

ACI 패브릭에서 계약 및 규칙 확인

목차

[소개](#)

[토폴로지](#)

[프로세스 개요](#)

[사용된 계약/조닝 규칙 식별](#)

[하드웨어 프로그래밍 확인](#)

[하드웨어 프로그래밍 문제 해결](#)

[유용한 문제 해결 명령](#)

[문제 해결 정보](#)

[규칙 ID에서 계약 이름 파싱](#)

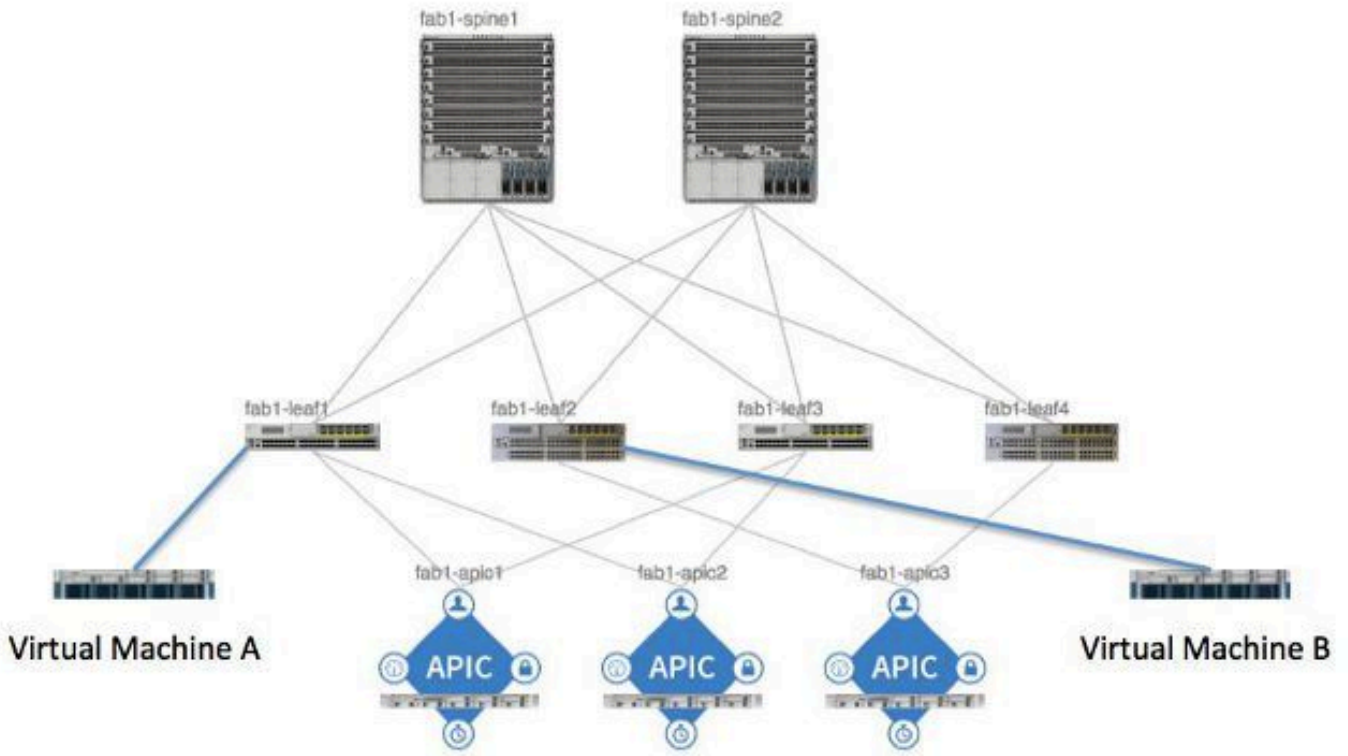
소개

이 문서에서는 ACI(Application Centric Infrastructure) 패브릭에서 계약이 구성되고 올바르게 작동하는지 확인하는 방법에 대해 설명합니다.

토폴로지

이 문서 전반에서 사용되는 예에서 VM(Virtual Machine-A)은 Leaf1에 연결되며, Leaf2에 연결된 VM-B와 통신할 수 있는 계약이 있습니다. 계약은 ICMP(Internet Control Message Protocol) 및 HTTP를 모두 허용합니다.

이 그림에서는 토폴로지를 보여줍니다.



프로세스 개요

이는 계약 및 규칙에 대한 정책 상호 작용 및 흐름입니다.

1. APIC(Application Policy Infrastructure Controller)의 정책 관리자는 스위치의 정책 요소 관리자와 통신합니다.
2. 스위치의 정책 요소 관리자는 스위치의 개체 저장소를 프로그래밍합니다.
3. 스위치의 정책 관리자는 스위치의 ACLQOS(Access Control List Quality of Service) 클라이언트와 통신합니다.
4. ACLQOS 클라이언트는 하드웨어를 프로그래밍합니다.

사용된 계약/조닝 규칙 식별

다음은 두 EPG(End Point Group)에 대한 계약이 추가되기 전에 leaf에서 실행된 show zoning-rule 명령의 예입니다.

```
<#root>
```

```
fab1_leaf1#
```

```
show zoning-rule
```

Rule ID	SrcEPG	DstEPG	FilterID	operSt	Scope	Action
---------	--------	--------	----------	--------	-------	--------

```

=====
4096      0      0      implicit enabled 16777200 deny,log
4097      0      0      implicit enabled 3080192 deny,log
4098      0      0      implicit enabled 2686976 deny,log
4099      0      49154 implicit enabled 2686976 permit
4102      0      0      implicit enabled 2097152 deny,log
4103      0      32771 implicit enabled 2097152 permit
4117     16387 16386 12      enabled 2097152 permit
4116     16386 16387 13      enabled 2097152 permit
4100     16386 49154 default enabled 2097152 permit
4101     49154 16386 default enabled 2097152 permit
4104      0      32770 implicit enabled 2097152 permit
4105     49155 16387 13      enabled 2097152 permit
4112     16387 49155 13      enabled 2097152 permit
4113     49155 16387 12      enabled 2097152 permit
4114     16387 49155 12      enabled 2097152 permit

```

[snip]

이는 계약이 추가된 후 두 EPG가 서로 통신할 수 있도록 출력되는 동일한 명령입니다.

```
<#root>
```

```
fab1_leaf1#
```

```
show zoning-rule
```

```

Rule ID  SrcEPG  DstEPG  FilterID  operSt  Scope  Action
=====
4096      0      0      implicit enabled 16777200 deny,log
4097      0      0      implicit enabled 3080192 deny,log
4098      0      0      implicit enabled 2686976 deny,log
4099      0      49154 implicit enabled 2686976 permit

```

```


4131      49155      32771      7          enabled  2686976  permit


4132      32771      49155      6          enabled  2686976  permit

4102      0           0           implicit  enabled  2097152  deny,log
4103      0           32771      implicit  enabled  2097152  permit
4117      16387      16386      12         enabled  2097152  permit
4116      16386      16387      13         enabled  2097152  permit
4100      16386      49154      default   enabled  2097152  permit
4101      49154      16386      default   enabled  2097152  permit
4104      0           32770      implicit  enabled  2097152  permit
4105      49155      16387      13         enabled  2097152  permit
4112      16387      49155      13         enabled  2097152  permit
4113      49155      16387      12         enabled  2097152  permit
4114      16387      49155      12         enabled  2097152  permit

```

[snip]

 참고: 추가된 새 규칙 ID(4131 및 4132), 필터 ID 7 및 6, 2686976 범위를 확인합니다.

 주의: 이 명령 출력을 사용하면 Lab 시스템에서 검사해야 하는 규칙을 쉽게 찾을 수 있습니다. 그러나 프로덕션 환경에서는 동적 변경이 발생하는 경우 이 작업이 번거로울 수 있습니다.

관심 규칙을 찾기 위해 사용할 수 있는 또 다른 방법은 Visore를 사용하는 것입니다. 컨텍스트 MO(Managed Object)에서 fvCtx에 대한 검색을 수행합니다. 그런 다음 화면에 표시된 대로 특정 컨텍스트 DN(Distinguished Name)을 검색할 수 있습니다.

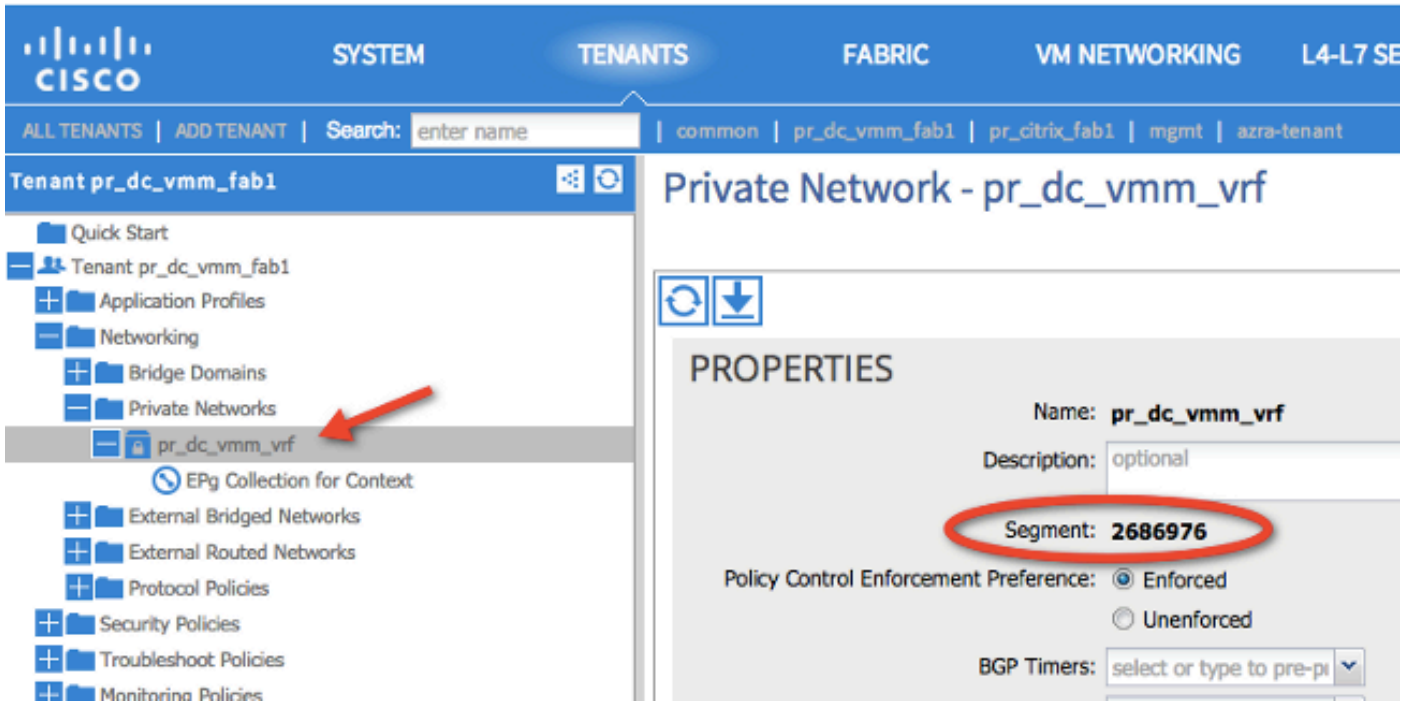


해당 컨텍스트의 범위를 확인합니다. 이 명령을 사용하여 show-zoning-rule 명령 출력에 매핑하면

쿼리해야 하는 규칙을 찾을 수 있습니다.

fvCtx		?
childAction		
descr		
dn	uni/tn-pr_dc_vmm_fab1/ctx-pr_dc_vmm_vrf < > MJ I 50	
knwMcastAct	permit	
lcOwn	local	
modTs	2014-09-03T09:32:36.625-04:00	
monPolDn	uni/tn-common/monepg-default < > MJ I 50	
name	pr_dc_vmm_vrf	
ownerKey		
ownerTag		
pcEntPref	enforced	
pcTag	32770	
scope	2686976	
seg	2686976	
status		
uid	15374	

여기에 표시된 대로 UI(사용자 인터페이스)에서 컨텍스트의 세그먼트 ID/범위를 식별할 수도 있습니다.



이 범위는 show zoning-rules 명령 출력에 표시된 것과 일치합니다.

4098	0	0	rule (4098) DN	implicit	enabled	2686976	deny, log
4099	0	49154	49154: 81553	implicit	enabled	2686976	permit
4131	49155	32771	(4099) DN	(sys/actrl/scope	enabled	2686976	permit
4132	32771	49155	ress: 0	6	enabled	2686976	permit

범위 ID 정보가 있고 규칙 및 필터 ID를 식별한 경우 다음 명령을 사용하여 새 필터에 도달했는지 (EPG 간의 암시적 거부 메시지는 아님) 확인할 수 있습니다. 암시적 거부 메시지는 기본적으로 EPG가 통신할 수 없도록 포함됩니다.

이 명령 출력에서 Leaf1, Filter-6(f-6)이 증가하고 있습니다.

```
<#root>
```

```
fab1_leaf1#
```

```
show system internal policy-mgr stats | grep 2686976
```

```
Rule (4098) DN (sys/actrl/scope-2686976/rule-2686976-s-any-d-any-f-implicit)
  Ingress: 0, Egress: 81553
```

```
Rule (4099) DN (sys/actrl/scope-2686976/rule-2686976-s-any-d-49154-f-implicit)
  Ingress: 0, Egress: 0
```

```
Rule (4131) DN (sys/actrl/scope-2686976/rule-2686976-s-49155-d-32771-f-7)
  Ingress: 0, Egress: 0
```

```
Rule (4132) DN (sys/actrl/scope-2686976/rule-2686976-s-32771-d-49155-f-6)
  Ingress: 1440, Egress: 0
```

<#root>

fab1_leaf1#

```
show system internal policy-mgr stats | grep 2686976
```

```
Rule (4098) DN (sys/actrl/scope-2686976/rule-2686976-s-any-d-any-f-implicit)
  Ingress: 0, Egress: 81553
```

```
Rule (4099) DN (sys/actrl/scope-2686976/rule-2686976-s-any-d-49154-f-implicit)
  Ingress: 0, Egress: 0
```

```
Rule (4131) DN (sys/actrl/scope-2686976/rule-2686976-s-49155-d-32771-f-7)
  Ingress: 0, Egress: 0
```

```
Rule (4132) DN (sys/actrl/scope-2686976/rule-2686976-s-32771-d-49155-f-6)
```

```
  Ingress: 1470, Egress: 0
```

이 명령 출력에서 Leaf2, Filter-7(f-7)이 증가하고 있습니다.

<#root>

fab1_leaf2#

```
show system internal policy-mgr stats | grep 268697
```

```
Rule (4098) DN (sys/actrl/scope-2686976/rule-2686976-s-any-d-any-f-implicit)
  Ingress: 0, Egress: 80257
```

```
Rule (4099) DN (sys/actrl/scope-2686976/rule-2686976-s-any-d-49153-f-implicit)
  Ingress: 0, Egress: 0
```

```
Rule (4117) DN (sys/actrl/scope-2686976/rule-2686976-s-32771-d-49155-f-6)
  Ingress: 0, Egress: 0
```

```
Rule (4118) DN (sys/actrl/scope-2686976/rule-2686976-s-49155-d-32771-f-7)
  Ingress: 2481, Egress: 0
```

<#root>

fab1_leaf2#

```
show system internal policy-mgr stats | grep 268697
```


```
Rule (4098) DN (sys/actrl/scope-2686976/rule-2686976-s-any-d-any-f-implicit)
  Ingress: 0, Egress: 80257
```

```
Rule (4099) DN (sys/actrl/scope-2686976/rule-2686976-s-any-d-49153-f-implicit)
```

Ingress: 0, Egress: 0

Rule (4117) DN (sys/actrl/scope-2686976/rule-2686976-s-32771-d-49155-f-6)
Ingress: 0, Egress: 0

Rule (4118) DN (sys/actrl/scope-2686976/rule-2686976-s-49155-d-32771-f-7)
Ingress: 2511, Egress: 0

 **팁:** 이 문제를 더 자세히 해결하려면 범위, 규칙 ID, 대상, 소스 pcTags 및 필터를 알아야 합니다. 또한 규칙 ID가 있는 EPG에 대한 지식이 있으면 유용합니다.

다음 그림과 같이 moquery 명령을 통해 DN 이름 fvAEPg와 grep를 사용하여 특정 pcTag에 대한 검색을 수행할 수 있습니다.

<#root>

admin@RTP_Apic1:~>

```
moquery -c fvAEPg | grep 49155 -B 5
```

dn : uni/tn-Prod/ap-commercespace/

epg-Web

lcOwn : local

matchT : AtleastOne

modTs : 2014-10-16T01:27:35.355-04:00

monPolDn : uni/tn-common/monepg-default

pcTag : 49155

다음 그림과 같이 filter 옵션을 moquery 명령과 함께 사용할 수도 있습니다.

<#root>

admin@RTP_Apic1:~>

```
moquery -c fvAEPg -f 'fv.AEPg.pcTag=="49155"'
```

Total Objects shown: 1

fv.AEPg

name : Web

childAction :

configIssues :

configSt : applied

descr :

dn : uni/tn-Prod/ap-commercespace/

epg-Web

lcOwn : local
matchT : AtleastOne
modTs : 2014-10-16T01:27:35.355-04:00
monPolDn : uni/tn-common/monepg-default

pcTag : 49155


prio : unspecified
rn : epg-Web
scope : 2523136
status :
triggerSt : triggerable
uid : 15374


하드웨어 프로그래밍 확인


이제 규칙에 대한 하드웨어 항목을 확인할 수 있습니다. 하드웨어 정보를 보려면 `show platform internal ns table mth_lux_slvz_DHS_SecurityGroupStatTable_memif_data ingress` 명령(vsh_lc 명령)을 입력합니다.

```
module-1# show platform internal ns table mth_lux_slvz_DHS_SecurityGroupStatTable_memif_data ingress
error opening file
: No such file or direct
Last login: Fri Sep  5 1
=====
[Restored]
TABLE INSTANCE : 0
=====
ENTRY[000010] = pkt_cnt=0x5176e
ENTRY[000011] = pkt_cnt=0x7d95
ENTRY[000014] = pkt_cnt=0x9d414
ENTRY[000016] = pkt_cnt=0x15208a
ENTRY[000017] = pkt_cnt=0x2975ce
ENTRY[000018] = pkt_cnt=0x662b
ENTRY[000021] = pkt_cnt=0x329f
ENTRY[000023] = pkt_cnt=0x40
ENTRY[000024] = pkt_cnt=0x21bf
ENTRY[000026] = pkt_cnt=0x556f0
ENTRY[000029] = pkt_cnt=0x5d7e2
ENTRY[000041] = pkt_cnt=0x6360
ENTRY[000050] = pkt_cnt=0x2a05
ENTRY[000052] = pkt_cnt=0x5ec
ENTRY[000054] = pkt_cnt=0xdfd
ENTRY[000055] = pkt_cnt=0xd
ENTRY[000068] = pkt_cnt=0xdac
ENTRY[000072] = pkt_cnt=0x91
ENTRY[000077] = pkt_cnt=0x35b
module-1# show platform internal ns table mth_lux_slvz_DHS_SecurityGroupStatTable_memif_data ingress
error opening file
: No such file or directory
=====
TABLE INSTANCE : 0
=====
ENTRY[000010] = pkt_cnt=0x517cf
ENTRY[000011] = pkt_cnt=0x7d9f
ENTRY[000014] = pkt_cnt=0x9d494
ENTRY[000016] = pkt_cnt=0x152262
ENTRY[000017] = pkt_cnt=0x29799e5
ENTRY[000018] = pkt_cnt=0x6631
ENTRY[000021] = pkt_cnt=0x329f
ENTRY[000023] = pkt_cnt=0x40
ENTRY[000024] = pkt_cnt=0x21c6
ENTRY[000026] = pkt_cnt=0x55771
ENTRY[000029] = pkt_cnt=0x5d7e2
ENTRY[000041] = pkt_cnt=0x64e0
ENTRY[000050] = pkt_cnt=0x2a05
ENTRY[000052] = pkt_cnt=0x5ec
ENTRY[000054] = pkt_cnt=0xdfd
ENTRY[000055] = pkt_cnt=0xd
ENTRY[000068] = pkt_cnt=0xdb8
ENTRY[000072] = pkt_cnt=0x92
ENTRY[000077] = pkt_cnt=0x35b
```

이 예에서는 하드웨어 항목 41(항목 [000041])이 증가하고 있습니다.

 참고: 표시된 이전 명령은 Northstar ASIC에 사용됩니다. Donner 또는 Donner+에 사용되는 명령은 show platform internal ns table mth_lush_slvy_DHS_SecurityGroupStatTable_memif_data입니다.

 참고: 이 명령은 프로덕션 환경에서는 실용적이지 않지만 이 섹션에 설명된 다른 명령을 대신

 사용할 수 있습니다.

규칙(4132) 및 범위(268976)를 기억하십시오.

4098	0	0	implicit	enabled	2686976	deny, log
4099	0	49154	implicit	enabled	2686976	permit
4131	49155	32771	7	enabled	2686976	permit
4132	32771	49155	6	enabled	2686976	permit

TCAM(Ternary Content-Addressable Memory) 하드웨어 인덱스 항목 매핑에 대한 규칙 ID를 결정하고 규칙 ID 및/또는 필터 ID를 기준으로 필터링하려면 다음 명령을 입력합니다.

<#root>

module-1#

show system internal aclqos zoning-rules

[snip]

=====
Rule ID: 4131 Scope 4 Src EPG: 49155 Dst EPG: 32771 Filter 7

Curr TCAM resource:

=====
unit_id: 0
=== Region priority: 771 (rule prio: 3 entry: 3)===
sw_index = 62 |

hw_index = 40

=== Region priority: 772 (rule prio: 3 entry: 4)===
sw_index = 63 |

hw_index = 45

=====
Rule ID: 4132 Scope 4 Src EPG: 32771 Dst EPG: 49155 Filter 6

Curr TCAM resource:

=====
unit_id: 0
=== Region priority: 771 (rule prio: 3 entry: 3)===
sw_index = 66 |

hw_index = 41

=== Region priority: 771 (rule prio: 3 entry: 3)===
sw_index = 67 |

hw_index = 42

[snip]

이 예에서 소스 및 목적지 EPG의 조합은 32771=0x8003, 49155=0xC003입니다. 따라서 규칙 ID(4131 및 4132) 및 필터 ID(6 및 7)와 일치하는 이러한 소스 및 대상 클래스에 대한 모든 TCAM 항목을 고려할 수 있습니다.

이 예에서는 이러한 TCAM 항목 중 일부가 덤프됩니다. 참고로, 이러한 EPG에 대해 ping 및 웹 트래픽을 허용하는 계약 컨피그레이션은 다음과 같습니다.

The screenshot shows the configuration of a filter named 'pr_dc_vmm_fab1'. The filter has two entries: 'ping' and 'web'. The 'ping' entry has IP protocol 'icmp' and the 'web' entry has IP protocol 'tcp'. The destination port for 'web' is 'http'.

NAME	ETHERTYPE	ARP FLAG	IP PROTOCOL	ALLOW FRAGMENT	SOURCE PORT / RANGE		DESTINATION PORT / RANGE	
					FROM	TO	FROM	TO
ping	IP		icmp	False				
web	IP		tcp	False	unspecified	unspecified	http	http

<#root>

module-1#

```
show platform internal ns table mth_lux_slvz_DHS_SecurityGroupKeyTable0
```

```
_memif_data 41
```

```
=====
TABLE INSTANCE : 0
=====
```

```
ENTRY[000041] =
    sg_label=0x4
    sclass=0x8003
    dclass=0xc003
    prot=0x1
(IP Protocol 0x01 = ICMP)
```

참고: 표시된 이전 명령은 Northstar ASIC에 사용됩니다. Doner 또는 Doner+에 사용되는 명령은 show platform internal ns table mth_lush_slvq_DHS_SecurityGroupKeyTable0_memif_data입니다.

Decimal	Keyword	Protocol	IPv6 Extension Header	
0	HOPOPT	IPv6 Hop-by-Hop Option	Y	[RFC2460]
1	ICMP	Internet Control Message		[RFC792]
2	IGMP	Internet Group Management		[RFC1112]

<#root>

```

sup_tx_mask=0x1
  src_policy_incomplete_mask=0x1
  dst_policy_incomplete_mask=0x1
  class_eq_mask=0x1
  aclass_mask=0x1ff
  port_dir_mask=0x1
  dport_mask=0xffff
  sport_mask=0xffff
  tcpflags_mask=0xff
  ip_opt_mask=0x1
  ipv6_route_mask=0x1
  ip_fragment_mask=0x1
  ip_frag_offset0_mask=0x1
  ip_frag_offset1_mask=0x1
  ip_mf_mask=0x1
  t4_partial_mask=0x1
  dst_local_mask=0x1
  routeable_mask=0x1
  spare_mask=0x7ff
  v4addr_key_mask=0x1
  v6addr_key_mask=0x1
  valid=0x1

```

module-1#

```
show platform internal ns table mth_lux_slvz_DHS_SecurityGroupKeyTable0
```

_memif_data 42

=====

TABLE INSTANCE : 0

=====

ENTRY[000042] =

sg_label=0x4

sclass=0x8003

dclass=0xc003

prot=0x6

<--

dport=0x50

<--

Decimal ⌵	Keyword ⌵	Protocol ⌵	IPv6 Extension Header ⌵	
0	HOPOPT	IPv6 Hop-by-Hop Option	Y	[RFC2460]
1	ICMP	Internet Control Message		[RFC792]
2	IGMP	Internet Group Management		[RFC1112]
3	GGP	Gateway-to-Gateway		[RFC823]
4	IPv4	IPv4 encapsulation		[RFC2003]
5	ST	Stream		[RFC1190] [RFC1819]
6	TCP	Transmission Control		[RFC793]
7	CBT	CBT		[Tony Ballardie]

Port ↕	TCP ↕	UDP ▲	Description
0	TCP		Programming technique for specifying system-allocated (dynamic) ports ^[3]
21	TCP		FTP control (command)
25	TCP		Simple Mail Transfer Protocol (SMTP)—used for e-mail routing between mail servers
43	TCP		WHOIS protocol
57	TCP		Mail Transfer Protocol (RFC 780 ↗)
70	TCP		Gopher protocol
71	TCP		NETRJS protocol
72	TCP		NETRJS protocol
73	TCP		NETRJS protocol
74	TCP		NETRJS protocol
79	TCP		Finger protocol
80	TCP		Hypertext Transfer Protocol (HTTP) ^[12]
81	TCP		Terminology: Onion routing

sup_tx_mask=0x1

src_policy_incomplete_mask=0x1

dst_policy_incomplete_mask=0x1

class_eq_mask=0x1

aclass_mask=0x1ff

port_dir_mask=0x1

sport_mask=0xffff

tcpflags_mask=0xff

ip_opt_mask=0x1

ipv6_route_mask=0x1

ip_fragment_mask=0x1


ip_frag_offset0_mask=0x1

ip_frag_offset1_mask=0x1

ip_mf_mask=0x1

l4_partial_mask=0x1

dst_local_mask=0x1

 **팁:** 각 TCAM 항목을 동일한 방법으로 확인할 수 있습니다.

하드웨어 프로그래밍 문제 해결

이 섹션에서는 몇 가지 유용한 문제 해결 명령 및 팁을 제공합니다.

유용한 문제 해결 명령

문제가 발생할 때 리프 정책 관리자 오류를 찾는 데 사용할 수 있는 몇 가지 유용한 명령은 다음과 같습니다.

```
<#root>
```

```
fab1_leaf1#
```

```
show system internal policy-mgr event-history errors
```

```
1) Event:E_DEBUG, length:84, at 6132 usecs after Mon Sep 8 13:15:56 2014
```

```
[103] policy_mgr_handle_ctx_mrules(779): ERROR: Failed to process prio(1537):  
(null)
```

```
2) Event:E_DEBUG, length:141, at 6105 usecs after Mon Sep 8 13:15:56 2014
```

```
[103] policy_mgr_process_mruler_prio_aces(646): ERROR: Failed to insert iptables  
rule for rule(4120) , fentry(5_0) with priority(1537): (null)
```

```
[snip]
```

```
fab1_leaf1#
```

```
show system internal policy-mgr event-history trace
```

```
[1409945922.23737] policy_mgr_ppf_hdl_close_state:562: Got close state callback
```

```
[1409945922.23696] policy_mgr_ppf_rdy_ntf_fun:239: StatStoreEnd returned: 0x0(SU  
CCESS)
```

```
[1409945922.23502] policy_mgr_ppf_rdy_ntf_fun:208: ppf ready notification: sess_  
id: (0xFF0104B400005B51)
```

```
[1409945922.23475] policy_mgr_ppf_rdy_ntf_fun:205: Got ready notification callba  
ck with statustype (4)
```

```
[1409945921.983476] policy_mgr_gwrap_handler:992: Dropped...now purging it...
```


[1409945921.982882] policy_mgr_ppf_goto_state_fun:481: Sess id (0xFF0104B400005B

[snip]

module-1#

show system internal aclqos event-history trace

T [Fri Sep 5 13:18:24.863283] ===== Session End =====

T [Fri Sep 5 13:18:24.862924] Commit phase: Time taken 0.62 ms, usr 0.00 ms,
sys 0.00 ms

T [Fri Sep 5 13:18:24.862302] ppf session [0xff0104b410000087] commit ... npi
nst 1

T [Fri Sep 5 13:18:24.861421] Verify phase: Time taken 0.77 ms, usr 0.00 ms,
sys 0.00 ms

T [Fri Sep 5 13:18:24.860615] ===== Session Begin =====

T [Fri Sep 5 13:18:24.830472] ===== Session End =====

T [Fri Sep 5 13:18:24.830062] Commit phase: Time taken 0.98 ms, usr 0.00 ms,
sys 0.00 ms

T [Fri Sep 5 13:18:24.829085] ppf session [0xff0104b410000086] commit ... npi
nst 1


T [Fri Sep 5 13:18:24.827685] Verify phase: Time taken 2.04 ms, usr 0.00 ms,
sys 0.00 ms

T [Fri Sep 5 13:18:24.825388] ===== Session Begin =====

T [Fri Sep 5 12:32:51.364225] ===== Session End =====

T [Fri Sep 5 12:32:51.363748] Commit phase: Time taken 0.64 ms, usr 0.00 ms,

[snip]

 **팁:** 일부 파일은 크기가 크기 때문에 부트플래시로 보내고 편집기에서 검사하는 것이 더 쉽습니다.

<#root>

module-1#

show system internal aclqos ?

asic Asic information
brcm Broadcam information
database Database
event-history Show various event logs of ACLQOS
mem-stats Show memory allocation statistics of ACLQOS
prefix External EPG prefixes
qos QoS related information
range-resource Zoning rules L4 destination port range resources
regions Security TCAM priority regions
span SPAN related information
zoning-rules Show zoning rules

module-1#

show system internal aclqos event-history ?

errors Show error logs of ACLQOS

msgs Show various message logs of ACLQOS
ppf Show ppf logs of ACLQOS
ppf-parse Show ppf-parse logs of ACLQOS
prefix Show prefix logs of ACLQOS
qos Show qos logs of ACLQOS
qos-detail Show detailed qos logs of ACLQOS
span Show span logs of ACLQOS
span-detail Show detailed span logs of ACLQOS

trace Show trace logs of ACLQOS

trace-detail Show detailed trace logs of ACLQOS

zoning-rules Show detailed logs of ACLQOS

문제 해결 정보

다음은 몇 가지 유용한 문제 해결 팁입니다.

- TCAM 소모 문제가 발생하는 경우 UI 또는 CLI에서 해당 규칙과 관련된 오류를 확인합니다. 이 결함은 다음과 같이 보고할 수 있습니다.

```
<#root>
```

```
Fault F1203 - Rule failed due to hardware programming error.
```

하나의 규칙이 ASIC(Application-Specific Integrated Circuit)에서 둘 이상의 TCAM 항목을 사용할 수 있습니다. ASIC의 항목 수를 보려면 다음 명령을 입력합니다.

```
<#root>
```

```
fab1-leaf1#
```

```
vsh_lc
```

```
module-1#
```

```
show platform internal ns table-health
```

```
VLAN STATE curr usage: 0 - size: 4096  
QQ curr usage: 0 - size: 16384  
SEG STATE curr usage: 0 - size: 4096  
SRC TEP curr usage: 0 - size: 4096  
POLICY KEY curr usage: 0 - size: 1  
SRC VP curr usage: 0 - size: 4096
```

```
SEC GRP curr usage: 43 - size: 4096
```



참고: 이 예에서는 43개의 항목이 있습니다. 이 사용량은 eqptCapacity 클래스의 APIC에도 보고됩니다.

- 일치하는 항목이 여러 개인 경우 TCAM 조회는 더 낮은 hw-index를 반환합니다. 인덱스를 확인하려면 다음 명령을 입력합니다.

```
<#root>
```

```
show system internal aclqos zoning-rule
```

트러블슈팅할 때 any-any-implicit 규칙에 의해 발생한 삭제를 관찰할 수 있습니다. 이 규칙은

항상 맨 아래에 있으며, 이는 규칙이 없기 때문에 패킷이 삭제됨을 의미합니다. 이는 컨피그레이션이 잘못되었거나 정책 요소 관리자가 예상대로 프로그래밍하지 않았기 때문입니다.

- pcTags는 로컬 또는 글로벌 범위를 가질 수 있습니다.

시스템 예약 pcTag - 이 pcTag는 시스템 내부 규칙(1~15)에 사용됩니다.

전역 범위의 pcTag - 이 pcTag는 공유 서비스(16-16385)에 사용됩니다.

로컬 범위 pcTag - 이 pcTag는 VRF당 로컬로 사용됩니다(16386~65535 범위).

문제를 해결할 때 값의 길이를 빠르게 살펴보면 해당 범위가 표시됩니다.

규칙 ID에서 계약 이름 파생

문제 해결 사례에서 엔지니어가 영역 지정 규칙을 검토하는 경우가 많습니다. 경우에 따라 EPG/pcTag에 계약이 많아 문제를 해결하는 데 번거로울 수 있습니다. 이 섹션에서는 스위치 CLI에 표시된 규칙 ID에서 EPG/pcTags 간에 사용되는 계약의 이름을 확인하는 방법을 살펴봅니다.

시작하려면 구체적인 계약/규칙 객체 `actrlRule`을 쿼리하려면 원하는 경우 속성별로 검색 범위를 좁힙니다. `id value: rule-d`

올바른 규칙이 발견되면 DN의 녹색 화살표를 클릭하여 `actrlRule` 객체 하위 항목을 확인합니다. 이들은 우리의 답이 있는 곳이다.

actrlRule ?	
action	permit
actrlCfgFailedBmp	
actrlCfgFailedTs	00:00:00:00.000
actrlCfgState	0
childAction	
dPcTag	16388 ←
descr	
direction	uni-dir
dn	topology/pod-1/node-101/sys/actrl/scope-2719746/rule-2719746-s-49164-d-16388-f-38 < > 📊 ⚠️ 🛡️
fltId	38 ←
id	4143 ←
lcOwn	local
markDscp	unspecified
modTs	2016-01-08T19:44:02.267+00:00
monPolDn	uni/tn-common/monepg-default < > 📊 ⚠️ 🛡️
name	
operSt	enabled
operStQual	
prio	fully_qual
qosGrp	unspecified
sPcTag	49164 ←
scopeId	2719746 ←
status	
type	tenant

여기서 하위 객체는 actrlRsToEpgConn입니다. 일반적으로 EPG마다 하나씩 두 개가 있을 수 있습니다. 이 객체의 DN은 계약이 적용되는 두 EPG와 방향(공급자 또는 소비자), 그리고 가장 중요한 계약 객체 이름을 표시합니다.

actrlRsToEpgConn	
childAction	
dn	topology/pod-1/node-101/sys/actrl/scope-2719746/rule-2719746-s-49164-d-16388-f-38/rstoEpgConn-[cdef-[uni/tn-dpita-tenant/brc-dpita-ssh]/epgCont-[uni/tn-dpita-tenant/ap-dpita-AP/epg-dpita-EPG1]/fr-[uni/tn-dpita-tenant/brc-dpita-ssh/dirass-prov-[uni/tn-dpita-tenant/ap-dpita-AP/epg-dpita-EPG1]-any-no]/to-[uni/tn-dpita-tenant/brc-dpita-ssh/dirass-cons-[uni/tn-dpita-tenant/ap-dpita-AP/epg-dpita-EPG2]-any-no]] < >   
forceResolve	no
leOwn	local
modTs	2016-01-08T19:44:02.267+00:00
rType	mo
state	unformed
stateQual	none
status	
tCl	vzToEPg
tDn	cdef-[uni/tn-dpita-tenant/brc-dpita-ssh]/epgCont-[uni/tn-dpita-tenant/ap-dpita-AP/epg-dpita-EPG1]/fr-[uni/tn-dpita-tenant/brc-dpita-ssh/dirass-prov-[uni/tn-dpita-tenant/ap-dpita-AP/epg-dpita-EPG1]-any-no]/to-[uni/tn-dpita-tenant/brc-dpita-ssh/dirass-cons-[uni/tn-dpita-tenant/ap-dpita-AP/epg-dpita-EPG2]-any-no]] < >   
tType	mo

강조 표시된 대로 이 경우 계약 이름은 brc-dpita-ssh입니다.

필요한 경우 vzBrCP를 쿼리하여 올바른 계약을 찾습니다.

vzBrCP

?

childAction	
configIssues	
descr	
dn	uni/tn-dpita-tenant/brc-dpita-ssh < > ! H
lcOwn	local
modTs	2015-06-25T16:21:10.003+00:00
monPolDn	uni/tn-common/monepg-default < > ! H
name	dpita-ssh
ownerKey	
ownerTag	
prio	unspecified
reevaluateAll	no
scope	context
status	
uid	15374

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.