

Catalyst 9000スイッチでのセキュリティACLの検証

内容

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[用語](#)

[ACLリソース使用率の例](#)

[例 1. IPv4 TCAM](#)

[例 2. IPv4 TCAM/L4OP/VCU](#)

[例 3. IPv6TCAM/L4OP/VCU](#)

[トポロジ](#)

[設定と確認](#)

[シナリオ 1. PACL \(IP ACL\)](#)

[IP ACLを使用したPACLの設定](#)

[PACLの確認](#)

[シナリオ 2. PACL \(MAC ACL\)](#)

[MAC ACLを使用したPACLの設定](#)

[PACLの確認](#)

[シナリオ 3. RAACL](#)

[RAACLの設定](#)

[RAACLの確認](#)

[シナリオ 4. VAACL](#)

[VAACLの設定](#)

[VAACLの確認](#)

[シナリオ 5. グループ/クライアントACL \(DAACL\)](#)

[GACLの設定](#)

[GACLの確認](#)

[シナリオ 6. ACL ロギング](#)

[トラブルシューティング](#)

[ACL統計情報](#)

[ACL統計情報のクリア](#)

[ACL TCAMが枯渇するとどうなりますか。](#)

[ACL TCAM枯渇](#)

[VCUの消耗](#)

[ACL syslogエラー](#)

[リソース不足のシナリオと回復アクション](#)

[ACLスケールの確認](#)

[カスタムSDMテンプレート \(TCAM再割り当て\)](#)

[関連情報](#)

はじめに

このドキュメントでは、Catalyst 9000シリーズスイッチのACL (アクセスコントロールリスト) の確認とトラブルシューティングの方法について説明します。

前提条件

要件


このドキュメントに関する固有の要件はありません。

使用するコンポーネント

このドキュメントの情報は、次のハードウェアのバージョンに基づくものです。

- C9200
- C9300
- C9400
- C9500
- C9600

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな (デフォルト) 設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

 注:他のシスコプラットフォームでこれらの機能を有効にするために使用するコマンドについては、該当するコンフィギュレーションガイドを参照してください。

背景説明

ACLは、ルータまたはスイッチを通過するトラフィックをフィルタし、指定されたインターフェイスを通過するパケットを許可または拒否します。ACLは、パケットに適用される許可条件と拒否条件の連続した集合です。パケットがインターフェイスで受信されると、スイッチは、アクセスリストで指定された基準に基づいて、パケットに転送が必要な権限がパケットにあることを確認するために、適用されているACLとパケット内のフィールドを比較します。アクセスリスト内の条件に対して、1つずつパケットをテストします。最初の一致によって、スイッチがパケットを受け入れるか拒否するかが決まります。スイッチは最初に一致した後でテストを停止するため、リスト内の条件の順序は重要です。条件が一致しない場合、スイッチはパケットを拒否します。制限がない場合、スイッチはパケットを転送し、制限がない場合はパケットをドロップします。スイッチは、転送するすべてのパケットでACLを使用できます。

アクセスリストを設定すると、ネットワークの基本的なセキュリティを確保できます。ACLを設

定しない場合は、スイッチを通過するすべてのパケットがすべてのネットワークパーツに対して許可されます。ACLを使用すると、ネットワークのさまざまな部分にアクセスできるホストを制御したり、ルーティングインターフェイスで転送またはブロックするトラフィックのタイプを決定したりできます。たとえば、電子メールトラフィックは転送できますが、Telnetトラフィックは転送できません。

用語

ACE	アクセスコントロールエントリ(ACE):ACL内の1つのルール/行
ACL	アクセスコントロールリスト(ACL) : ポートに適用されるACEのグループ
DAACL	ダウンロード可能ACL(DAACL):ISEセキュリティポリシーによって動的にプッシュされるACL
PAACL	ポートACL(PAACL) : レイヤ2インターフェイスに適用されるACL
RAACL	ルーテッドACL(RAACL) : レイヤ3インターフェイスに適用されるACL
VACL	VLAN ACL(VACL):VLANに適用されるACL
GACL	グループACL(GACL) : ユーザグループまたはクライアントのIDに基づいて動的に割り当てられるACL
IP ACL	IPv4/IPv6パケットの分類に使用されます。これらのルールには、さまざまなレイヤ3およびレイヤ4パケットフィールドと属性が含まれています。たとえば、送信元と宛先のIPv4アドレス、TCP/UDP送信元と宛先ポート、TCPフラグ、DSCPなどが含まれます。
MAACL	Mac Address ACL(MAACL) : 非IPパケットの分類に使用されます。ルールには、さまざまなレイヤ2フィールドと、送信元/宛先MACアドレス、イーサネットタイプなどの属性が含まれています。
L4OP	レイヤ4オペレータポート(L4OP):EQ (等しい) 以外のロジックに一致します。GT (より大きい) 、LT (より小さい) 、NE (等しくない) 、およびRANGE(FROM-TO)
VCU	値比較ユニット(VCU) : レイヤ4ヘッダーで分類を実行するために、L4OPがVCUに変換されます。

VMR	値マスク結果(VMR):ACEエントリはVMRとしてTCAMに内部的にプログラムされます。
CGD	クラスグループデータベース(CGD):FMAN-FPがACLコンテンツを保存する場所
クラス	CGDでのACEの識別方法
CG	Class Group (CG ; クラスグループ) :CGD内でACLがどのように識別されるかについてのクラスのグループ
CGE	クラスグループエントリ(CGЕ) : クラスグループ内に保存されるACEエントリ
FMAN	Forwarding Manager(FMAN):Cisco IOS® XEとハードウェア間のプログラミングレイヤ
FED	フォワーディングエンジンドライバ(FED) : デバイスのハードウェアをプログラムするコンポーネント

ACLリソース使用率の例

ここでは、ACLがTCAM、L4OP、およびVCUをどのように消費するかを示すために、3つの例を示します。

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エントリ	L4OP	VCU
消費	5	0	0

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 VCUs

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エントリ	L4OP	VCU
消費	4	5	7

例 3.IPv6 TCAM/L4OP/VCU

IPv6 ACEは2つのTCAMエントリを使用しますが、一方はIPv4用です。この例では、4つのACEが

4つではなく8つのTCAMを消費します。

```
<#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

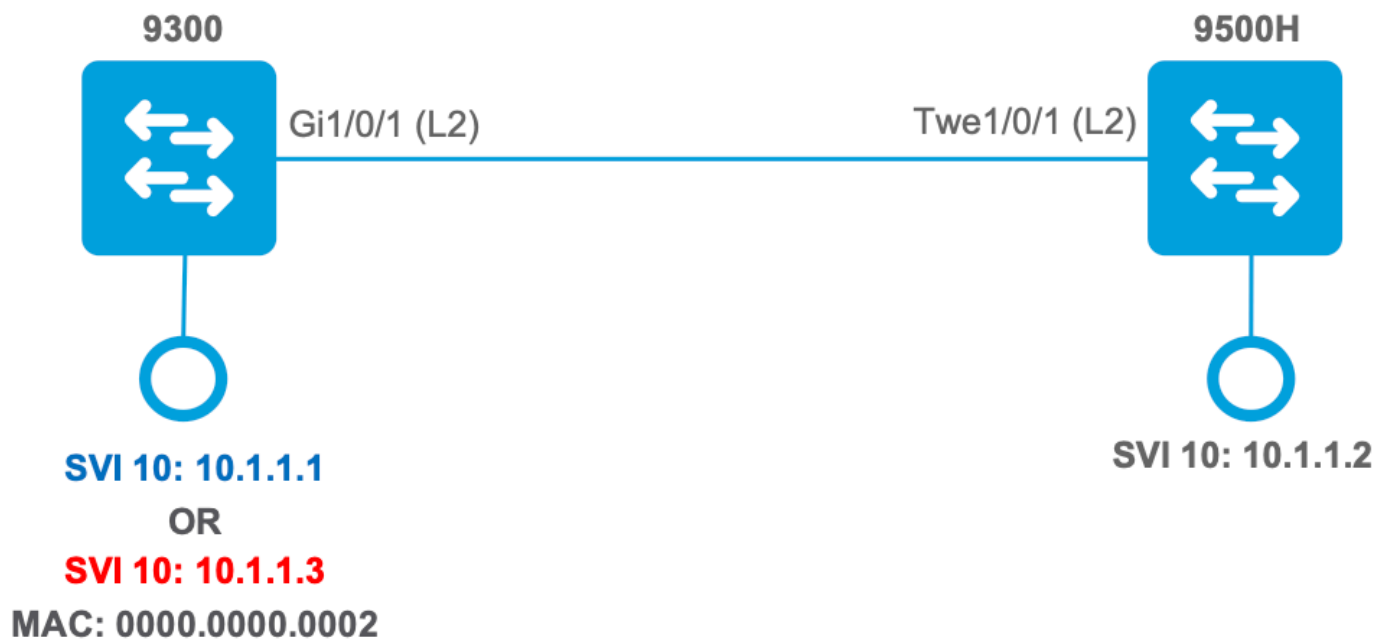
host 2001:DB8:C18:2:1::1

<-- One L4OP & VCU
```

	TCAMエントリ	L4OP	VCU
消費	8	2	2

トポロジ

9300 VLAN 10 SVIは、この図に示す2つのIPアドレスのいずれかを使用します。これは、この例に転送と廃棄のどちらの結果が示されているかに基づきます。



設定と確認

このセクションでは、ソフトウェアおよびハードウェアでのACLプログラミングの確認とトラブルシューティングの方法について説明します。

シナリオ 1.PACL(IP ACL)

PAACLはレイヤ2インターフェイスに割り当てられます。

- セキュリティ境界：ポートまたはVLAN
- アタッチメント：レイヤ2インターフェイス
- 方向：入力または出力（一度に1つずつ）
- サポートされるACLタイプ：MAC ACLおよびIP ACL（標準または拡張）

IP ACLを使用したPAACLの設定

```
<#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
```

PACLの確認

インターフェイスに関連付けられたIF_IDを取得します。

```
<#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
```

IF_IDにバインドされているクラスグループID(CG ID)を確認します。

```
<#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

CG IDに関連付けられたACL情報。

<#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

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

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

CG IDに関するポリシー情報、およびCG IDを使用するインターフェイス。

<#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-----
```

PACLが機能していることを確認します。

 注： を入力すると、 show ip access-lists privileged EXEC コマンドを使用すると、表示される match countには、ハードウェアでアクセス制御されているパケットは含まれません。スイッチドパケットとルーテッドパケットに関する基本的なハードウェアACLの統計情報を取得するには、show platform software fed switch {switch_num|active|standby}acl counters hardware特権 EXECコマンドを使用します。

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

シナリオ 2.PACL(MAC ACL)

PACLはレイヤ2インターフェイスに割り当てられます。

- セキュリティ境界：ポートまたはVLAN

- アタッチメント：レイヤ2インターフェイス
- 方向：入力または出力（一度に1つずつ）
- サポートされるACLタイプ：MAC ACLおよびIP ACL（標準または拡張）

MAC ACLを使用したPAACLの設定

```
<#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
```

PAACLの確認

インターフェイスに関連付けられたIF_IDを取得します。

```
<#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

IF_IDにバインドされているクラスグループID(CG ID)を確認します。

<#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

CG IDに関連付けられたACL情報。

<#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

CG IDに関するポリシー情報、およびCG IDを使用するインターフェイス。

<#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-----
```

PACLが機能していることを確認します。

- MACLは送信元アドレス0001.aaaa.aaaaのみを許可します。
- これはMAC ACLであるため、非IP ARPパケットはドロップされ、pingが失敗します。

<#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 is blocked
```

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

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

シナリオ 3.RACL

RACLは、SVIやルーテッドインターフェイスなどのレイヤ3インターフェイスに割り当てられます。

- セキュリティ境界：異なるサブネット
- アタッチメント：レイヤ3インターフェイス
- 方向：入力または出力
- サポートされるACLタイプ：IP ACL (標準または拡張)

RACLの設定

```
<#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

```

RACLの確認

インターフェイスに関連付けられたIF_IDを取得します。

```
<#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

IF_IDにバインドされているクラスグループID(CG ID)を確認します。

<#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

CG IDに関連付けられたACL情報。

<#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

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

CG IDに関するポリシー情報、およびCG IDを使用するインターフェイス。

<#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 Interface 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-----
```

RACLが機能していることを確認します。

 注： を入力すると、 show ip access-lists privileged EXEC コマンドを使用すると、表示されるmatch countには、ハードウェアでアクセス制御されているパケットは含まれません。 show platform software fed switch{switch_num|active|standby}acl counters hardwareコマンドを使用します。特権EXECコマンドを発行して、スイッチドパケットとルーテッドパケットに関する基本的なハードウェアACLの統計情報を取得します。

<#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 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...>
```

シナリオ 4.VACL

VACLはレイヤ2 VLANに割り当てられます。

- セキュリティ境界：VLAN内またはVLAN間
- 添付ファイル：VLAN/VLANマップ
- 方向：入力と出力の両方を同時に処理
- サポートされるACLタイプ：MAC ACLおよびIP ACL (標準または拡張)

VACLの設定

```
<#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
```

VACLの確認

インターフェイスに関連付けられたIF_IDを取得します。

```
<#root>
```

```
9500H#
```

```
show platform software fed active ifm interfaces vlan
```

```
Interface
```

```
IF_ID
```

```
State
```

```
Vlan10
```

```
0x00420010
```

READY

IF_IDにバインドされているクラスグループID(CG ID)を確認します。

<#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 interface

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

CGグループIDに関連付けられたACL情報。

同じ名前付きVACLポリシーで使用される2つのACLが、このacl-groupにグループ化されています

<#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 14: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
```

CG IDに関するポリシー情報、およびCG IDを使用するインターフェイス。

<#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 in
```

```
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 : 0xf0000091
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-----

```

VACLが機能していることを確認します。

- トラブルシューティングは、PACLおよびRACLセクションと同じシナリオです。pingテストの詳細については、次のセクションを参照してください。
- 10.1.1.3から10.1.1.2へのpingは、適用されるACLポリシーによって拒否されます。
- platform dropコマンドをチェックします。

<#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...>

シナリオ 5.グループ/クライアントACL(DACL)

グループ/クライアントACLは、ユーザグループまたはクライアントのIDに基づいて動的に適用されます。これらはDACLとも呼ばれます。

- セキュリティ境界：クライアント (クライアントインターフェイスレベル)
- 添付ファイル：クライアントインターフェイス単位
- 方向：入力のみ
- サポートされるACLタイプ：MAC ACLおよびIP ACL (標準または拡張)

GACLの設定

```
<#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 authentication)

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

GACLの確認

iif-idにバインドされたグループ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

グループGC IDに関連付けられたACL情報。

<#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
```

```
-----  
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  
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
```

```
<-- Permits 1
```

シナリオ 6.ACL ロギング

デバイスソフトウェアは、標準のIPアクセスリストで許可または拒否されたパケットに関するsyslogメッセージを提供できます。ACLに一致するパケットがあると、パケットに関する情報ログメッセージがコンソールに送信されます。コンソールに記録されるメッセージのレベルは、ロギングコンソールsyslogメッセージを制御するコマンド。

- ACLログメッセージは、Unicast Reverse Path Forwarding(uRPF)で使用するACLではサポートされません。RACLでのみサポートされます。
- 出力方向のACLログは、デバイスのコントロールプレーンから生成されたパケットではサポートされません。
- ルーティングはハードウェアとソフトウェアで行われるため、多数のパケットがlogkeywordを含むpermitまたはdeny ACEに一致する場合、ソフトウェアはハードウェアの処理速度を一致させることができず、すべてのパケットをログに記録できるわけではありません。
- ACLをトリガーする最初のパケットによって、すぐにログメッセージが生成され、後続のパケットは5分間隔で収集されてから表示または記録されます。ログメッセージには、アクセスリスト番号、パケットが許可されたか拒否されたか、パケットの送信元IPアドレス、および直前の5分インターバルの間に許可または拒否されたその送信元からのパケットの数が含まれます。
- ACLログの動作と制限事項の詳細については、「関連情報」セクションに記載されている適切なセキュリティ設定ガイド、Cisco IOS XEを参照してください。

ログ例PAACL:

次の例は、ACLタイプとlogキーワードと一緒に機能しない負のケースを示しています。

```
<#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
```

ログの例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
```

```
### 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
```

ログ例RACL (許可):

log文をpermit文に使用すると、ソフトウェアカウンタのヒット数は送信されたパケット数の2倍になります。

<#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)
```

トラブルシューティング

ACL統計情報

ACLの問題のトラブルシューティングを行う際には、ACL統計情報がデバイスによってどのように、どこで測定されるのかを理解することが重要です。

- ACL統計情報は、ACEレベルごとではなく、集約レベルで収集されます。
- ハードウェアには、ACEごとまたはACLごとの統計情報を許可する機能がありません。
- 拒否、ログ、CPU転送パケットなどの統計情報が収集されます。
- MAC、IPv4、およびIPv6パケットの統計情報は個別に収集されます。
- `show platform software fed switch active acl counters hardware` 集約統計情報を表示するために使用できます。

ACL統計情報のクリア

ACLの問題のトラブルシューティングを行う際には、新しいベースラインカウントを取得するために、さまざまなACLカウンタをクリアすると役立つ場合があります。

- これらのコマンドを使用すると、ソフトウェアおよびハードウェアのACLカウンタの統計情報をクリアできます。
- ACLの一致/ヒットイベントをトラブルシューティングする場合は、関連するACLをクリアして、最新または関連する一致をベースラインにすることが推奨されます。

```
<#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が枯渇するとどうなりますか。

- ACLは常にハードウェアTCAMで適用されます。以前に設定されたACLによってTCAMがすでに使用されている場合、新しいACLはプログラムに必要なACLリソースを取得しません。
- TCAMが枯渇した後にACLが追加されると、接続されているインターフェイスのすべてのパケットがドロップされます。
- ソフトウェアでACLを保持するアクションは、アンロードと呼ばれます。
- リソースが使用可能になると、スイッチは自動的にACLをハードウェアにプログラムしようとしています。成功すると、ACLがハードウェアにプッシュされ、パケットの転送が開始されます。
- ソフトウェアで保持されたACLをTCAMにプログラミングする動作をリロードと呼びます。
- PAACL、VACL、RAACL、およびGACLは、互いに独立してアンロード/リロードできます。

ACL TCAM枯渇

- 新しく追加されたACLが適用されたインターフェイスは、ハードウェアリソースが使用可能になるまでパケットの廃棄を開始します。
- GACLクライアントはUnAuth状態になります。

VCUの消耗

- L4OPの制限を超えた場合、またはVCUの範囲外になった場合、ソフトウェアはACL拡張を実行し、VCUを使用せずに同等のアクションを実行するために新しいACEエントリを作成します。
- これが発生すると、TCAMはこれらの追加されたエントリから使い果たされる可能性があります。

ACL syslogエラー

特定のセキュリティACLリソースが不足すると、SYSLOGメッセージがシステムによって生成されます (インターフェイス、VLAN、ラベルなど、値が異なる場合があります)。

ACLログメッセージ	定義	回復アクション
%ACL_ERRMSG-4-UNLOADED : スイッチ1 fed : インターフェイス<interface>の入力 <ACL>はハードウェアでプログラムされておらず、トラフィックはドロップされます。	ACLがアンロードされる (ソフトウェアに保持される)	TCAMスケールを調べます。規模を超える場合は、ACLを再設計します。

<p>%ACL_ERRMSG-6-REMOVED: 1 fed : インターフェイス<interface>の入力<ACL>のアンロード設定が、ラベル<label>asic<number>に対して削除されました。</p>	<p>アンロードされたACL設定がインターフェイスから削除される</p>	<p>ACLはすでに削除されています。実行するアクションはありません。</p>
<p>%ACL_ERRMSG-6-RELOADED: 1 fed : インターフェイス<interface>の入力<ACL>が、asic<number>のラベル<label>に対してハードウェアにロードされました。</p>	<p>ACLがハードウェアにインストールされました。</p>	<p>ACLに関する問題はハードウェアで解決され、対処は不要になりました</p>
<p>%ACL_ERRMSG-3-ERROR: 1 fed : 入力<ACL> IP ACL <NAME>設定がバインド順<number>の<interface>に適用されていません。</p>	<p>その他のタイプのACLエラー (dot1x ACLインストールの失敗など)</p>	<p>ACL設定がサポートされ、TCAMがスケールを超えていないことを確認する</p>
<p>%ACL_ERRMSG-6-GACL_INFO : スイッチ1 R0/0: fed:GACLではロギングはサポートされていません。</p>	<p>GACLにログオプションが設定されている</p>	<p>GACLはログをサポートしていません。GACLからログステートメントを削除します。</p>
<p>%ACL_ERRMSG-6-PACL_INFO : スイッチ1 R0/0: fed : ロギングはPACLではサポートされていません。</p>	<p>PACLにはlogオプションが設定されています</p>	<p>PACLはログをサポートしていません。PACLからログステートメントを削除します。</p>
<p>%ACL_ERRMSG-3-ERROR: Switch 1 R0/0: fed: Input IPv4 Group ACL implicit_deny:<name> : 設定がクライアントMAC 0000.0000.0000に適用されていません。</p>	<p>(dot1x)ACLがターゲットポートに適用されない</p>	<p>ACL設定がサポートされ、TCAMがスケールを超えていないことを確認する</p>

リソース不足のシナリオと回復アクション

シナリオ 1.ACLバインド	回復アクション
<ul style="list-style-type: none"> ACLが作成され、インターフェイスまたはVLANに適用されます。 TCAMの枯渇など、「リソース不足」状態が原因でバインドが失敗します。 ACL内のACEをTCAMにプログラムすることはできません。ACLはUNLOADED状態のままになります。 	<p>TCAMの使用率を下げるため、ACLを再設計します。</p>

<ul style="list-style-type: none"> UNLOADED状態では、問題が修正されるまで、すべてのトラフィック（制御パケットを含む）がインターフェイスでドロップされます。 	
<p style="text-align: center;">シナリオ 2.ACLの編集</p>	<p style="text-align: center;">回復アクション</p>
<ul style="list-style-type: none"> ACLが作成されてインターフェイスに適用され、インターフェイスに適用されている間に、このACLにさらにACEエントリが追加されます。 TCAMにリソースがない場合、編集操作は失敗します。 ACL内のACEをTCAMにプログラムすることはできません。ACLはUNLOADED状態のままになります。 UNLOADED状態では、問題が解決するまで、インターフェイス上のすべてのトラフィック（制御パケットを含む）がドロップされます。 既存のACLエントリも、これが修正されるまでUNLOADED状態で失敗します。 	<p>TCAMの使用率を下げるため、ACLを再設計します。</p>
<p style="text-align: center;">シナリオ 3.ACLの再バインド</p>	<p style="text-align: center;">回復アクション</p>
<ul style="list-style-type: none"> ACLの再バインドとは、ACLをインターフェイスに割り当て、その後、最初のACLをデタッチせずに、別のACLを同じインターフェイスに割り当てることです。 最初のACLが作成され、正常にアタッチされました。 同じプロトコル(IPv4/IPv6)を持つ別の名前の大きいACLが作成され、同じインターフェイスに接続されます。 デバイスは最初のACLのアタッチを正常に解除し、新しいACLをこのインターフェイスにアタッチしようとします。 TCAMにリソースがない場合、再バインド操作は失敗します。 ACL内のACEをTCAMにプログラムすることはできません。ACLはUNLOADED状態のままになります。 UNLOADED状態では、問題が修正されるま 	<p>TCAMの使用率を下げるため、ACLを再設計します。</p>

<p>で、すべてのトラフィック (制御パケットを含む) がインターフェイスでドロップされます。</p>	
<p>シナリオ 4.空 (ノル) ACLのバインド</p>	<p>回復アクション</p>
<ul style="list-style-type: none"> • ACEエントリを持たないACLが作成され、インターフェイスに接続されます。 • このACLは、permit 「any ACE」 を使用して内部的に作成され、ハードウェア内のインターフェイスに関連付けられます (この状態ではすべてのトラフィックが許可されます) 。 • ACEエントリは、同じ名前または番号でACLに追加されます。各ACEが追加されると、システムはTCAMをプログラムします。 • ACEエントリを追加するときにTCAMのリソースが不足すると、ACLはUNLOADED状態に移行します。 • UNLOADED状態では、問題が修正されるまで、すべてのトラフィック (制御パケットを含む) がインターフェイスでドロップされます。 • 既存のACLエントリも、これが修正されるまでUNLOADED状態で失敗します。 	<p>TCAMの使用率を下げるため、ACLを再設計します。</p>

ACLスケールの確認

このセクションでは、ACLのスケールとTCAMの使用率を判別するためのコマンドについて説明します。

FMANアクセスリストの要約：

設定されているACLと、ACLごとの合計ACE数を特定します。

```
<#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の使用 :

<#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の使用(17.x):

TCAM usageコマンドには、16.xトレインと17.xトレインの間に大きな違いがあります。

<#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		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の使用(16.x):

TCAM usageコマンドには、16.xトレインと17.xトレインの間に大きな違いがあります。

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

カスタムSDMテンプレート (TCAM再割り当て)

Cisco IOS XEベンガロール17.4.1を使用して、acl機能用のカスタムSDMテンプレートを設定するには、sdm prefer custom aclコマンドを使用して、アップグレードを実行します。

この機能を設定および確認する方法の詳細については、『[システム管理コンフィギュレーションガイド、Cisco IOS XEベンガロール17.4.x \(Catalyst 9500スイッチ \)](#)』を参照してください。

このセクションでは、いくつかの基本設定と検証について説明します。

現在のSDMテンプレートを確認します。

```
<#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
```

現在のSDMテンプレートを変更します。

- 9500H(config)#sdmはカスタムaclを優先する
9500H(config-sdm-acl)#acl-ingress 26 priority 1 <- 新しい26K値を適用します。(設定ガイドで説明されているプライオリティ)
9500H(config-sdm-acl)#acl-egress 20プライオリティ2
9500H(config-sdm-acl)#exit
利用 show sdm prefer custom 提案された値と sdm prefer custom commit このCLIを使用して「view the changes」を適用します。
- SDMプロファイルへの変更を確認します。
- 9500H番号show sdm prefer custom

SDMテンプレート情報の表示 :

これは、詳細が記載されたカスタムテンプレートです。

入力セキュリティアクセスコントロールエントリ*:12288 (現在) - 26624 (提案) <- 現在および提案された使用方法 (26,000提案)

出力セキュリティアクセスコントロールエントリ*: 15360 (現行) ~ 20480 (提案)

9500H番号show sdm prefer custom user-input

ACL機能のユーザ入力

ユーザ入力値

=====

機能名の優先順位 スケール

入力セキュリティアクセスコントロールエントリ : 1 26*1024 <- ユーザ入力によって26 x 1024(26K)に変更

Egress Security Access Control Entries: 2 20*1024 <- ユーザ入力によって20 x 1024(20K)に変更される

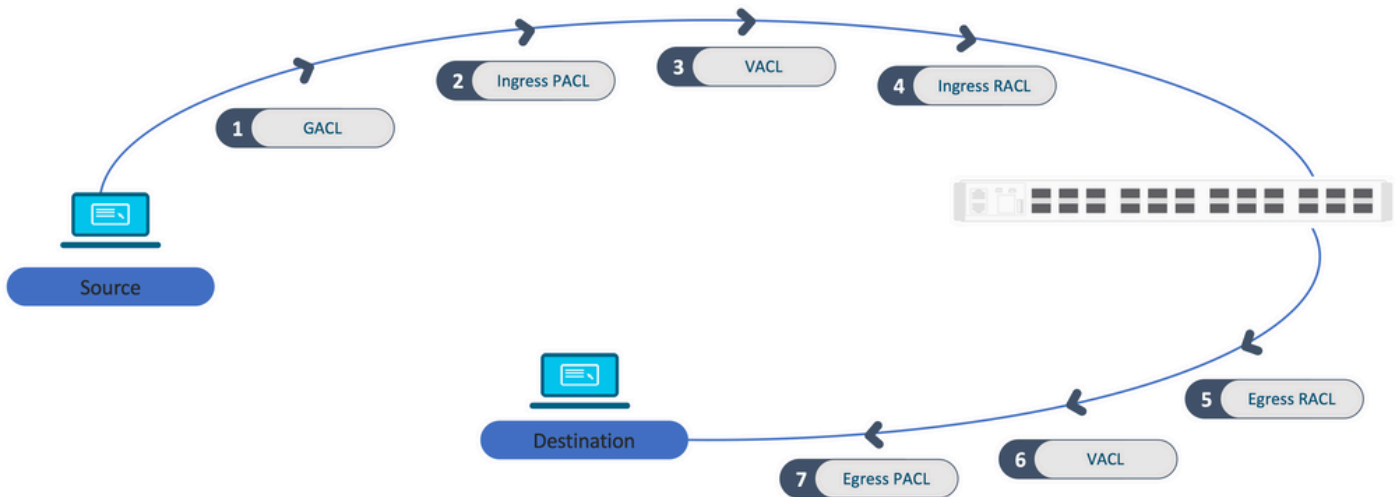
- SDMプロファイルに変更を適用します。
- 9500H(config)#sdmはカスタムコミットを好む
実行中のSDM設定に対する変更は保存され、次回のリロード時に有効になります。 <- リ

ロードすると、ACL TCAMがカスタム値に割り当てられます。

参考資料:

ACLの処理順序 :

ACLは、送信元から宛先へのこの順序で処理されます。



スタックにプログラムされたACL:

- ポートベースではないACL (VACL、RAACLなど) は、任意のスイッチ上のトラフィックに適用され、スタック内のすべてのスイッチにプログラムされます。
- ポートベースのACLは、ポート上のトラフィックにのみ適用され、インターフェイスを所有するスイッチ上でのみプログラムされます。
- ACLはアクティブスイッチによってプログラムされ、その後メンバスイッチに適用されます。
- ISSU/SVLなどの他の冗長オプションにも同じ規則が適用されます。

ACLの拡張 :

- ACLの拡張は、デバイスがL4OP、Label、またはVCUを使い果たした場合に行われます。デバイスは、同じロジックを実現し、TCAMを迅速に枯渇させるために、同等のACEを複数作成する必要があります。
- ### L4OPが拡張され、このACLが作成されます##
9500H(config)#ip access-list extended TEST
9500H(config-ext-nacl)#permit tcp 10.0.0.0 0.255.255.255 any gt 150 < - ポート151以上と一致します。

###これは、L4OP ###を使用しない複数のACEに拡張する必要があります。

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....

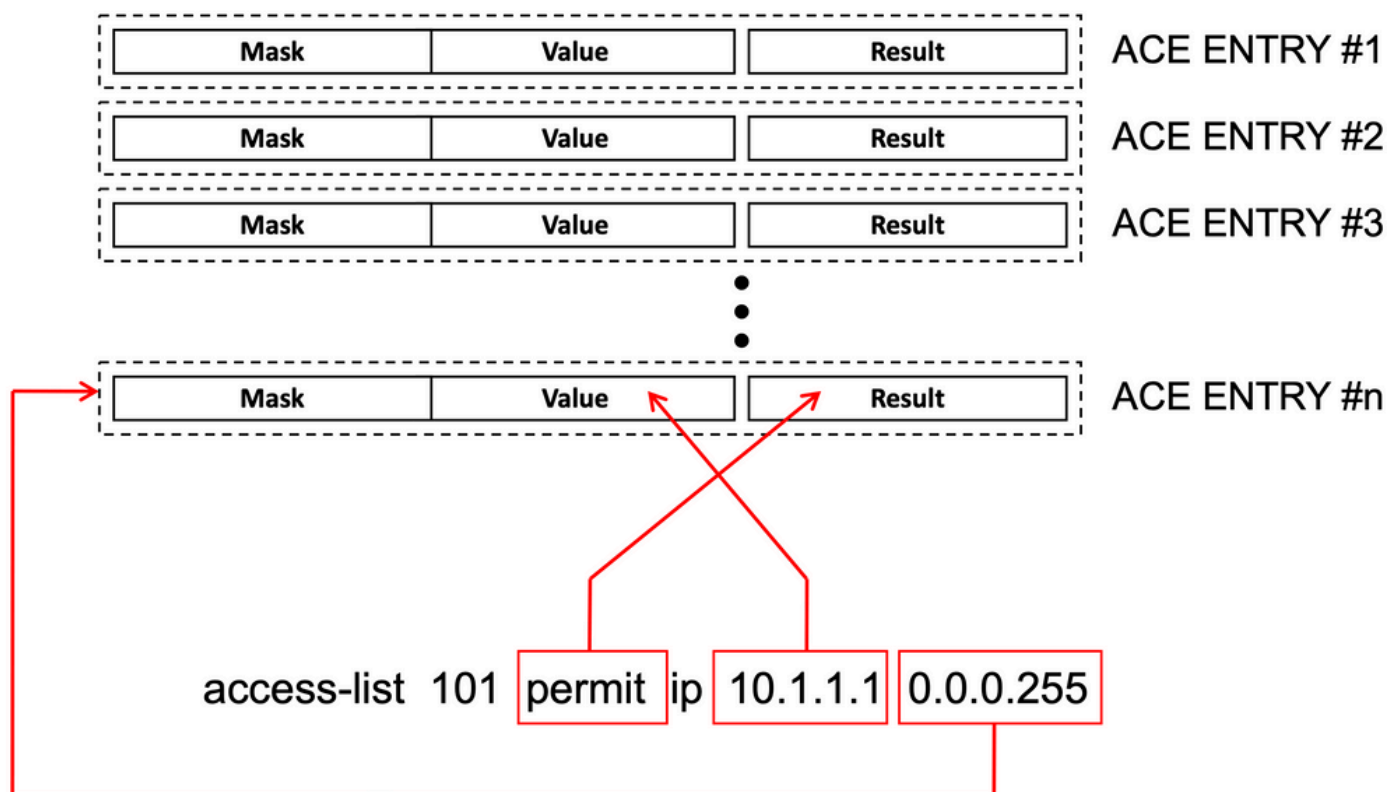
```

TCAMの使用とラベルの共有 :

- 各ACLポリシーは、内部でラベルによって参照されます。
- ACLポリシー (GACL、PAACL、VAACL、RAACLなどのセキュリティACL) が複数のインターフェイスまたはVLANに適用される場合、同じラベルが使用されます。
- 入力ACLと出力ACLでは、異なるラベルスペースが使用されます。
- IPv4、IPv6、およびMAC ACLは、他のラベルスペースを使用します。
- 同じPAACLがインターフェイスAの入力とインターフェイスAの出力に適用されます。TCAMにはPAACLのインスタンスが2つあり、それぞれ入力と出力に固有のラベルがあります。
- 各コアに存在する複数の入力インターフェイスにL4OPと同じPAACLが適用される場合、TCAMにプログラムされた同じPAACLのインスタンスが2つ (各コアに1つ) あります。

VMRの説明

ACEはTCAM内で「VMR」として内部的にプログラムされます (値、マスク、結果とも呼ばれます)。各ACEエントリはVMRとVCUを消費できます。



ACLの拡張性 :

セキュリティACLリソースは、セキュリティACL専用です。これらは他の機能と共有されません

ACL TCAMリ ソース	Cisco Catalyst 9600	Cisco Catalyst 9500	Cisco Catalyst 9400	Cisco Catalyst 9300	Cisco Catalyst 9200			
IPv4エ ントリ	入力 : 12000*	出力: 15000*	C9500:18000*	C9500ハ イパフォ ーマンス 入力 : 12000* 出力 : 15000*	18000*	C9300: 5000	C9300B: 18000	C9300X:8
IPv6エ ントリ	IPv4エントリの 半分		IPv4エントリの半分		IPv4エ ントリ の半分	IPv4エントリの半分		
1つのタ イプの IPv4 ACLエ ントリ は次の 値を超 えるこ とはで きませ ん。	12000		C9500:18000	C9500ハ イパフォ ーマンス : 15000	18000	C9300: 5000	C9300B:18000	C9300X:8000
1つのタ イプの IPv6 ACLエ ントリ は次の 値を超 えるこ とはで	6000		C9500: 9000	C9500ハ イパフォ ーマンス : 7500	9000	2500/9000/4000		

きま せん。						
L4OP/ラ ベル	8	8	8	8		
入力 VCU	192	192	192	192		
出力 VCU	96	96	96	96		

関連情報

- [セキュリティコンフィギュレーションガイド、Cisco IOS XE Amsterdam 17.3.x \(Catalyst 9200スイッチ \)](#)
- [セキュリティ設定ガイド、Cisco IOS XE Amsterdam 17.3.x \(Catalyst 9300スイッチ \)](#)
- [セキュリティ設定ガイド、Cisco IOS XE Amsterdam 17.3.x \(Catalyst 9400スイッチ \)](#)
- [セキュリティ設定ガイド、Cisco IOS XE Amsterdam 17.3.x \(Catalyst 9500スイッチ \)](#)
- [セキュリティコンフィギュレーションガイド、Cisco IOS XE Amsterdam 17.3.x \(Catalyst 9600スイッチ \)](#)
- 『System Management Configuration Guide, Cisco IOS XE Bengaluru 17.4.x』 (Catalyst 9500スイッチ)
- [シスコテクニカルサポートおよびダウンロード](#)

debugコマンドとtraceコマンド

Num	コマンド	備考
1	show platform hardware fed [switch] active fwd-asic drops exceptions asic <0>	ASIC #Nの例外カウンタをダンプします。
2	show platform software fed [switch] active acl	このコマンドは、インターフェイスおよびポリシー情報とともに、ボックス上のすべての設定済みACLに関する情報を出力します。
3	show platform software fed [switch] active acl policy 18	このコマンドでは、ポリシー18に関する情報だけが出力されます。このポリシーIDは、コマンド2から取得できます。

4	show platform software fed [switch] active acl interface intftype pacl	このコマンドは、インターフェイスタイプ (pacl/vacl/racl/gacl/sgaclなど) に基づいてACLに関する情報を出力します。
5	show platform software fed [switch] active acl interface intftype pacl acltype ipv4	このコマンドは、インターフェイスタイプ (pacl/vacl/racl/gacl/sgaclなど) に基づいてACLに関する情報を出力し、プロトコルに基づいてフィルタリングします (ipv4/ipv6/macなど)。
6	show platform software fed [switch] active acl interface intftype pacl acltype ipv4	このコマンドは、インターフェイスに関する情報を出力します。
7	show platform software fed [switch] active acl interface 0x9	このコマンドは、IIF-ID (6からのコマンド) に基づいて、インターフェイスに適用されたACLの短い情報を出力します。
8	show platform software fed [switch] active acl definition	このコマンドは、ボックスで設定され、CGD内に存在するACLに関する情報を出力します。
9	show platform software fed [switch] active acl iifid 0x9	このコマンドは、IIF-IDに基づいて、インターフェイスに適用されたACLの詳細情報を出力します。
10	show platform software fed [switch] active acl usage	このコマンドは、フィーチャタイプに基づいて、各ACLが使用するVMRの数を出力します。
11	show platform software fed [switch] active acl policy intftype pacl vcu	このコマンドは、ポリシー情報と、インターフェイスタイプ (pacl/vacl/racl/gacl/sgaclなど) に基づくVCU情報も提供します。
12	show platform software fed [switch] active acl policy intftype pacl cam	このコマンドは、インターフェイスタイプ (pacl/vacl/racl/gacl/sgaclなど) に基づいて、CAM内のVMRに関するポリシー情報と詳細を提供します。
13	show platform software interface [switch] [active] R0 brief	このコマンドは、ボックス上のインターフェイスに関する詳細情報を表示します。
14	show platform software fed [switch] active port if_id 9	このコマンドは、IIF-IDに基づいてポートの詳細を出力します。

15	show platform software fed [switch] active vlan 30	このコマンドは、VLAN 30に関する詳細を出力します。
16	show platform software fed [switch] active acl cam asic 0	このコマンドは、使用されているASIC 0の完全なACL camを出力します。
17	show platform software fed [switch] active acl counters hardware	このコマンドは、ハードウェアからのすべてのACLカウンタを出力します。
18	show platform hardware fed [switch] active fwd-asic resource tcam table pbr record 0 format 0	PBRセクションのエントリを印刷すると、PBRの代わりにACLやCPPなどの異なるセクションを指定できます。
19	show platform software fed [switch] active punt cpuq [1 2 3 ...]	CPUキューの1つでアクティビティを確認するには、デバッグ用にキューの統計情報をクリアするオプションもあります。
20	show platform software fed [switch] active ifm mappings gpn	IIF-IDとGPNを使用してインターフェイスのマッピングを印刷します。
21	show platform software fed [switch active ifm if-id	インターフェイス設定に関する情報とASICとの関係を印刷します。このコマンドは、ASICとCOREがどのインターフェイス上に存在するかを確認するのに役立ちます。
22	set platform software trace fed [switch] active acl/asic_vmr/asic_vcu/cgacl/sgacl [debug error ...]	FEDの特定の機能のトレースを設定します。
23	request platform software trace rotate all	トレースバッファをクリアしています。
24	show platform software trace message fed [switch] active	FEDのトレースバッファを印刷しています。
25	set platform software trace forwarding-manager [switch] [active] f0 fman [debug error ...]	FMANのトレースを有効にします。
26	show platform software trace message forwarding-manager [switch] [active] f0	FMANのトレースバッファを印刷しています。

27	debug platform software infrastructure punt detail	PUNTのデバッグを設定します。
28	debug ip cef packet all input rate 100	CEFパケットのデバッグがオンになっています。

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。