Applying Access Control to Router Interfaces

Contents

Access Control for Interfaces on the ProCurve Secure Router ............... 5-3
Access Control Mechanisms ........................................... 5-4
Using ACLs Alone to Configure Access Control ......................... 5-6
  Configure ACLs ......................................................... 5-6
    ACL Entries ......................................................... 5-6
    Types of ACLs ...................................................... 5-7
    Creating an ACL ..................................................... 5-9
    Creating a Standard ACL .......................................... 5-9
    Creating an Extended ACL ....................................... 5-11
    Entry Order ........................................................ 5-16
    Adding a Descriptive Tag to an ACL .............................. 5-17
    Editing an Existing ACL .......................................... 5-18
    Deleting an Existing ACL ......................................... 5-18
Applying the ACL to an Interface ....................................... 5-19
  Selecting the Packet and Controlling the Action .................... 5-20
Controlling FTP, HTTP, and Telnet Access to the Router ................. 5-21
  Restricting FTP Access ............................................. 5-22
  Restricting HTTP Access ........................................... 5-22
  Restricting Telnet Access .......................................... 5-23
Examples of Applying ACLs ............................................. 5-23
Using ACPs to Control Access to Router Interfaces ...................... 5-26
  Enable the Firewall ................................................ 5-26
Configure ACLs ......................................................... 5-27
  Types of ACLs ......................................................... 5-27
  Creating an ACL ...................................................... 5-29
  Creating a Standard ACL .......................................... 5-29
  Creating an Extended ACL ....................................... 5-32
Applying Access Control to Router Interfaces

Contents

Configure ACPs .................................................. 5-35
Action ........................................... 5-35
Selector ........................................ 5-36
Creating an ACP ........................................ 5-36
Creating Entries in the ACP ....................... 5-37
Editing ACPs ........................................ 5-37
Deleting an ACP ........................................ 5-37
Assigning the ACP to an Interface .................. 5-38
Using the reload Command ......................... 5-38
Processing ACPs ........................................ 5-39
ACP Action Summary ................................ 5-42
Traffic Flow through Interfaces with ACPs ........ 5-44
Inbound Interface Has an ACP; Outbound Interface Does Not Have an ACP .......... 5-45
Inbound Interface Has an ACP; Outbound Interface Has a Different ACP .......... 5-45
Inbound Interface Does Not Have an ACP; Outbound Interface Has an ACP .......... 5-46
Traffic in and out Through a Single Interface .... 5-47
Examples of ACPs ........................................ 5-47
Viewing ACLs and ACPs ............................... 5-51
Displaying ACLs ........................................ 5-51
Displaying ACPs ........................................ 5-52
Viewing Access Policy Sessions ...................... 5-53
Viewing Access Policy Statistics .................... 5-54
Troubleshooting ........................................... 5-56
show Commands ........................................ 5-56
Monitoring Packets Matched to an ACP ............. 5-56
Clearing Existing Policy Sessions ................... 5-56
Clear ACL Counters ..................................... 5-58
Debug ACLs ........................................... 5-59
Quick Start .............................................. 5-60
Enabling the Built-in Firewall ....................... 5-60
Configuring an ACL and Applying It Directly to an Interface .......... 5-61
Configuring ACPs ....................................... 5-63
Access Control for Interfaces on the ProCurve Secure Router

In addition to blocking known cyber attacks with its stateful-inspection firewall, the ProCurve Secure Router OS can filter both inbound and outbound traffic, enabling you to control the traffic that enters and exits your corporate network. You can filter traffic based on specific types of criteria:

- source IP address
- source and destination IP addresses
- IP protocols
- UDP and TCP port numbers

For example, you can block certain types of traffic, such as FTP or Telnet traffic, from entering your corporate network, or you can permit traffic from only certain IP addresses. You can also limit WAN traffic to certain network segments on your internal network. This granularity allows you to configure the router to enforce your company’s security guidelines—no matter what those guidelines are.

Before you begin to configure access controls on the router, you should evaluate your security guidelines and determine the type of traffic that you want to allow or discard on each interface. Then when you begin to configure your access policies, you will not mistakenly block traffic that should be permitted to enter an interface, and you will not allow traffic that could be a security risk to your company.

You can use Table 5-1 to begin the process of evaluating traffic that is entering and exiting each interface. A general example is provided for you, illustrating the process of determining the traffic that users need to transmit and the traffic that could create a security risk. The more detailed you are in evaluating traffic, the more equipped you are to apply access controls on each interface.
Access Control Mechanisms

The ProCurve Secure Router OS provides two mechanisms for filtering traffic:

■ access control lists (ACLs)
■ access control policies (ACPs)

You can use ACLs alone or in combination with ACPs. By themselves, ACLs are easier to configure than ACPs. If you have experience configuring access control on Cisco routers, you will find that the process of configuring ACLs on the ProCurve Secure Router is very similar.

In addition, when you use ACLs by themselves, you can filter both incoming and outgoing traffic on an interface. When you use ACPs, on the other hand, you can filter only incoming traffic on an interface.

In general, however, ACPs provide more flexibility in configuring access controls than ACLs do. For example, using an ACP, you can apply more than two ACLs to an interface. You can essentially apply an unlimited number of ACLs to an interface.
ACPs also allow you to perform certain actions on traffic that ACLs do not. For example, you must use an ACP to configure Network Address Translation (NAT) on the ProCurve Secure Router. (For more information about NAT, see Chapter 6: Configuring Network Address Translation.)

Table 5-2 lists the main differences between ACLs and ACPs.

**Table 5-2. Comparing ACLs and ACPs**

<table>
<thead>
<tr>
<th>Access Control Mechanism</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| ACLs                     | • easier to configure  
                          • filter both incoming and outgoing traffic on an interface  
                          • support only one ACL to filter the incoming traffic on an interface and one ACL to filter the outgoing traffic on an interface | |
| ACPs                     | • provide more flexibility  
                          • allow you to NAT traffic  
                          • allow you to apply an unlimited number of ACLs to an interface | • can be more difficult to understand and configure |

The remainder of this chapter is divided into four main sections:

- using ACLs alone to configure access controls on router interfaces
- using ACPs to configure access controls on router interfaces
- viewing ACLs and ACPs
- troubleshooting ACLs and ACPs
Using ACLs Alone to Configure Access Control

When you use ACLs alone to configure access controls on router interfaces, you must complete two main steps:

1. Configure the ACL.
2. Apply the ACL directly to an interface.

Note

You do not have to enable the ProCurve Secure Router OS firewall if you use ACLs alone to configure access controls. If you use ACPs, you must enable the firewall.

Configure ACLs

You can create and apply two ACLs to each interface:
- one ACL to control incoming traffic
- one ACL to control outgoing traffic

If you apply ACLs directly to router interfaces, the ProCurve Secure Router uses the ACL to both select the traffic and to perform the action on that traffic.

ACL Entries

ACLs are composed of an ordered list of entries, and each entry contains two parts: an action and a packet pattern.

Action. You can define one of two actions for an entry:
- permit
- deny

Packet Pattern. You can define patterns based on:
- source IP address
- source and destination IP addresses
- IP protocols
- TCP or UDP ports
Using ACLs Alone to Configure Access Control

For example, an ACL could include entries such as:

deny host 192.168.115.91
deny host 192.168.44.53
permit 192.168.115.0 0.0.0.255
permit 192.168.44.0 0.0.0.255

The first two entries deny access to the devices with the IP addresses 192.168.115.91 and 192.168.44.53. The last two entries permit access to two subnets: 192.168.115.0 /24 and 192.168.44.0 /24.

Types of ACLs

The ProCurve Secure Router supports two types of ACLs:
- standard
- extended

A standard ACL matches only one packet pattern: the source IP address. An extended ACL matches more complex packet patterns:
- source address and destination address
- IP protocols
- TCP and UDP ports

You should create a standard ACL if you want to select traffic based only on the source IP address. (See Figure 5-1.) If you want to select traffic based on other fields in the IP, TCP, or UDP header or if you want the Secure Router OS to filter traffic based on the both the source and destination IP addresses, you must create an extended ACL. (See Figure 5-2.)
Applying Access Control to Router Interfaces
Using ACLs Alone to Configure Access Control

Figure 5-1. With Standard ACLs, the ProCurve Secure Router Checks Only the Source Address

Figure 5-2. With Extended ACLs, the ProCurve Secure Router Checks Both the Source and the Destination Address and, Optionally, the Protocol and the Port
Creating an ACL

To create an ACL, you enter the `ip access-list` command from the global configuration mode context:

**Syntax:** `ip access-list [standard | extended] <listname>`

Enter either the **standard** or **extended** option, depending on the type of ACL you are configuring, and replace `<listname>` with an alphanumeric descriptor that is meaningful to you. The listname is case sensitive.

Creating a Standard ACL

To create a standard ACL, enter:

ProCurve(config)# ip access-list standard <listname>

After you enter this command, you are moved to the standard ACL configuration mode context, as shown below:

ProCurve(config-std-nacl)#

**Permit or Deny Traffic.** You can now begin to enter permit and deny entries. The ACL is empty until you add these entries.

To create permit and deny entries for standard ACLs, you use the following command syntax:

**Syntax:** `[permit | deny] <source address> [log]`

To specify the `<source address>` use the following syntax:

[any | host [A.B.C.D] | hostname] | A.B.C.D <wildcard bits>

Table 5-3 explains the options for specifying the source address.

**Table 5-3. Options for Specifying the Source Address**

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>any</td>
<td>matches all hosts</td>
</tr>
<tr>
<td>host [A.B.C.D]</td>
<td>hostname]</td>
</tr>
<tr>
<td>A.B.C.D</td>
<td></td>
</tr>
<tr>
<td>A.B.C.D &lt;wildcard bits&gt;</td>
<td></td>
</tr>
</tbody>
</table>
For example, if you want to permit all traffic that enters through the Ethernet interface, you create a permit entry in the ACL:

```
ProCurve(config-std-nacl)# permit any
```

You can also permit or deny a specific host:

```
ProCurve(config-std-nacl)# permit host <A.B.C.D>
ProCurve(config-std-nacl)# deny host <A.B.C.D>
```

For example, if you want to deny a host with the IP address of 192.168.115.90, enter:

```
ProCurve(config-std-nacl)# deny host 192.168.115.90
```

If you want to permit a host with the hostname user1.procurve.com, enter:

```
ProCurve(config-std-nacl)# permit hostname user1.procurve.com
```

You can also omit the `host` keyword to permit or deny a specific IP address:

```
ProCurve(config-std-nacl)# permit 192.168.115.80
ProCurve(config-std-nacl)# deny 192.168.115.80
```

**Use Wildcard Bits.** You can use wildcard bits to permit or deny a range of IP addresses. Wildcard bits define which address bits the Secure Router OS should match and which address bits it should ignore. Essentially, you use the wildcard bits to specify the subnet to which you want the Secure Router OS to match packets.

When you enter wildcard bits, you use a 0 to indicate that the Secure Router OS should match the corresponding bit in the IP address. You use a 1 to indicate that the Secure Router OS can ignore the corresponding bit in the IP address. In other words, the Secure Router OS does not have to match that bit.

For example, you might enter:

```
ProCurve(config-std-nacl)# deny 192.168.1.0 0.0.0.255
```

If you enter 192.168.1.90 with the wildcard bits 0.0.0.255, the Secure Router OS will not match any address bits in the fourth octet of the IP address. The Secure Router OS will match incoming packets to the IP subnet address 192.115.1.0 /24. (because it will not match the bits in the fourth octet). (See Figure 5-3.)
Applying Access Control to Router Interfaces
Using ACLs Alone to Configure Access Control

As a general rule, you should specify the network address for the subnet you are using the wildcard bits to select. Adding the wildcard bits to the network address gives you the last address in the range. For example, enter `permit 192.168.1.0 0.0.0.255` to permit traffic with any source address between 192.168.1.0 and 192.168.1.255.

![Figure 5-3. Understanding Wildcard Bits](image)

Implicit “deny any” Entry. Each ACL includes an implicit “deny any” entry at the end of the list. If a packet does not match any entry in the ACL you create, it matches the implicit “deny any” entry.

When you configure a standard ACL and apply it to an interface, you should permit at least one host. Otherwise, you will, in effect, shut down the interface, preventing any traffic from entering it.

Selecting the log Option. Include the `log` option if you want the Secure Router OS to log a message when these two conditions are met:

- `debug access-list` is enabled for this ACL
- a packet matches this ACL

Exit the ACL

After you have finished creating the ACL, enter `exit` to return to the global configuration mode context, as shown below:

```
ProCurve(config-std-nacl)# exit
ProCurve(config)#
```

Creating an Extended ACL

To create an extended ACL, you enter the following command from the global configuration mode context:

**Syntax:** `ip access-list extended <listname>`
Replace `<listname>` with an alphanumeric descriptor that is meaningful to you. The name is case sensitive.

After you enter this command, you are moved to the extended ACL configuration mode context, as shown below:

ProCurve(config-ext-nacl)#

**Permit or Deny Traffic.** You can now begin to enter permit and deny entries. The ACL is empty until you add these entries.

To create permit and deny entries for extended ACLs, you use the following command syntax:

**Syntax:** `[permit | deny] <protocol> <source address> <source port> <destination address> <destination port> [<packet bits>] [log | log-input]

You must specify a `<protocol>`, `<source address>`, and `<destination address>`. However, the following are optional:

- `<source port>` for TCP or UDP traffic
- `<destination port>` for TCP or UDP traffic
- `<packet bits>`
- `[log | log-input]`

All of the command options are explained in the sections that follow.

**Specify a Protocol.** When you configure extended ACLs, you must specify a protocol. Valid protocols include:

- AH (ahp)
- ESP (esp)
- GRE (gre)
- ICMP (icmp)
- IP (ip)
- TCP (tcp)
- UDP (udp)

You can also specify the number of the protocol. Valid numbers include any number between 0 and 255.

**Defining the Source and Destination Addresses.** You must configure both a source and a destination address for each entry. When you create entries in an extended ACL, remember that you always specify the source address first, and then you specify the destination address.
Applying Access Control to Router Interfaces
Using ACLs Alone to Configure Access Control

To specify a source or destination address, you use the following syntax:

\[
\text{[any | host } <A.B.C.D> \text{ | hostname } <\text{hostname}> \text{ | } <A.B.C.D> <\text{wildcard bits}>]\]

Table 5-4 lists the options you have for specifying both the source address and the destination address.

### Table 5-4. Options for Specifying Source and Destination Addresses

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>any</td>
<td>matches all hosts</td>
</tr>
<tr>
<td>host &lt;A.B.C.D&gt;</td>
<td>specifies a single IP address or a single host</td>
</tr>
<tr>
<td>hostname &lt;hostname&gt;</td>
<td>specifies a single host, using its hostname rather than its IP address</td>
</tr>
<tr>
<td>&lt;A.B.C.D&gt; &lt;wildcard bits&gt;</td>
<td>specifies a range of IP addresses</td>
</tr>
</tbody>
</table>

For example, if you want to permit all TCP traffic from any source to any destination, you enter:

\[\text{ProCurve(config-ext-nacl)# permit tcp any}\]

To deny all ICMP traffic from a specific host, such as host 192.168.1.1, to any destination, you enter:

\[\text{ProCurve(config-ext-nacl)# deny icmp host 192.168.1.1 any}\]

To deny ICMP traffic from a range of IP addresses to a specific destination, enter:

**Syntax:** deny icmp <A.B.C.D> <wildcard bits> host <A.B.C.D>

Replace the first <A.B.C.D> with the IP address that represents the range of IP address that, in this case, you want to block. For example, you may want to block IP addresses from 192.168.1.0/24. Then replace <wildcard bits> with a reverse logic mask so that the router will check the appropriate part of the IP address.

For example, if you want to block the entire 192.168.1.0/24 network, you might enter the wildcard bit 0.0.0.255. (For more information about wildcard bits, see Figure 5-3 on page 5-11.)
Applying Access Control to Router Interfaces
Using ACLs Alone to Configure Access Control

Replace the second \(<A.B.C.D>\) with the IP address for the destination device. For example, if you want to block all traffic from the 192.168.1.0 /24 network to the server with the IP address 10.15.1.1, you would replace \(<A.B.C.D>\) with 10.15.1.1.

**Specifying a Source or Destination Port for TCP and UDP.** If you are configuring ACL entries to select TCP or UDP traffic, you can also specify source and destination ports—although this is optional. For example, you could specify the well-known port 80 for HTTP traffic if you wanted to permit only this type of traffic on this port.

There is a drawback to using a port number, however. The Secure Router OS will match the type of traffic only on that port. If a device transmits the traffic you are targeting on another port, the Secure Router OS will not match that traffic to your ACL.

To view the options available for specifying ports, enter one of the following:

```
ProCurve(config-ext-nacl)# permit tcp any ?
ProCurve(config-ext-nacl)# deny tcp any ?
ProCurve(config-ext-nacl)# permit udp any ?
ProCurve(config-ext-nacl)# deny udp any ?
```

In practice, you would use the `any` keyword only if you want to match all traffic from a particular port. You can also view options for selecting the port by entering the `? help` command after specifying a particular source or destination. For example:

```
ProCurve(config-ext-nacl)# permit tcp 192.168.1.0 0.0.0.255 ?
ProCurve(config-ext-nacl)# deny udp any host 192.168.10.1 ?
```

Table 5-5 shows the options you have for specifying ports.
Applying Access Control to Router Interfaces
Using ACLs Alone to Configure Access Control

Table 5-5. Specifying Ports in Extended ACLs

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>eq &lt;port number&gt;</td>
<td>matches a specific port</td>
</tr>
<tr>
<td>gt &lt;port number&gt;</td>
<td>matches all ports that are a larger number than the port number you specify (not including the specified port)</td>
</tr>
<tr>
<td>lt &lt;port number&gt;</td>
<td>matches all ports that are a smaller number than the port number you specify (not including the specified port)</td>
</tr>
<tr>
<td>range &lt;first port number last port number&gt;</td>
<td>matches the range of ports you specify</td>
</tr>
<tr>
<td>neq &lt;port number&gt;</td>
<td>matches all ports except the port number you specify</td>
</tr>
</tbody>
</table>

To view a list of well-known ports, enter the ? help command after one of the port commands (such as eq, gt, or neq). The list of options is displayed in alphabetical order.

You can enter one of the ports listed by the CLI or the corresponding number. For example, the following commands perform the same function:

ProCurve(config-ext-nacl)# permit tcp host 192.168.1.11 eq 80 any

or

ProCurve(config-ext-nacl)# permit tcp host 192.168.1.11 eq www any

Specifying Bits in the Packets. To protect your network against attacks and hackers scanning your network for information, you can block packets based on certain bits set in the packet. You can specify the following bits:

- ack
- fin
- psh
- rst
- syn
- urg

Selecting the log Option. Include the log option if you want the Secure Router OS to log a message when these two conditions are met:

- debug access-list is enabled for this ACL
- a packet matches this ACL
Enter the **log-input** option if you want the log to include the interface on which the matching packet was received.

**Entry Order**

The order in which you add entries to an ACL is important. The Secure Router OS processes entries one-by-one in the order in which they are listed. When comparing a packet to an ACL that is applied directly to an interface, the Secure Router OS first attempts to match that packet to the first entry in the ACL. If the packet matches the first entry, the Secure Router OS stops processing the rest of the ACL and takes the action specified in that entry. If the packet does not match the first entry, the Secure Router OS tries to match the packet with the second entry, then the third entry, until it finds a match.

When you are creating entries in an ACL, you should put the most specific entries first. For example, if you want to deny a particular host but permit the subnet on which that host resides, you should first enter the deny entry and then the permit entry. If you enter the permit entry first, the Secure Router OS will process that entry first, and the packet from the host will be permitted.

As mentioned earlier, each ACL contains an implicit "deny any" entry at the end of the list of entries. (For an extended ACL, this entry is properly a "deny ip any any" entry as it denies all packets to and from all hosts.) The **any** keyword matches any IP address. If a packet does not match any entry in the ACL, it automatically matches the "deny any" entry. In this case, the Secure Router OS automatically denies, or blocks, that packet.

In Figure 5-4, for example, the device with the IP address 168.44.1.10 is trying to send a packet to a device on the LAN attached to Router A. The network administrator has configured a standard ACL called WAN and assigned this ACL to incoming traffic on the PPP 1 interface.

```plaintext
ip access-list standard WAN
  deny host 192.168.115.91
  deny host 192.168.44.53
  permit 192.168.115.0  0.0.0.255
  permit 192.168.44.0  0.0.0.255
```
Applying Access Control to Router Interfaces
Using ACLs Alone to Configure Access Control

Figure 5-4. Processing Entries in an ACL

When the device with the IP address 172.16.1.10 sends a packet to the LAN that is connected to Router A, Router A begins to check the entries in the WAN ACL to determine if the packet is allowed.

Router A checks the first entry, but the packet from 172.16.1.10 does not match this entry. Router A then checks the second entry, but again the packet from 172.16.1.10 does not match the entry. Next, the Router A checks entry three, and the packet does not match that entry. Nor does the packet match the fourth entry.

However, the packet does match the implicit "deny any" at the end of the ACL. As a result, the packet is denied.

Adding a Descriptive Tag to an ACL

To document why you created an ACL, you can use the remark command to add a descriptive tag to either a standard or an extended ACL. This tag can be up to 80 alphanumeric characters.

Syntax: remark <description>

For example, you may want to add a descriptive tag so that other network administrators will immediately understand the purpose of the ACL.

Syntax: remark This ACL permits inbound Internet traffic
Applying Access Control to Router Interfaces
Using ACLs Alone to Configure Access Control

You and other network administrators can view this remark by entering one of the following commands from the enable mode context:

ProCurve# show running-config
ProCurve# show access-lists

Figure 5-5 displays the output from the `show access-lists` command.

```
Extended IP access list WAN1
    permit udp any any eq rip   (42 matches)
    deny tcp any any eq telnet  (0 matches)
    permit tcp host 192.168.20.1 any   (0 matches)
    remark ACL for WAN connection to Phoenix
```

Remark is displayed when you enter `show access-lists`

Figure 5-5. `show access-lists`

Editing an Existing ACL

If you want to edit an existing ACL, you access it by entering the `ip access-list` command that references the name of the ACL, as shown below:

**Syntax:**

```
ip access-list [standard | extended] <listname>
```

You can then use the `no` command to remove entries, or you can use the `deny` and `permit` commands to add entries.

Deleting an Existing ACL

To delete an entire ACL, move to the global configuration mode context and enter:

**Syntax:**

```
no ip access-list [extended | standard] <listname>
```

Replace `<listname>` with the name of the list you want to delete. For example, if you wanted to delete an extended ACL list called `Inside`, you would enter:

```
ProCurve (config)# no ip access-list extended Inside
```
Applying the ACL to an Interface

After you configure an ACL, it will have not control access to an interface until you apply it to one of the following:

- interface
- ACP

As discussed above, you can also apply an ACL to all FTP, HTTP, and Telnet traffic destined to the router.

An ACL can also select traffic for a variety of router functions. You can apply an ACL to any of the following:

- quality of service (QoS) map
- crypto map for a Virtual Private Network (VPN)
- route map for policy-based routing (PBR)
- demand interface to select the traffic that triggers a dial-up connection
- rendezvous point (RP) address to select the multicast groups that a PIM-SM RP supports

This section describes how you apply ACLs to an interface. Table 5-6 lists the page number where you can find information about how to apply ACLs for other functions.

Table 5-6. Locating Information about Applying ACLs

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>applying ACLs to an ACP</td>
<td>5-26</td>
</tr>
<tr>
<td>applying ACLs to FTP, HTTP, and Telnet traffic destined to the router</td>
<td>5-21</td>
</tr>
<tr>
<td>applying ACLs to allow routing updates</td>
<td>5-25</td>
</tr>
<tr>
<td>applying ACLs to an ACP for NAT</td>
<td>6-16</td>
</tr>
<tr>
<td>applying ACLs to a QoS map</td>
<td>8-23, 8-39</td>
</tr>
<tr>
<td>applying ACLs to a crypto map for a VPN</td>
<td>10-38</td>
</tr>
<tr>
<td>applying ACLs to a route map for PBR</td>
<td>15-129</td>
</tr>
<tr>
<td>applying an ACL to a demand interface to select interesting traffic</td>
<td>8-18</td>
</tr>
<tr>
<td>applying an ACL to an RP to select the multicast groups that it can support</td>
<td>13-32</td>
</tr>
</tbody>
</table>
Applying Access Control to Router Interfaces
Using ACLs Alone to Configure Access Control

Selecting the Packet and Controlling the Action

When you assign an ACL directly to an interface, the Secure Router OS uses it to both to select traffic and to determine which action it should take on this traffic. The Secure Router OS permits traffic selected by a permit entry and blocks traffic selected by a deny entry.

For each interface, you can assign one ACL to control inbound traffic and one ACL to control outbound traffic. To assign an ACL to an interface, enter the following command from the configuration mode context for the appropriate interface:

Syntax: ip access-group <listname> [in | out]

Replace <listname> with the name of the ACL. Use the in option to have the Secure Router OS apply the ACL to traffic when it is received on the interface; use the out option to have the Secure Router OS apply the ACL to traffic when it is sent from the interface.

For example, to assign the WAN ACL to outgoing traffic on the PPP 1 interface, you would move to the PPP 1 interface configuration mode context and enter:

ProCurve(config-ppp 1)# ip access-group WAN out

If you wanted to configure the Secure Router OS to allow only Telnet traffic and traffic to subnet 192.168.115.0 /24 to enter the Ethernet 0/1 interface, you could create an extended ACL and apply it to this interface:

ProCurve(config)# ip access-list extended Outside
ProCurve(config-ext-nacl)# permit tcp any any eq telnet
ProCurve(config-ext-nacl)# permit tcp any 192.168.115.0 0.0.0.255
ProCurve(config-ext-nacl)# exit
ProCurve(config)# int eth 0/1
ProCurve(config-eth 0/1)# ip access-group Outside in
Applying Access Control to Router Interfaces
Using ACLs Alone to Configure Access Control

You may also want to create an ACL to control traffic to your company’s two Web servers: one is an Internet server, accessible to anyone on the Internet, and one is an intranet server, accessible only to company users. You want to permit all HTTP traffic to the Internet server, but for the intranet server, you want to permit HTTP traffic only from the company’s two private networks at remote VPN sites. To do this, you would create an extended ACL, such as the following:

```
ProCurve(config)# ip access-list extended webservers
ProCurve(config-ext-nacl)# permit tcp any host 192.168.1.12 eq www
ProCurve(config-ext-nacl)# permit tcp 192.168.1.0 0.0.0.255 host 192.168.1.15 eq www
ProCurve(config-ext-nacl)# permit tcp 192.168.16.0 0.0.0.255 host 192.168.1.15 eq www
ProCurve(config-ext-nacl)# exit
```

In this ACL, the first entry permits HTTP traffic from any source to the Internet server with the IP address 192.168.1.12. The second entry permits HTTP traffic from the 192.168.1.0 /24 network to the intranet server with the IP address 192.168.1.15. Finally, the third entry permits HTTP traffic from the 192.168.15.0 /24 network to the intranet server with the IP address 192.168.1.15. After you create the ACL, you must apply it to the appropriate interfaces.

For example, the PPP 1 interface connects to the Internet. Traffic both from Internet users and users at the remote VPN sites arrives on this interface. Enter:

```
ProCurve(config)# interface ppp 1
ProCurve(config-ppp 1)# ip access-group webservers in
```

Controlling FTP, HTTP, and Telnet Access to the Router

The ProCurve Secure Router allows you to control FTP, HTTP, and Telnet access globally for the entire router. This feature greatly simplifies the effort required to manage FTP, HTTP, and Telnet access. For example, rather than configuring entries to restrict FTP access in all of the ACLs applied to router interfaces, you can create one ACL for FTP access and apply it globally.
Applying Access Control to Router Interfaces
Using ACLs Alone to Configure Access Control

Restricting FTP Access

To control access to the FTP server on the router, you first create a standard ACL that permits the FTP traffic you want to access the router and denies the FTP traffic that you want to block. For example, if you want to permit FTP access only from network 192.168.1.0/24, you would create a standard ACL:

ProCurve(config)# ip access-list standard FTPaccess
ProCurve(config-std-nacl)# permit 192.168.1.0 0.0.0.255
ProCurve(config-std-nacl)# exit

To apply the ACL globally to all incoming FTP traffic, enter this command from the global configuration mode context:

Syntax: ip ftp access-class <listname> in

Replace <listname> with the name of the ACL you configured for FTP access. For example, if you created a standard ACL called FTPaccess, you would enter:

ProCurve(config)# ip ftp access-class FTPaccess in

Restricting HTTP Access

To configure HTTP access to the ProCurve Secure Router, you must configure a standard ACL. For example, suppose you want to permit HTTP access only from the company’s two subnetworks. You would create a standard ACL, such as the following:

ProCurve(config)# ip access-list standard webaccess
ProCurve(config-std-nacl)# permit 192.168.1.0 0.0.0.255
ProCurve(config-std-nacl)# permit 192.168.115.0 0.0.0.255
ProCurve(config-std-nacl)# exit

In this ACL, the first entry permits HTTP traffic from network 192.168.1.0/24, and the second entry permits HTTP traffic from network 192.168.115.0/24. Because each ACL contains an implicit “deny any” at the end of the list, this will be the only HTTP traffic that is allowed to access the Web browser interface once the ACL is applied to the router.

To apply the ACL that controls HTTP access to the router, enter the following command from the global configuration mode context:

Syntax: ip http access-class <listname> in
Applying Access Control to Router Interfaces
Using ACLs Alone to Configure Access Control

For example, if you wanted to apply an ACL called webaccess, you would enter:

ProCurve(config)# ip http access-class webaccess in

Restricting Telnet Access

Restricting Telnet access to the router is similar to restricting access to an interface. You configure an ACL and then apply that ACL to the appropriate Telnet line configuration mode context. To restrict Telnet access, however, you must configure a standard ACL. From the Telnet line configuration mode context, enter:

**Syntax:** access-class <listname> in

For example, to control access to Telnet line 0, you must move to the Telnet line 0 configuration mode context and then apply the Telnet ACL to incoming Telnet traffic:

ProCurve(config)# line telnet 0
ProCurve(config-telnet0)# access-class Telnet in

If you assign the ACL to only one Telnet line but you have configured a password for other lines, the hosts that you want to block may gain access. If a host is denied access to one Telnet line, the Secure Router OS will assign the host to the next available Telnet line.

Unless you only want to restrict access on a single Telnet line, you must apply the standard ACL to all of the Telnet lines that you have enabled on the router. If you want use the Telnet ACL to control access to all five Telnet lines, you would enter:

ProCurve(config)# line telnet 0 4
ProCurve(config-telnet0-4)# access-class Telnet in

The ProCurve Secure Router would then allow only the Telnet traffic that matches the criteria permitted by the specified ACL.

Examples of Applying ACLs

When you create ACLs, you may want to first record the transport protocol, source IP address, source port, destination IP address, and destination port for each type of traffic that you want to control, specifying also whether this traffic will be permitted or denied. In the long run, attention to detail will simplify configuring ACLs for your environment. It will also help you keep track of the entries needed to permit all necessary traffic.
This section contains some sample ACLs to help you understand both the type
of ACLs that may be required for your network and the way you configure
them.

**Block Telnet Traffic.** To strengthen security on your WAN, you may want
to deny any Telnet traffic entering your WAN interfaces. You can control this
access by creating an extended ACL. Enter the following commands to
configure an extended ACL called Telnet that prohibits Telnet access:

```
ProCurve(config)# ip access-list extended Telnet
ProCurve(config-ext-nacl)# deny tcp any any eq telnet
```

You would then apply this ACL to all the WAN interfaces that are activated on
the router. Users would still be able to initiate a Telnet session from the
Ethernet interfaces.

You can also configure an ACL to restrict Telnet access to the router. For more
information, see “Restricting Telnet Access” on page 5-23.

**Permit HTTP, Mail, and POP3 Traffic.** Some companies may want to
restrict incoming traffic on a WAN interface to HTTP, Simple Mail Transfer
Protocol (SMTP), POP3, and FTP traffic. To do so, you must configure an
extended ACL, as shown below:

```
ProCurve(config)# ip access-list extended Internet
ProCurve(config-ext-nacl)# permit tcp any any eq www
ProCurve(config-ext-nacl)# permit tcp any any eq smtp
ProCurve(config-ext-nacl)# permit tcp any any eq pop3
ProCurve(config-ext-nacl)# permit tcp any any eq ftp
ProCurve(config-ext-nacl)# permit tcp any any eq ftp-data
```

*Note*

If the Secure Router OS firewall and the FTP ALG are enabled, you do not
have to configure an entry to allow traffic on FTP data port (21). The FTP ALG
automatically allows the return traffic for established FTP sessions. For more
information about ALGs, see *Chapter 4: ProCurve Secure Router OS Fire-
wall—Protecting the Internal, Trusted Network.*

You may also want to permit Domain Name System (DNS) traffic on WAN
interfaces that are connected to the Internet. To permit DNS traffic, enter:

```
ProCurve(config-ext-nacl)# permit tcp any any eq domain
```

You would apply this ACL to the WAN interfaces on which you want to enforce
this access control.
Applying Access Control to Router Interfaces
Using ACLs Alone to Configure Access Control

**Permit Routing Updates.** When you configure ACLs, remember that any traffic that you do not explicitly permit will match the implicit “deny any” entry at the end of the ACL. If you have configured a routing protocol and routing updates are being sent to a router interface, you should ensure that these routing updates are permitted by the ACL you assign to that interface. For example, to permit routing information protocol (RIP) updates, enter:

```
ProCurve(config-ext-nacl)# permit udp any any eq rip
```

To permit border gateway protocol (BGP) updates, enter

```
ProCurve(config-ext-nacl)# permit tcp any any eq bgp
```

You would apply the ACL to the interface on which you want to permit routing updates.

**Permit Traffic from Specific Networks.** You may want to restrict access to specific networks. For example, you may want to permit traffic from 10.1.1.0 /30, but deny traffic from 192.168.115.0 /24. To configure entries for this access, enter:

```
ProCurve(config-ext-nacl)# permit ip 10.1.1.0 0.0.0.3 any
ProCurve(config-ext-nacl)# deny ip 192.168.115.0 0.0.0.255 any
```

Again, you would apply the ACL to the appropriate interface.
Using ACPs to Control Access to Router Interfaces

By themselves, ACLs have some limitations: you can assign only one ACL to each interface to control inbound traffic and one ACL to control outbound traffic. In addition, the Secure Router OS can use the ACL only to permit or discard traffic.

ACPs provide more flexibility than ACLs in enforcing your company's security guidelines. For example, ACPs allow you to apply multiple ACLs to an interface, and they allow you to NAT traffic.

To provide this flexibility, the ACP must take over some of the tasks that are typically associated with ACLs. For example, when you assign an ACL directly to an interface, the ACL both selects traffic and determines the action taken on this traffic. When you use an ACP, however, the ACL only selects traffic. The ACP takes over the role of determining which action is taken on the selected traffic. Dividing up the tasks in this way allows you to configure a single ACP in which the Secure Router OS firewall takes various actions on multiple types of traffic.

Configuring ACPs involves four steps:

1. Enable the Secure Router OS firewall (if it is not already enabled).
2. Configure at least one ACL.
3. Configure an ACP.
4. Apply the ACP to an interface.

Enable the Firewall

To use ACPs to filter traffic entering router interfaces, you must first enable the Secure Router OS firewall. Unless the firewall is enabled, any ACPs you apply to router interfaces will not take effect. From the global configuration mode context, enter:

ProCurve(config)# ip firewall
If you do not enable the firewall, you can still configure ACPs. However, when you try to apply an ACP to an interface, the ProCurve Secure Router displays a message similar to the following:

Firewall is disabled, access policy commands applied but not used

Configure ACLs

ACLs are composed of an ordered list of entries, and each entry contains two parts: an action and a packet pattern.

**Action.** You can define one of two actions for an entry:
- permit
- deny

**Packet Pattern.** You can define patterns based on:
- source IP address
- source and destination IP addresses
- IP protocol
- TCP or UDP ports

When you create ACLs that are used in ACPs, the permit and deny actions take on new meanings. Permit means that the traffic is selected for the action specified in the ACP entry. If a packet matches the permit entry in the ACL, the Secure Router OS firewall will perform the action specified in the ACP entry.

Deny means that the traffic is excluded from the action specified in the ACP entry. If a packet matches a deny entry in the ACL, the Secure Router OS will not perform the action specified. Instead, the Secure Router OS will stop processing that particular ACL and the related ACP entry and move to the next entry in the ACP (if there is another entry). This process is explained in more depth throughout the rest of this section.

Types of ACLs

The ProCurve Secure Router supports two types of ACLs:
- standard
- extended
A standard ACL matches only one packet pattern: the source IP address. An extended ACL matches more complex packet patterns:

- source and a destination address
- most fields in the IP, TCP, and UDP header, including IP protocol and TCP or UDP source or destination port

You should create a standard ACL if you want to select traffic based only on the source IP address. (See Figure 5-6.) If you want to select traffic based on other fields in the IP, TCP, or UDP header or if you want the firewall to filter traffic based on the destination IP address of traffic, you must create an extended ACL. (See Figure 5-7.)
Creating an ACL

To create an ACL, you enter the **ip access-list** command from the global configuration mode context:

**Syntax:** `ip access-list [standard |extended] <listname> [log]`

Enter either the **standard** or **extended** option, depending on the type of ACL you are configuring, and replace `<listname>` with an alphanumeric descriptor that is meaningful to you. The listname is case sensitive.

Creating a Standard ACL

To create a standard ACL, enter:

```
ProCurve(config)# ip access-list standard <listname>
```

After you enter this command, you are moved to the standard ACL configuration mode context:

```
ProCurve(config-std-nacl)#
```

You can now begin to enter permit and deny entries. The ACL is empty until you add these entries.
Using Permit and Deny Entries to Select Traffic. To create permit and deny entries for standard ACLs, you use the following command syntax:

**Syntax:** [permit | deny] [any | host {<A.B.C.D> | <hostname>} | <A.B.C.D> <wildcard bits>]

Table 5-7 lists the options for specifying the source address.

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>any</td>
<td>matches all hosts</td>
</tr>
<tr>
<td>host {&lt;A.B.C.D&gt;</td>
<td>&lt;hostname&gt;}</td>
</tr>
<tr>
<td>&lt;A.B.C.D&gt;</td>
<td>specifies a single IP address</td>
</tr>
<tr>
<td>&lt;A.B.C.D&gt; &lt;wildcard bits&gt;</td>
<td>specifies a range of IP addresses</td>
</tr>
</tbody>
</table>

For example, if you want to select all hosts for the action specified in the ACP, you create a permit entry in the ACL:

```
ProCurve(config-std-nacl)# permit any
```

If you want to select all hosts and exclude them from the action specified in the ACP, enter:

```
ProCurve(config-std-nacl)# deny any
```

You can also create an entry to select a specific host. To select a host for the action specified in the ACP, enter:

```
ProCurve(config-std-nacl)# permit host [<A.B.C.D> | <hostname>]
```

To select a host and exclude it from the action specified in the ACP, enter:

```
ProCurve(config-std-nacl)# deny host <A.B.C.D>
```

For example, if you want to select a host with the IP address of 192.168.115.90 and exclude it from the action you will later specify in the ACP, enter:

```
ProCurve(config-std-nacl)# deny host 192.168.115.90
```

You can also use a hostname to select the host for the action. Enter:

```
ProCurve(config-std-nacl)# permit hostname user1.procurve.com
```
You can also omit the host keyword to select a specific IP address:

ProCurve(config-std-nacl)# permit 192.168.115.80
ProCurve(config-std-nacl)# deny 192.168.115.80

**Using Wildcard Bits.** Finally, you can use wildcard bits to permit or deny a range of IP addresses. Wildcard bits define which address bits the Secure Router OS should match and which address bits it should ignore. Essentially, you use the wildcard bits to specify the subnet to which you want the Secure Router OS to match packets.

When you enter wildcard bits, you use a 0 to indicate that the Secure Router OS should match the corresponding bit in the IP address. You use a 1 to indicate that the Secure Router OS can ignore the corresponding bit in the IP address. In other words, the Secure Router OS does not have to match a packet’s address to that bit.

For example, you might enter:

ProCurve(config-std-nacl)# deny 192.168.1.0 0.0.0.255

In this case, the Secure Router OS will not match any address bits in the fourth octet of the IP address. The Secure Router OS will match incoming packets destined to any address on the IP subnet 192.168.1.0 /24 (because it will not match the bits in the fourth octet). (See Figure 5-8.)

![Figure 5-8. Understanding Wildcard Bits](image)

**Implicit “deny any” Entry.** Each ACL includes an implicit “deny any” entry at the end of the list. If traffic does not match an entry in the ACL, the Secure Router OS firewall does not perform the action specified by the related entry in the ACP. Instead the firewall moves to the next entry in the ACP (if there is one).
Applying Access Control to Router Interfaces
Using ACPs to Control Access to Router Interfaces

**Selecting the log Option.** Include the `log` option if you want the Secure Router OS to log a message when these two conditions are met:

- `debug access-list` is enabled for this ACL
- a packet matches this ACL

**Exit the ACL.** After you have finished creating the ACL, enter `exit` to return to the global configuration mode context, as shown below:

ProCurve(config-std-nacl)# exit
ProCurve(config)#

Creating an Extended ACL

To create an extended ACL, you enter the following command from the global configuration mode context:

ProCurve(config)# ip access-list extended `<listname>`

Replace `<listname>` with an alphanumeric descriptor that is meaningful to you. The name is case sensitive.

After you enter this command, you are moved to the extended ACL configuration mode context:

ProCurve(config-ext-nacl)#

**Using Permit and Deny Entries to Select Traffic.** You can now begin to enter permit and deny entries. Because you are applying the ACL to an ACP, you will use a permit entry to select the traffic for the action you specify in the ACP. You will use a deny entry to exclude the traffic from the action you specify in the ACP. The ACL is empty until you add these entries.

To create permit and deny entries for extended ACLs, use the following command syntax:

**Syntax:** `[permit | deny] <protocol> <source address> <source port> <destination address> <destination port> [packet bits] [log | log-input]`

You must specify a `<protocol>`, `<source address>`, and `<destination address>`. However, the following are optional:

- `<source port>` for TCP or UDP traffic
- `<destination port>` for TCP or UDP traffic
- `<packet bits>`
- `[log | log-input]`
Applying Access Control to Router Interfaces
Using ACPs to Control Access to Router Interfaces

All of the command options are explained in the sections that follow.

**Specifying a Protocol.** When you configure extended ACLs, you must specify a protocol. Valid protocols include:

- AH (ahp)
- ESP (esp)
- GRE (gre)
- ICMP (icmp)
- IP (ip)
- TCP (tcp)
- UDP (udp)

You can also specify the number of the protocol. Valid numbers include any number between 0 and 255.

**Defining the Source and Destination Addresses.** You must configure both a source and a destination address for each entry. Table 5-8 shows the options you have for specifying these addresses.

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>any</td>
<td>matches all hosts</td>
</tr>
<tr>
<td>host &lt;A.B.C.D&gt;</td>
<td>specifies a single host or a single IP address</td>
</tr>
<tr>
<td>hostname &lt;hostname&gt;</td>
<td>specifies a single host, using its hostname rather than its IP address</td>
</tr>
<tr>
<td>&lt;A.B.C.D&gt; &lt;wildcard bits&gt;</td>
<td>specifies a range of IP addresses</td>
</tr>
</tbody>
</table>

For example, suppose you want to select all TCP traffic from any source to any destination and have the Secure Router OS take the action specified in the ACP entry that you will configure later. Enter:

ProCurve(config-ext-nacl)# permit tcp any any

To exclude traffic from the action that you will later specify in the ACP, create a deny entry. For example to exclude all ICMP traffic from a specific host, such as host 192.168.1.1, to any destination, enter:

ProCurve(config-ext-nacl)# deny icmp host 192.168.1.1 any
Applying Access Control to Router Interfaces

Using ACPs to Control Access to Router Interfaces

To exclude ICMP traffic from a range of IP addresses to a specific destination, enter:

`ProCurve(config-ext-nacl)# deny icmp <A.B.C.D> <wildcard bits> host <A.B.C.D>`

**Specifying a Source or Destination Port for TCP and UDP.** If you are configuring ACL entries to select TCP or UDP traffic, you can also specify source and destination ports—although this is optional. For example, you could specify the well-known port 80 for HTTP traffic if you wanted to select HTTP traffic for an action.

There is a drawback to including a port number, however. The Secure Router OS firewall will match the type of traffic only on that port. If a device transmits the traffic you are targeting on another port, the firewall will not match that traffic to your ACL.

To view the options available for specifying ports, enter:

`ProCurve(config-ext-nacl)# [permit | deny] [tcp | udp] any ?`

In practice, you would use the `any` keyword only if you want to match all traffic from a particular port. When you actually enter the `permit` or `deny` command, you can specify any host, a specific host, a specific IP address, or a range of IP addresses. Then enter the `?` help command to view the options for specifying ports.

Table 5-9 shows the options for specifying ports in an extended ACL.

**Table 5-9. Specifying Ports in Extended ACLs**

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>eq &lt;port number&gt;</code></td>
<td>matches a specific port</td>
</tr>
<tr>
<td><code>gt &lt;port number&gt;</code></td>
<td>matches all ports that are a larger number than the port number specify (not including the specified port)</td>
</tr>
<tr>
<td><code>lt &lt;port number&gt;</code></td>
<td>matches all ports that are a smaller number than the port number specify (not including the specified port)</td>
</tr>
<tr>
<td><code>range &lt;first port number last port number&gt;</code></td>
<td>matches a range of ports</td>
</tr>
<tr>
<td><code>neq &lt;port number&gt;</code></td>
<td>matches all ports except the port number you specify</td>
</tr>
</tbody>
</table>
To view a list of well-known ports, enter the `help` command after one of the port commands (such as `eq`, `gt`, or `neq`). The list of options is displayed in alphabetical order. You can either one of the keywords displayed or a specific port number between 0 and 65535.

**Specifying Bits in the Packets.** To protect your network against attacks and hackers scanning your network for information, you can block packets based on certain bits set in the packet. You can specify the following bits:

- `ack`
- `fin`
- `psh`
- `rst`
- `syn`
- `urg`

**Selecting the log Option.** Include the `log` option if you want the Secure Router OS to log a message when these two conditions are met:

- `debug access-list` is enabled for this ACL
- A packet matches this ACL

Enter the `log-input` option if you want the log to include the interface on which packets that matched the entry were received.

**Configure ACPs**

You create entries in the ACP to specify how the Secure Router OS firewall treats packets selected by ACLs. Each ACP entry contains two parts:

- the action
- the selector

**Action**

The action specifies exactly how the Secure Router OS firewall handles packets selected by an ACL. ACPs can perform one of three actions on selected packets:

- allow the packets
- discard the packets
- manipulate the packets, using NAT to translate the source or destination IP address to a different IP address
Applying Access Control to Router Interfaces
Using ACPs to Control Access to Router Interfaces

Each ACP contains an implicit “discard all” at the end. Packets are discarded if they do not match any ACL listed in the ACP.

This chapter explains how to create entries that allow or discard packets. For information about NAT, see Chapter 6: Configuring Network Address Translation.

Selector

For the selector, you specify one of the ACLs that you have already configured. If a packet matches a permit entry, the Secure Router OS firewall takes the action specified in the ACP entry (allow, discard, or NAT). If a packet matches a deny entry, the Secure Router OS firewall selects the packet but does not take the action specified by the ACL. Instead, it stops processing the ACL and attempts to match the packet to the ACL specified in the next entry in the ACP.

Note

ACPs use ACLs in much the same way that NAT or crypto maps use ACLs on a Cisco router.

Creating an ACP

To create an ACP, you enter the following command from the global configuration mode context:

**Syntax:** `ip policy-class <policynname> [max-sessions <number>]`

Replace `<policynname>` with a unique name that is a maximum of 255 alphanumeric characters. Include the optional `max-sessions` keyword if you want to limit the number of sessions that can be created for packets matching this ACP. For the ProCurve Secure Router 7102dl, you can specify a number between 1 and 4000. For the ProCurve Secure Router 7203dl, you can specify a number between 1 and 30000.

For example, to create an ACP called WAN, you would enter:

```
ProCurve(config)# ip policy-class WAN
```

The `ip policy-class` command moves you to the policy class configuration mode context and creates an empty ACP.
Creating Entries in the ACP

From the policy class configuration mode context, you can begin to enter allow, discard, and NAT entries. To create an allow entry, enter:

**Syntax:** allow list <listname> [self]

For example, if you want to allow the packets selected by the Outside ACL, you would enter:

```
ProCurve(config-policy-class)# allow list Outside
```

The optional `self` keyword designates the internal IP stack as the destination in the specified ACL. The Secure Router OS firewall performs the specified action on traffic that matches the ACL and is destined for the router itself.

To discard the packets that are selected by an ACL, enter:

**Syntax:** discard list <listname> [self]

For example, to discard the packets that are selected by the Inside ACL, enter:

```
ProCurve(config-policy-class)# discard list Inside
```

You can also NAT the packets specified in an ACL. This option is discussed in Chapter 6: Configuring Network Address Translation.

Editing ACPs

If you need to edit the entries in an ACP, you use the `ip policy-class` command to access that ACP:

**Syntax:** ip policy-class <policyname>

You can then use the `no` command to remove entries, or you can add entries as needed.

Deleting an ACP

To delete an entire ACP, you must know the name of that ACP. You can then enter:

**Syntax:** no ip policy-class <policyname>
Assigning the ACP to an Interface

An ACP does not become active until you assign it to an interface (and enable the firewall). Then it affects only the incoming traffic on the interface to which it is assigned.

Therefore, to filter traffic that is generated on your company's LAN, you assign an ACP to each of the Ethernet interfaces. To filter traffic that originates from the WAN, you assign an ACP to the logical WAN interfaces that you have configured on the router.

Valid logical interfaces to which you can assign an ACP include:

- Ethernet interfaces
- PPP interfaces
- Frame Relay subinterfaces
- HDLC interfaces
- ATM subinterfaces (for ADSL connections)
- demand interfaces

If you enable support for VLANs, you must assign the ACP to the Ethernet subinterfaces, rather than to the Ethernet interface.

To assign an ACP to an interface, move to the appropriate interface configuration mode context and enter:

**Syntax:** access-policy <policyname>

**Note**

If you are accessing the ProCurve Secure Router through a Telnet session, you should be especially careful when configuring and applying ACPs. Before you assign an ACP to the interface, you should verify that the ACP will not interfere with your Telnet connection. If you apply an ACP to an interface and that ACP does not allow Telnet traffic to the interface you are using to connect to the ProCurve Secure Router, your connection will be terminated.

Using the reload Command

When configuring ACPs, you may want to use the **reload** command as a safeguard in case you accidentally block your access to the ProCurve Secure Router. The **reload** command schedules a system reboot at the time you specify.
Applying Access Control to Router Interfaces
Using ACPs to Control Access to Router Interfaces

For example, if you configure an ACP that blocks your Telnet access to the ProCurve Secure Router, you will lose your ability to manage the router through a Telnet session and must use another access method to correct your error. You may have to access the router through a console session.

To prevent this from happening, you can enter the `reload` command to schedule the router to reboot after the specified amount of time has elapsed. From the enable mode context, enter:

**Syntax:** `reload in <mmm>`

or

**Syntax:** `reload in <hh:mm>`

Replace `<mmm>` with up to three digits to specify minutes. Replace `<hh:mm>` with hours and minutes. For example, you may want to specify 15 minutes. Then, you can configure your ACLs without saving your configurations. Before you apply your ACPs, you enter the `reload` command. After 15 minutes, the router will reload using the previously saved startup-config file. If applying your ACPs has made you lose access to the router, you can now re-access it and re-configure the ACPs correctly.

When you are sure that the ACPs you have applied do not prevent you from accessing the ProCurve Secure Router, you can save your configurations and cancel the `reload` command:

**Syntax:** `reload cancel`

**Note**

You can also enable SafeMode to protect your Telnet or Secure Shell (SSH) session. (Enter `safe-mode` from the global configuration mode context.) When in SafeMode, the Secure Router OS periodically asks you to reset a timer. If you not do so before the timer expires, the Secure Router OS assumes that you have been locked out of the router and reboots the router. See Chapter 1: Overview for more information about SafeMode.

**Processing ACPs**

Both ACLs and ACPs are order dependent. That is, the order of the entries is important because the Secure Router OS firewall executes the entries one at a time, from the top of the list to the bottom. The firewall stops processing entries after it executes an action on a packet, so, even if more than one entry matches a packet, only the first entry will affect the packet. As a general rule, you should include the most specific entries at the top of the list and the most general entries at the end of the list.
When a packet enters an interface that has been assigned an ACP, the Secure Router OS firewall checks the first entry in the ACP. The firewall then reads the associated ACL to determine if the packet matches the IP address and any other fields that are specified. If the packet matches a permit entry in the ACL, the firewall performs the action specified for that entry in the ACP. If the packet matches a deny entry in the ACL, the Secure Router OS firewall does not perform the action specified for that entry in the ACP. Instead, the Secure Router OS firewall moves to the next entry in the ACP and checks whether the packet matches that entry.

In Figure 5-9, for example, device 192.168.1.14 sends a packet that is destined for the internal network attached to Router B. The access-policy (ACP) Private has been applied to the PPP 2 interface on Router B, so Router B will try to match all incoming traffic on that interface to the Private ACP.

Router B first attempts to match the packet to the Group 1 ACL. It checks each entry in the Group 1 ACL, one-by-one, but the packet does not match any of the entries. Router B then checks the next entry in the ACP and tries to match the packet to the first entry in the Group 2 ACL. Because the packet does not match the first entry, Router B moves to the second entry. When the packet does not match the second entry, Router B moves to the next entry.

The packet matches the third entry in the Group 2 ACL: permit host 192.168.1.14. Because it is a permit entry, Router B selects the packet and then takes the action specified in the ACP: it discards the packet.
Applying Access Control to Router Interfaces
Using ACPs to Control Access to Router Interfaces

Figure 5-9. Using ACLs with ACPs

In Figure 5-10, device 10.10.10.1 sends a packet to server 192.168.1.10. Router B forwards the packet to Router A, which receives the packet on its PPP 1 interface. The WAN ACP has been assigned to PPP 1, so the Secure Router OS firewall begins to process the entries in that ACP.

The Secure Router OS firewall first tries to match the packet from 10.10.10.1 to the allow list Web entry. It checks the entry in the Web ACL, but the packet does not match this entry.

The Secure Router OS then checks the second entry in the WAN ACP: discard list Host. It tries to match the packet from 10.10.10.1 to the first entry in the Host ACL and then to the second entry. There is no match.

Next, the Secure Router OS checks the last entry in the WAN ACP: allow list MatchAll. It tries to match the packet from 10.10.10.1 to the first entry in the MatchAll ACL. Then, it tries to match the packet to the second entry, and this time the packet matches the packet pattern.
However, the action specified in the ACL is deny, and when an ACL is part of an ACP, deny means do not take the action specified in the ACP.

The **allow list MatchAll** entry is the last in the ACP. Because each ACP includes an implicit “discard all” at the end of the list, the Secure Router OS discards the packet from 10.10.10.1.

---

**Figure 5-10. Processing ACPs**

ACP Action Summary

Table 5-10 outlines the actions that the Secure Router OS firewall takes, based on the entries configured in the ACL and the ACP.
Applying Access Control to Router Interfaces
Using ACPs to Control Access to Router Interfaces

Table 5-10. Actions Based on ACP Configuration

<table>
<thead>
<tr>
<th>ACL</th>
<th>ACP</th>
<th>Action</th>
</tr>
</thead>
</table>
| deny  | does not matter | Secure Router OS firewall:  
|       |       | • does not take the specified action on the packet  
|       |       | • stops processing this ACL  
|       |       | • tries to match the packet to the next entry in the ACP (if there is one)  |
| permit| allow | Secure Router OS firewall allows the packet  |
| permit| discard| Secure Router OS firewall discards the packet  |
| permit| nat   | Secure Router OS firewall NATs the packet  |

ACL Deny Entries. When a packet matches a deny entry in an ACL, it does not matter what action the corresponding ACP entry specifies. The Secure Router OS firewall does not perform that action, whatever it is. When a packet matches an ACLs deny entry, the Secure Router OS firewall immediately stops processing the ACL and the corresponding ACP entry. It advances to the next entry in the ACP and searches the associated ACL for another match for that packet. (This process is different from that implemented with the Cisco access-group command, which automatically discards traffic denied to the ACL.)

If the packet matches only deny entries, it will eventually be discarded due to the implicit “discard all” at the end of the ACP.

ACL Permit Entries. Permit entries in an ACL select packets for the action specified in the ACP entry. If a packet matches a permit entry in the ACL, the Secure Router OS firewall performs the action specified in the ACP entry. It will either allow, discard, or NAT the packet. After performing this action, the Secure Router OS firewall will not continue searching the ACP to identify other possible matches for that packet.

ACP Flow Chart. Figure 5-11 outlines the process that the Secure Router OS firewall follows when the router receives a packet on an interface. The firewall first determines if an ACP has been assigned to the interface. If there is an ACP, the Secure Router OS firewall begins the process of trying to match the packet to an entry in the first ACL listed in the ACP.
Applying Access Control to Router Interfaces
Using ACPs to Control Access to Router Interfaces

Figure 5-11.  ACP Flow Chart

Traffic Flow through Interfaces with ACPs

As mentioned earlier, ACPs control only the traffic arriving on the interface. For example, if you assign an ACP to the PPP 1 interface, the ProCurve Secure Router uses that ACP to control traffic that it receives on the PPP 1 interface. The ACP does not affect traffic that is sent from that interface.

Before you begin to use ACPs to enforce your company’s security guidelines, you must understand the traffic flow through interfaces on the ProCurve Secure Router and apply ACPs to the appropriate interfaces.
Applying Access Control to Router Interfaces
Using ACPs to Control Access to Router Interfaces

Inbound Interface Has an ACP; Outbound Interface Does Not Have an ACP

When you assign an ACP to an interface, the Secure Router OS firewall uses that ACP to filter inbound traffic—traffic arriving on the interface. Only traffic allowed by the ACP is forwarded to another interface to become outbound traffic that is sent on to its destination. (See Figure 5-12.)

![Diagram of ACP applied to the inbound interface filters incoming traffic](image)

Figure 5-12. ACP Applied to the Inbound Interface Filters Incoming Traffic

Because the traffic is forwarded through the second interface, the ProCurve Secure Router does not check that interface for ACPs. It does not matter that an ACP has not been assigned to the interface that will handle the outbound traffic.

For example, you might configure an ACP for the Ethernet 0/1 interface, but you might not configure one for the PPP interface. If a packet is received on the Ethernet 0/1 interface, the ProCurve Secure Router checks the ACP assigned to the Ethernet 0/1 interface. If the packet is allowed, the ProCurve Secure Router sends the packet to the PPP interface, and the packet is then forwarded to the next hop en route to its destination.

Inbound Interface Has an ACP; Outbound Interface Has a Different ACP

If both the inbound and outbound interfaces have been assigned ACPs, the ProCurve Secure Router applies only the ACP assigned to the inbound interface. In this case, the ACP configured on the outbound interface does not affect the traffic.

For example, suppose you assign the Inside ACP to the Ethernet 0/1 interface and the Outside ACP to the PPP 1 interface. When traffic arrives on the Ethernet 0/1 interface, the Secure Router OS firewall will use the Inside ACP to filter the traffic. The Outside ACP will not affect traffic flow. (See Figure 5-13.)
Applying Access Control to Router Interfaces
Using ACPs to Control Access to Router Interfaces

However, if traffic arrives on the PPP 1 interface, the roles are reversed: the Secure Router OS firewall will use the Outside ACP to filter traffic. The Inside ACP will not affect this traffic. (See Figure 5-14.)

Inbound Interface Does Not Have an ACP; Outbound Interface Has an ACP

When traffic arrives on an interface without a configured ACP, the ProCurve Secure Router does not apply an ACP. If the ProCurve Secure Router then forwards the packets through an interface for which an ACP has been configured, it still does not apply an ACP because ACPs apply only to incoming traffic. The same concept applies if the outbound interface does not have an ACP; all traffic proceeds normally. (See Figure 5-15.)
Applying Access Control to Router Interfaces
Using ACPs to Control Access to Router Interfaces

Figure 5-15. No ACP Applied to the Inbound Interface, so all Traffic Is Allowed

If you have enabled the firewall on the ProCurve Secure Router, it will still check this traffic for known attacks and block those attacks. However, all other traffic will be allowed.

Traffic in and out Through a Single Interface
A packet can be received and sent out on the same interface. For example, this may happen in a multi-netted environment. By default, the ProCurve Secure Router immediately forwards such traffic without applying the ACP configured for the interface. You can, however, enable the Secure Router OS firewall to apply an ACP to traffic that enters and leaves on the same interface. Enter the following command from the global configuration mode context:

```
ProCurve(config)# ip firewall check reflexive-traffic
```

For more information, see Chapter 4: ProCurve Secure Router OS Firewall—Protecting the Internal, Trusted Network.

Examples of ACPs
When you create ACPs, you can first record the transport protocol, source IP address, source port, destination IP address, and destination port for the types of traffic that you want to control. You should also record whether this traffic will be permitted or denied in the ACL and allowed or discarded in the ACPs. In the long run, attention to detail will simplify configuring both the ACLs and ACPs required to control access to your WAN. It will also help you keep track of the entries needed to permit all necessary traffic.

This section contains some examples of ACLs and ACPs. These examples serve a two-fold purpose: they provide examples of the type of traffic that many companies will want to allow or discard. They also illustrate how you configure ACLs and ACPs to work together to control access to your WAN.
**Applying Access Control to Router Interfaces**

Using ACPs to Control Access to Router Interfaces

**Block Telnet Traffic.** To strengthen security on your WAN, you may want to deny any Telnet session that users attempt to establish with the ProCurve Secure Router. You must first create an extended ACL and give it a name, such as Telnet. When you create the entry for Telnet traffic, you must use a permit entry because you want the Secure Router OS to both select the traffic and to take the action configured in the ACP. Enter:

```
ProCurve(config)# ip access-list extended Telnet
ProCurve(config-ext-nacl)# permit tcp any any eq telnet
ProCurve(config-ext-nacl)# exit
```

Next, you must create an ACP and give it a unique name, such as Manage:

```
ProCurve(config)# ip policy-class Manage
ProCurve(config-policy-class)# discard list Telnet self
ProCurve(config-policy-class)# exit
```

The `self` option designates the destination as the internal IP stack—the router itself.

After you create the ACP, you must then use the `access-policy` command to assign it to the appropriate interface on the router.

**Permit HTTP, Mail, and POP3 Traffic.** Some companies may want to restrict incoming traffic on a WAN interface to HTTP, Simple Mail Transfer Protocol (SMTP), POP3, and FTP traffic. To do so, you must configure an extended ACL, as shown below:

```
ProCurve(config)# ip access-list extended Internet
ProCurve(config-ext-nacl)# permit tcp any any eq www
ProCurve(config-ext-nacl)# permit tcp any any eq smtp
ProCurve(config-ext-nacl)# permit tcp any any eq pop3
ProCurve(config-ext-nacl)# permit tcp any any eq ftp
ProCurve(config-ext-nacl)# permit tcp any any eq ftp-data
```

**Note**

If the Secure Router OS firewall and the FTP application-level gateway (ALG) are enabled, you do not have to configure an entry to allow traffic on FTP data port (21). The FTP ALG automatically allows the return traffic for an established FTP session. For more information about ALGs, see *Chapter 4: ProCurve Secure Router OS Firewall—Protecting the Internal, Trusted Network*. 

---

5-48
Applying Access Control to Router Interfaces
Using ACPs to Control Access to Router Interfaces

You may also want to permit Domain Name System (DNS) traffic on WAN interfaces that are connected to the Internet. To permit DNS traffic, enter:

   `ProCurve(config-ext-nacl)# permit tcp any any eq domain`

You can then create an ACP, as shown below:

   `ProCurve(config)# ip policy-class WAN`
   `ProCurve(config-policy-class)# allow list Internet`
   `ProCurve(config-policy-class)# exit`

Finally, you use the `access-policy` command to apply the ACP to the appropriate WAN interface.

Because the ACP contains an implicit “discard any” entry at the end, any traffic that is not explicitly allowed is dropped.

**Permit Routing Updates.** When you configure ACPs, remember that any traffic that you do not explicitly allow will match the implicit “discard any” at the end of the ACP. If you have configured a routing protocol and routing updates are being sent to a router interface, you should ensure that these routing updates are allowed by the ACP you assign to that interface. For example, to allow RIP updates, enter:

   `ProCurve(config)# ip access-list extended Route`
   `ProCurve(config-ext-nacl)# permit udp any any eq rip`

To permit BGP updates, enter:

   `ProCurve(config-ext-nacl)# permit tcp any any eq bgp`

You must then assign this ACL to an ACP. For example, you may want to add entries to the WAN ACP you created in the previous section. You would enter:

   `ProCurve(config)# ip policy-class WAN`
   `ProCurve(config-policy-class)# allow list Internet`
   `ProCurve(config-policy-class)# allow list Route`
   `ProCurve(config-policy-class)# exit`

After configuring the ACP, you must use the `access-policy` command to apply the ACP to the appropriate WAN interface.

**Permit Traffic from Specific Networks.** You may want to restrict access to specific networks. For example, you may want to permit traffic from 10.1.1.0 /30, but deny traffic from 192.168.115.0 /24.
When you are using ACLs with ACPs, remember that you must use a permit entry to both select traffic and to have the Secure Router OS firewall take the action configured in the ACP. If you want to explicitly deny access to a subnet, you must create a permit entry in the ACL and then create a discard entry in the ACP.

Because you want to permit some traffic but deny other traffic, you should create two different ACLs. Enter:

```
ProCurve(config)# ip access-list extended Allow
ProCurve(config-ext-nacl)# permit ip 10.1.1.0 0.0.0.3 any
ProCurve(config-ext-nacl)# exit

ProCurve(config)# ip access-list extended Discard
ProCurve(config-ext-nacl)# permit ip 192.168.115.0 0.0.0.255 any
ProCurve(config-ext-nacl)# exit

ProCurve(config)# ip policy-class WAN
ProCurve(config-policy-class)# allow list Allow
ProCurve(config-policy-class)# discard list Discard
ProCurve(config-ext-nacl)# exit
```

Again, you must use the `access-policy` command to apply the ACP to the appropriate WAN interface.
Viewing ACLs and ACPs

Table 5-11 lists the show commands that you can use to view and troubleshoot ACLs and ACPs.

<table>
<thead>
<tr>
<th>Command</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>show access-lists</td>
<td>displays all of the ACLs configured on the ProCurve Secure Router</td>
</tr>
<tr>
<td>show ip access-lists</td>
<td>displays all of the IP ACLs configured on the ProCurve Secure Router</td>
</tr>
<tr>
<td>show ip policy-class</td>
<td>displays all of the ACPs configured on the ProCurve Secure Router</td>
</tr>
<tr>
<td>show ip policy-sessions</td>
<td>displays the total number of sessions associated with ACPs, the number of sessions per ACP, and detailed information about each device that has established a session</td>
</tr>
<tr>
<td>show ip policy-sessions &lt;policyname&gt;</td>
<td>displays the number of sessions associated with the specified ACP and information about each device that has established a session</td>
</tr>
<tr>
<td>show ip policy-stats</td>
<td>displays statistics about the policies, such as the number of sessions and the number of sessions allowed</td>
</tr>
</tbody>
</table>

Displaying ACLs

To view all of the ACLs that are configured on the ProCurve Secure Router, move to the enable mode context and enter:

**Syntax:** show access-lists

If you are in a different context (other than the basic mode context), you can use the do command:

**Syntax:** do show access-lists
Applying Access Control to Router Interfaces
Viewing ACLs and ACPs

As Figure 5-16 shows, this command lists the following information for each ACL:

- type of ACL—standard or extended
- all entries in the ACLs
- number of packets matched to each entry

<table>
<thead>
<tr>
<th>Command</th>
<th>Output</th>
</tr>
</thead>
</table>
| ProCurve# show access-lists | Extended IP access list Internet  
  permit tcp any any eq www (300 matches)  
  permit tcp any any eq smtp (1 matches)  
  permit tcp any any eq pop3 (0 matches)  
  permit tcp any any eq ftp (0 matches)  
  permit tcp any any eq ftp-data (0 matches)  
  permit tcp any any eq domain (0 matches)  
|  
| | Extended IP access list WAN1  
  permit udp any any eq rip (42 matches)  
  deny tcp any any eq telnet (0 matches)  
  permit tcp host 192.168.20.1 any (0 matches) |

Figure 5-16. Displaying All the ACLs Configured on the Router

You can use this information to review the ACLs that are configured and to ensure that they are configured correctly.

Displaying ACPs

To view all of the ACPs that are configured on the ProCurve Secure Router, move to the enable mode context and enter:

**Syntax:** show ip policy-class

If you are in any other mode context (except the basic mode context), you can enter:

**Syntax:** do show ip policy-class

As Figure 5-17 shows, entries for each ACP are displayed in order so you can determine whether you need to reconfigure the policy. When an ACP is not enforcing your policies in the way you expected, you may have entered commands in the wrong order.
Applying Access Control to Router Interfaces
Viewing ACLs and ACPs

For example, in Figure 5-17 the “allow list Region” entry is entered before the “discard list MatchAll.” If the “discard list MatchAll” was the first entry and the ACL MatchALL included the entry “permit any,” the Secure Router OS would process that entry first and discard all traffic entering the interface. Because the “allow list Region” and the “allow list InWeb” entries are listed first, however, the Secure Router OS will process those entries first and allow any traffic that matches permit entries in these ACLs.

If traffic does not match the “allow list Region” and the “allow list InWeb” entries, it will match the “discard list MatchAll” and be blocked.

Viewing Access Policy Sessions

After you enable the firewall and assign an ACP to an interface, the Secure Router OS firewall checks all the packets entering that interface. When a packet matches an ACL, the Secure Router OS treats it as specified in the ACP. If the ACP allows the packet, then the Secure Router OS firewall can establish a connection (also called a session) between the packet’s source and its destination.

The ProCurve Secure Router records information about that session. To view this information, move to the enable mode context and enter:

**Syntax:** show ip policy-sessions

The Secure Router OS lists each ACP, or policy class, by name. Under a specific policy, you can view the traffic that matched this policy as it arrived on the interface. You can also view information about the traffic, such as:

- source IP address
- source port
- destination IP address
- destination port
Applying Access Control to Router Interfaces

Viewing ACLs and ACPs

If the traffic has been manipulated using NAT, the NAT IP address and port are also listed.

Figure 5-18 illustrates a sample display of sessions.

<table>
<thead>
<tr>
<th>ProCurve# show ip policy-sessions</th>
<th>Src IP Address</th>
<th>Src Port</th>
<th>Dest IP Address</th>
<th>Dst Port</th>
<th>NAT IP Address</th>
<th>NAT Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy class &quot;Inside&quot;:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tcp (80)</td>
<td>192.168.20.1</td>
<td>2001</td>
<td>172.16.1.1</td>
<td>80</td>
<td>d 10.10.3.10</td>
<td>80</td>
</tr>
<tr>
<td>Policy class &quot;Outside&quot;:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tcp (20)</td>
<td>192.168.100.99</td>
<td>1908</td>
<td>172.16.3.10</td>
<td>80</td>
<td>d 10.10.3.10</td>
<td>80</td>
</tr>
<tr>
<td>Policy class &quot;self&quot;:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>icmp (50)</td>
<td>0.0.0.0</td>
<td>10</td>
<td>192.168.100.1</td>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 5-18. Displaying IP Policy Sessions**

If you want to view information about the sessions associated with a specific ACP, enter:

**Syntax:** `show ip policy-sessions <policyname>`

Replace `<policyname>` with the name of the specific ACL.

Viewing Access Policy Statistics

You can also display a summary of ACP statistics by entering the following command from the enable mode context:

**Syntax:** `show ip policy-stats`

The Secure Router OS displays the total number of current sessions. It also lists all ACPs assigned to interfaces and the entries in each ACP. It displays:

- the number of sessions for each ACP (policy class)
- the maximum number of sessions allowed
- the number of hits for each entry in the ACP (policy class)
- the number of bytes for traffic associated with sessions created using the entry
See Figure 5-19 for a sample display.

ProCurve# show ip policy-stats
Global 0 current sessions (255300 max)
Policy-class "Inside":
  121 current sessions (85100 max)
  Entry 1 - allow list MatchAll
    1424221 in bytes, 14222323 out bytes, 123 hits

Policy-class "Outside":
  554 current sessions (85100 max)
  Entry 1 - allow list Region
    2345352 in bytes, 56363536 out bytes, 554 hits
  Entry 2 - allow list InWeb
    0 in bytes, 0 out bytes, 0 hits
  Entry 2 - discard list MatchAll
    0 in bytes, 0 out bytes, 0 hits

**Figure 5-19. Displaying IP Policy Statistics**
Troubleshooting

show Commands

In addition to using `show` commands to view information about ACLs and ACPs and to verify that your configuration is correct, you can use these commands for troubleshooting. For example, suppose that several users call you, complaining that they cannot send traffic to a remote site. However, the PPP 1 interface, which provides the connection to that site, is up, and other users are successfully sending traffic across the interface. You can use the `show ip policy-sessions` command to determine whether or not the traffic is being blocked by an ACP. You can then change the appropriate ACP as required.

Monitoring Packets Matched to an ACP

The Secure Router OS firewall tracks the number of sessions established using each ACP that is configured on the router. By default, the firewall generates a log message after it creates 100 sessions (connections) for an ACP.

You can customize the number of connections made before a log message is generated. For example, you may want to be notified after 50 connections have been made. If you have a large network, on the other hand, you may want to be notified when 200 sessions have been established. To change the default setting, move to the global configuration mode context and enter:

**Syntax:** `ip firewall policy-log threshold <sessions>`

You can specify a number between 0 and 4294967295.

Clearing Existing Policy Sessions

Whenever you change your ACP configurations, you are prompted to clear the existing sessions. This enables you to apply your new configurations. Otherwise, an existing session that violates an ACP that you just configured will remain active.

To clear all of the policy sessions on the router, move to the enable mode context and enter:

`ProCurve# clear ip policy-sessions`
You can also clear a particular policy session. For example, if you enter the `show ip policy-sessions` command and determine that an existing session should be terminated, you can use one of the following commands:

**Syntax:**
```
clear ip policy-sessions <policyname> [ahp | esp | gre | icmp | tcp | udp | <protocol number>] <source A.B.C.D> <source port> <destination A.B.C.D> <destination port>
```

or

```
clear ip policy-sessions <policyname> [ahp | esp | gre | icmp | tcp | udp | <protocol number>] <source A.B.C.D> <source port> <destination A.B.C.D> <destination port> [destination | source] <nat A.B.C.D> <nat port>
```

Enter the command as follows:

- Replace `<policyname>` with the name of the policy class (ACP) associated with that IP policy session.
- Specify the protocol: `ahp`, `esp`, `gre`, `icmp`, `tcp`, `udp`, or a protocol number.
- Replace `<source A.B.C.D>` with the source IP address.
- Replace `<source port>` with the port specified by the source. Use hexadecimal format for `ahp`, `esp`, and `gre`; use the decimal format for all other protocols.
- Replace `<destination A.B.C.D>` with the destination IP address.
- Replace `<destination port>` with the destination port. Use hexadecimal format for `AHP`, `ESP`, and `GRE`; use the decimal format for all other protocols.

The remaining options apply only to NAT:

- Include the `destination` option to select a NAT session that is based on the destination address. Include the `source` option to select a NAT session that is based on the source IP address.
- Replace `<nat A.B.C.D>` with the NAT IP address that replaced the original IP address.
- Replace `<nat port>` with the port used by NAT. Use hexadecimal format for `AHP`, `ESP`, and `GRE`; use decimal format for all other protocols.

**Note**

Rather than input this entire command, you can enter the `show ip policy-sessions` command to display the current sessions and then copy the second part of the command, beginning with the source IP address, from the display. (See Figure 5-20.)
Applying Access Control to Router Interfaces
Troubleshooting

Figure 5-20. Using the Information from the show ip policy-sessions Command to Clear a Specific Session

Clear ACL Counters

When troubleshooting problems with the ACLs that are configured on a router, clear the counters for matches to ACL entries so that you can isolate the problem. From the enable mode context, enter:

Syntax: clear access-list [<listname>]

If you want to clear all counters, enter:

ProCurve# clear access-list

If you want to clear counters for a particular ACL, use the <listname> option:

ProCurve# clear access-list <listname>

For example, if you want to clear the counters for the Inside ACL, enter:

ProCurve# clear access-list Inside

<table>
<thead>
<tr>
<th>Src IP Address</th>
<th>Src Port</th>
<th>Dest IP Address</th>
<th>Dst Port</th>
<th>NAT IP Address</th>
<th>NAT Port</th>
</tr>
</thead>
</table>
| Policy class "Inside":
tcp (80)        | 192.168.20.1 | 2001          | 172.16.1.1 | 80           | d 10.10.3.10 | 80     |

Policy class "Outside":
tcp (20)
192.168.100.99 1908 172.16.3.10 80 d 10.10.3.10 80

Policy class "self":
icmp (50)
0.0.0.0 10 192.168.100.1 10

Policy class "Private":

Highlight and copy the entire line into your command
Debug ACLs

You can debug events associated with a particular ACL. From the enable mode context, enter:

**Syntax:** `debug access-list <listname>`

Replace `<listname>` with the name of the ACL you want to debug.

For example, if you want to debug the Inside ACL, enter:

```
ProCurve# debug access-list Inside
```

To end the debug, enter one of the following commands:

**Syntax:** `no debug access-list <listname>`

**Syntax:** `undebug all`

The `undebug all` command ends all debug sessions on the ProCurve Secure Router.

**Note**

Debug commands are processor intensive and could compromise performance.
Quick Start

This section provides the commands you will need to quickly configure and apply access controls to interfaces on the ProCurve Secure Router. There are two access control mechanisms on the ProCurve Secure Router:

- access control lists (ACLs)
- access control policies (ACPs)

ACLs can be used alone or in combination with ACPs. This “Quick Start” section first explains how to use just ACLs to control access. Specifically, it explains how to configure an ACL and apply it directly to an interface.

This section then outlines the steps for configuring an ACP. Specifically, it explains:

- how to configure an ACL
- how to configure an ACP
- how to apply the ACP to an interface

Only a minimal explanation is provided in the “Quick Start” section. For a detailed explanation of ACLs and ACPs, read the entire chapter.

Note

If you are not familiar with ACLs and ACPs, ProCurve Networking strongly recommends that you read the entire chapter before you begin configuring and applying access controls to the interfaces on your ProCurve Secure Router. If you do not thoroughly understand how ACLs and ACPs work, you could inadvertently allow traffic that should be blocked or, conversely, block traffic that should be allowed.

Enabling the Built-in Firewall

Before you begin configuring access control on router interfaces, you should enable the ProCurve Secure Route OS firewall. From the global configuration mode context, enter:

ProCurve(config)# ip firewall

Note

If you are using ACLs only, you do not have to enable the Secure Router OS firewall. If you are using ACPs, you must enable the firewall.
Configuring an ACL and Applying It Directly to an Interface

This section explains how to use ACLs by themselves to enforce access control on particular interfaces. If you use ACLs in this way, you can apply two ACLs to each interface: one ACL to control incoming traffic and one ACL to control outgoing traffic.

If you apply ACLs directly to router interfaces, the ProCurve Secure Router uses the ACL to both select the traffic and to perform the action on that traffic. If you have experience configuring access control on Cisco routers, you will find that this process is similar to that used on the Cisco router.

An ACL contains two parts:
- action
- packet pattern

You can define one of two actions: permit or deny. The packet pattern you specify depends on the type of ACL you create: a standard ACL or an extended ACL. A standard ACL allows you to match traffic based on source IP address. An extended ACL allows you to match traffic based on:
- source address and destination address
- other fields in the IP header

Before you begin configuring an ACL, you must determine if you want to configure a standard ACL or an extended ACL.

To configure an ACL and apply it to an interface, complete the following steps:

1. Create the ACL. From the global configuration mode context, enter:
   
   **Syntax:** ip access-list [standard |extended] <listname>
   
   For example, to create an extended ACL, enter:
   
   ProCurve(config)# ip access-list extended Inside

2. From the ACL configuration mode context, configure permit or deny entries.
   
   a. If you are configuring a standard ACL, enter:
      
      **Syntax:** [permit | deny] any [host {<A.B.C.D> | <hostname>} | <A.B.C.D> <wildcard bits>]
      
      For example, to permit any packet, enter:
      
      ProCurve(config-std-nacl)# permit any
Applying Access Control to Router Interfaces

Quick Start

To permit or deny a specific host, use the host keyword. For example, enter:

```
ProCurve(config-std-nacl)# deny host 192.168.115.90
```

b. If you are configuring an extended ACL, enter:

**Syntax:** permit | deny <protocol> <source address> <source port> <destination address> <destination port>

Replace `<protocol>` with one of the following:
- ahp
- esp
- gre
- icmp
- ip
- tcp
- udp

To specify a source or destination address, use the following syntax:

**Syntax:** any | host <A.B.C.D> | hostname <hostname> | <A.B.C.D> <wildcard bits>

For example, if you want to permit all TCP traffic from any source to any destination, enter:

```
ProCurve(config-ext-nacl)# permit tcp any any
```

To deny all ICMP traffic from a specific host, such as host 192.168.115.90, to any destination, enter:

```
ProCurve(config-ext-nacl)# deny icmp host 192.168.115.90 any
```

To deny ICMP traffic from a range of IP addresses to a specific destination, enter:

```
ProCurve(config-ext-nacl)# deny icmp <A.B.C.D> <wildcard bits> host <A.B.C.D>
```

**Note**
The entries are processed in the order in which you enter them. In addition, each ACL contains an implicit “deny any” entry at the end of the list. If you do not create an entry to allow a specific type of traffic, it will be denied.

3. After configuring the entries for the ACL, enter:

**Syntax:** exit

4. To apply the ACL to an interface, move to the configuration mode context for that interface.

```
ProCurve(config)# interface <interface> <number>
```
Applying Access Control to Router Interfaces
Quick Start

Valid interfaces include PPP interfaces, Frame Relay subinterfaces, ATM subinterfaces, HDLC interfaces, Ethernet interfaces, and demand interfaces. (If you have enabled support for virtual LANs [VLANs], you must apply the ACL to an Ethernet subinterface.)

5. Apply the ACL to the interface by entering the following command from the appropriate interface configuration mode context:

**Syntax:** ip access-group <listname> [in | out]

For example, if you want to apply the Inside ACL to the Ethernet 0/1 interface to control traffic incoming from the LAN to the WAN, enter:

ProCurve(config-eth 0/1)# ip access-group Inside out

For more information about using ACLs alone to control access to a router interface, read the detailed explanation in this chapter.

Configuring ACPs

When you configure ACPs, you use ACLs to select the traffic. However, the ACP, rather than the ACL, dictates the action that the ProCurve Secure Router takes.

Although ACPs are a little more complicated to configure and apply, they provide greater flexibility than ACLs do by themselves. With ACPs, you can apply more than two ACLs to an interface. Each ACP can include an unlimited number of entries, which reference an unlimited number of ACLs.

There is another major difference between ACLs and ACPs: you can use ACPs to configure network address translation (NAT), which ACLs, by themselves, do not support.

You can apply one ACP to each interface, and that ACP will affect only incoming traffic on the interface. In addition, you can configure a maximum of five ACPs on the ProCurve Secure Router.

**Selecting the Traffic.** Creating an ACL to select the traffic may at first seem confusing because an ACL entry itself includes an action as well as a packet pattern. The action can be either deny or permit, and the packet pattern can vary, depending on the type of ACL you are creating:

- standard ACL—packet patterns based on source IP address
- extended ACL—packet patterns based on protocol, source and destination IP addresses, and, optionally, UDP or TCP port
Applying Access Control to Router Interfaces
Quick Start

When an ACL is used in conjunction with an ACP, a permit entry means that the traffic defined by the packet pattern is selected for the action specified in the ACP. A deny entry, on the other hand, means that the traffic is excluded from the action specified in the ACP. If the ProCurve Secure Router detects traffic that matches a deny entry in the ACL, it does not take the action specified in the ACP entry. Instead, the router stops processing the ACL and the related entry in the ACP and moves to the next entry in the ACP. For more information about matching traffic to ACLs in ACPs, see “Processing ACPs” on page 5-39.

ACPs support three types of actions:
- allow traffic selected by the ACL
- discard traffic selected by the ACL
- manipulate traffic selected by the ACL for NAT

This chapter focuses on creating ACPs to allow or discard traffic that is selected by the ACL. NAT is discussed in Chapter 6: Configuring Network Address Translation.

**Note**
Remember that you must enable the Secure Router OS firewall before the ACPs that you apply to interfaces can take effect.

To configure an ACL and apply it to an ACP, complete the following steps:

1. Create the ACL. From the global configuration mode context, enter:
   ```
   Syntax: ip access-list [standard | extended] <listname>
   ```
   For example, to create an extended ACL, enter:
   ```
   ProCurve(config)# ip access-list extended Inside
   ```

2. From the ACL configuration mode context, configure permit or deny entries.
   a. If you are configuring a standard ACL, enter:
      ```
      Syntax: [permit | deny] [any | host <A.B.C.D> | <hostname>] <wildcard bits>
      ```
      For example, to select any packet, enter:
      ```
      ProCurve(config-std-nacl)# permit any
      ```
      To select a specific host, enter:
      ```
      ProCurve(config-std-nacl)# permit host 192.168.115.90
      ```
To exclude a specific host from the action that you will specify in the ACP, enter:

ProCurve(config-std-nacl)# deny host 192.168.115.90

b. If you are configuring an extended ACL, enter:

Syntax: permit | deny <protocol> <source address> <source port> <destination address> <destination port>

Replace <protocol> with one of the following:

- AHP
- ESP
- GRE
- ICMP
- IP
- TCP
- UDP

To specify a source or destination address, use the following syntax:

Syntax: any | host <A.B.C.D> | hostname <hostname> | <A.B.C.D> <wildcard bits>

For example, if you want to select TCP traffic from any source to any destination, enter:

ProCurve(config-ext-nacl)# permit tcp any any

If you want to exclude all ICMP traffic from a specific host, such as host 192.168.115.90, to any destination, enter:

ProCurve(config-ext-nacl)# deny icmp host 192.168.115.90 any

To exclude ICMP traffic from a range of IP addresses to a specific destination, enter:

Syntax: deny icmp <A.B.C.D> <wildcard bits> host <A.B.C.D>

**Note**

The entries are processed in the order in which you enter them. In addition, each ACL contains an implicit “deny any” entry at the end of the list. If you do not create an entry to allow a specific type of traffic, it will be denied. That is, the traffic will be excluded from the action specified in the related entry in the ACP.

3. After configuring the entries for the ACL, exit the ACL.

Syntax: exit
4. From the global configuration mode context, enter the following command to create an ACP:

**Syntax:** `ip policy-class <policyname>`

Replace `<policyname>` with a unique name that is a maximum of 255 alphanumeric characters.

You are moved to the policy class configuration mode context.

5. Create entries to allow packets selected by an ACL or to deny packets selected by an ACL:

**Syntax:**
- `allow list <listname>`
- `discard list <listname>`

**Note**
You can also NAT the packets selected by an ACL. This option is discussed in Chapter 6: Configuring Network Address Translation.

The order in which you create entries in the ACP affects how the policy is enforced. The first entry you create is the first entry processed. After the Secure Router OS firewall matches a packet to an ACL and performs the action specified in the corresponding ACP entry, it stops processing the ACP.

6. To apply the ACP to an interface, move to the configuration mode context for that interface:

```
ProCurve(config)# interface <interface> <number>
```

Valid interfaces include PPP interfaces, Frame Relay subinterfaces, ATM subinterfaces, HDLC interfaces, Ethernet interfaces, and demand interfaces. (If you have enabled support for virtual LANs [VLANs], you must apply the ACL to an Ethernet subinterface.)

7. Apply the ACP to the interface by entering the following command from the appropriate interface configuration mode context:

**Syntax:** `access-policy <policyname>`

For example, if you want to apply the WAN ACP to the PPP 1 interface, enter:

```
ProCurve(config-ppp 1)# access-policy WAN
```

For more information about using ACPs to control access to a router interface, read the detailed explanation in this chapter.