

验证Catalyst 9000交换机上的安全ACL

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[术语](#)

[ACL资源利用率示例](#)

[示例 1.IPv4 TCAM](#)

[示例 2.IPv4 TCAM/L4OP/VCU](#)

[示例 3.IPv6TCAM/L4OP/VCU](#)

[拓扑](#)

[配置和验证](#)

[场景 1.PACL\(IP ACL\)](#)

[使用IP ACL配置PACL](#)

[检验PACL](#)

[场景 2 : PACL\(MAC ACL\)](#)

[使用MAC ACL配置PACL](#)

[检验PACL](#)

[场景 3 : RAACL](#)

[配置RAACL](#)

[检验RAACL](#)

[场景 4.VACL](#)

[配置VACL](#)

[检验VACL](#)

[方案 5.组/客户端ACL\(DACL\)](#)

[配置GACL](#)

[检验GACL](#)

[方案 6.ACL 记录](#)

[故障排除](#)

[ACL统计信息](#)

[清除ACL统计信息](#)

[ACL TCAM用尽后会发生什么情况？](#)

[ACL TCAM耗尽](#)

[VCU耗尽](#)

[ACL系统日志错误](#)

[资源不足情形和恢复操作](#)

[检验ACL规模](#)

[自定义SDM模板 \(TCAM重新分配 \)](#)

[相关信息](#)

[Debug和Trace命令](#)

简介

本文档介绍如何验证Catalyst 9000系列交换机上的ACL (访问控制列表) 并对其进行故障排除。

先决条件

要求

本文档没有任何特定的要求。

使用的组件

本文档中的信息基于以下硬件版本：

- C9200
- C9300
- C9400
- C9500
- C9600

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始 (默认) 配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。



注意：请参阅相应的配置指南，了解在其他思科平台上启用这些功能所使用的命令。

背景信息

ACL过滤通过路由器或交换机的流量，并允许或拒绝通过指定接口的数据包。ACL是适用于数据包的允许和拒绝条件的顺序集合。当在接口上收到数据包时，交换机根据访问列表中指定的条件，将数据包中的字段与任何应用的ACL进行比较，以验证数据包是否具有转发所需的权限。它会逐个根据访问列表中的条件测试数据包。第一个匹配项决定交换机是接受还是拒绝数据包。由于交换机在第一次匹配后停止测试，因此列表中的条件顺序至关重要。如果没有匹配的条件，交换机将拒绝该数据包。如果没有限制，交换机将转发数据包；否则，交换机将丢弃数据包。交换机可以对其转发的所有数据包使用ACL。

您可以配置访问列表以便为网络提供基本安全性。如果不配置ACL，则允许通过交换机的所有数据包到达所有网络部分。您可以使用ACL来控制哪些主机可以访问网络的不同部分，或者决定在路由器接口上转发或阻止哪些类型的流量。例如，您可以转发电子邮件流量，但不能转发Telnet流量。

术语

ACE	访问控制条目(ACE)- ACL中的单个规则/行
-----	--------------------------

ACL	访问控制列表(ACL) — 应用于端口的一组ACE
DAACL	可下载ACL(DAACL) — 通过ISE安全策略动态推送的ACL
PAACL	端口ACL(PAACL) — 应用于第2层接口的ACL
RAACL	路由ACL(RAACL) — 应用于第3层接口的ACL
VACL	VLAN ACL(VACL) — 应用于VLAN的ACL
GACL	组ACL(GACL) — 根据用户组或客户端的身份动态分配的ACL
IP ACL	用于对IPv4/IPv6数据包进行分类。这些规则包含各种第3层和第4层数据包字段和属性，包括但不限于源和目的IPv4地址、TCP/UDP源和目的端口、TCP标志和DSCP等。
MAACL	Mac地址ACL(MAACL) — 用于对非IP数据包进行分类。规则包含各种第2层字段和属性，包括源/目标MAC地址、以太网类型等。
L4OP	第4层运营商端口(L4OP) — 匹配除EQ (等于) 以外的逻辑。GT (大于)、LT (小于)、NE (不等于) 和RANGE (从至)
VCU	值比较单元(VCU)- L4OP转换为VCU以便对第4层报头执行分类
VMR	值掩码结果(VMR)- ACE条目在TCAM中作为VMR进行内部编程。
CGD	类组数据库(CGD)- FMAN-FP在其中存储ACL内容
类	如何在CGD中识别ACE
CG	Class Group(CG) — 关于如何在CGD中标识ACL的一组类
CGE	类组条目(CGE) — 存储在类组中的ACE条目
FMAN	转发管理器(FMAN)- Cisco IOS® XE与硬件之间的编程层
美联储	转发引擎驱动程序(FED) — 用于编程设备硬件的组件

ACL资源利用率示例

此处提供了三个示例以演示ACL如何使用TCAM、L4OP和VCU。

示例 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条目	L4OP	VCU
消费	5	0	0

示例 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
```

Each range L4OPs consume two VCU

Source and destination L4OPs consume 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条目	L4OP	VCU
消费	4	5	7

示例 3.IPv6 TCAM/L4OP/VCU

IPv6 ACE使用两个TCAM条目，而IPv4使用一个。在本示例中，四个ACE消耗八个TCAM，而不是四个TCAM。

```

<#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

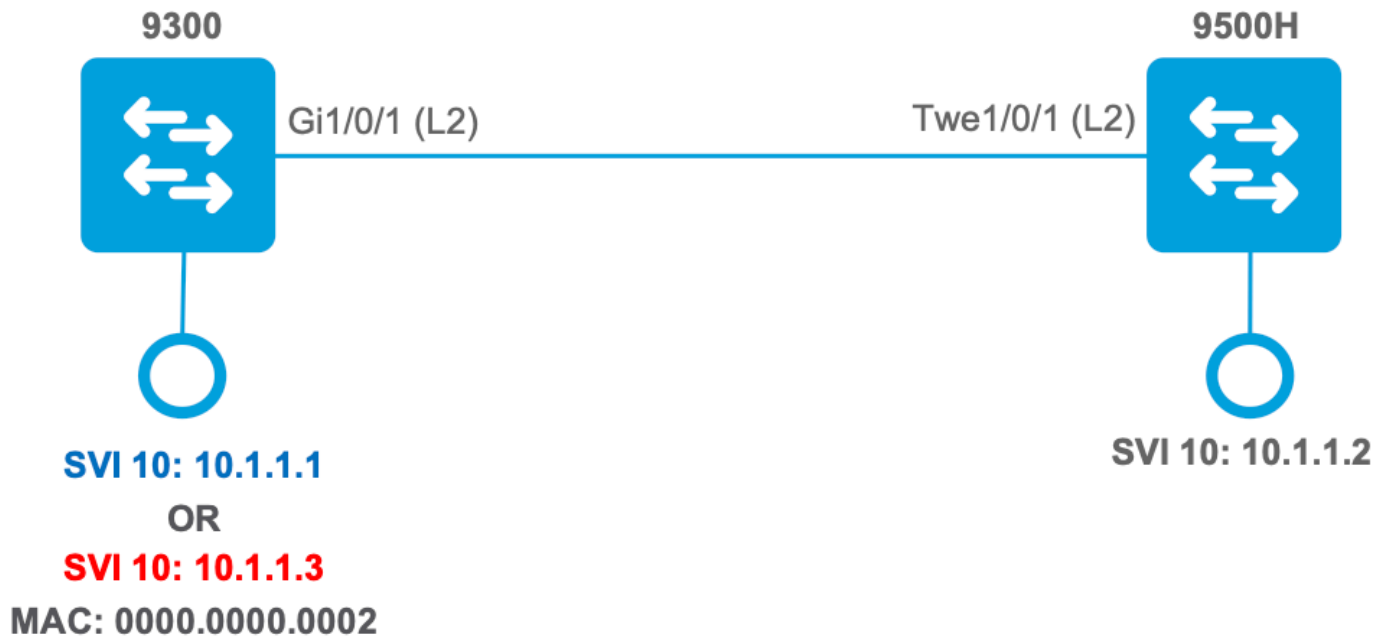
<-- One L4OP & VCU

```

	TCAM条目	L4OP	VCU
消费	8	2	2

拓扑

根据示例中显示的是转发还是丢弃结果，9300 VLAN 10 SVI使用本图中所示的两个IP地址之一。



配置和验证

本节介绍如何在软件和硬件中验证ACL编程并对其进行故障排除。

场景 1.PACL(IP ACL)

PACL分配给第2层接口。

- 安全边界：端口或VLAN
- 附件：第2层接口
- 方向：入口或出口（一次一个）
- 支持的ACL类型：MAC ACL和IP ACL（标准或扩展）

使用IP ACL配置PACL

```
<#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

检验PACL

检索与接口关联的IF_ID。

<#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
```

验证绑定到IF_ID的类组ID(CG 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
```

与CG ID关联的ACL信息。

```
<#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

14:N

matchall:N deny:N

<-- #flds: 2 = two fields in entry | 14: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)

有关CG ID以及哪些接口使用CG ID的策略信息。

<#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 PAACL

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-----
```

确认PACL工作正常。

 **注意：** 当您输入 `show ip access-lists privileged EXEC` 命令，显示的匹配计数不考虑硬件中受访问控制的数据包。使用 `show platform software feed switch {switch_num|active|standby} acl counters hardware` 特权EXEC命令可获取交换和路由数据包的一些基本硬件ACL统计信息。

```
<#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...>

场景 2 : PAcl(MAC ACL)

PAcl分配给第2层接口。

- 安全边界 : 端口或VLAN
- 附件 : 第2层接口
- 方向 : 入口或出口 (一次一个)
- 支持的ACL类型 : MAC ACL和IP ACL (标准或扩展)

使用MAC ACL配置PAcl

<#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
```

检验PAACL

检索与接口关联的IF_ID。

<#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

验证绑定到IF_ID的类组ID(CG 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

与CG ID关联的ACL信息。

<#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-TEST

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 destination"

```
mac_src: value = 0x1aaaaaaaa
```

```
,
```

```
mask = 0xffffffffffff
```

```
<-- Mac source: 0x1aaaaaaaa | hex with leading zeros omitted (0001.aaaa.aaaa) & mask 0xffffffffffff is 1
```

有关CG ID以及哪些接口使用CG ID的策略信息。

```
<#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-----

```

确认PAcl工作正常：

- MACl仅允许源地址0001.aaaa.aaaa。
- 由于这是MAC ACL，非IP ARP数据包将被丢弃，从而导致ping失败。

<#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^FAR
```

```
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^FAR 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^FAR 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^FAR 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 request
```

```
Egress MAC PACL Drop (0x0200004c): 0 frames
```

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

场景 3 : RACL

RACL分配给第3层接口，例如SVI或路由接口。

- 安全边界：不同子网
- 附件：第3层接口
- 方向：入口或出口
- 支持的ACL类型：IP ACL (标准或扩展)

配置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

检验RACL

检索与接口关联的IF_ID。

<#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

验证绑定到IF_ID的类组ID(CG ID)。

<#root>

9500H#

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

```
#####
#####
##### 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

与CG ID关联的ACL信息。

<#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

14:N

matchall:N deny:N

```
<-- #flds: 2 = two fields in entry | 14: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
```

```
14:Y
```

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

```
<-- #flds: 4 = four fields in entry | 14: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
```

```
14_src: start = 1000, end = 1000
```

```
<-- matches eq 1000 (equal UDP port 1000)
```

有关CG ID以及哪些接口使用CG ID的策略信息。

```
<#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-----

```

确认RAACL工作正常。

 **注意：** 当您输入 `show ip access-lists privileged EXEC` 命令，显示的匹配计数不考虑硬件中受访问控制的数据包。使用 `show platform software feed switch{switch_num|active|standby}acl counters hardware` 以获得交换数据包和路由数据包的一些基本硬件ACL统计信息。

```
<#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 deny)
```

```
.....
```

```
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...>
```

场景 4.VACL

VACL分配给第2层VLAN。

- 安全边界：在VLAN内或跨VLAN
- 附件：VLAN/VLAN映射
- 方向：入口和出口同时
- 支持的ACL类型：MAC ACL和IP ACL（标准或扩展）

配置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
```

检验VACL

检索与接口关联的IF_ID。

```
<#root>
9500H#
show platform software fed active ifm interfaces vlan
```

```
Interface
IF_ID
State
-----
Vlan10                0x00420010
READY
```

验证绑定到IF_ID的类组ID(CG 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

与CG组ID关联的ACL信息。

同一命名VACL策略中使用两个ACL，它们分组到此acl组中

<#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 VACL

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 example)

```

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

      GCE#:20 #flds: 2 14: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 14: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 "per

```

有关CG ID以及哪些接口使用CG ID的策略信息。

<#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 id
```

```
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 Interface 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-----
```

确认VACL工作正常。

- 故障排除与PACL和RACL部分相同。有关ping测试的详细信息，请参阅以下各节。
- 从10.1.1.3对10.1.1.2执行ping操作被应用的ACL策略拒绝。
- 检查平台丢弃命令。

```
<#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...>
```

方案 5.组/客户端ACL(DACL)

组/客户端ACL根据用户组或客户端的身份动态应用到用户组。这些有时也称为DAACL。

- 安全边界：客户端（客户端接口级别）
- 附件：每个客户端接口
- 方向：仅入口
- 支持的ACL类型：MAC ACL和IP ACL（标准或扩展）

配置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 authenticat
```

```
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 any
```

检验GACL

绑定到iif-id的组CG 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
```

与组GC ID关联的ACL信息。

```
<#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 14: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 14:N matchall:N deny:N
Result: 0x04000000
ipv4_src: value = 0x00000000, mask = 0x00000000
ipv4_dst: value = 0x00000000, mask = 0x00000000
```

方案 6.ACL 记录

设备软件可以提供有关标准IP访问列表允许或拒绝的数据包的系统日志消息。任何与ACL匹配的数据包都会导致有关数据包的信息日志消息发送到控制台。记录到控制台的消息级别由日志控制台命令控制Syslog消息。

- 单播逆向路径转发(uRPF)所使用的ACL不支持ACL日志消息。仅支持RAACL。
- 从设备控制平面生成的数据包不支持出口方向的ACL日志。
- 在硬件中完成路由并在软件中记录，因此，如果大量数据包匹配包含logkeyword的permit或

deny ACE，则软件无法匹配硬件处理速率，并且无法记录所有数据包。

- 触发ACL的第一个数据包会立即生成日志消息，后续数据包会在出现或记录之前以5分钟为间隔收集。日志消息包括访问列表编号、数据包是被允许还是被拒绝、数据包的源IP地址以及在前5分钟间隔内允许或拒绝的来自该源的数据包数。
- 有关ACL日志行为和限制的完整详细信息，请参阅相关信息部分中说明的相应的安全配置指南Cisco IOS XE。

日志示例PACL:

此示例显示一个负面的情况，其中ACL type和log关键字不能同时工作。

```
<#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
```

日志示例RAACL (拒绝) :

```
<#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

### Originate 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
```

日志示例RACL (允许) :

当log语句用于permit语句时，软件计数器的命中数显示发送的数据包数翻倍。

```
<#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)

故障排除

ACL统计信息

在对ACL问题进行故障排除时，必须了解设备如何测量ACL统计信息以及在何处测量ACL统计信息。

- ACL统计信息在汇聚级别收集，而不是按ACE级别收集。
- 硬件无法允许每个ACE或每个ACL统计信息。
- 系统会收集拒绝数据包、日志数据包和CPU转发数据包等统计信息。
- MAC、IPv4和IPv6数据包的统计信息是单独收集的。
- `show platform software fed switch active acl counters hardware` 可用于显示聚合统计信息。

清除ACL统计信息

在对ACL问题进行故障排除时，清除各种ACL计数器以获得新的基线计数很有帮助。

- 这些命令可用于清除软件和硬件ACL计数器统计信息。
- 对ACL匹配/命中事件进行故障排除时，建议清除相关ACL以查找最新或相关的基线匹配。

```
<#root>
```

```
clear platform software fed active acl counters hardware
```

(clears the hardware matched counters)

```
clear ip access-list counters
```

(clears the software matched counters - IPv4)

```
clear ipv6 access-list counters
```

(clears the software matched counters - IPv6)

ACL TCAM用尽后会发生什么情况？

- ACL始终应用于硬件TCAM。如果之前配置的ACL已使用TCAM，则新的ACL无法获得编程所需的ACL资源。
- 如果在耗尽TCAM后添加ACL，则会丢弃它所连接的接口的所有数据包。
- 在软件中保留ACL的操作称为卸载。
- 当资源可用时，交换机将自动尝试将ACL编程到硬件中。如果成功，ACL将推送到硬件并且数据包开始转发。
- 将软件保留的ACL编程到TCAM的操作称为Reloading。
- PAACL、VACL、RAACL和GACL可以相互独立卸载/重新加载。

ACL TCAM耗尽

- 应用新添加的ACL的接口开始丢弃数据包，直到硬件资源可用。
- GACL客户端进入UnAuth状态。

VCU耗尽

- 一旦超过L4OP限制或超过VCU，软件将执行ACL扩展并创建新的ACE条目，以便在不使用VCU的情况下执行等效操作。
- 一旦发生这种情况，TCAM可能会从这些添加的条目中耗尽。

ACL系统日志错误

如果特定安全ACL资源耗尽，则系统生成SYSLOG消息（接口、VLAN、标签等，值可能不同）。

ACL日志消息	定义	恢复操作
%ACL_ERRMSG-4-UNLOADED: 馈送交换机 1: 接口<interface>上的输入<ACL>未在硬件中编程，且流量被丢弃。	ACL已卸载（保留在软件中）	研究TCAM规模。如果超出规模，请重新设计ACL。
%ACL_ERRMSG-6-REMOVED: 1 fed: 已为标签<label>asic<number>删除接口<interface>上输入<ACL>的卸载配置。	卸载的ACL配置会从接口删除	ACL已删除，无需执行任何操作

%ACL_ERRMSG-6-RELOADED: 1 fed: Input <ACL> on interface <interface>现已加载到asic<number>标签<label>的硬件中。	ACL现在已安装在硬件中	ACL的问题现已通过硬件解决，无需采取任何措施
%ACL_ERRMSG-3-ERROR: 1 fed: Input <ACL> IP ACL <NAME> configuration is not applied on <interface> at bind order <number>。	其他类型的ACL错误（例如dot1x ACL安装失败）	确认是否支持ACL配置，以及TCAM是否超出规模
%ACL_ERRMSG-6-GACL_INFO：交换机1 R0/0：馈送：GACL不支持日志记录。	GACL配置了日志选项	GACL不支持日志。从GACL中删除日志语句。
%ACL_ERRMSG-6-PACL_INFO：交换机1 R0/0：馈送：PAACL不支持日志记录。	PAACL配置了日志选项	PAACL不支持日志。从PAACL中删除日志语句。
%ACL_ERRMSG-3-ERROR：交换机1 R0/0：已提供：输入IPv4组ACL隐式拒绝：<name>：配置未应用于客户端MAC 0000.0000.0000。	(dot1x)ACL无法应用到目标端口	确认是否支持ACL配置，以及TCAM是否超出规模

资源不足情形和恢复操作

场景 1.ACL绑定	恢复操作
<ul style="list-style-type: none"> 创建ACL并将其应用于接口或VLAN。 由于“资源不足”的情况（例如TCAM耗尽），绑定失败。 ACL中没有ACE可以编程到TCAM中。ACL仍处于UNLOADED状态。 在UNLOADED状态下，所有流量（包括控制数据包）都会在接口上丢弃，直到问题得到解决。 	重新设计ACL以降低TCAM的利用率。
场景 2：ACL编辑	恢复操作
<ul style="list-style-type: none"> 创建一个ACL并将其应用到接口，并且在应用到接口时，会向此ACL添加更多ACE条目。 如果TCAM没有资源，则编辑操作失败。 ACL中没有ACE可以编程到TCAM中。ACL仍处于UNLOADED状态。 	重新设计ACL以降低TCAM的利用率。

<ul style="list-style-type: none"> 在UNLOADED状态下，所有流量（包括控制数据包）都会在接口上丢弃，直到问题得到解决。 现有的ACL条目也在UNLOADED状态中失败，直到这种情况得到修复。 	
<p style="text-align: center;">场景 3：ACL重新绑定</p>	<p style="text-align: center;">恢复操作</p>
<ul style="list-style-type: none"> ACL重新绑定是指将ACL附加到接口，然后将另一个ACL附加到同一接口而不分离第一个ACL的操作。 第一个ACL已成功创建并连接。 使用不同的名称和相同的协议(IPv4/IPv6)创建较大的ACL，并将其连接到同一接口。 设备成功分离第一个ACL并尝试将新的ACL附加到此接口。 如果TCAM没有资源，重新绑定操作将失败。 ACL中没有ACE可以编程到TCAM中。ACL仍处于UNLOADED状态。 在UNLOADED状态下，所有流量（包括控制数据包）都会在接口上丢弃，直到问题得到解决。 	<p>重新设计ACL以降低TCAM的利用率。</p>
<p style="text-align: center;">场景 4.绑定空（空）ACL</p>	<p style="text-align: center;">恢复操作</p>
<ul style="list-style-type: none"> 创建没有ACE条目的ACL并将其附加到接口。 系统使用允许“任何ACE”在内部创建此ACL，并将其连接到硬件中的接口（在此状态下允许所有流量）。 然后，使用相同的名称或编号将ACE条目添加到ACL。系统会在添加每个ACE时对TCAM进行编程。 如果TCAM在添加ACE条目时耗尽资源，则ACL将变为UNLOADED状态。 在UNLOADED状态下，所有流量（包括控制数据包）都会在接口上丢弃，直到问题得到解决。 现有的ACL条目也在UNLOADED状态中失败，直到这种情况得到修复。 	<p>重新设计ACL以降低TCAM的利用率。</p>

检验ACL规模

本节介绍用于确定ACL规模和TCAM利用率的命令。

FMAN访问列表摘要：

确定已配置的ACL和每个ACL的ACE总数。

<#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用法：

<#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 cor

```
=====
```

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

TCAM使用情况(17.x):

TCAM usage命令在16.x和17.x系列之间存在显著差异。

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

```

<...snip...>

```

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

```

TCAM使用情况(16.x):

TCAM usage命令在16.x和17.x系列之间存在显著差异。

<#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...>

自定义SDM模板 (TCAM重新分配)

使用Cisco IOS XE Bengaluru 17.4.1, 您可以使用SDM模板来配置ACL功能 `sdm prefer custom acl` 命令。

有关如何配置和验证此功能的详细信息, 请参阅[系统管理配置指南, Cisco IOS XE Bengaluru 17.4.x \(Catalyst 9500交换机 \)](#)。

本节介绍一些基本配置和验证。

验证当前SDM模板：

```
<#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

修改当前的SDM模板：

- 9500H (配置) #sdm prefer custom acl
9500H(config-sdm-acl)#acl-ingress 26 priority 1 <— 应用新的26K值。(在配置指南中讨论优先级)
- 9500H(config-sdm-acl)#acl-egress 20 priority 2
- 9500H(config-sdm-acl)#退出
- 使用 show sdm prefer custom 要查看建议的值和 sdm prefer custom commit 以便通过此CLI应用“查看更改”。
- 检验对SDM配置文件的更改。
- 9500H#show sdm prefer custom

显示SDM模板信息：

这是自定义模板及其详细信息。

入口安全访问控制条目*: 12288 (当前) — 26624 (建议) <— 当前和建议使用 (建议26K)

出口安全访问控制条目*: 15360 (当前) — 20480 (建议)

```
9500H#show sdm prefer自定义用户输入
```

ACL功能用户输入

用户输入值

=====

功能名称优先级 扩展

入口安全访问控制条目：1 26*1024 <— 由用户输入修改为26 x 1024(26K)

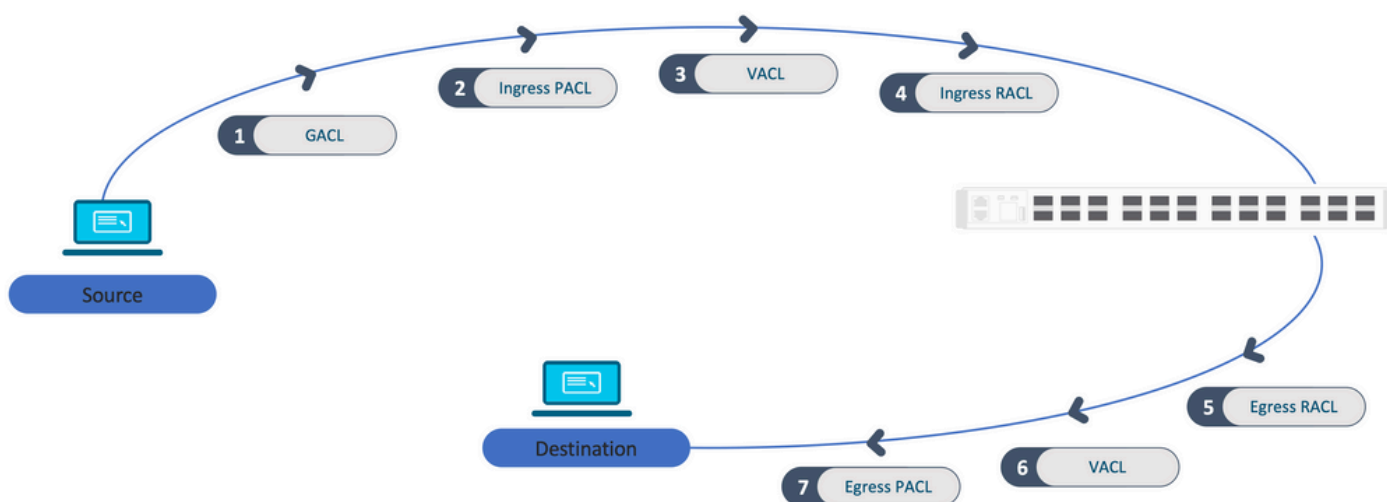
出口安全访问控制条目：2 20*1024 <— 由用户输入修改为20 x 1024(20K)

- 将更改应用于SDM配置文件。
- 9500H (配置) #sdm prefer custom commit
对正在运行的SDM首选项所做的更改会存储起来，并在下次重新加载时生效。 <— 重新加载后，ACL TCAM将分配给自定义值。

进一步阅读：

ACL处理顺序：

ACL按照从源到目的地的顺序进行处理。



堆栈中编程的ACL:

- 非基于端口的ACL (例如，VACL、RACL) 应用于任何交换机上的流量，并且编程在堆栈中的所有交换机上。
- 基于端口的ACL仅应用于端口上的流量，并且只在拥有接口的交换机上进行编程。
- ACL由活动交换机编程，随后应用于成员交换机。
- 相同的规则适用于其他冗余选项，例如ISSU/SVL。

ACL扩展：

- 当设备耗尽L4OP、Lables或VCU时，会发生ACL扩展。设备必须创建多个等效ACE才能完成

相同的逻辑，并快速耗尽TCAM。

- `### L4OP已大规模建立，此ACL已创建##`
`9500H(config)#ip access-list extended TEST`
`9500H(config-ext-nacl)#permit tcp 10.0.0.0 0.255.255.255 any gt 150 <— 匹配端口151及更高版本`

`###必须将此扩展为多个不使用L4OP ###的ACE`

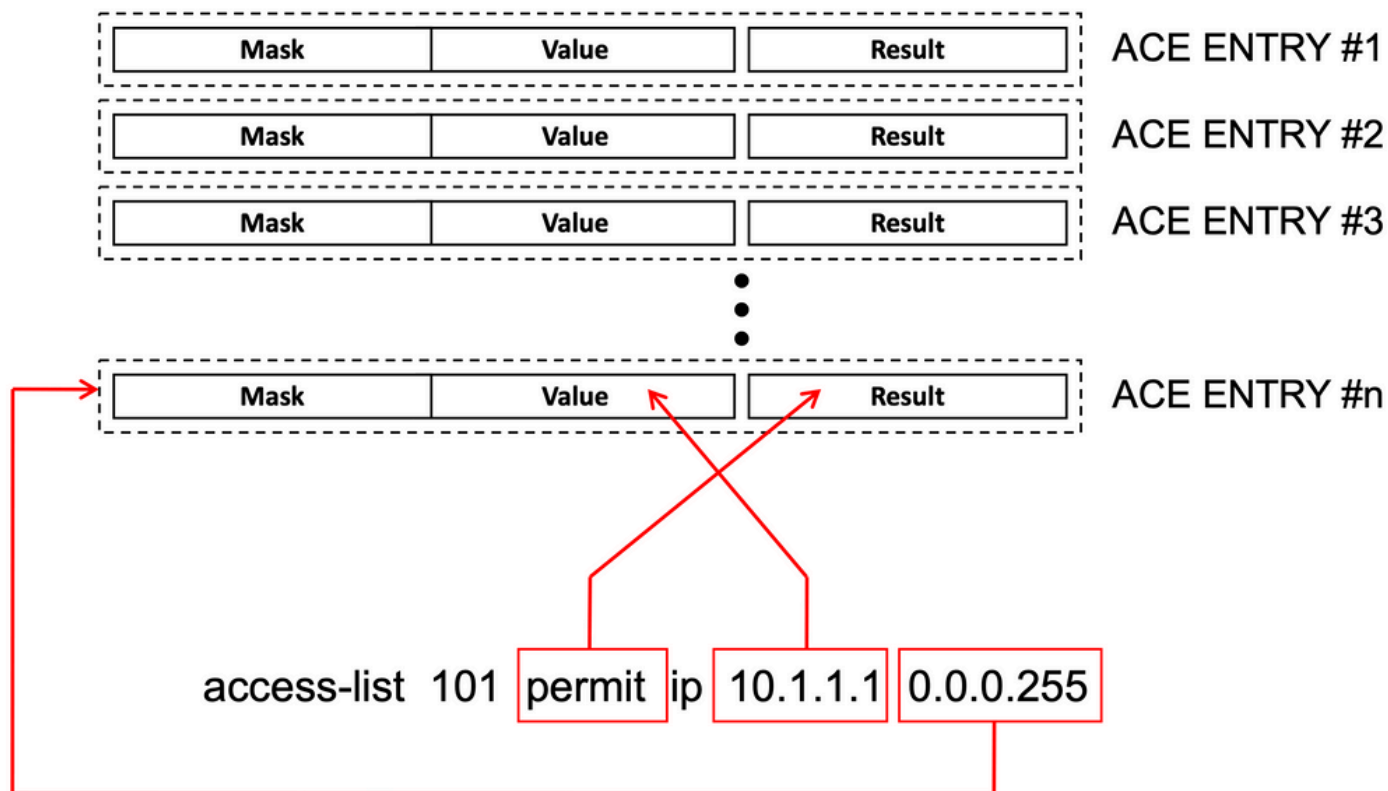
```
9500H(config-ext-nacl)#permit tcp 10.0.0.0 0.255.255.255 any eq 151
9500H(config-ext-nacl)#permit tcp 10.0.0.0 0.255.255.255 any eq 152
9500H(config-ext-nacl)#permit tcp 10.0.0.0 0.255.255.255 any eq 153
9500H(config-ext-nacl)#permit tcp 10.0.0.0 0.255.255.255 any eq 154
...等等....
```

TCAM消费和标签共享：

- 每个ACL策略都由一个标签在内部引用。
- 当ACL策略（安全ACL，如GACL、PACL、VACL、RACL）应用于多个接口或VLAN时，它使用相同的标签。
- 入口/出口ACL使用不同的标签空间。
- IPv4、IPv6和MAC ACL使用其他标签空间。
- 同一PACL应用于接口A的入口和接口A的出口。TCAM中有两个PACL实例，每个实例具有唯一的入口和出口标签。
- 如果将L4OP的相同PACL应用于每个核心上存在的多个入口接口，则在TCAM中编程的相同PACL有两个实例，每个核心一个。

VMR描述：

ACE在TCAM中内部编程为“VMR”，也称为值、掩码、结果。每个ACE条目可以消耗VMR和VCU。



ACL可扩展性：

安全ACL资源专用于安全ACL。它们不与其他功能共享。

ACL TCAM资源	Cisco Catalyst 9600	Cisco Catalyst 9500	Cisco Catalyst 9400	Cisco Catalyst 9300	Cisco Catalyst 9200			
IPv4条目	入口 : 12000*	出口 : 15000 *	C9500:18000*	C9500高性能 入口 : 12000* 出口 : 15000*	18000 *	C9300: 5000	C9300B: 18000	C9300X:8000
IPv6条目	IPv4条目的一半		IPv4条目的一半		一半的IPv4条目	一半的IPv4条目		
一种IPv4	12000		C9500:18000	C9500高性能 : 18000	18000	C9300: 5000	C9300B:18000	C9300X:8000

ACL条目不能超过			15000				
一种IPv6 ACL条目不能超过	6000	C9500: 9000	C9500高性能: 7500	9000	2500/9000/4000		
L4OP/标签	8	8		8	8		
入口VCU	192	192		192	192		
出口VCU	96	96		96	96		

相关信息

- [安全配置指南, Cisco IOS XE Amsterdam 17.3.x \(Catalyst 9200交换机 \)](#)
- [安全配置指南, Cisco IOS XE Amsterdam 17.3.x \(Catalyst 9300交换机 \)](#)
- [安全配置指南, Cisco IOS XE Amsterdam 17.3.x \(Catalyst 9400交换机 \)](#)
- [安全配置指南, Cisco IOS XE Amsterdam 17.3.x \(Catalyst 9500交换机 \)](#)
- [安全配置指南, Cisco IOS XE Amsterdam 17.3.x \(Catalyst 9600交换机 \)](#)
- [系统管理配置指南, Cisco IOS XE Bengaluru 17.4.x \(Catalyst 9500交换机 \)](#)
- [思科技术支持和下载](#)

Debug和Trace命令

数字	命令	备注
1	show platform hardware fed [switch] active fwd-asic drops exceptions asic <0>	转储ASIC服务器上的异常计#N器。
2	show platform software fed [switch] active acl	此命令可打印机箱上所有已配置ACL的相关信息以及接口和策略信息。

3	show platform software fed [switch] active acl policy 18	此命令仅打印有关策略18的信息。您可以从命令2获取此策略ID。
4	show platform software fed [switch] active acl interface intftype pacl	此命令根据接口类型 (pacl/vacl/racl/gacl/sgacl等) 打印有关ACL的信息。
5	show platform software fed [switch] active acl interface intftype pacl acltype ipv4	此命令根据接口类型 (pacl/vacl/racl/gacl/sgacl等) 打印有关ACL的信息，并过滤基于协议的信息 (ipv4/ipv6/mac等)。
6	show platform software fed [switch] active acl interface intftype pacl acltype ipv4	此命令打印有关接口的信息。
7	show platform software fed [switch] active acl interface 0x9	此命令根据IIF-ID (来自6的命令) 打印接口上应用的ACL的简短信息。
8	show platform software fed [switch] active acl definition	此命令可打印有关机箱上配置的ACL及其存在于CGD中的信息。
9	show platform software fed [switch] active acl iifid 0x9	此命令根据IIF-ID打印接口上应用的ACL的详细信息。
10	show platform software fed [switch] active acl usage	此命令根据功能类型打印每个ACL使用的VMR数量。
11	show platform software fed [switch] active acl policy intftype pacl vcu	此命令根据接口类型 (pacl/vacl/racl/gacl/sgacl等) 为您提供策略信息和VCU信息。
12	show platform software fed [switch] active acl policy intftype pacl cam	此命令根据接口类型 (pacl/valc/racl/gacl/sgacl等) 提供有关CAM中VMR的策略信息和详细信息。
13	show platform software interface [switch] [active] R0 brief	此命令可提供有关机箱上接口的详细信息。
14	show platform software fed [switch] active port if_id 9	此命令根据IIF-ID打印端口的详细信息。
15	show platform software fed [switch] active vlan	此命令可打印有关VLAN 30的详细信息。

	30	
16	show platform software fed [switch] active acl cam asic 0	此命令在正在使用的ASIC 0上打印完整的ACL cam。
17	show platform software fed [switch] active acl counters hardware	此命令打印硬件中的所有ACL计数器。
18	show platform hardware fed [switch] active fwd- asic resource tcam table pbr record 0 format 0	在打印PBR部分的条目时，您可以指定不同的部分，例如ACL和CPP，而不是PBR。
19	show platform software fed [switch] active punt cpuq [1 2 3 ...]	为了检查某个CPU队列上的活动，您还可以选择清除用于调试的队列统计信息。
20	show platform software fed [switch] active ifm mappings gpn	使用IIF-ID和GPN打印接口映射
21	show platform software fed [switch active ifm if- id	打印有关接口配置和与ASIC的关联性的信息。此命令有助于检查ASIC和CORE的接口。
22	set platform software trace fed [switch] active acl/asic_vmr/asic_vcu/cgac1/sgacl [debug error ...]	为FED中的特定功能设置跟踪。
23	request platform software trace rotate all	正在清除跟踪缓冲区。
24	show platform software trace message fed [switch] active	正在打印FED的跟踪缓冲区。
25	set platform software trace forwarding-manager [switch] [active] f0 fman [debug error ...]	启用FMAN跟踪。
26	show platform software trace message forwarding-manager [switch] [active] f0	正在打印FMAN的跟踪缓冲区。
27	debug platform software infrastructure punt detail	在PUNT上设置调试。

28	debug ip cef packet all input rate 100	CEF数据包调试已启用。
----	--	--------------

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。