well-Known Port Name:

Use the TCP or UDP port number required by your application. The switch also accepts these well-known TCP or UDP port names as an alternative to their corresponding port numbers:

- ***TCP***: bgp, dns, ftp, http, imap4, ldap, nntp, pop2, pop3, smtp, ssl, telnet
- ***UDP***: bootpc, dns, ntp, radius, radius-old, rip, snmp, snmp-trap, tftp

To list the above names, press the **[Shift] [?]** *key combination after entering an operator. For a comprehensive listing of port numbers, visit www.iana.org/assignments/port-numbers.*

< any | host < *dest-ip-addr* > | *ip-addr/mask*-length >

In an extended ACL, this parameter defines the destination IP address (DA) that a packet must carry in order to have a match with the ACE. The options are the same as shown for < ***src-ip-addr*** >.

[< *dest-port tcp/udp-id* >]

In an extended ACL, this parameter defines the TCP or UDP destination port number a packet must carry in order to have a match with the extended ACE. The options are the same as shown above on the preceding page for the source IP address.

[log]

Optional; generates an ACL log message if:

- *The action is* **deny.** ***(This option is not configurable for*** **Permit.*)***
- *There is a match.*
- *ACL logging is enabled on the switch. (Refer to "Enabling ACL Logging on the Switch" on page 9-60)*

Syntax: vlan < *vid* > ip access-group < *list-#* | *ascii-str* > < in | out >

Assigns an ACL, designated by an ACL list number or ASCII string (alphanumeric list name), to a VLAN to filter either inbound or outbound IP traffic on that VLAN. To configure named ACLs, refer to "Configuring a Named ACL" on page 9-44.

Example of an Extended ACL. Suppose that you want to implement these policies on a Series 5300XL switch configured for IP routing and membership in VLANs 10, 20, and 30:

A. Permit Telnet traffic from 10.10.10.44 to 10.10.20.78, deny all other IP traffic from network 10.10.10.0 (VLAN 10) to 10.10.20.0 (VLAN 20), and permit all other IP traffic from any source to any destination. (See "A" in figure 9-13, below.)

B. Permit FTP traffic from IP address 10.10.20.100 (on VLAN 20) to 10.10.30.55 (on VLAN 30). Deny FTP traffic from other hosts on network10.10.20.0 to any destination, but permit all other traffic.

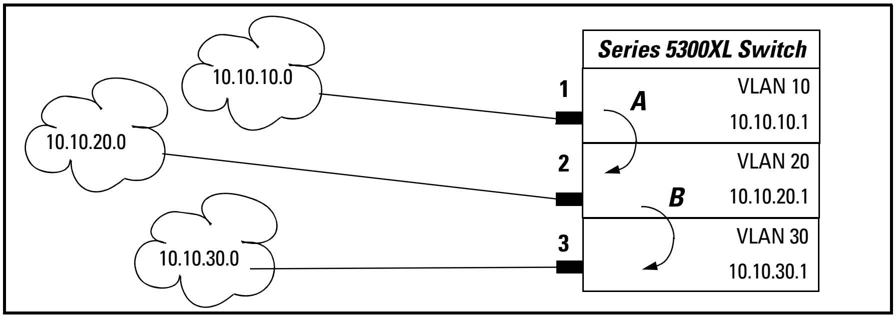
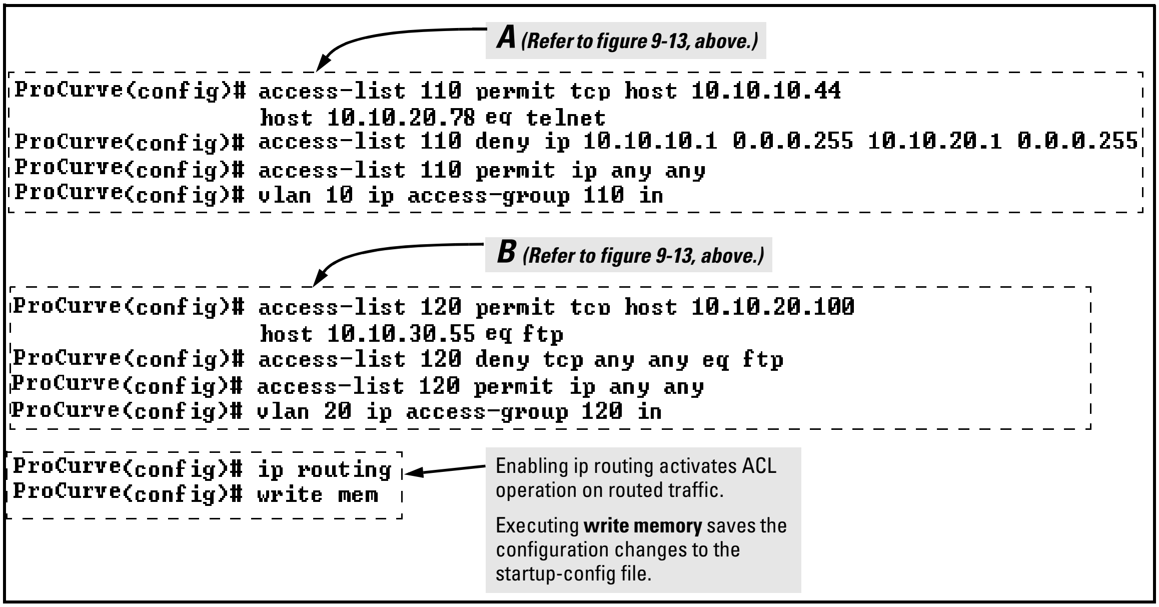


Figure 9-13. Example of an Extended ACL



***A** (Refer to figure 9-13, above.)*

```
ProCurve(config)# access-list 110 permit tcp host 10.10.10.44
                  host 10.10.20.78 eq telnet
ProCurve(config)# access-list 110 deny ip 10.10.10.1 0.0.0.255 10.10.20.1 0.0.0.255
ProCurve(config)# access-list 110 permit ip any any
ProCurve(config)# vlan 10 ip access-group 110 in
```

***B** (Refer to figure 9-13, above.)*

```
ProCurve(config)# access-list 120 permit tcp host 10.10.20.100
                  host 10.10.30.55 eq ftp
ProCurve(config)# access-list 120 deny tcp any any eq ftp
ProCurve(config)# access-list 120 permit ip any any
ProCurve(config)# vlan 20 ip access-group 120 in
```

```
ProCurve(config)# ip routing
ProCurve(config)# write mem
```

Enabling ip routing activates ACL operation on routed traffic.

Executing **write memory** saves the configuration changes to the startup-config file.

Figure 9-14. Example of Configuration Commands for an Extended ACL

Configuring a Named ACL

You can use the “Named ACL” context to configure a standard or extended ACL with an alphanumeric name instead of a number. Note that the command structure for configuring a named ACL differs from that for a numbered ACL.

Syntax:

```
ip access-list standard < name-str | 1-99 >
  < deny | permit >
  < any | host < src-ip-addr > | ip-addr / mask-length >
  [log]

ip access-list extended < name-str | 100-199 >
  < deny | permit > ip
  < any | host < src-ip-addr > | ip-addr / mask-length >
  < any | host < dest-ip-addr > | ip-addr / mask-length >
  [log]

ip access-list extended < name-string >
  < deny | permit > < tcp | udp >
      < any | host < src-ip-addr > | ip-addr / mask-length >
          [ oper < src-port tcp/udp-id >]
      < any | host < dest-ip-addr > | ip-addr / mask-length >
           [ oper < dest-port tcp/udp-id >]
    [log]
```

These commands create an ACE in the named ACL list and:

- *Indicate the action (deny or permit) to take on a packet if there is a match between a packet and the criteria in the complete ACE.*
- *Specify the packet protocol type (IP, TCP, or UDP) and (if TCP or UDP) the comparison operator.*
- *Specify the source and destination addressing options required for a match.*
- *Allow optional ACL logging where a packet has a match with a* **deny** *ACE. The* **log** *option does not appear when* **permit** *is the action.*

If the ACL does not already exist, these commands create the specified ACL and its first ACE. If the ACL already exists, these commands add a new, explicit ACE to the end of the ACL. For a match to occur, the packet must have the source and destination IP addressing criteria specified by this command, as well as any protocol-specific (TCP or UDP port number) criteria specified by the command.

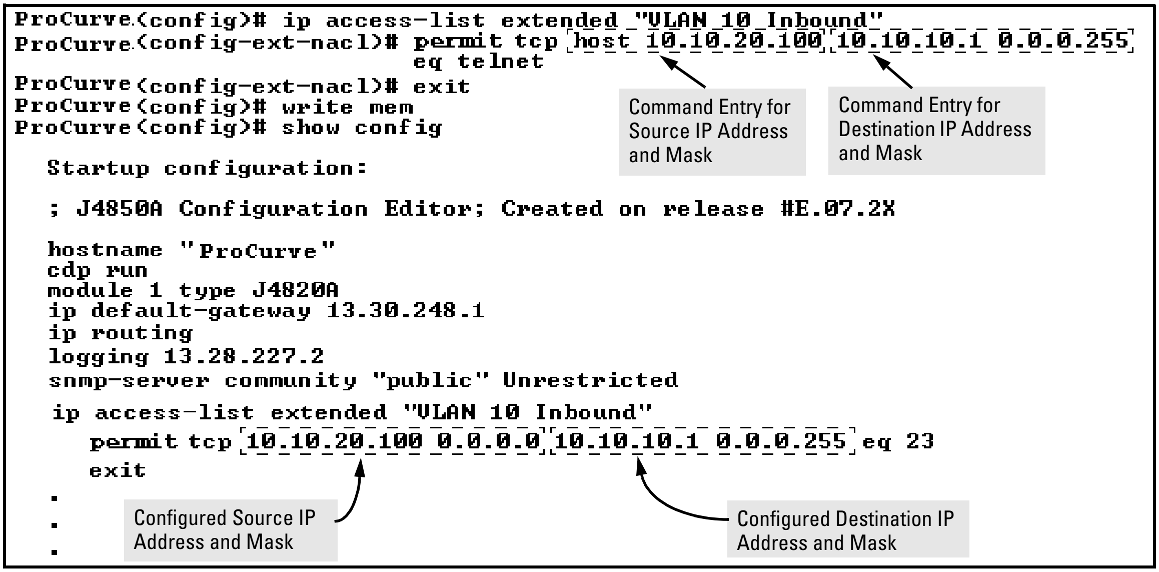
< *name-str* | 1-99 | 100-199 >

> *Consists of an alphanumeric string of up to 64 case-sensitive characters. If you include a space in the string, you must also enclose the string with quotes. For example,* **"ACL # 1"**. *You can also enter numbers in the ranges associated with standard (1-99) and extended (100-199) ACLs.*

For explanations of the individual parameters in the preceding syntax statements, refer to the syntax descriptions under "Configuring and Assigning a Numbered, Standard ACL" on page 9-33 or "Configuring and Assigning a Numbered, Extended ACL" on page 9-38.

For example, figure 9-15 shows the commands for creating an ACL in the "Named ACL" context with these parameters:

ACL Name:	VLAN 10 Inbound
Action:	Deny
Protocol:	TCP
Source IP Address and Mask	10.10.20.100 0.0.0.0
Destination IP Address and Mask	10.10.10.1 0.0.0.255
Protocol Operator and Port Number at Destination	eq telnet



```
ProCurve(config)# ip access-list extended "VLAN 10 Inbound"
ProCurve(config-ext-nacl)# permit tcp host 10.10.20.100 10.10.10.1 0.0.0.255
                           eq telnet
ProCurve(config-ext-nacl)# exit
ProCurve(config)# write mem
ProCurve(config)# show config

   Startup configuration:

   ; J4850A Configuration Editor; Created on release #E.07.2X

   hostname "ProCurve"
   cdp run
   module 1 type J4820A
   ip default-gateway 13.30.248.1
   ip routing
   logging 13.28.227.2
   snmp-server community "public" Unrestricted
   ip access-list extended "VLAN 10 Inbound"
      permit tcp 10.10.20.100 0.0.0.0 10.10.10.1 0.0.0.255 eq 23
      exit
   .
   .
   .
```

Command Entry for Source IP Address and Mask

Command Entry for Destination IP Address and Mask

Configured Source IP Address and Mask

Configured Destination IP Address and Mask

Figure 9-15. Using the "Named ACL" Context To Configure an ACL

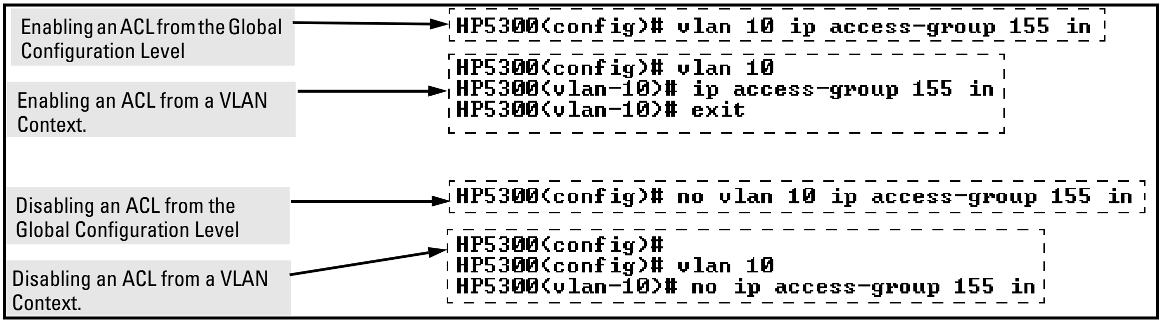
Enabling or Disabling ACL Filtering on a VLAN

For a given interface, you can configure one ACL to filter inbound traffic and one ACL to filter outbound traffic. You can also use the same ACL for both inbound and outbound traffic, and for assignment to multiple VLANs. For limits and operating rules, refer to "ACL Configuration and Operating Rules" on page 9-18.

Syntax: [no] vlan < *vid* > ip access-group < *ascii-string* > < in | out >
where: < *ascii-string* > = either a ACL name or an ACL ID number.

Assigns an ACL to a VLAN. You can use either the global configuration level or the VLAN context level to assign an ACL to a VLAN or remove an ACL from a VLAN.

***Note:** The switch allows you to assign a nonexistent ACL name or number to a VLAN. In this case, if you subsequently configure an ACL with that name or number, it will automatically become active on the assigned VLAN. Also, if you delete an assigned ACL from the switch without subsequently using the "*no*" form of this command to remove the assignment to a VLAN, the ACL assignment remains and will automatically activate any new ACE if you create with the same ACL name.*



Enabling an ACL from the Global Configuration Level
Enabling an ACL from a VLAN Context.
Disabling an ACL from the Global Configuration Level
Disabling an ACL from a VLAN Context.

```
HP5300(config)# vlan 10 ip access-group 155 in

HP5300(config)# vlan 10
HP5300(vlan-10)# ip access-group 155 in
HP5300(vlan-10)# exit

HP5300(config)# no vlan 10 ip access-group 155 in

HP5300(config)#
HP5300(config)# vlan 10
HP5300(vlan-10)# no ip access-group 155 in
```

Figure 9-16. Methods for Enabling and Disabling ACLs

Deleting an ACL from the Switch

Syntax: no ip access-list standard < *name-str* | 1-99 >

no ip access-list extended < *name-str* | 100-199 >

Removes the specified ACL from the switch's running-config file.

Note: *Deleting an ACL does not delete any assignment of that ACL to a specific VLAN. If you need to delete an ACL assignment, refer to "Enabling or Disabling ACL Filtering on a VLAN" on page 9-46.*

Displaying ACL Data

ACL Commands	Function	Page
show access-list	View a brief listing of all ACLs on the switch.	9-48
show access-list config	Display the CLI commands for generating the ACL commands configured in the switch.	9-49
show access-list vlan < ***vid*** >	List the name and type of ACLs assigned to a particular VLAN on the switch.	9-50
show access-list < ***acl-id*** >	Display detailed content information for a specific ACL.	9-50
show config	**show config** includes configured ACLs and assignments existing in the startup-config file.	
show running	**show running** includes configured ACLs and assignments existing in the running-config file.	

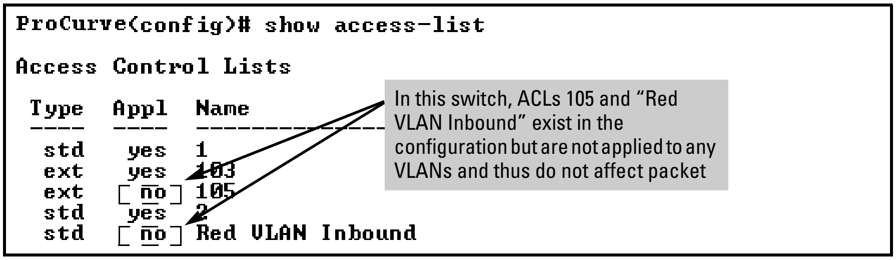
Display an ACL Summary

This command lists the configured ACLs, regardless of whether they are assigned to any VLANs.

Syntax: show access-list

List a summary table of the name, type, and application status of all ACLs configured on the switch.

For example:



```
ProCurve(config)# show access-list

Access Control Lists

  Type  Appl  Name
  ----  ----  ---------------
   std   yes  1
   ext   yes  103
   ext   no   105
   std   yes  2
   std   no   Red VLAN Inbound
```

In this switch, ACLs 105 and "Red VLAN Inbound" exist in the configuration but are not applied to any VLANs and thus do not affect packet

Figure 9-17. Example of a Summary Table of Access lists

Term	Meaning
Type	Shows whether the listed ACL is **std** (Standard; source-address only) or **ext** (Extended; protocol, source, and destination data).
Appl	Shows whether the listed ACL has been applied to a VLAN (**yes**/**no**).
Name	Shows the name or ID number assigned to each ACL configured in the switch.

Display the Content of All ACLs on the Switch

This command lists the configuration details for every ACL configured in the running-config file, regardless of whether you have assigned any to filter traffic on VLANs configured on the switch.

Syntax: show access-list config

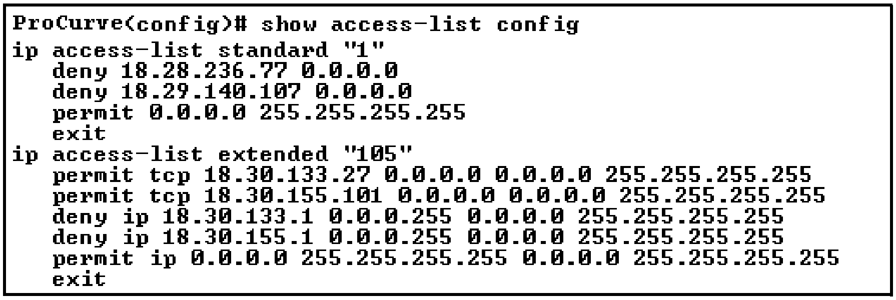
List the configured syntax for all ACLs currently configured on the switch.

Note

Notice that you can use the output from this command for input to an offline text file in which you can edit, add, or delete ACL commands. Refer to "Editing ACLs and Creating an ACL Offline" on page 9-53.

This information also appears in the **show running** display. If you executed **write memory** after configuring an ACL, it appears in the **show config** display.

For example, with two ACLs configured in the switch, you will see results similar to the following:



```
ProCurve(config)# show access-list config
ip access-list standard "1"
   deny 18.28.236.77 0.0.0.0
   deny 18.29.140.107 0.0.0.0
   permit 0.0.0.0 255.255.255.255
   exit
ip access-list extended "105"
   permit tcp 18.30.133.27 0.0.0.0 0.0.0.0 255.255.255.255
   permit tcp 18.30.155.101 0.0.0.0 0.0.0.0 255.255.255.255
   deny ip 18.30.133.1 0.0.0.255 0.0.0.0 255.255.255.255
   deny ip 18.30.155.1 0.0.0.255 0.0.0.0 255.255.255.255
   permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
   exit
```

Figure 9-18. Example of an ACL Configured Syntax Listing

Display the ACL Assignments for a VLAN

This command briefly lists the identification and type(s) of ACLs currently assigned to a particular VLAN in the running-config file. (The switch allows up to two ACL assignments per VLAN; one inbound and one outbound.)

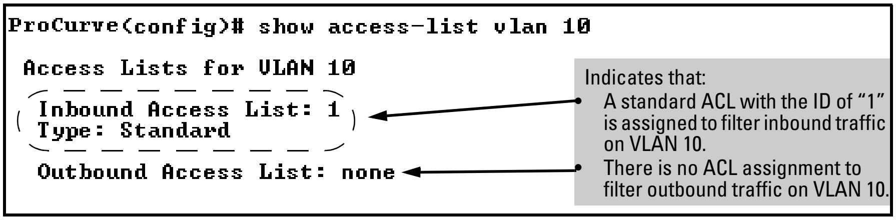
Syntax: show access-list vlan < *vid* >

List the ACLs assigned to a VLAN in the running config file.

Note

This information also appears in the **show running** display. If you executed **write memory** after configuring an ACL, it appears in the **show config** display.

For example, if you assigned a standard ACL with an ACL-ID of “1” to filter inbound traffic on VLAN 10, you could quickly verify this assignment as follows:



```
ProCurve(config)# show access-list vlan 10

 Access Lists for VLAN 10

  Inbound Access List: 1
  Type: Standard

  Outbound Access List: none
```

Indicates that:
- A standard ACL with the ID of “1” is assigned to filter inbound traffic on VLAN 10.
- There is no ACL assignment to filter outbound traffic on VLAN 10.

Figure 9-19. Example of Listing the ACL Assignments for a VLAN

Displaying the Content of a Specific ACL

This command displays a specific ACL configured in the running config file in an easy-to-read tabular format.

Note

This information also appears in the **show running** display. If you executed **write memory** after configuring an ACL, it appears in the **show config** display.

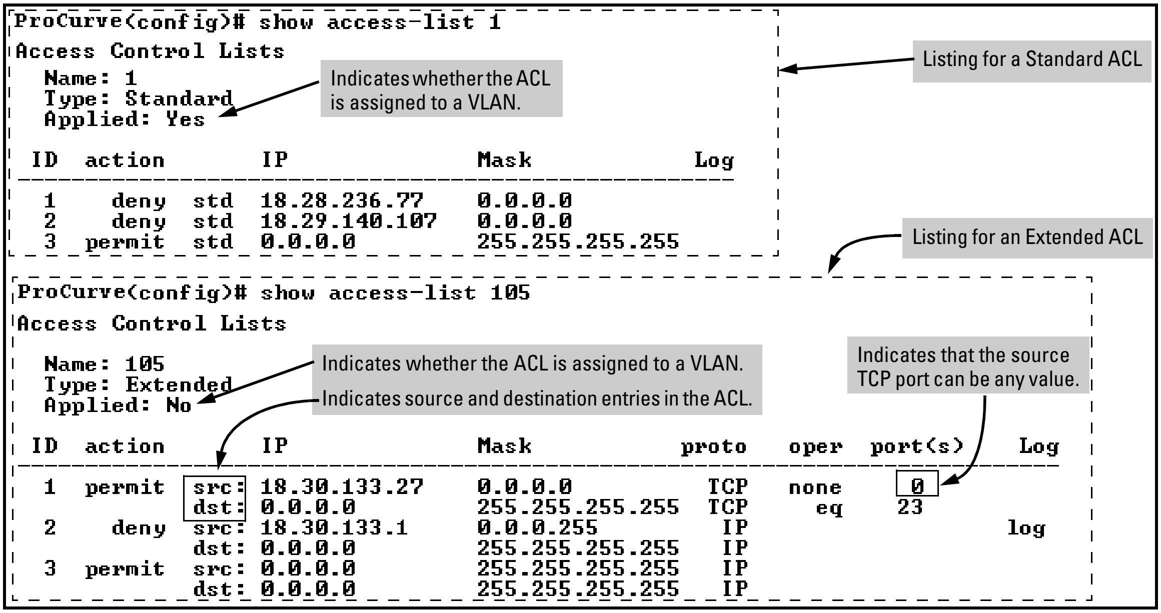
Syntax: show access-list < *acl-id* >

Display detailed information on the content of a specific ACL configured in the running-config file.

For example, suppose you configured the following two ACLs in the switch:

ACL ID	ACL Type	Desired Action
1	Standard	• Deny IP traffic from 18.28.236.77 and 18.29.140.107. • Permit IP traffic from all other sources.
105	Extended	• Permit any TCP traffic from 18.30.133.27 to any destination. • Deny any other IP traffic from 18.30.133.(1-255). • Permit all other IP traffic from any source to any destination.

Inspect the ACLs as follows:



```
ProCurve(config)# show access-list 1
Access Control Lists
   Name: 1
   Type: Standard
   Applied: Yes

 ID  action       IP               Mask               Log
 -----------------------------------------------------------
  1    deny  std  18.28.236.77     0.0.0.0
  2    deny  std  18.29.140.107    0.0.0.0
  3  permit  std  0.0.0.0          255.255.255.255
```

```
ProCurve(config)# show access-list 105
Access Control Lists

   Name: 105
   Type: Extended
   Applied: No

 ID  action       IP               Mask               proto   oper   port(s)    Log
 ------------------------------------------------------------------------------------
  1  permit  src: 18.30.133.27     0.0.0.0              TCP   none      0
             dst: 0.0.0.0          255.255.255.255      TCP     eq     23
  2    deny  src: 18.30.133.1      0.0.0.255             IP                     log
             dst: 0.0.0.0          255.255.255.255       IP
  3  permit  src: 0.0.0.0          255.255.255.255       IP
             dst: 0.0.0.0          255.255.255.255       IP
```

Listing for a Standard ACL

Indicates whether the ACL is assigned to a VLAN.

Listing for an Extended ACL

Indicates whether the ACL is assigned to a VLAN.

Indicates source and destination entries in the ACL.

Indicates that the source TCP port can be any value.

Figure 9-20. Examples of Listings Showing the Content of Standard and Extended ACLs

Table 9-6. Descriptions of Data Types Included in Show Access-List < *acl-id* > Output

Field	Description
Name	The ACL identifier. Can be a number from 1 to 199, or a name.
Type	Standard or Extended. The former uses only source IP addressing. The latter uses both source and destination IP addressing and also allows TCP or UDP port specifiers.
Applied	“Yes” means the ACL has been applied to a VLAN. “No” means the ACL exists in the switch configuration, but has not been applied to any VLANs, and is therefore not in use.
ID	The sequential number of the Access Control Entry (ACE) in the specified ACL.
action	Permit (forward) or deny (drop) a packet when it is compared to the criteria in the applicable ACE and found to match.
IP	**In Standard ACLs:** The source IP address to which the configured mask is applied to determine whether there is a match with a packet. **In Extended ACLs:** The source and destination IP addresses to which the corresponding configured masks are applied to determine whether there is a match with a packet.
Mask	The mask configured in an ACE and applied to the corresponding IP address in the ACE to determine whether a packet matches the filtering criteria.
proto	Used only in extended ACLs to specify the packet protocol type to filter. Must be either IP, TCP, or UDP.
oper	Used only in extended ACLs where a TCP or UDP port type and number have been entered. Specifies how to compare the corresponding TCP or UDP port number in a packet to the port number in the ACE.
port(s)	Used only in extended ACLs to show any TCP or UDP port number that has been entered in the ACE.
Log	Shows the status of logging for the entry (ACE). A blank space indicates ACL logging is not enabled for that ACE.

Display All ACLs and Their Assignments in the Switch Startup-Config File and Running-Config File

The **show config** and **show running** commands include in their listings any configured ACLs and any ACL assignments to VLANs. Refer to figure 9-11 (page 9-36) and figure 9-12 (page 9-37) for examples. Remember that **show config** lists the startup-config file and **show running** lists the running-config file.

Editing ACLs and Creating an ACL Offline

Earlier sections of this chapter describe how to use the CLI to create an ACL. Beginning with “Using the CLI To Edit ACLs”, below, describes how to use the CLI to edit existing ACLs. However, you can also create or edit an ACL offline, then use a TFTP server to upload the ACL as a command file. The offline method (page 9-55) provides a useful alternative to using the CLI for creating or editing large ACLs.

Using the CLI To Edit ACLs

The switch applies individual ACEs in the order in which they occur in an ACL. You can use the CLI to delete individual ACEs from anywhere in an ACL and to append new ACEs to the end of an ACL. However, the CLI method does not allow you to insert a new ACE between two existing ACEs.

Using the CLI To Edit a Short ACL. To insert a new ACE between existing ACEs in a short ACL, you may want to delete the ACL and then re-configure it by entering your updated list of ACEs in the correct order.

Using the CLI to Edit a Longer ACL. To insert a new ACE between existing ACEs in a longer ACL:

a. Delete the first ACE that is out of sequence and all following ACEs through the end of the ACL.

b. Re-Enter the desired ACEs in the correct sequence.

General Editing Rules

■ You can delete any ACE from an ACL by repeating the ACE’s entry command, preceded by the “no” statement. When you enter a new ACE, the switch inserts it as the last entry of the specified ACL.

- Deleting the last ACE from a *numeric* ACL, removes the ACL from the configuration. Deleting the last ACE from a *named* ACL leaves the ACL in memory. In this case, the ACL is "empty" and cannot perform any filtering tasks. (In any ACL the implicit "deny any" does not apply unless the ACL includes at least one explicit ACE.)
- When you create a new ACL, the switch inserts it as the last ACL in the startup-config file. (Executing **write memory** saves the running-config file to the startup-config file.)

Deleting Any ACE from an ACL

You can delete an ACE from an ACL by repeating the ACE's entry command, preceded by the "**no**" statement.

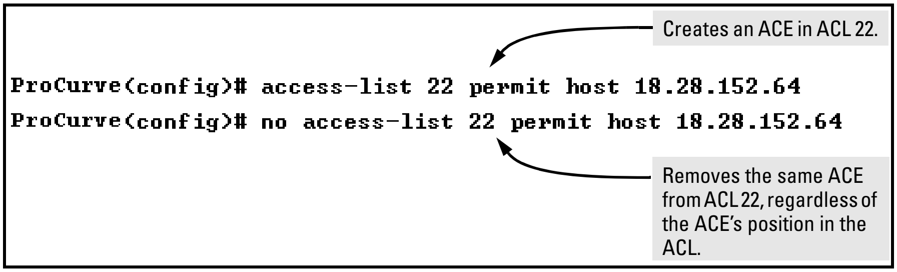
Syntax: no access-list < *acl-id* > < permit | deny > < any | host | *ip-addr/mask-length* >

Deletes an ACE from a standard ACL. All variable parameters in the command must be an exact match with their counterparts in the ACE you want to delete.

no access-list < *acl-id* > < permit | deny > < ip | tcp | udp >
< src-addr: any | host | *ip-addr/mask-length* > [operator < src-port-num >]
< dest-addr: any | host | ip-addr-mask-length > [operator < dest-port-num >
[log]

Deletes an ACE from a standard ACL. All variable parameters in the command must be an exact match with their counterparts in the ACE you want to delete.

For example, the first of the following two commands creates an ACE in ACL 22 and the second deletes the same ACE:



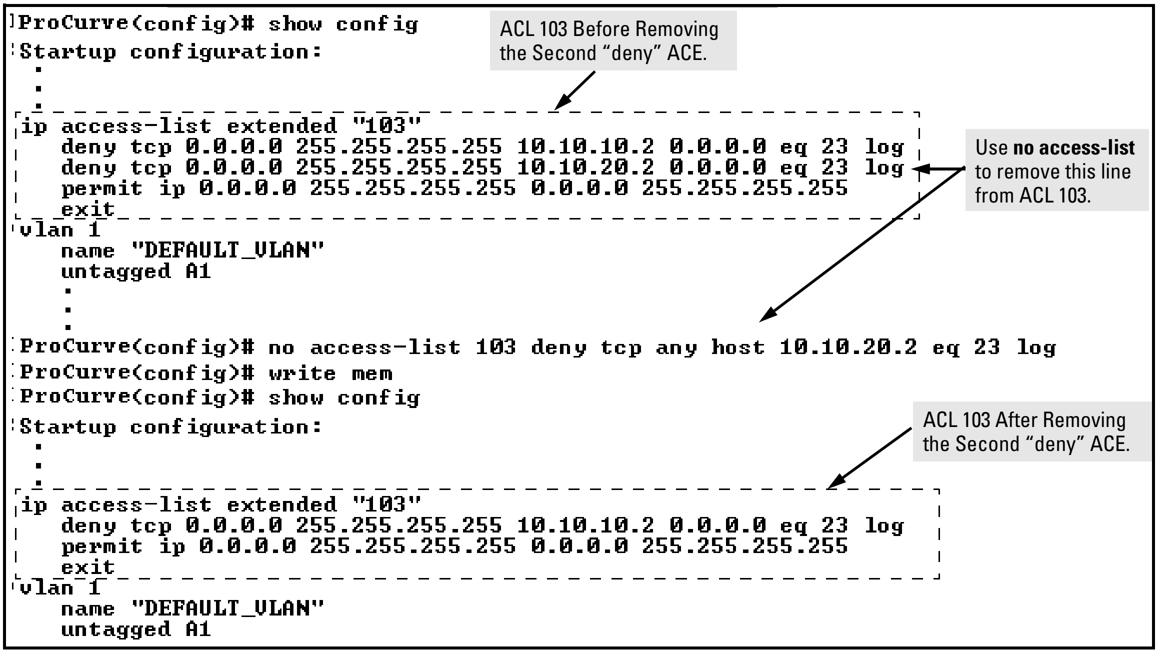
Creates an ACE in ACL 22.

```
ProCurve(config)# access-list 22 permit host 18.28.152.64
ProCurve(config)# no access-list 22 permit host 18.28.152.64
```

Removes the same ACE from ACL 22, regardless of the ACE's position in the ACL.

Figure 9-21. Example of Deleting an ACE from a Standard ACL

Figure 9-22 shows an example of deleting an ACE from an extended ACL.



```
ProCurve(config)# show config
Startup configuration:
 .
 .
 .
ip access-list extended "103"
   deny tcp 0.0.0.0 255.255.255.255 10.10.10.2 0.0.0.0 eq 23 log
   deny tcp 0.0.0.0 255.255.255.255 10.10.20.2 0.0.0.0 eq 23 log
   permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
   exit
vlan 1
   name "DEFAULT_VLAN"
   untagged A1
   .
   .
   .
ProCurve(config)# no access-list 103 deny tcp any host 10.10.20.2 eq 23 log
ProCurve(config)# write mem
ProCurve(config)# show config
Startup configuration:
 .
 .
 .
ip access-list extended "103"
   deny tcp 0.0.0.0 255.255.255.255 10.10.10.2 0.0.0.0 eq 23 log
   permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
   exit
vlan 1
   name "DEFAULT_VLAN"
   untagged A1
```

ACL 103 Before Removing the Second “deny” ACE.

Use **no access-list** to remove this line from ACL 103.

ACL 103 After Removing the Second “deny” ACE.

Figure 9-22. Example of Deleting an ACE from an ACL

Working Offline To Create or Edit an ACL

For longer ACLs that would be difficult or time-consuming to accurately create or edit in the CLI, you can use the offline method:

1. Begin by doing one of the following:
   - To edit one or more existing ACLs, use **copy command-output tftp** to copy the current version of the ACL configuration to a file in your TFTP server. For example, to copy the ACL configuration to a file named **acl02.txt** in the TFTP directory on a server at 18.28.227.2:

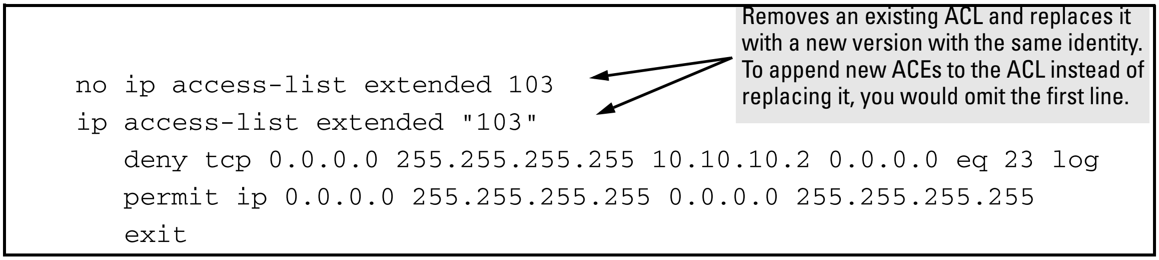
     ```
     ProCurve# copy command-output 'show access-list
     config' tftp 18.28.227.2 acl02.txt pc
     ```
   - To create a new ACL, just open a text file in the appropriate directory on a TFTP server accessible to the switch.
2. Use the text editor to create or edit the ACL(s).
3. Use **copy tftp command-file** to download the file as a list of commands to the switch.

Creating an ACL Offline

Use a text editor that allows you to create an ASCII text file (.txt).

If you are replacing an ACL on the switch with a new ACL that uses the same number or name syntax, begin the command file with a "no" command to remove the earlier version of the ACL from the switch's running-config file. Otherwise, the switch will append the new ACEs in the ACL you download to the existing ACL. For example, if you plan to use the Copy command to *replace* ACL "103", you would place this command at the beginning of the edited file:

```
no ip access-list extended 103
```



```
no ip access-list extended 103
ip access-list extended "103"
   deny tcp 0.0.0.0 255.255.255.255 10.10.10.2 0.0.0.0 eq 23 log
   permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
   exit
```

Removes an existing ACL and replaces it with a new version with the same identity. To append new ACEs to the ACL instead of replacing it, you would omit the first line.

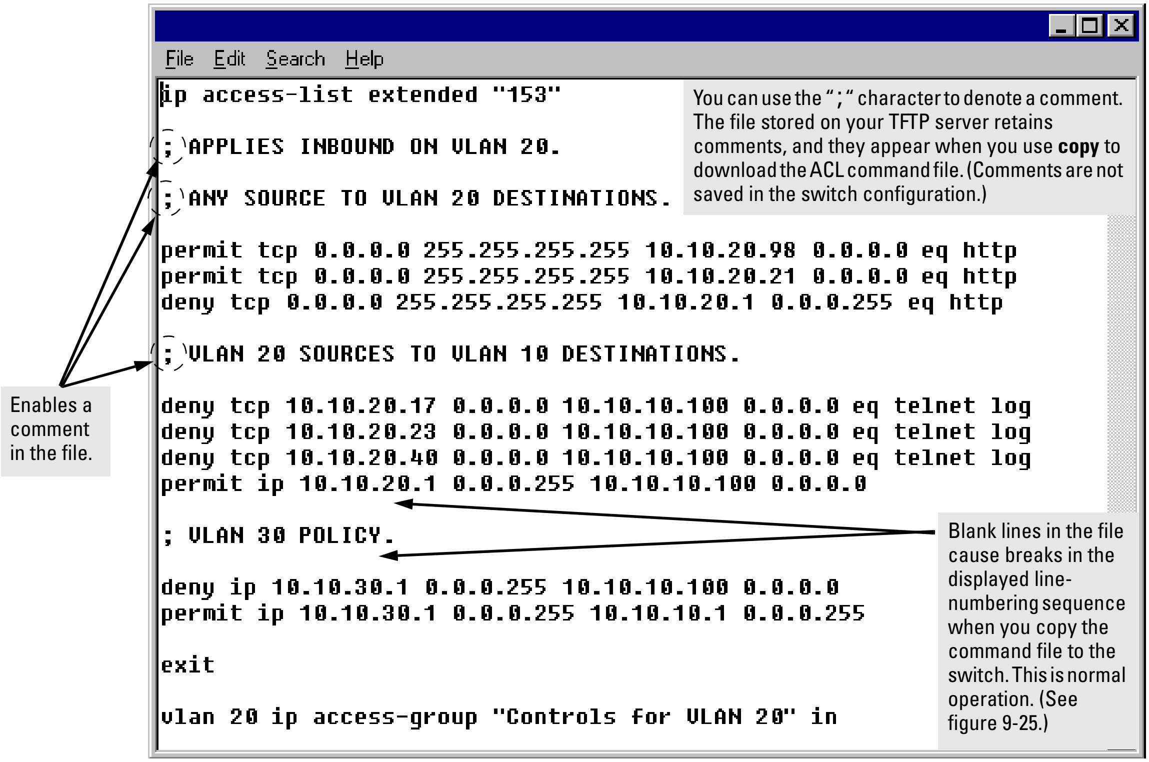
Figure 9-23. Example of an Offline ACL File Designed To Replace An Existing ACL

For example, suppose that you wanted to create an extended ACL to fulfill the following requirements (Assume a subnet mask of 255.255.255.0.):

- ID: "Controls for VLAN 20"
- Deny Telnet access to a server at 10.10.10.100 on VLAN 10 from these three IP addresses on VLAN 20 (with ACL logging):
  - 10.10.20.17
  - 10.10.20.23
  - 10.10.20.40
- Allow any access to the server from all other addresses on VLAN 20:
- Permit internet access to these two IP address on VLAN 20, but deny access to all other addresses on VLAN 20 (without ACL logging).
  - 10.10.20.98
  - 10.10.20.21
- Deny all other traffic from VLAN 20 to VLAN 10.

- Deny all traffic from VLAN 30 (10.10.30.0) to the server at 10.10.10.100 on VLAN 10 (without ACL logging), but allow any other traffic from VLAN 30 to VLAN 10.
- Deny all other inbound traffic to VLAN 20. (Hint: The implicit "deny any" can achieve this objective.)

1. You would create a **.txt** file with the content shown in figure 9-24.



```
ip access-list extended "153"

; APPLIES INBOUND ON VLAN 20.

; ANY SOURCE TO VLAN 20 DESTINATIONS.

permit tcp 0.0.0.0 255.255.255.255 10.10.20.98 0.0.0.0 eq http
permit tcp 0.0.0.0 255.255.255.255 10.10.20.21 0.0.0.0 eq http
deny tcp 0.0.0.0 255.255.255.255 10.10.20.1 0.0.0.255 eq http

; VLAN 20 SOURCES TO VLAN 10 DESTINATIONS.

deny tcp 10.10.20.17 0.0.0.0 10.10.10.100 0.0.0.0 eq telnet log
deny tcp 10.10.20.23 0.0.0.0 10.10.10.100 0.0.0.0 eq telnet log
deny tcp 10.10.20.40 0.0.0.0 10.10.10.100 0.0.0.0 eq telnet log
permit ip 10.10.20.1 0.0.0.255 10.10.10.100 0.0.0.0

; VLAN 30 POLICY.

deny ip 10.10.30.1 0.0.0.255 10.10.10.100 0.0.0.0
permit ip 10.10.30.1 0.0.0.255 10.10.10.1 0.0.0.255

exit

vlan 20 ip access-group "Controls for VLAN 20" in
```

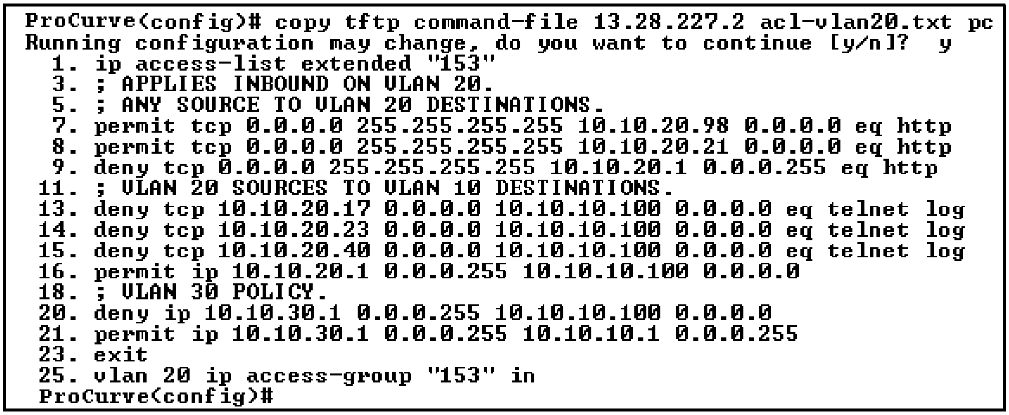
You can use the ";" character to denote a comment. The file stored on your TFTP server retains comments, and they appear when you use **copy** to download the ACL command file. (Comments are not saved in the switch configuration.)

Enables a comment in the file.

Blank lines in the file cause breaks in the displayed line-numbering sequence when you copy the command file to the switch. This is normal operation. (See figure 9-25.)

Figure 9-24. Example of a.txt File Designed for Creating an ACL

2. After you copy the above .txt file to a TFTP server the switch can access, you would then execute the following command:



```
ProCurve(config)# copy tftp command-file 13.28.227.2 acl-vlan20.txt pc
Running configuration may change, do you want to continue [y/n]?  y
  1. ip access-list extended "153"
  3. ; APPLIES INBOUND ON VLAN 20.
  5. ; ANY SOURCE TO VLAN 20 DESTINATIONS.
  7. permit tcp 0.0.0.0 255.255.255.255 10.10.20.98 0.0.0.0 eq http
  8. permit tcp 0.0.0.0 255.255.255.255 10.10.20.21 0.0.0.0 eq http
  9. deny tcp 0.0.0.0 255.255.255.255 10.10.20.1 0.0.0.255 eq http
 11. ; VLAN 20 SOURCES TO VLAN 10 DESTINATIONS.
 13. deny tcp 10.10.20.17 0.0.0.0 10.10.10.100 0.0.0.0 eq telnet log
 14. deny tcp 10.10.20.23 0.0.0.0 10.10.10.100 0.0.0.0 eq telnet log
 15. deny tcp 10.10.20.40 0.0.0.0 10.10.10.100 0.0.0.0 eq telnet log
 16. permit ip 10.10.20.1 0.0.0.255 10.10.10.100 0.0.0.0
 18. ; VLAN 30 POLICY.
 20. deny ip 10.10.30.1 0.0.0.255 10.10.10.100 0.0.0.0
 21. permit ip 10.10.30.1 0.0.0.255 10.10.10.1 0.0.0.255
 23. exit
 25. vlan 20 ip access-group "153" in
 ProCurve(config)#
```

Figure 9-25. Example of Using "copy tftp command-file" To Configure an ACL in the Switch

Note If a transport error occurs, the switch does not execute the command and the ACL is not configured.

3. Next, assign the new ACL to the intended VLAN which, in this example, is for inbound traffic on VLAN 20.

```
ProCurve(config)# vlan 20 ip access-group "Controls
for VLAN 20" in
```

4. Inspect the new running configuration:

```
ProCurve(config)# show running
```

5. If the configuration appears satisfactory, save it to the startup-config file:

```
ProCurve(config)# write memory
```

Enable ACL "Deny" Logging

ACL logging enables the switch to generate a message when IP traffic meets the criteria for a match with an ACE that results in an explicit "deny" action. You can use ACL logging to help:

- Test your network to ensure that your ACL configuration is detecting and denying the traffic you do not want forwarded
- Receive notification when the switch detects attempts to transmit traffic you have designed your ACLs to reject

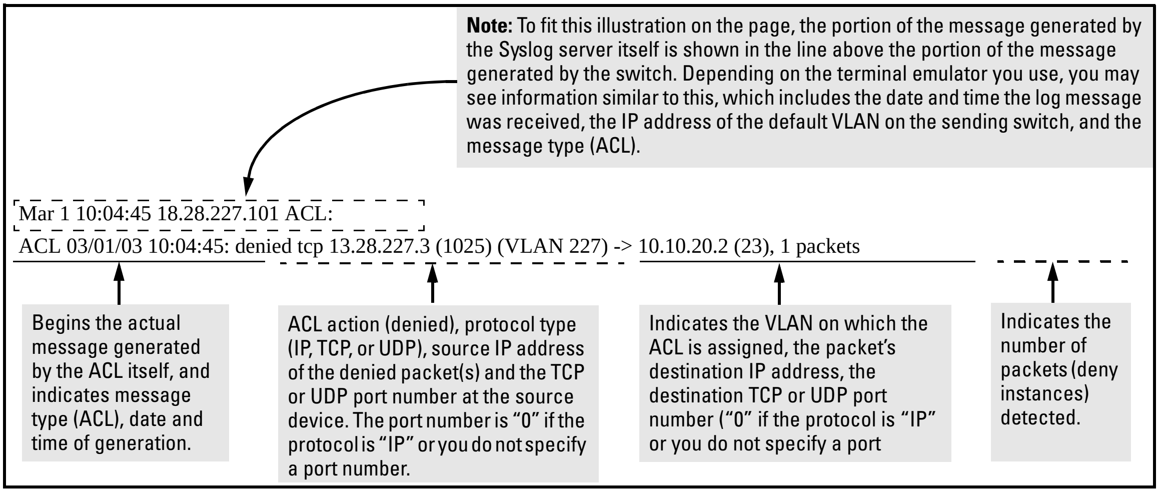
The switch sends ACL messages to Syslog and optionally to the current console, Telnet, or SSH session. You can configure up to six Syslog server destinations.

Requirements for Using ACL Logging

- The switch configuration must include an ACL (1) assigned to a static VLAN and (2) containing an ACE configured with the **deny** action and the **log** option.
- To screen routed packets with destination IP addresses outside of the switch, IP routing must be enabled.
- For ACL logging to a Syslog server, the server must be accessible to the switch and identified (with the **logging <** ***ip-addr*** **>** command) in the switch configuration.
- Debug must be enabled for ACLs and one or both of the following:
  - logging (for sending messages to Syslog)
  - Session (for sending messages to the current console interface)

ACL Logging Operation

When the switch detects a packet match with an ACE and the ACE includes both the **deny** action and the optional **log** parameter, an ACL log message is sent to the designated debug destination. The first time a packet matches an ACE with **deny** and **log** configured, the message is sent immediately to the destination and the switch starts a wait-period of approximately five minutes. (The exact duration of the period depends on how the packets are internally routed.) At the end of the collection period, the switch sends a single-line summary of any additional "deny" matches for that ACE (and any other "deny" ACEs for which the switch detected a match). If no further log messages are generated in the wait-period, the switch suspends the timer and resets itself to send a message as soon as a new "deny" match occurs. The data in the message includes the information illustrated in figure 9-26.



Note: To fit this illustration on the page, the portion of the message generated by the Syslog server itself is shown in the line above the portion of the message generated by the switch. Depending on the terminal emulator you use, you may see information similar to this, which includes the date and time the log message was received, the IP address of the default VLAN on the sending switch, and the message type (ACL).

```
Mar 1 10:04:45 18.28.227.101 ACL:
ACL 03/01/03 10:04:45: denied tcp 13.28.227.3 (1025) (VLAN 227) -> 10.10.20.2 (23), 1 packets
```

Begins the actual message generated by the ACL itself, and indicates message type (ACL), date and time of generation.

ACL action (denied), protocol type (IP, TCP, or UDP), source IP address of the denied packet(s) and the TCP or UDP port number at the source device. The port number is "0" if the protocol is "IP" or you do not specify a port number.

Indicates the VLAN on which the ACL is assigned, the packet's destination IP address, the destination TCP or UDP port number ("0" if the protocol is "IP" or you do not specify a port

Indicates the number of packets (deny instances) detected.

Figure 9-26. Content of an ACL-Generated Message

Enabling ACL Logging on the Switch

1. Use the debug command to:
   a. Configure one or more log destinations.
   b. If you are using a Syslog server, use the **logging** command to configure the server's IP address. (You can configure up to six Syslog servers.)
   c. Ensure that the switch can access any Syslog servers you specify.
2. Configure one or more ACLs with the deny action and the log option.

For example, suppose that you want to:

- On VLAN 100 configure an extended ACL with an ACL-ID of 143 to deny Telnet traffic from IP address 18.38.100.127 on VLAN 100.
- Configure the switch to send an ACL log message to the console and to a Syslog server at IP address 18.38.110.54 on VLAN 110 if the switch detects a match denying Telnet access from 18.38.100.127.

(This example assumes that IP routing is already configured on the switch.)

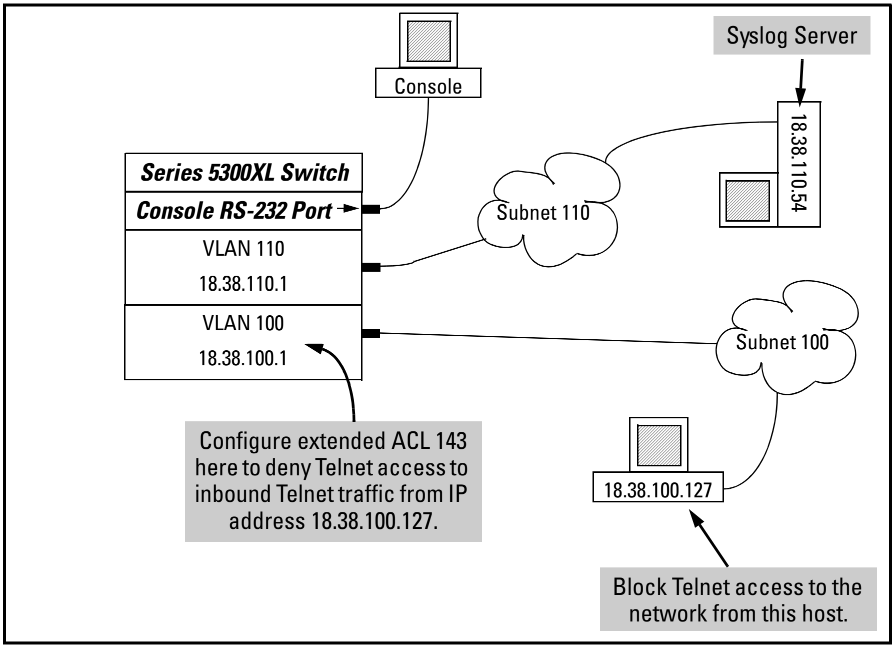
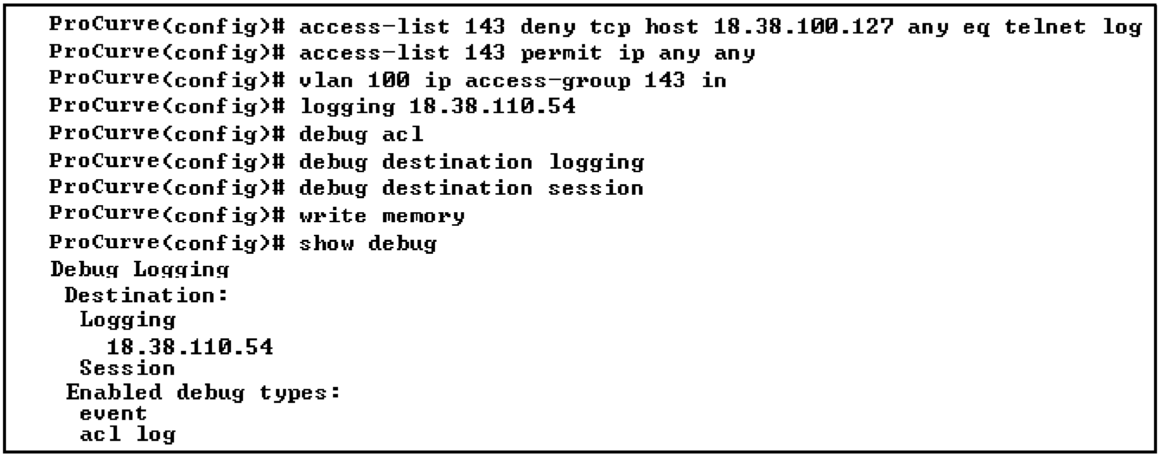

Console
Syslog Server
18.38.110.54
Series 5300XL Switch
Console RS-232 Port
Subnet 110
VLAN 110
18.38.110.1
VLAN 100
18.38.100.1
Subnet 100
Configure extended ACL 143 here to deny Telnet access to inbound Telnet traffic from IP address 18.38.100.127.
18.38.100.127
Block Telnet access to the network from this host.

Figure 9-27. Example of an ACL Log Application



```
ProCurve(config)# access-list 143 deny tcp host 18.38.100.127 any eq telnet log
ProCurve(config)# access-list 143 permit ip any any
ProCurve(config)# vlan 100 ip access-group 143 in
ProCurve(config)# logging 18.38.110.54
ProCurve(config)# debug acl
ProCurve(config)# debug destination logging
ProCurve(config)# debug destination session
ProCurve(config)# write memory
ProCurve(config)# show debug
Debug Logging
 Destination:
  Logging
    18.38.110.54
  Session
 Enabled debug types:
  event
  acl log
```

Figure 9-28. Commands for Applying an ACL with Logging to Figure 9-27

Operating Notes for ACL Logging

- The ACL logging feature generates a message only when packets are explicitly denied as the result of a match, and not when explicitly permitted or implicitly denied. To help test ACL logging, configure an ACL with an explicit **deny any** and **log** statements at the end of the list, and apply the ACL to an appropriate VLAN.

- Logging enables you to selectively test specific devices or groups. However, excessive logging can affect switch performance. For this reason, HP recommends that you remove the logging option from ACEs for which you do not have a present need. Also, avoid configuring logging where it does not serve an immediate purpose. (Note that ACL logging is not designed to function as an accounting method.) See also "Apparent Failure To Log All "Deny" Matches" in the section titled “ACL Problems”, found in appendix C, “Troubleshooting” of the Management and Configuration Guide for your switch.

- When configuring logging, you can reduce excessive use by configuring the appropriate ACEs to match with specific hosts instead of entire subnets.

General ACL Operating Notes

ACLs do not provide DNS hostname support.

Protocol Support: ACL criteria includes IP, TCP, and UDP. ACLs do not use these protocols:

- TOS (Type-of-Service)
- Precedence
- MAC information
- QoS

ACLs do not affect switch serial port access.

When the ACL configuration includes TCP or UDP options, the switch operates in "strict" TCP and UDP mode for increased control. The switch compares all TCP and UDP packets against the ACLs. (In the ProCurve Series 9300 Routing Switches, the Strict TCP and Strict UDP modes are optional and must be specifically invoked.)

Replacing or Adding To an Active ACL Policy. If you assign an ACL to a VLAN and subsequently add or replace ACEs in that ACL, each new ACE becomes active when you enter it.

Note

When an ACE becomes active, it screens the packets resulting from new traffic connections. It does not screen packets resulting from currently open traffic connections. If you invoke a new ACE to screen packets in a currently open traffic connection, you must force the connection to close before the ACE can begin screening packets from that source.

ACL Screening of Traffic Generated by the Switch. Outbound ACLs on a switch do not screen traffic (such as broadcasts, Telnet, Ping, and ICMP replies) *generated by the switch itself*. Note that ACLs do screen this type of traffic when other devices generate it. Similarly, ACLs can screen responses from other devices to unscreened traffic the switch generates.

— This page is intentionally unused. —