Validate Security ACLs on Catalyst 9000 Switches

Contents

Introduction

This document describes how to verify and troubleshoot ACLs (access control lists) on Catalyst 9000 series switches.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on these hardware versions:

- C9200
- C9300
- C9400
- C9500
- C9600

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

> **Note**: Consult the appropriate configuration guide for the commands used to enable these features on other Cisco platforms.

Background Information

ACLs filter traffic as it passes through a router or switch and permit or deny packets that cross specified interfaces. An ACL is a sequential collection of permit and deny conditions that apply to packets. When a packet is received on an interface, the switch compares the fields in the packet against any applied ACLs in order to verify that the packet has the required permissions to be forwarded, based on the criteria specified in the access lists. One by one, it tests packets against the conditions in an access list. The first match decides whether the switch accepts or rejects the packets. Because the switch stops testing after the first match, the order of conditions in the list is critical. If no conditions match, the switch rejects the packet. If there are no restrictions, the switch forwards the packet; otherwise, the switch drops the packet. The switch can use ACLs on all packets it forwards.

You can configure access lists in order to provide basic security for your network. If you do not configure ACLs, all packets that pass through the switch can be allowed onto all network parts. You can use ACLs in order to control which hosts can access different parts of a network or to decide which types of traffic are forwarded or blocked at router interfaces. For example, you can forward e-mail traffic but not Telnet traffic.

Terminology

ACE	Access Control Entry (ACE) - A single rule/line within an ACL
ACL	Access Control List (ACL) - A group of ACEs applied to a port
DACL	Downloadable ACL (DACL) - An ACL pushed dynamically via the ISE security policy

PACL	Port ACL (PACL) - An ACL applied to a Layer 2 interface
RACL	Routed ACL (RACL) - An ACL applied to a Layer 3 interface
VACL	VLAN ACL (VACL) - An ACL applied to a VLAN
GACL	Group ACL (GACL) - An ACL dynamically assigned to a user group or client based on their identity
IP ACL	Is used to classify IPv4/IPv6 packets. These rules contain various Layer-3 and Layer-4 packet fields and attributes including but not limited to source and destination IPv4 addresses, TCP/UDP source and destination ports, TCP flags and DSCP, and so on.
MACL	Mac Address ACL (MACL) - Used to classify non-IP packets. Rules contain various Layer-2 fields and attributes including source/dest MAC address, ether type, and so on.
L4OP	Layer 4 Operator Port (L4OP) - Matches logic that is other than EQ (Equal To). GT (greater than), LT (less than), NE (not equal to), and RANGE (from-to)
VCU	Value Comparison Unit (VCU) - L4OPs are translated into VCU in order to perform classification on Layer 4 headers
VMR	Value Mask Result (VMR) - An ACE entry is internally programmed in TCAM as a VMR
CGD	Class Group Database (CGD) - Where FMAN-FP stores ACL content
Classes	How ACEs are identified in CGD
CG	Class Group (CG) - A group of classes on how ACLs are identified in CGD
CGE	Class Group Entry (CGE) - An ACE entry stored within a Class Group
FMAN	Forwarding Manager (FMAN) - The programming layer between Cisco IOS® XE and hardware
FED	Forwarding Engine Driver (FED) - The component that programs the hardware of the device

ACL Resource Utilization Examples

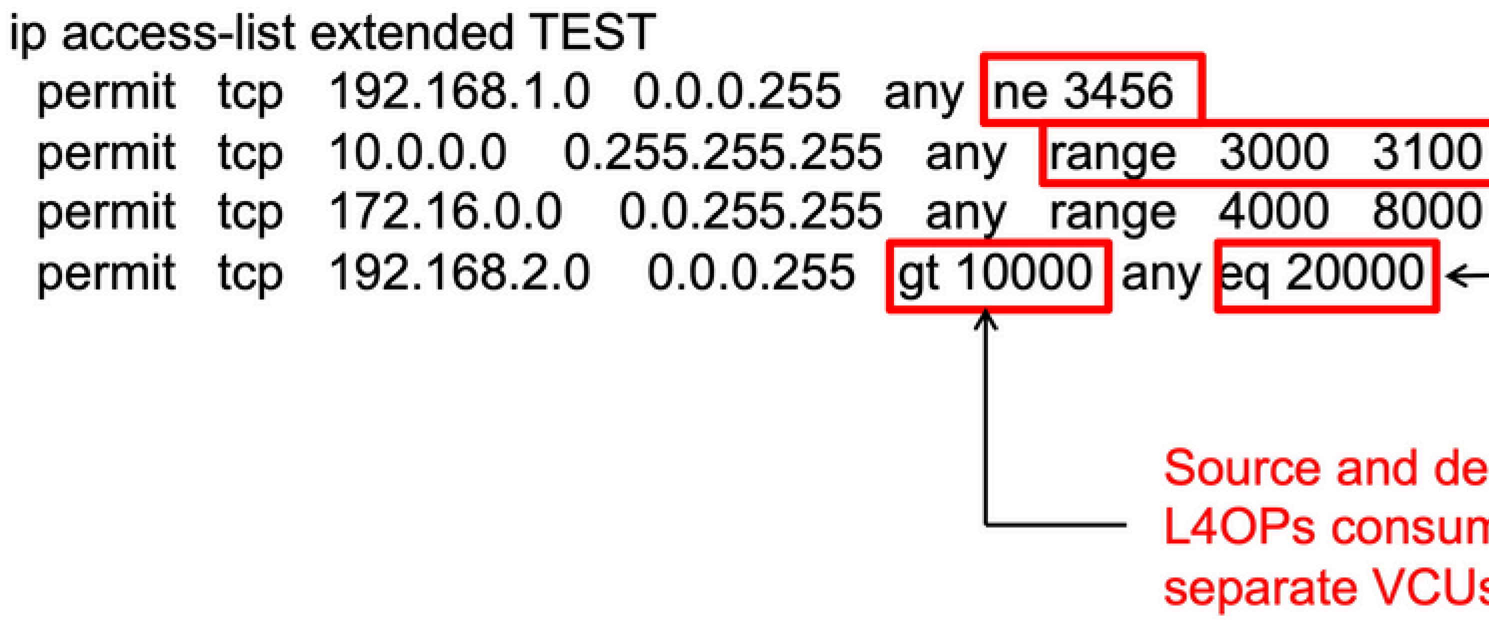
Three examples are given here in order to demonstrate how ACLs consume TCAM, L4OPs, and VCUs.

Example 1. IPv4 TCAM

```
access-list 101 permit ip any 10.1.1.0 0.0.0.255
access-list 101 permit ip any 10.1.2.0 0.0.0.255
access-list 101 permit ip any 10.1.3.0 0.0.0.255
access-list 101 permit ip any 10.1.4.0 0.0.0.255
access-list 101 permit ip any 10.1.5.0 0.0.0.255
```

	TCAM Entries	L4OPs	VCUs
Consumption	5	0	0

Example 2. IPv4 TCAM/L4OP/VCU



```
ip access-list extended TEST
  permit   tcp   192.168.1.0   0.0.0.255   any  ne 3456
  permit   tcp   10.0.0.0    0.255.255.255   any   range   3000   3100
  permit   tcp   172.16.0.0    0.0.255.255   any   range   4000   8000
  permit   tcp   192.168.2.0    0.0.0.255   gt 10000  any  eq 20000
```

Source and de
L4OPs consum
separate VCUs

<#root>

```
ip access-list extended TEST
10 permit tcp 192.168.1.0 0.0.0.255 any
```

```
neq 3456
```

```
<-- 1 L4OP, 1 VCU
```

```
20 permit tcp 10.0.0.0 0.255.255.255 any
```

```
range 3000 3100  <-- 1 L4OP, 2 VCU


30 permit tcp 172.16.0.0 0.0.255.255 any

range 4000 8000  <-- 1 L4OP, 2 VCU


40 permit tcp 192.168.2.0 0.0.0.255

gt 10000

 any

eq 20000 <-- 2 L4OP, 2 VCU
```

	TCAM Entries	L4OPs	VCUs
Consumption	4	5	7

Example 3. IPv6 TCAM/L4OP/VCU

IPv6 ACEs use two TCAM entries versus one for IPv4. In this example, four ACEs consume eight TCAM instead of four.

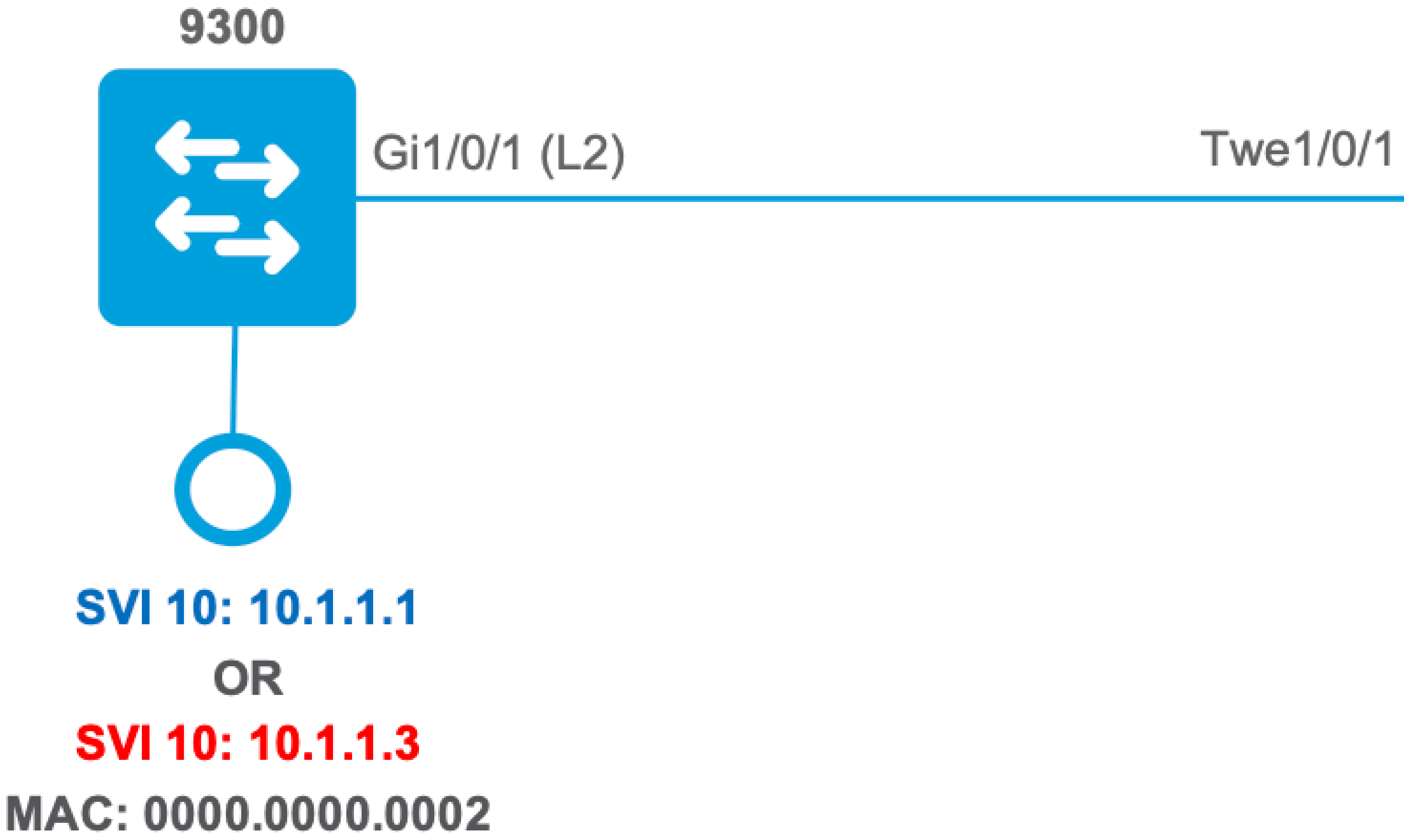
<#root>

```
ipv6 access-list v6TEST
sequence 10 deny ipv6 any 2001:DB8:C18::/48 fragments
sequence 20 deny ipv6 2001:DB8::/32 any
sequence 30 permit tcp host 2001:DB8:C19:2:1::F host 2001:DB8:C18:2:1::1

eq bgp  <-- One L4OP & VCU


sequence 40 permit tcp host 2001:DB8:C19:2:1::F

eq bgp

 host 2001:DB8:C18:2:1::1

<-- One L4OP & VCU
```

	TCAM Entries	L4OPs	VCUs
Consumption	8	2	2

Topology

The 9300 VLAN 10 SVI uses one of the two IP addresses shown in this image, based on whether a forward or drop result is shown in the examples.



Configure and Verify

This section covers how to verify and troubleshoot ACL programming in software and hardware.

Scenario 1. PACL (IP ACL)

PACLs are assigned to a Layer 2 interface.

- Security Boundary: Ports or VLANs
- Attachment: Layer 2 Interface
- Direction: Ingress or Egress (one at a time)
- Supported ACL Types: MAC ACL & IP ACLs (standard or extended)

Configure PACL with IP ACL

<#root>

```
9500H(config)#
ip access-list extended TEST          <-- Create a named extended ACL

9500H(config-ext-nacl)#
permit ip host 10.1.1.1 any
9500H(config-ext-nacl)#
permit udp host 10.1.1.1 eq 1000 host 10.1.1.2
```

```
9500H#

show access-lists TEST                        <-- Display the ACL configured


Extended IP access list TEST
    10 permit ip host 10.1.1.1 any
    20 permit udp host 10.1.1.1 eq 1000 host 10.1.1.2

9500H(config)#

interface twentyFiveGigE 1/0/1        <-- Apply ACL to Layer 2 interface


9500H(config-if)#

ip access-group TEST in



9500H#

show running-config interface twentyFiveGigE 1/0/1


Building configuration...

Current configuration : 63 bytes
!
interface TwentyFiveGigE1/0/1

 ip access-group TEST in                             <-- Display the ACL applied to the interface


end
```

Verify PACL

Retrieve the IF_ID associated with the interface.

<#root>

```
9500H#

show platform software fed active ifm interfaces ethernet


Interface

 IF_ID

               State
----------------------------------------------------------------

TwentyFiveGigE1/0/1



0x00000008

          READY
```

```
<-- IF_ID value for Tw1/0/1
```

Verify the Class group ID (CG ID) bound to the IF_ID.

```
<#root>

9500H#

show platform software fed active acl interface 0x8       <-- IF_ID with leading zeros omitted


########################################################
########                              ##################
#######   Printing Interface Infos     #################
########                              ##################
########################################################

INTERFACE:

TwentyFiveGigE1/0/1                                  <-- Confirms the interface matches the IF_ID


MAC 0000.0000.0000
########################################################
    intfinfo: 0x7f8cfc02de98
    Interface handle: 0x7e000028


Interface Type: Port                                        <-- Type: Port indicates Layer 2 interface



if-id: 0x0000000000000008                                   <-- IF_ID 0x8 is correct


    Input IPv4: Policy Handle: 0x5b000093


Policy Name: TEST                                       <-- The named ACL bound to this interface



    CG ID: 9                                           <-- Class Group ID for this entry



CGM Feature: [0] acl                                     <-- Feature is ACL


        Bind Order: 0
```

ACL information associated with the CG ID.

```
<#root>

9500H#
```

```
show platform software fed active acl info acl-cgid 9  <-- The CG ID associated to the ACL TEST

########################################################
#########                             ##################
########      Printing CG Entries      #################
#########                             ##################
########################################################
===================================

ACL CG (acl/9): TEST type: IPv4     <-- feature ACL/CG ID 9: ACl name TEST : ACL type IPv4


Total Ref count 1
---------------------------------

1 Interface



<-- ACL is applied to one interface


---------------------------------
  region reg_id: 10
    subregion subr_id: 0
      GCE#:1

#flds: 2



l4:N

 matchall:N deny:N

<-- #flds: 2 = two fields in entry | l4:N (no Layer 4 port match)


        Result: 0x01010000


ipv4_src: value

 =

0x0a010101

,

mask = 0xffffffff



<-- src 0x0a010101 hex = 10.1.1.1 | mask 0xffffffff = exact host match




ipv4_dst: value

 =

0x00000000, mask = 0x00000000



<--
```

```
dst & mask = 0x00000000 = match any

       GCE#:1 #flds: 4

l4:Y

 matchall:N deny:N

<-- #flds: 4 = four fields in entry | l4:Y (ACE uses UDP port L4 match)


        Result: 0x01010000


ipv4_src: value = 0x0a010101, mask = 0xffffffff  <-- Exact match (host) 10.1.1.1



ipv4_dst: value = 0x0a010102, mask = 0xffffffff  <-- Exact match (host) 10.1.1.2



ip_prot: start = 17, end = 17                    <-- protocol 17 is UDP



l4_src: start = 1000, end = 1000                 <-- matches eq 1000 (equal UDP port 1000)
```

Policy information on the CG ID, as well as what interfaces use the CG ID.

<#root>

```
9500H#

show platform software fed active acl policy 9      <-- Use the CG ID value


########################################################
#########                            ##################
########     Printing Policy Infos     #################
#########                            ##################
########################################################


INTERFACE: TwentyFiveGigE1/0/1                            <-- Interface with ACL applied


MAC 0000.0000.0000
########################################################
    intfinfo: 0x7f8cfc02de98
    Interface handle: 0x7e000028
    Interface Type: Port


if-id: 0x0000000000000008                             <-- The Interface IF_ID 0x8


------------
```

```
Direction: Input                                          <-- ACl is applied in the ingress direction

Protocol Type:IPv4                                        <-- Type is IPv4

    Policy Intface Handle: 0x880000c1
    Policy Handle: 0x5b000093

########################################################
#########                             ##################
########      Policy information       #################
#########                             ##################
########################################################
Policy handle      : 0x5b000093

Policy name        : TEST                                 <-- ACL Name TEST

ID                 : 9                                    <-- CG ID for this ACL entry

Protocol           : [3] IPV4

Feature            : [1] AAL_FEATURE_PACL                 <-- ASIC feature is PACL

Number of ACLs     : 1

########################################################
## Complete policy ACL information
########################################################
Acl number    : 1
=====================================
Acl handle      : 0x320000d2
Acl flags       : 0x00000001

Number of ACEs

  : 3

<-- 3 ACEs: two explicit and the implicit deny entry

  Ace handle [1] : 0xb700010a
  Ace handle [2] : 0x5800010b

Interface(s):

        TwentyFiveGigE1/0/1                               <-- The interface ACL is applied

########################################################
#########                             ##################
########  Policy instance information  #################
#########                             ##################
########################################################
Policy intf handle    : 0x880000c1
Policy handle         : 0x5b000093
ID                    : 9
Protocol              : [3] IPV4
```

```
Feature                : [1] AAL_FEATURE_PACL
Direction              : [1] Ingress
Number of ACLs         : 1
Number of VMRs         : 3------------
```

Confirm PACL is working.

> **Note**: When you enter the show ip access-lists privileged EXEC command, the match count displayed does not account for packets that are access controlled in hardware. Use the show platform software fed switch {*switch_num*|active|standby}acl counters hardware privileged EXEC command in order to obtain some basic hardware ACL statistics for switched and routed packets.

<#root>

```
### Ping originated from neighbor device with source 10.1.1.1 ###


C9300#

ping 10.1.1.2 source g 1/0/1


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

Packet sent with a source address of 10.1.1.1                        <--- Ping source is permitted and p



!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms <-- 100% ping success




### Ping originated from neighbor device with source 10.1.1.3 ###

C9300#

ping 10.1.1.2 source g 1/0/1


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

Packet sent with a source address of 10.1.1.3                        <-- Ping source is denied (implicit


.....

Success rate is 0 percent (0/5)                                      <-- 0% ping success


### Confirm PACL drop ###

9500H#

show access-lists TEST


Extended IP access list TEST
```

```
    10 permit ip host 10.1.1.1 any                                <-- Counters in this command do not
    20 permit udp host 10.1.1.1 eq 1000 host 10.1.1.2


9500H#

show platform software fed active acl counters hardware | i PACL Drop
Ingress IPv4 PACL Drop            (0x77000005):         11 frames    <-- Hardware level command displays

Ingress IPv6 PACL Drop            (0x12000012):          0 frames


<...snip...>
```

Scenario 2. PACL (MAC ACL)

PACLs are assigned to a Layer 2 interface.

- Security Boundary: Ports or VLANs
- Attachment: Layer 2 Interface
- Direction: Ingress or Egress (one at a time)
- Supported ACL Types: MAC ACL & IP ACLs (standard or extended)

Configure PACL with MAC ACL

<#root>

```
9500H#

show run | sec mac access-list


mac access-list extended

MAC-TEST           <-- MAC ACL named MAC-TEST




permit host 0001.aaaa.aaaa any             <-- permit host MAC to any dest MAC



9500H#

show access-lists MAC-TEST


Extended MAC access list MAC-TEST
    permit host 0001.aaaa.aaaa any

9500H#

show running-config interface twentyFiveGigE 1/0/1


Building configuration...

interface TwentyFiveGigE1/0/1
```

```
switchport access vlan 10
switchport mode access

mac access-group MAC-TEST in               <-- Applied MACL to layer 2 interface
```

Verify PACL

Retrieve the IF_ID associated with the interface.

<#root>

```
9500H#

show platform software fed active ifm interfaces ethernet


Interface

 IF_ID

               State
----------------------------------------------------------------

TwentyFiveGigE1/0/1



0x00000008

          READY

<-- IF_ID value for Tw1/0/1
```

Verify the Class group ID (CG ID) bound to the IF_ID.

<#root>

```
9500H#

show platform software fed active acl interface 0x8       <-- IF_ID with leading zeros omitted


########################################################
########                              ##################
#######   Printing Interface Infos     #################
########                              ##################
########################################################


INTERFACE: TwentyFiveGigE1/0/1                                <-- Confirms the interface matches the IF


MAC 0000.0000.0000
########################################################
    intfinfo: 0x7f489404e408
    Interface handle: 0x7e000028


Interface Type: Port                                        <-- Type: Port indicates Layer 2 interface
```

```
if-id: 0x0000000000000008                                  <-- IF_ID 0x8 is correct

    Input MAC: Policy Handle: 0xde000098

 Policy Name: MAC-TEST                                    <-- The named ACL bound to this interface



CG ID: 20                                           <-- Class Group ID for this entry



CGM Feature: [0] acl                                      <-- Feature is ACL

        Bind Order: 0
```

ACL information associated with the CG ID.

<#root>

```
9500H#

show platform software fed active acl info acl-cgid 20  <-- The CG ID associated to the ACl MAC-TEST

########################################################
#########                             ##################
########      Printing CG Entries      #################
#########                             ##################
########################################################
===================================

ACL CG (acl/20): MAC-TEST type: MAC                          <-- feature ACL/CG ID 20: ACL name MAC-TES

Total Ref count 1
---------------------------------

1 Interface                                                  <-- Applied to one interface


---------------------------------
  region reg_id: 3
    subregion subr_id: 0
       GCE#:1 #flds: 2 l4:N matchall:N deny:N
         Result: 0x01010000

 mac_dest: value = 0x00, mask = 0x00                  <-- Mac dest: hex 0x00 mask 0x00 is "any destinati



mac_src: value = 0x1aaaaaaaa

,
mask = 0xffffffffffff


<-- Mac source: 0x1aaaaaaaa | hex with leading zeros omitted (0001.aaaa.aaaa) & mask 0xffffffffffff is h
```

Policy information on the CG ID, as well as what interfaces use the CG ID.

<#root>

```
9500H#

show platform software fed active acl policy 20      <-- Use the CG ID value


########################################################
#########                             ##################
########     Printing Policy Infos     #################
#########                             ##################
########################################################


INTERFACE: TwentyFiveGigE1/0/1                          <-- Interface with ACL applied


MAC 0000.0000.0000
########################################################
    intfinfo: 0x7f8cfc02de98
    Interface handle: 0x7e000028
    Interface Type: Port


if-id: 0x0000000000000008                           <-- The Interface IF_ID 0x8


------------

Direction:  Input                                       <-- ACl is applied in the ingress direction



Protocol Type:MAC                                       <-- Type is MAC


    Policy Intface Handle: 0x30000c6
    Policy Handle: 0xde000098

########################################################
#########                             ##################
########         Policy information    #################
#########                             ##################
########################################################
Policy handle         : 0xde000098

Policy name           : MAC-TEST                        <-- ACL name is MAC-TEST



ID                    : 20                              <-- CG ID for this ACL entry


Protocol              : [1] MAC

Feature               : [1] AAL_FEATURE_PACL            <-- ASIC Feature is PACL
```

```
Number of ACLs        : 1

########################################################
## Complete policy ACL information
########################################################
Acl number : 1
=====================================
Acl handle : 0xd60000dc
Acl flags : 0x00000001

Number of ACEs : 2                                        <-- 2 ACEs: one permit, and one implicit deny


  Ace handle [1] : 0x38000120
  Ace handle [2] : 0x31000121


Interface(s):



        TwentyFiveGigE1/0/1                               <-- Interface the ACL is applied


########################################################
#########                             ##################
########  Policy instance information  #################
#########                             ##################
########################################################
Policy intf handle    : 0x030000c6
Policy handle         : 0xde000098
ID                    : 20
Protocol              : [1] MAC
Feature               : [1] AAL_FEATURE_PACL
Direction             : [1] Ingress
Number of ACLs        : 1
Number of VMRs        : 3------------
```

Confirm PACL is working:

- The MACL only permits source address 0001.aaaa.aaaa.
- Since this is a MAC ACL, a non-IP ARP packet is dropped and thereby causing the ping to fail.

<#root>

```
### Ping originated from neighbor device with Source MAC 0000.0000.0002 ###

C9300#

ping 10.1.1.2 source vlan 10

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
.....

Success rate is 0 percent (0/5)
```

```
C9300#

show ip arp

Protocol  Address          Age (min)  Hardware Addr    Type    Interface
Internet  10.1.1.2               0

Incomplete

       ARPA

<-- ARP is unable to complete on Source device

### Monitor capture configured on Tw 1/0/1 ingress ###

9500H#

monitor capture 1 interface TwentyFiveGigE 1/0/1 in match any


9500H#

show monitor cap

Status Information for Capture 1
  Target Type:


 Interface: TwentyFiveGigE1/0/1, Direction: IN


9500H#sh monitor capture 1 buffer brief | inc ARP

    5   4.767385 00:00:00:00:00:02 b^F^R

ff:ff:ff:ff:ff:ff ARP 60 Who has 10.1.1.2? Tell 10.1.1.1


    8   8.767085 00:00:00:00:00:02 b^F^R ff:ff:ff:ff:ff:ff ARP 60 Who has 10.1.1.2? Tell 10.1.1.1
   11  10.767452 00:00:00:00:00:02 b^F^R ff:ff:ff:ff:ff:ff ARP 60 Who has 10.1.1.2? Tell 10.1.1.1
   13  12.768125 00:00:00:00:00:02 b^F^R ff:ff:ff:ff:ff:ff ARP 60 Who has 10.1.1.2? Tell 10.1.1.1


<-- 9300 (10.1.1.1) sends ARP request, but since there is no reply 4 more ARP requests are sent



9500H#

show platform software fed active acl counters hardware | inc MAC PACL Drop
Ingress MAC PACL Drop                       (0x73000021): 937 frames        <-- Confirmed that ARP requ

Egress MAC PACL Drop                        (0x0200004c): 0 frames

<...snip...>
```

Scenario 3. RACL

RACL is assigned to a Layer 3 interface such as an SVI or Routed interface.

- Security Boundary: Different Subnets
- Attachment: Layer 3 Interface

- Direction: Ingress or Egress
- Supported ACL Types: IP ACLs (standard or extended)

Configure RACL

<#root>

```
9500H(config)#

ip access-list extended TEST          <-- Create a named extended ACL


9500H(config-ext-nacl)#

permit ip host 10.1.1.1 any

9500H(config-ext-nacl)#

permit udp host 10.1.1.1 eq 1000 host 10.1.1.2



9500H#

show access-lists TEST                        <-- Display the ACL configured


Extended IP access list TEST
    10 permit ip host 10.1.1.1 any
    20 permit udp host 10.1.1.1 eq 1000 host 10.1.1.2

9500H(config)#

interface Vlan 10                     <-- Apply ACL to Layer 3 SVI interface


9500H(config-if)#

ip access-group TEST in



9500H#

show running-config interface Vlan 10


Building configuration...

Current configuration : 84 bytes
!
interface Vlan10

 ip access-group TEST in                            <-- Display the ACL applied to the interface


end
```

Verify RACL

Retrieve the IF_ID associated with the interface.

<#root>

```
9500H#

show platform software fed active ifm mappings l3if-le  <-- Retrieve the IF_ID for a Layer 3 SVI type po

Mappings Table

L3IF_LE                    Interface                                    IF_ID            Type
----------------------------------------------------------------------------------------------------
0x00007f8d04983958

Vlan10



0x00000026

        SVI_L3_LE

<-- IF_ID value for SVI 10
```

Verify the Class group ID (CG ID) bound to the IF_ID.

<#root>

```
9500H#

show platform software fed active acl interface 0x26  <-- IF_ID for SVI Vlan 10 with leading zeros omitt

########################################################
########                              #################
#######     Printing Interface Infos   ################
########                              #################
########################################################


INTERFACE: Vlan10                                           <-- Confirms the interface matches the IF_ID


MAC 0000.0000.0000
########################################################
    intfinfo: 0x7f8cfc02de98
    Interface handle: 0x6e000047


Interface Type: L3                                      <-- Type: L3 indicates Layer 3 type interface




if-id: 0x0000000000000026                               <-- IF_ID 0x26 is correct


    Input IPv4: Policy Handle: 0x2e000095


Policy Name: TEST                                   <-- The named ACL bound to this interface




  CG ID: 9                                       <-- Class Group ID for this entry
```

```
CGM Feature: [0] acl                                    <-- Feature is ACL
```

```
        Bind Order: 0
```

ACL information associated with the CG ID.

<#root>

```
9500H#
```

```
show platform software fed active acl info acl-cgid 9  <-- The CG ID associated to the ACL TEST
```

```
########################################################
#########                            ##################
########      Printing CG Entries      #################
#########                            ##################
########################################################
===================================
```

```
ACL CG (acl/9): TEST type: IPv4
```

```
<-- feature ACL/CG ID 9: ACl name TEST : ACl type IPv4
```

```
Total Ref count 2
---------------------------------
```

```
2 Interface                           <-- Interface count is 2. Applied to SVI 10 and as PACL to Tw1/0/
```

```
---------------------------------
region reg_id: 10
  subregion subr_id: 0
    GCE#:1
```

```
#flds: 2
```

```
l4:N
```

```
 matchall:N deny:N
```

```
<-- #flds: 2 = two fields in entry | l4:N (no Layer 4 port match)
```

```
      Result: 0x01010000
```

```
      ipv4_src: value
```

```
 =
```

```
0x0a010101
```

```
,
```

```
mask = 0xffffffff
```

```
<-- src 0x0a010101 hex = 10.1.1.1 | mask 0xffffffff = exact host match

      ipv4_dst: value
 =
0x00000000, mask = 0x00000000

<--

dst & mask = 0x00000000 = match any
    GCE#:1 #flds: 4
l4:Y
 matchall:N deny:N
<-- #flds: 4 = four fields in entry | l4:Y (ACE uses UDP port L4 match)

      Result: 0x01010000
      ipv4_src: value = 0x0a010101, mask = 0xffffffff <-- Exact match (host) 10.1.1.1

      ipv4_dst: value = 0x0a010102, mask = 0xffffffff <-- Exact match (host) 10.1.1.2

      ip_prot: start = 17, end = 17                   <-- protocol 17 is UDP

      l4_src: start = 1000, end = 1000                <-- matches eq 1000 (equal UDP port 1000)
```

Policy information on the CG ID, as well as what interfaces use the CG ID.

<#root>

```
9500H#
show platform software fed active acl policy 9      <-- Use the CG ID Value
########################################################
#########                             #################
########     Printing Policy Infos     ################
#########                             #################
########################################################

INTERFACE: Vlan10                                          <-- Interface with ACL applied

MAC 0000.0000.0000
```

```
########################################################
    intfinfo: 0x7f8cfc02de98
    Interface handle: 0x6e000047
    Interface Type: L3

    if-id: 0x0000000000000026                             <-- Interface IF_ID 0x26


------------

Direction:  Input                                         <-- ACL applied in the ingress direction



Protocol Type:IPv4                                        <-- Type is IPv4



    Policy Intface Handle: 0x1c0000c2
    Policy Handle: 0x2e000095



########################################################
#########                             ##################
########        Policy information     #################
#########                             ##################
########################################################
Policy handle        : 0x2e000095

Policy name          : TEST                               <-- ACL name TEST



ID                   : 9



<-- CG ID for this ACL entry



Protocol             : [3] IPV4

Feature              : [27] AAL_FEATURE_RACL              <-- ASIC feature is RACL


Number of ACLs       : 1

########################################################
## Complete policy ACL information
########################################################
Acl number       : 1
=====================================
Acl handle         : 0x7c0000d4
Acl flags          : 0x00000001

Number of ACEs     : 5                                    <-- 5 Aces: 2 explicit, 1 implicit deny, 2 ???


  Ace handle [1] : 0x0600010f
  Ace handle [2] : 0x8e000110
  Ace handle [3] : 0x3b000111
  Ace handle [4] : 0xeb000112
  Ace handle [5] : 0x79000113


Interface(s):
        Vlan10                                          <-- The interface the ACL is applied

########################################################
#########                             ##################
########  Policy instance information  #################
#########                             ##################
########################################################
Policy intf handle     : 0x1c0000c2
Policy handle          : 0x2e000095
ID                     : 9
Protocol               : [3] IPV4
Feature                : [27] AAL_FEATURE_RACL
Direction              : [1] Ingress
Number of ACLs         : 1
Number of VMRs         : 4------------
```

Confirm RACL is working.

> **Note**: When you enter the show ip access-lists privileged EXEC command, the match count displayed does not account for packets that are access controlled in hardware. Use the show platform software fed switch{*switch_num*|active|standby}acl counters hardwareprivileged EXEC command in order to obtain some basic hardware ACL statistics for switched and routed packets.

<#root>

```
### Ping originated from neighbor device with source 10.1.1.1 ###


C9300#

ping 10.1.1.2 source g 1/0/1


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

Packet sent with a source address of 10.1.1.1                        <--- Ping source is permitted and p



!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms <-- 100% ping success




### Ping originated from neighbor device with source 10.1.1.3 ###

C9300#

ping 10.1.1.2 source g 1/0/1


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

Packet sent with a source address of 10.1.1.3                        <-- Ping source is denied (implici


.....
```

```
Success rate is 0 percent (0/5)                                     <-- 0% ping success


### Confirm RACL drop ###

9500H#

show access-lists TEST


Extended IP access list TEST


    10 permit ip host 10.1.1.1 any                                  <-- Counters in this command do not
    20 permit udp host 10.1.1.1 eq 1000 host 10.1.1.2


9500H#

show platform software fed active acl counters hardware | i RACL Drop
Ingress IPv4 RACL Drop              (0xed000007):         100 frames  <-- Hardware level command display


<...snip...>
```

Scenario 4. VACL

VACLs are assigned to a Layer 2 VLAN.

- Security Boundary: Within OR across a VLAN
- Attachment: VLAN/VLAN Map
- Direction: Both Ingress and Egress at once
- Supported ACL Types: MAC ACL & IP ACLs (standard or extended)

Configure VACL

<#root>

```
ip access-list extended TEST


 10 permit ip host 10.1.1.1 any
 20 permit ip any host 10.1.1.1


ip access-list extended ELSE


 10 permit ip any any


vlan access-map VACL 10


 match ip address TEST
 action forward

vlan access-map VACL 20
```

```
 match ip address ELSE
 action drop

vlan filter VACL vlan-list 10
```

```
9500H#

sh vlan access-map VACL

Vlan access-map "VACL" 10
  Match clauses:
    ip address: TEST

 Action:

    forward

Vlan access-map "VACL" 20
  Match clauses:
    ip address: ELSE

 Action:

    drop

9500H#

sh vlan filter access-map VACL

VLAN Map VACL is filtering VLANs:

10
```

Verify VACL

Retrieve the IF_ID associated with the interface.

<#root>

```
9500H#

show platform software fed active ifm interfaces vlan

Interface

IF_ID

               State
----------------------------------------------------------------------

Vlan10                            0x00420010
```

```
        READY
```

Verify the Class group ID (CG ID) bound to the IF_ID.

<#root>

```
9500H#

show platform software fed active acl interface 0x420010  <-- IF_ID for the Vlan


########################################################
########                              ##################
#######   Printing Interface Infos     #################
########                              ##################
########################################################


INTERFACE: Vlan10                                                <-- Can be L2 only, with no vlan interfa


MAC 0000.0000.0000
########################################################
    intfinfo: 0x7fc8cc7c7f48
    Interface handle: 0xf1000024
    Interface Type: Vlan
    if-id: 0x0000000000420010


Input IPv4:

 Policy Handle: 0xd10000a3

<-- VACL has both Ingress and Egress actions



Policy Name: VACL                                        <-- Name of the VACL used



CG ID: 530                                          <-- Class Group ID for entry



 CGM Feature: [35] acl-grp                                <-- Feature is ACL group, versus ACl


        Bind Order: 0


Output IPv4:

 Policy Handle: 0xc80000a4

<-- VACL has both Ingress and Egress actions


         Policy Name: VACL
              CG ID: 530
         CGM Feature: [35] acl-grp
    Bind Order: 0
```

ACL information associated with the CG Group ID.

There are two ACLs used in the same named VACL policy, grouped into this acl-group

<#root>

```
9500H#

show platform software fed active acl info acl-grp-cgid 530  <-- use the group-id command versus gc ID


########################################################
#########                             ##################
########    Printing CG Entries        #################
#########                             ##################
########################################################
===================================

ACL CG (acl-grp/530): VACL type: IPv4                              <-- feature acl/group ID 530: name VA


Total Ref count 2
---------------------------------

2 VACL                                                             <-- Ingress and egress ACL direction


---------------------------------
  region reg_id: 12
    subregion subr_id: 0
      GCE#:10 #flds: 2 l4:N matchall:N deny:N
        Result: 0x06000000


ipv4_src: value = 0x0a010101, mask = 0xffffffff           <-- permit from host 10.1.1.1 (see PACL exampl




ipv4_dst: value = 0x00000000, mask = 0x00000000           <-- to any other host


      GCE#:20 #flds: 2 l4:N matchall:N deny:N
        Result: 0x06000000


ipv4_src: value = 0x00000000, mask = 0x00000000           <-- permit from any host




ipv4_dst: value = 0x0a010101, mask = 0xffffffff           <-- to host 10.1.1.1


      GCE#:10 #flds: 2 l4:N matchall:N deny:N
        Result: 0x05000000


 ipv4_src: value = 0x00000000, mask = 0x00000000           <-- This is the ACL named 'ELSE' which is per
```

```
        ipv4_dst: value = 0x00000000, mask = 0x00000000          <-- with VACL, the logic used was "pe
```

Policy information on the CG ID, as well as what interfaces use the CG ID.

```
<#root>

9500H#

show platform software fed active acl policy 530     <-- use the acl-grp ID


########################################################
#########                             ##################
########     Printing Policy Infos     #################
#########                             ##################
########################################################

INTERFACE: Vlan10
MAC 0000.0000.0000
########################################################
    intfinfo: 0x7fa15802a5d8
    Interface handle: 0xf1000024


Interface Type: Vlan                                  <-- Interface type is the Vlan, not a specific i




if-id: 0x0000000000420010                             <-- the Vlan IF_ID matches Vlan 10


------------

Direction: Input                                          <-- VACL in the input direction


Protocol Type:IPv4
    Policy Intface Handle: 0x44000001
    Policy Handle: 0x29000090

########################################################
#########                             ##################
########       Policy information      #################
#########                             ##################
########################################################
Policy handle       : 0x29000090

Policy name         : VACL                                <-- the VACL policy is named 'VACL'


ID                  : 530
Protocol            : [3] IPV4

Feature             : [23] AAL_FEATURE_VACL               <-- ASIC feature is VACL



Number of ACLs      : 2                                   <-- 2 ACL used in the VACL: "TEST & ELSE"
```

```
########################################################
## Complete policy ACL information
########################################################
Acl number : 1
=====================================
Acl handle : 0xa6000090
Acl flags : 0x00000001
Number of ACEs : 4
  Ace handle [1] : 0x87000107
  Ace handle [2] : 0x30000108
  Ace handle [3] : 0x73000109
  Ace handle [4] : 0xb700010a

Acl number : 2
=====================================
Acl handle : 0x0f000091
Acl flags : 0x00000001
Number of ACEs : 1
  Ace handle [1] : 0x5800010b

Interface(s):
        Vlan10
########################################################
#########                             ##################
########  Policy instance information  #################
#########                             ##################
########################################################
Policy intf handle    : 0x44000001
Policy handle         : 0x29000090

ID                    : 530                             <-- 530 is the acl group ID


Protocol              : [3] IPV4
Feature               : [23] AAL_FEATURE_VACL

Direction             : [1] Ingress                     <-- Ingress VACL direction


Number of ACLs        : 2
Number of VMRs        : 4------------
Direction:  Output
Protocol Type:IPv4
    Policy Intface Handle: 0xac000002
    Policy Handle: 0x31000091

########################################################
#########                             ##################
########        Policy information     #################
#########                             ##################
########################################################
Policy handle        : 0x31000091
Policy name          : VACL
ID                   : 530
Protocol             : [3] IPV4
Feature              : [23] AAL_FEATURE_VACL
Number of ACLs       : 2

########################################################
## Complete policy ACL information
########################################################
Acl number    : 1
```

```
=====================================
Acl handle      : 0xe0000092
Acl flags       : 0x00000001
Number of ACEs  : 4
  Ace handle [1] : 0xf500010c
  Ace handle [2] : 0xd800010d
  Ace handle [3] : 0x4c00010e
  Ace handle [4] : 0x0600010f

Acl number    : 2
=====================================
Acl handle      : 0x14000093
Acl flags       : 0x00000001
Number of ACEs : 1
  Ace handle [1] : 0x8e000110

Interface(s):
        Vlan10
########################################################
#########                             ##################
########  Policy instance information  #################
#########                             ##################
########################################################
Policy intf handle    : 0xac000002
Policy handle         : 0x31000091
ID                    : 530                            <-- 530 is the acl group ID


Protocol              : [3] IPV4
Feature               : [23] AAL_FEATURE_VACL
Direction             : [2] Egress                     <-- Egress VACL direction


Number of ACLs        : 2
Number of VMRs        : 4------------
```

Confirm VACL is working.

- Troubleshoot is the same scenario as PACL and RACl sections. Refer to these sections for details on the ping test.
- Ping from 10.1.1.3 to 10.1.1.2 denied by the ACL policy applied.
- Check the platform drop command.

<#root>

```
9500H#

show platform software fed active acl counters hardware | inc VACL Drop



Ingress IPv4 VACL Drop

              (0x23000006):

1011 frames      <-- Hardware level command displays drops against VACL



<...snip...>
```

Scenario 5. Group/Client ACL (DACL)

Group/Client ACLs are applied dynamically to a user group or client based on their identity. These are also sometimes called DACL.

- Security Boundary: Client (Client interface level)
- Attachment: Per client interface
- Direction: Ingress only
- Supported ACL Types: MAC ACL and IP ACLs (standard or extended)

Configure GACL

<#root>

```
Cat9400#

show run interface gigabitEthernet 2/0/1

Building configuration...

Current configuration : 419 bytes
!
interface GigabitEthernet2/0/1
 switchport access vlan 10
 switchport mode access
 switchport voice vlan 5


ip access-group ACL-ALLOW in                    <-- This is the pre-authenticated ACL (deny ip any any)

 authentication periodic
 authentication timer reauthenticate server
 access-session control-direction in
 access-session port-control auto
 no snmp trap link-status
 mab
 dot1x pae authenticator
 spanning-tree portfast


service-policy type control subscriber ISE_Gi2/0/1

end

Cat9400#

show access-session interface gigabitEthernet 2/0/1 details

            Interface:  GigabitEthernet2/0/1


 IIF-ID:  0x1765EB2C                <-- The IF_ID used in this example is dynamic



 MAC Address:  000a.aaaa.aaaa            <-- The client MAC

         IPv6 Address:  Unknown
         IPv4 Address:  10.10.10.10
            User-Name:  00-0A-AA-AA-AA-AA
```

```
Status:  Authorized                <-- Authorized client

              Domain:  VOICE
      Oper host mode:  multi-auth
    Oper control dir:  in
     Session timeout:  300s (server), Remaining: 182s
      Timeout action:  Reauthenticate
   Common Session ID:  27B17A0A000003F499620261
     Acct Session ID:  0x000003e7
              Handle:  0x590003ea
      Current Policy:  ISE_Gi2/0/1


Server Policies:

             ACS ACL:

xACSACLx-IP-MAB-FULL-ACCESS-59fb6e5e

  <-- The ACL pushed from ISE server



Method status list:
       Method           State
        dot1x           Stopped


 mab           Authc Success                                    <-- Authenticated via MAB (Mac authent


Cat9400#

show ip access-lists xACSACLx-IP-MAB-FULL-ACCESS-59fb6e5e

Extended IP access list xACSACLx-IP-MAB-FULL-ACCESS-GOOD-59fb6e5e


 1 permit ip any any                                                  <-- ISE pushed a permit ip any an
```

Verify GACL

Group CG ID bound to the iif-id.

<#root>

```
Cat9400#

show platform software fed active acl interface 0x1765EB2C                 <-- The IF_ID from the access

########################################################
########                              ##################
#######    Printing Interface Infos    #################
########                              ##################
########################################################

INTERFACE: Client MAC

000a.aaaa.aaaa
```

```
                                                <-- Client MAC matches the access-session output

MAC

000a.aaaa.aaaa

########################################################
    intfinfo: 0x7f104820cae8
    Interface handle: 0x5a000110


Interface Type: Group                                                            <-- This is a group ident



IIF ID: 0x1765eb2c

    Input IPv4: Policy Handle: 0x9d00011e


  Policy Name: ACL-ALLOW:xACSACLx-IP-MAB-FULL-ACCESS-59fb6e5e

:

<-- DACL name matches




CG ID: 127760                                                           <-- The ACL group ID

       CGM Feature: [35]

acl-grp

        Bind Order: 0
```

ACL information associated with the group GC ID.

<#root>

```
Cat9400#

show platform software fed active acl info acl-grp-cgid 127760                          <-- the CG ID

########################################################
#########                             ##################
########      Printing CG Entries      #################
#########                             ##################
########################################################
===================================
ACL CG (

acl-grp/127760

):

ACL-ALLOW:xACSACLx-IP-MAB-FULL-ACCESS-59fb6e5e

: type: IPv4

<-- Group ID & ACL name are correct


Total Ref count 1
```

```
---------------------------------

1 CGACL                                                                                                   <-- 1

---------------------------------
  region reg_id: 1
    subregion subr_id: 0
      GCE#:1 #flds: 2 l4:N matchall:N deny:N
        Result: 0x04000000


 ipv4_src: value = 0x00000000, mask = 0x00000000                                                  <-- Permits I
        ipv4_dst: value = 0x00000000, mask = 0x00000000

      GCE#:10 #flds: 2 l4:N matchall:N deny:N
        Result: 0x04000000
        ipv4_src: value = 0x00000000, mask = 0x00000000
        ipv4_dst: value = 0x00000000, mask = 0x00000000
```

Scenario 6. ACL Logging

The device software can provide syslog messages about packets permitted or denied by a standard IP access list. Any packet that matches the ACL causes an informational log message about the packet to be sent to the console. The level of messages logged to the console is controlled by thelogging consolecommands controlling the Syslog messages.

- ACL log messages are not supported for ACLs used with Unicast Reverse Path Forwarding (uRPF). It is only supported for RACL.
- ACL log in the egress direction is not supported for packets that are generated from the control plane of the device.
- Routing is done in hardware and logging in software, so if a large number of packets match a permit or deny ACE containing a logkeyword, the software is unable to match the hardware processing rate, and not all packets can be logged.
- The first packet that triggers the ACL causes a log message right away, and subsequent packets are collected over 5-minute intervals before they appear or are logged. The log message includes the access list number, whether the packet was permitted or denied, the source IP address of the packet, and the number of packets from that source permitted or denied in the prior 5-minute interval.
- See the appropriate Security Configuration Guide, Cisco IOS XE as noted in the Related Information section for complete details on ACL log behavior and restrictions.

Log Example PACL:

This example shows a negative case, where the ACL type and log keyword do not work together.

<#root>

```
9500H#

show access-lists TEST


Extended IP access list TEST
    10 permit ip host 10.1.1.1 any

log             <-- Log keyword applied to ACE entry
```

```
    20 deny ip host 10.1.1.3 any
```

log

```
9500H(config)#
```

interface twentyFiveGigE 1/0/1

```
9500H(config-if)#
```

```
ip access-group TEST in           <-- apply logged ACL
Switch Port ACLs are not supported for LOG!        <-- message indicates this is an unsupported combinat
```

Log Example RACL (Deny):

<#root>

```
9500H#
```

show access-lists TEST

```
Extended IP access list TEST
    10 permit ip host 10.1.1.1 any
```

```
log            <-- Log keyword applied to ACE entry
```

```
    20 deny ip host 10.1.1.3 any
```

log

```
9500H(config)#
```

interface vlan 10

```
9500H(config-if)#
```

```
ip access-group TEST in          <-- ACL applied to SVI
```

### Orginate ICMP from 10.1.1.3 to 10.1.1.2 (denied by ACE) ###

```
C9300#
```

ping 10.1.1.2 source vlan 10 repeat 110

```
Type escape sequence to abort.


Sending 10, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.3
..........
```

Success rate is 0 percent (0/110)

```
9500H#
```

```
show access-list TEST


Extended IP access list TEST
    10 permit ip host 10.1.1.1 any log


 20 deny ip host 10.1.1.3 any log (110 matches)  <-- Matches increment in show access-list command


9500H#

show platform software fed active acl counters hardware | inc RACL

Ingress IPv4 RACL Drop          (0xed000007):            0 frames

Ingress IPv4 RACL Drop and Log  (0x93000009):          110 frames   <-- Aggregate command shows hits on

%SEC-6-IPACCESSLOGDP: list TEST denied icmp 10.1.1.3 -> 10.1.1.2 (8/0), 10 packets  <-- Syslog message i
```

Log Example RACL (Permit):
When a log statement is used for a permit statement, the software counter hits show double the number of packets sent.

<#root>

```
C9300#

ping 10.1.1.2 source vlan 10 repeat 5          <-- 5 ICMP Requests are sent


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!

Success rate is 100 percent (5/5)

, round-trip min/avg/max = 1/1/1 ms

9500H#

show access-lists TEST


Extended IP access list TEST


 10 permit ip host 10.1.1.1 any log (10 matches)  <-- Hit counter shows 10


    20 deny ip host 10.1.1.3 any log (115 matches)
```

Troubleshoot

ACL Statistics

When you troubleshoot an ACL issue, it is essential to understand how and where ACL statistics are measured by the device.

- ACL Statistics are collected at an aggregate level, and not per ACE level.
- Hardware does not have the capability to allow per ACE or per ACL stats.
- Statistics such as Deny, Log, and CPU forwarded packets are collected.
- Statistics for MAC, IPv4, and IPv6 packets are collected separately.
- show platform software fed switch active acl counters hardware can be used in order to display aggregate statistics.

Clearing ACL Statistics

When troubleshooting an ACL issue, it can be helpful to clear the various ACL counters in order to get fresh baseline counts.

- These commands allow you to clear software and hardware ACL counter statistics.
- When you troubleshoot ACL match/hit events, it is recommended to clear the relevant ACL to baseline matches that are recent or relevant.

```
<#root>

clear platform software fed active acl counters hardware

 (clears the hardware matched counters)

clear ip access-list counters <acl_name>

 (clears the software matched counters - IPv4)

clear ipv6 access-list counters <acl_name>

 (clears the software matched counters - IPv6)
```

What happens when ACL TCAM is exhausted?

- ACLs are always applied in hardware TCAM. If TCAM is already used by previously configured ACLs, the new ACLs do not get the required ACL resources needed to program.

- If an ACL is added after TCAM is exhausted, all packets are dropped for the interface it is attached.

- The action of holding an ACL in software is called **Unloading**.
- When resources become available, the switch automatically tries to program the ACLs into the hardware. If successful, the ACLs are pushed to hardware and packets start to forward.
- The action of programming a software-held ACL into TCAM is called **Reloading**.

- PACL, VACL, RACL, and GACL can be unloaded/reloaded independently of each other.

ACL TCAM Exhaustion

- The interface to which the newly added ACL is applied starts dropping packets until hardware resources become available.
- GACL clients are put into the UnAuth state.

VCU Exhaustion

- Once over the L4OPs limit or out of VCUs, the software performs ACL expansion and creates new ACE entries in order to perform equivalent action without using VCUs.
- Once this happens TCAM can become exhausted from these added entries.

ACL Syslog Errors

If you run out of a particular Security ACL resource, SYSLOG messages are generated by the system (interface, VLAN, label, and so on, values can differ).

ACL Log message	Definition	Recovery Action
%ACL_ERRMSG-4-UNLOADED: Switch 1 fed: Input <ACL> on interface <interface> is not programmed in hardware and traffic is dropped.	ACL is Unloaded (held in software)	Investigate the TCAM scale. If beyond scale, redesign ACLs.
%ACL_ERRMSG-6-REMOVED: 1 fed: The unloaded configuration for Input <ACL> on interface <interface> has been removed for label <label>asic<number>.	Unloaded ACL configuration is removed from the interface	ACL is already removed, no action to take
%ACL_ERRMSG-6-RELOADED: 1 fed: Input <ACL> on interface <interface> has now been loaded into the hardware for label <label> on asic<number>.	ACL is now installed in Hardware	The issue with ACL is now in hardware resolved, no action to take
%ACL_ERRMSG-3-ERROR: 1 fed: Input <ACL> IP ACL <NAME> configuration is not applied on <interface> at bind order <number>.	Other types of ACL error (such as dot1x ACL install failure)	Confirm ACL configuration is supported, and TCAM is not beyond scale
%ACL_ERRMSG-6-GACL_INFO: Switch 1 R0/0: fed: Logging is not supported for GACL.	GACL has a log option configured	GACL does not support logs. Remove log statements from GACL.
%ACL_ERRMSG-6-PACL_INFO: Switch 1 R0/0: fed: Logging is not supported for PACL.	PACL has a log option configured	PACL does not support logs. Remove log statements from PACL.
%ACL_ERRMSG-3-ERROR: Switch 1 R0/0: fed: Input IPv4 Group ACL implicit_deny:<name>: configuration is not applied on Client MAC 0000.0000.0000.	(dot1x) ACL fails to apply on the target port	Confirm ACL configuration is supported, and TCAM is not beyond scale

Out of Resource Scenarios and Recovery Actions

Scenario 1. ACL Bind	**Recovery Action**
• ACL is created and applied to an interface or VLAN. • Bind fails due to 'out of resource' conditions, such as TCAM exhaustion. • No ACEs within the ACL can be programmed into TCAM. ACL remains in **UNLOADED** state. • In the **UNLOADED** state, all traffic (including control packets) drops on the interface until the issue is fixed.	Re-design the ACL in order to reduce the utilization of TCAM.
Scenario 2. ACL Edit	**Recovery Action**
• An ACL is created and applied to an interface, and more ACE entries are added to this ACL while applied to the interface(s). • If TCAM does not have resources the edit operation fails. • No ACEs within the ACL can be programmed into TCAM. ACL remains in **UNLOADED** state. • In the **UNLOADED** state all traffic (including control packets) drops on the interface until the issue is fixed. • The existing ACL entries also fail in the **UNLOADED** state until this is fixed.	Re-design the ACL in order to reduce the utilization of TCAM.
Scenario 3. ACL Re-bind	**Recovery Action**
• ACL Re-bind is the action of attaching an ACL to an interface, then attaching another ACL to the same interface without detaching the first ACL. • First ACL is created and attached successfully. • A larger ACL with a different name and the same protocol (IPv4/IPv6) is created and attached to the same interface. • The device detaches the first ACL successfully and attempts to attach the new ACL to this interface. • If TCAM does not have resources the re-bind operation fails. • No ACEs within the ACL can be programmed into TCAM. ACL remains in **UNLOADED** state. • In the **UNLOADED** state, all traffic (including control packets) drops on the interface until the issue is fixed.	Re-design the ACL in order to reduce the utilization of TCAM.
Scenario 4. Bind Empty (Null) ACL	**Recovery Action**

• An ACL that has no ACE entries is created and attached to an interface. • The system creates this ACL internally with a permit 'any ACE', and attaches it to the interface in hardware (all traffic is permitted in this state). • ACE entries are then added to the ACL with the same name or number. The system programs TCAM as each ACE is added. • If TCAM runs out of resources when adding ACE entries, ACL is moved to the **UNLOADED** state. • In the **UNLOADED** state, all traffic (including control packets) drops on the interface until the issue is fixed. • The existing ACL entries also fail in the **UNLOADED** state until this is fixed.	Re-design the ACL in order to reduce the utilization of TCAM.

Verify ACL Scale

This section covers commands in order to determine the ACL scale and TCAM utilization.

FMAN Access-list Summary:

Identify configured ACLs and total ACE count per ACL.

<#root>

```
9500H#

show platform software access-list f0 summary



Access-list

                         Index        Num Ref

 Num ACEs


--------------------------------------------------------------------------

TEST

                                 1            1                2

<-- ACL TEST contains 2 ACE entries


ELSE                                     2            1                1
DENY                                     3            0                1
```

ACL Usage:

<#root>

```
9500H#

show platform software fed active acl usage

########################################################
########                              ##################
#######      Printing Usage Infos      #################
########                              ##################
########################################################
#####

ACE Software VMR max:196608 used:283                   <-- Value/Mask/Result entry usage


########################################################
==================================================================================================

Feature Type



ACL Type



Dir



Name



 Entries Used


VACL                   IPV4            Ingress          VACL                    4

<-- Type of ACL Feature, type of ACL, Direction ACL applied, name of ACL, and number of TCAM entries con


==================================================================================================
Feature Type           ACL Type        Dir             Name                    Entries Used
RACL                   IPV4            Ingress         TEST                    5
```

TCAM Usage (17.x):

TCAM usage command has significant differences between 16.x and 17.x trains.

<#root>

```
9500H#

show platform hardware fed active fwd-asic resource tcam utilization

Codes: EM - Exact_Match,

I - Input

,

O - Output
```

```
, IO - Input & Output, NA - Not Applicable


CAM Utilization for ASIC [0]

 Table                  Subtype

Dir



Max

     Used

%Used

       V4        V6    MPLS   Other
 ------------------------------------------------------------------------------------------------------------

 Security ACL Ipv4

      TCAM

I



 7168

      16

 0.22%

     16        0       0        0
 Security ACL Non Ipv4   TCAM        I        5120       76    1.48%        0       36       0       40
 Security ACL Ipv4       TCAM

  O

        7168       18    0.25%       18        0       0        0
 Security ACL Non Ipv4   TCAM        O        8192       27    0.33%        0       22       0        5

<...snip...>

<-- Percentage used and other counters about ACL consumption
<-- Dir = ACL direction (Input/Output ACl)
```

TCAM Usage (16.x):

TCAM usage command has significant differences between 16.x and 17.x trains.

<#root>

```
C9300#

show platform hardware fed switch active fwd-asic resource tcam utilization

CAM Utilization for ASIC [0]
 Table                                              Max Values

Used Values
```

```
--------------------------------------------------------------------------------


 Security Access Control Entries                       5120

 126         <-- Total used of the Maximum
<...snip...>
```

Custom SDM Template (TCAM Reallocation)

Using Cisco IOS XE Bengaluru 17.4.1, you can configure a custom SDM template for ACL features using the sdm prefer custom aclcommand.

Details on how to configure and verify this feature are covered in System Management Configuration Guide, Cisco IOS XE Bengaluru 17.4.x (Catalyst 9500 Switches).

Some basic configuration and verification are noted in this section.

Verify the current SDM template:

```
<#root>

9500H#

show sdm prefer


Showing SDM Template Info


This is the Core template.                                                                  <-- Core SD


  Security Ingress IPv4 Access Control Entries*:         7168  (current) - 7168  (proposed)  <-- IPv4 AC


  Security Ingress Non-IPv4 Access Control Entries*:     5120  (current) - 5120  (proposed)
  Security Egress IPv4 Access Control Entries*:          7168  (current) - 7168  (proposed)
  Security Egress Non-IPv4 Access Control Entries*:      8192  (current) - 8192  (proposed)

  <...snip...>


9500H#

show sdm prefer custom user-input

Custom Template Feature Values are not modified

<-- No customization to SDM
```

Modify the current SDM template:

- 9500H(config)#**sdm prefer custom acl**
  9500H(config-sdm-acl)#**acl-ingress 26 priority 1 <-- apply new 26K value. (priority discussed in the configuration guide)**

```
9500H(config-sdm-acl)#acl-egress 20 priority 2
9500H(config-sdm-acl)#exit
```

Use show sdm prefer custom in order to see the proposed values and sdm prefer custom commit in order to apply 'view the changes' via this CLI.

- Verify changes to the SDM profile.
- 9500H#**show sdm prefer custom**

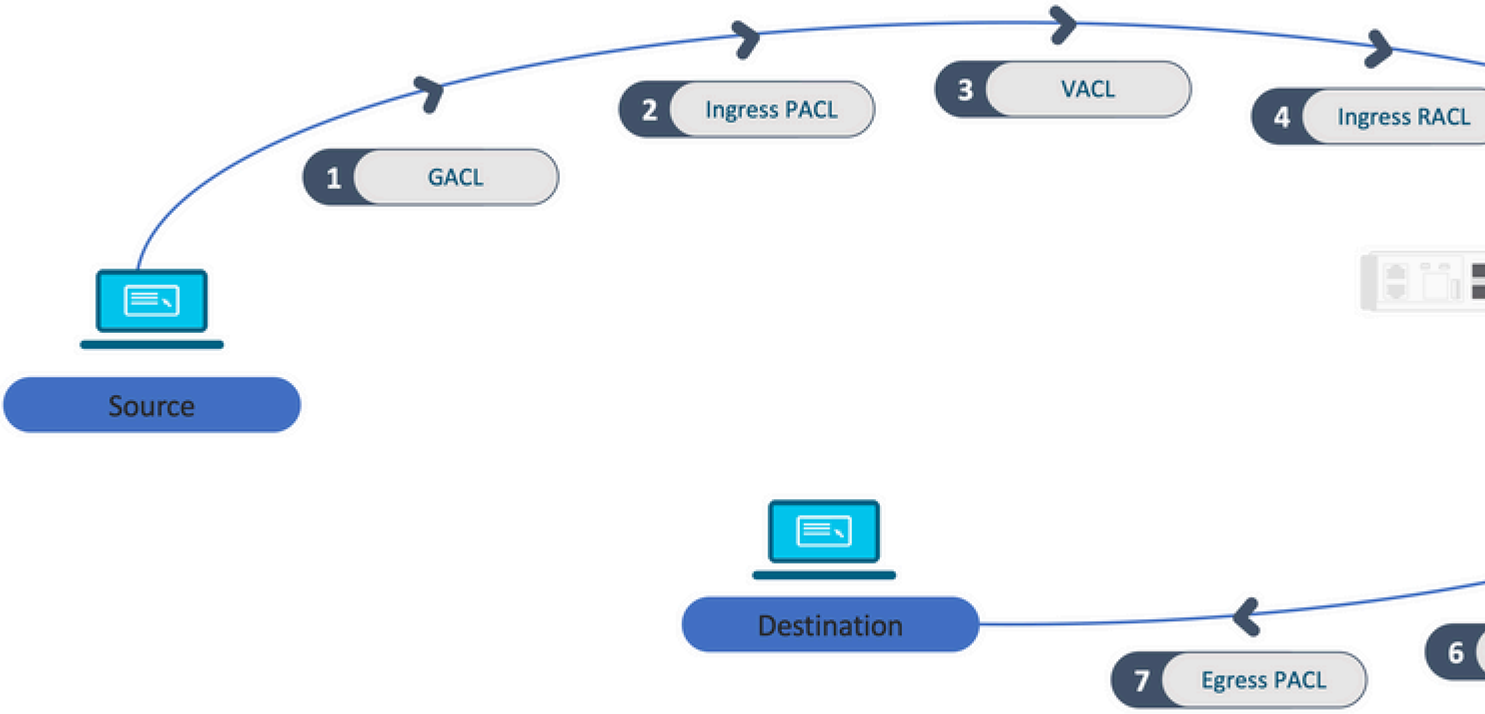
```
Showing SDM Template Info:

This is the custom template with its details.
Ingress Security Access Control Entries*: 12288 (current) - 26624 ( proposed) <-- Current and proposed usage (26K proposed)
Egress Security Access Control Entries*: 15360 (current) - 20480 (proposed)


9500H#show sdm prefer custom user-input

ACL FEATURE USER INPUT
User Input values
==========================

FEATURE NAME PRIORITY SCALE
------------------------------------------------------------------------
Ingress Security Access Control Entries: 1 26*1024 <-- Modifed by user input to 26 x 1024 (26K)
Egress Security Access Control Entries: 2 20*1024 <-- Modifed by user input to 20 x 1024 (20K)
```

- Apply changes to the SDM profile.
- 9500H(config)#**sdm prefer custom commit**
  Changes to the running SDM preferences are stored and take effect on the next reload. **<-- Once reloaded, ACL TCAM allocated to custom value.**

Further Reading:

ACL Processing Order:

ACLs are processed in this order from Source to Destination.



ACLs Programmed in a Stack:

- ACLs that are not port-based (for example, VACL, RACL) are applied to traffic on any switch and are programmed on all switches in the stack.
- Port-based ACLs are applied only to the traffic on a port and are programmed only on the switch that owns the interface.
- ACLs are programmed by the Active switch and subsequently applied to Member switches.
- The same rules apply to other redundancy options, such as ISSU/SVL.

ACL Expansion:

- ACL expansion happens when the device runs out of L4OPs, Lables, or VCUs. The device must create multiple equivalent ACEs in order to accomplish the same logic, and in order to rapidly exhaust TCAM.
- **### L4OPs are at scale and this ACL is created ##**

```
9500H(config)#ip access-list extended TEST
9500H(config-ext-nacl)#permit tcp 10.0.0.0 0.255.255.255 any gt 150 <-- matches ports 151 and higher


### This must be expanded into mutiple ACEs that do not use an L4OP ###
9500H(config-ext-nacl)#permit tcp 10.0.0.0 0.255.255.255 any eq 151
9500H(config-ext-nacl)#permit tcp 10.0.0.0 0.255.255.255 any eq 152
9500H(config-ext-nacl)#permit tcp 10.0.0.0 0.255.255.255 any eq 153
9500H(config-ext-nacl)#permit tcp 10.0.0.0 0.255.255.255 any eq 154
... and so on ....
```

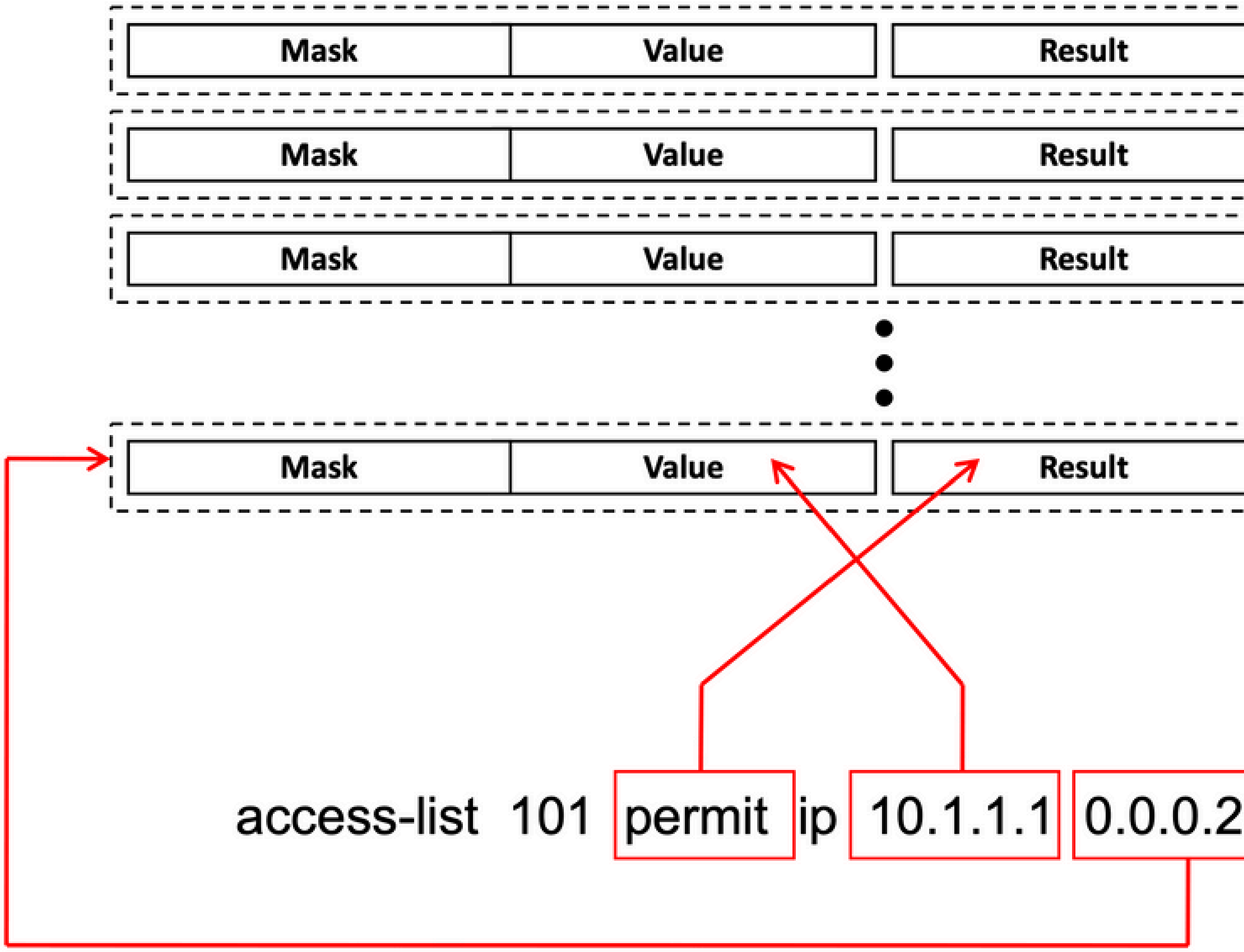
TCAM Consumption and Label Sharing:

- Each ACL policy is referenced internally by a label.

- When ACL policy (Security ACL like GACL, PACL, VACL, RACL) is applied to multiple interfaces or VLAN, it uses the same label.
- Ingress/Egress ACL uses different label spaces.
- IPv4, IPv6, and MAC ACL use other label spaces.
- The same PACL is applied to the ingress of interface-A and egress of interface-A. There are two instances of the PACL in the TCAM, each one with a unique label for Ingress and Egress.
- If the same PACL with an L4OP is applied to multiple ingress interfaces that exist on each core, there are two instances of the same PACL programmed in TCAM, one per each core.

VMR Description:

An ACE is internally programmed in TCAM as a 'VMR' â€“ also known as Value, Mask, Result. Each ACE entry can consume VMRs and can consume VCUs.



ACL Scalability:

Security ACL Resources are dedicated to Security ACLs. They are not shared with other features.

ACL TCAM resources	Cisco Catalyst 9600	Cisco Catalyst 9500	Cisco Catalyst 9400	Cisco Catalyst 9300	Cisco Catalyst 9200				
IPv4 entries	Ingress: 12000*	Egress: 15000*	C9500: 18000*	C9500 High Performance Ingress: 12000* Egress: 15000*	18000*	C9300: 5000	C9300B: 18000	C9300X:8000	1000
IPv6 entries	Half the IPv4 entries		Half the IPv4 entries		Half of the IPv4 entries	Half of the IPv4 entries			Half o IPv4 e
One type of IPv4 ACL Entries cannot Exceed	12000		C9500: 18000	C9500 High Performance: 15000	18000	C9300: 5000	C9300B: 18000	C9300X: 8000	1000
One type of IPv6 ACL Entries cannot Exceed	6000		C9500: 9000	C9500 High Performance: 7500	9000	2500/9000/4000			500
L4OPs/Label	8		8		8	8			8
Ingress VCUs	192		192		192	192			192
Egress VCUs	96		96		96	96			96

Related Information

- Security Configuration Guide, Cisco IOS XE Amsterdam 17.3.x (Catalyst 9200 Switches)
- Security Configuration Guide, Cisco IOS XE Amsterdam 17.3.x (Catalyst 9300 Switches)
- Security Configuration Guide, Cisco IOS XE Amsterdam 17.3.x (Catalyst 9400 Switches)
- Security Configuration Guide, Cisco IOS XE Amsterdam 17.3.x (Catalyst 9500 Switches)
- Security Configuration Guide, Cisco IOS XE Amsterdam 17.3.x (Catalyst 9600 Switches)
- System Management Configuration Guide, Cisco IOS XE Bengaluru 17.4.x (Catalyst 9500 Switches)
- Cisco Technical Support & Downloads

Debug and Trace Commands

Num	Command	Remark
1	show platform hardware fed [switch] active fwd-asic drops exceptions asic <0>	Dump the Exception counters on the ASIC #N.
2	show platform software fed [switch] active acl	This command prints the information about all the configured ACLs on the box along with interface and policy information.
3	show platform software fed [switch] active acl policy 18	This command prints the information about policy 18 only. You can get this policy ID from the command 2.
4	show platform software fed [switch] active acl interface intftype pacl	This command prints the information about the ACL based on interface type (pacl/vacl/racl/gacl/sgacl and so on).
5	show platform software fed [switch] active acl interface intftype pacl acltype ipv4	This command prints the information about the ACL based on interface type (pacl/vacl/racl/gacl/sgacl and so on) and also filters protocol-wise (ipv4/ipv6/mac and so on).
6	show platform software fed [switch] active acl interface intftype pacl acltype ipv4	This command prints the information about interfaces.
7	show platform software fed [switch] active acl interface 0x9	This command prints the short info of ACL applied on the interface, based on the IIF-ID (command from 6).
8	show platform software fed [switch] active acl definition	This command prints the information about the ACLs configured on the box and whose presence is in the CGD.
9	show platform software fed [switch] active acl iifid 0x9	This command prints the Detailed info of ACL applied on the interface, based on the IIF-ID.
10	show platform software fed [switch] active acl usage	This command prints the number of VMRs each ACL uses based on the Feature Type.
11	show platform software fed [switch] active acl policy intftype pacl vcu	This command gives you the policy information and also the VCU information based on the interface type (pacl/vacl/racl/gacl/sgacl and so on).
12	show platform software fed [switch] active acl policy intftype pacl cam	This command gives you the policy information and details about the VMRs in the CAM, based on the interface type

		(pacl/valc/racl/gacl/sgacl and so on).
13	show platform software interface [switch] [active] R0 brief	This command gives you details about the interface on the box.
14	show platform software fed [switch] active port if_id 9	This command prints the details about the port based on the IIF-ID.
15	show platform software fed [switch] active vlan 30	This command prints the details about the VLAN 30.
16	show platform software fed [switch] active acl cam asic 0	This command prints the complete ACL cam on ASIC 0 which is being used.
17	show platform software fed [switch] active acl counters hardware	This command prints all the ACL Counters from the hardware.
18	show platform hardware fed [switch] active fwd-asic resource tcam table pbr record 0 format 0	Printing the entries for the PBR section, you can give different sections like ACL and CPP instead of PBR.
19	show platform software fed [switch] active punt cpuq [1|2|3 â€¦]	In order to check the activity on one of the CPU Queues, you also have options to clear the queue stats for debugging.
20	show platform software fed [switch] active ifm mappings gpn	Print the interface mapping with the IIF-ID and GPNs
21	show platform software fed [switch active ifm if-id <iif-d>	Print the information about the interface configuration, and affinity with the ASIC. This command is helpful in order to check on which interface the ASIC and CORE are.
22	set platform software trace fed [switch] active acl/asic_vmr/asic_vcu/cgacl/sgacl [debug|error â€¦]	Setting the trace for a specific feature in FED.
23	request platform software trace rotate all	Clearing the trace buffer.
24	show platform software trace message fed [switch] active	Printing the trace buffer for FED.
25	set platform software trace forwarding-manager [switch] [active] f0 fman [debug|error â€¦]	Enabling the traces for FMAN.
26	show platform software trace message forwarding-	Printing the trace buffer for FMAN.

	manager [switch] [active] f0	
27	debug platform software infrastructure punt detail	Set the debugging on the PUNT.
28	debug ip cef packet all input rate 100	CEF packet debugging is on.