Understanding Cisco IOS ACL Support

This chapter describes Cisco IOS ACL support on the Cisco 7600 series routers:

For complete information about configuring Cisco IOS ACLs, refer to the *Cisco IOS Security Configuration Guide*, Release 12.2 at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/scfrpf.html

Cisco IOS ACL Configuration Guidelines and Restrictions

The following guidelines and restrictions apply to Cisco IOS ACL configurations:

- You can apply Cisco IOS ACLs directly to Layer 3 ports and to VLAN interfaces.
- You can apply VLAN ACLs (VACLs) to VLANs (refer to Chapter 38, "Configuring VLAN ACLs").
- Each type of ACL (IP, IPX, and MAC) filters only traffic of the corresponding type. A Cisco IOS MAC ACL never matches IP or IPX traffic.
- The PFC does not provide hardware support for Cisco IOS IPX ACLs. Cisco IOS IPX ACLs are supported in software on the MSFC.
- By default, the MSFC sends Internet Control Message Protocol (ICMP) unreachable messages when a packet is denied by an access group.

  With the **ip unreachables** command enabled (which is the default), the supervisor engine drops most of the denied packets in hardware and sends only a small number of packets to the MSFC to be dropped (10 packets per second, maximum), which generates ICMP-unreachable messages.

  To eliminate the load imposed on the MSFC CPU by the task of dropping denied packets and generating ICMP-unreachable messages, you can enter the **no ip unreachables** interface configuration command to disable ICMP unreachable messages, which allows all access group-denied packets to be dropped in hardware.
- ICMP unreachable messages are not sent if a packet is denied by a VACL.
- Access Control Entries (ACEs) matching IP options are not supported in the hardware forwarding path.

Hardware and Software ACL Support

Access control lists (ACLs) can be processed in hardware by the Policy Feature Card (PFC), a Distributed Forwarding Card (DFC), or in software by the Multilayer Switch Feature Card (MSFC). The following behavior describes software and hardware handling of ACLs:

- The PFC provides more efficient hardware support for named ACLs than it can for numbered ACLs.
- ACL flows that match a “deny” statement in standard and extended ACLs (input and output) are dropped in hardware if “ip unreachables” is disabled.
- ACL flows that match a “permit” statement in standard and extended ACLs (input and output) are processed in hardware.
- VLAN ACL (VACL) flows are processed in hardware. If a field specified in a VACL is not supported by hardware processing that field is ignored (for example, the **log** keyword in an ACL) or the whole configuration is rejected (for example, a VACL containing IPX ACL parameters).
- VACL logging is processed in software.
- Dynamic ACL flows are processed in hardware.
- Idle timeout is processed in software.

Note Idle timeout is not configurable. Cisco 7600 series routers do not support the **access-enable host timeout** command.

- IP accounting for an ACL access violation on a given port is supported by forwarding all denied packets for that port to the MSFC for software processing without impacting other flows.
- The PFC does not provide hardware support for Cisco IOS IPX ACLs. Cisco IOS IPX ACLs are supported in software on the MSFC.
- Extended name-based MAC address ACLs are supported in hardware.
- The following ACL types are processed in software:
  - Internetwork Packet Exchange (IPX) access lists
  - Standard XNS access list
  - Extended XNS access list
  - DECnet access list
  - Extended MAC address access list
  - Protocol type-code access list

Note IP packets with a header length of less than five will not be access controlled.

- Unless you configure optimized ACL logging (OAL), flows that require logging are processed in software without impacting nonlogged flow processing in hardware (see the “Optimized ACL Logging with a PFC3” section on page 36-3).
- The forwarding rate for software-processed flows is substantially less than for hardware-processed flows.
- When you enter the **show ip access-list** command, the match count displayed does not include packets processed in hardware.

Optimized ACL Logging with a PFC3

These sections describe OAL:

Understanding OAL

Optimized ACL Logging (OAL) provides hardware support for ACL logging. Unless you configure OAL, packets that require logging are processed completely in software on the MSFC. OAL permits or drops packets in hardware on the PFC3 and uses an optimized routine to send information to the MSFC3 to generate the logging messages.

OAL Guidelines and Restrictions

The following guidelines and restrictions apply to OAL:

- OAL and VACL capture are incompatible. Do not configure both features on the router. With OAL configured, use SPAN to capture traffic.
- OAL is supported only on the PFC3.
- OAL supports only IPv4 unicast packets.
- OAL supports VACL logging of permitted ingress traffic
- OAL does not provide hardware support for the following:
  - Reflexive ACLs
  - ACLs used to filter traffic for other features (for example, QoS)
  - Exception packets (for example, TTL failure and MTU failure)
  - Packets with IP options
  - Packets addressed at Layer 3 to the router
  - Packets sent to the MSFC3 to generate ICMP unreachable messages
  - Packets being processed by features not accelerated in hardware
- To provide OAL support for denied packets, enter the **mls rate-limit unicast ip icmp unreachable acl-drop 0** command.

Configuring OAL

These sections describe how to configure OAL:

Note

- For complete syntax and usage information for the commands used in this section, refer to the *Cisco 7600 Series Router Cisco IOS Command Reference*.
- To provide OAL support for denied packets, enter the **mls rate-limit unicast ip icmp unreachable acl-drop 0** command.

Configuring OAL Global Parameters

To configure global OAL parameters, perform this task:

Command	Purpose
`Router(config)# logging ip access-list cache {{entries number_of_entries} | {interval seconds} | {rate-limit number_of_packets} | {threshold number_of_packets}}`	Sets OAL global parameters.
`Router(config)# no logging ip access-list cache {entries | interval | rate-limit | threshold}`	Reverts OAL global parameters to defaults.

When configuring OAL global parameters, note the following information:

- **entries** *number_of_entries*:
  - Sets the maximum number of entries cached.
  - Range: 0–1,048,576 (entered without commas).
  - Default: 8000.
- **interval** *seconds*:
  - Sets the maximum time interval before an entry is sent to be logged. Also if the entry is inactive for this duration it is removed from the cache.
  - Range: 5–86,400 (1440 minutes or 24 hours, entered without commas).
  - Default: 300 seconds (5 minutes).
- **rate-limit** *number_of_packets*:
  - Sets the number of packets logged per second in software.
  - Range: 10–1,000,000 (entered without commas).
  - Default: 0 (rate limiting is off and all packets are logged).
- **threshold** *number_of_packets*:
  - Sets the number of packet matches before an entry is logged.
  - Range: 1–1,000,000 (entered without commas).
  - Default: 0 (logging is not triggered by the number of packet matches).

Configuring OAL on an Interface

To configure OAL on an interface, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {{type[1] slot/port}`	Specifies the interface to configure.
Step 2	`Router(config-if)# logging ip access-list cache in`	Enables OAL for ingress traffic on the interface.
	`Router(config-if)# no logging ip access-list cache`	Disables OAL on the interface.
Step 3	`Router(config-if)# logging ip access-list cache out`	Enables OAL for egress traffic on the interface.
	`Router(config-if)# no logging ip access-list cache`	Disables OAL on the interface.

1. *type* = any that supports Layer 3-switched traffic.

Displaying OAL Information

To display OAL information, perform this task:

Command	Purpose
`Router # show logging ip access-list cache`	Displays OAL information.

Clearing Cached OAL Entries

To clear cached OAL entries, perform this task:

Command	Purpose
`Router # clear logging ip access-list cache`	Clears cached OAL entries.

Guidelines and Restrictions for Using Layer 4 Operators in ACLs

These sections describe guidelines and restrictions when configuring ACLs that include Layer 4 port operations:

- Determining Layer 4 Operation Usage, page 36-5
- Determining Logical Operation Unit Usage, page 36-6

Determining Layer 4 Operation Usage

You can specify these types of operations:

- gt (greater than)
- lt (less than)
- neq (not equal)

- eq (equal)
- range (inclusive range)

We recommend that you do not specify more than *nine different* operations on the same ACL. If you exceed this number, each new operation might cause the affected ACE to be translated into more than one ACE.

Use the following two guidelines to determine Layer 4 operation usage:

- Layer 4 operations are considered different if the operator or the operand differ. For example, in this ACL there are three different Layer 4 operations ("gt 10" and "gt 11" are considered two different Layer 4 operations):

```
... gt 10 permit
... lt 9 deny
... gt 11 deny
```

Note There is no limit to the use of "eq" operators as the "eq" operator does not use a logical operator unit (LOU) or a Layer 4 operation bit. See the "Determining Logical Operation Unit Usage" section on page 36-6 for a description of LOUs.

- Layer 4 operations are considered different if the same operator/operand couple applies once to a source port and once to a destination port. For example, in this ACL there are two different Layer 4 operations because one ACE applies to the source port and one applies to the destination port.

```
... Src gt 10 ...
... Dst gt 10
```

Determining Logical Operation Unit Usage

Logical operation units (LOUs) are registers that store operator-operand couples. All ACLs use LOUs. There can be up to 32 LOUs; each LOU can store two different operator-operand couples with the exception of the range operator. LOU usage per Layer 4 operation is as follows:

- gt uses 1/2 LOU
- lt uses 1/2 LOU
- neq uses 1/2 LOU
- range uses 1 LOU
- eq does not require a LOU

For example, this ACL would use a single LOU to store two different operator-operand couples:

```
... Src gt 10 ...
... Dst gt 10
```

A more detailed example follows:

```
ACL1
... (dst port) gt 10 permit
... (dst port) lt 9 deny
... (dst port) gt 11 deny
... (dst port) neq 6 permit
... (src port) neq 6 deny
... (dst port) gt 10 deny

ACL2
... (dst port) gt 20 deny
```

```
... (src port) lt 9 deny
... (src port) range 11 13 deny
... (dst port) neq 6 permit
```

The Layer 4 operations and LOU usage is as follows:

- ACL1 Layer 4 operations: 5
- ACL2 Layer 4 operations: 4
- LOUs: 4

An explanation of the LOU usage follows:

- LOU 1 stores “gt 10” and “lt 9”
- LOU 2 stores “gt 11” and “neq 6”
- LOU 3 stores “gt 20” (with space for one more)
- LOU 4 stores “range 11 13” (range needs the entire LOU)