

Controleer contracten en regels in het ACI-fabric

Inhoud

[Inleiding](#)

[Topologie](#)

[Overzicht van processen](#)

[Identificeer de gebruikte contract/Zoningregel](#)

[Controleer de hardwareprogramma's](#)

[Probleemoplossing voor hardwareprogrammeerproblemen](#)

[Opdrachten voor nuttige probleemoplossing](#)

[Tips bij het oplossen van problemen](#)

[Contractnaam aan regel-id afleiden](#)

Inleiding

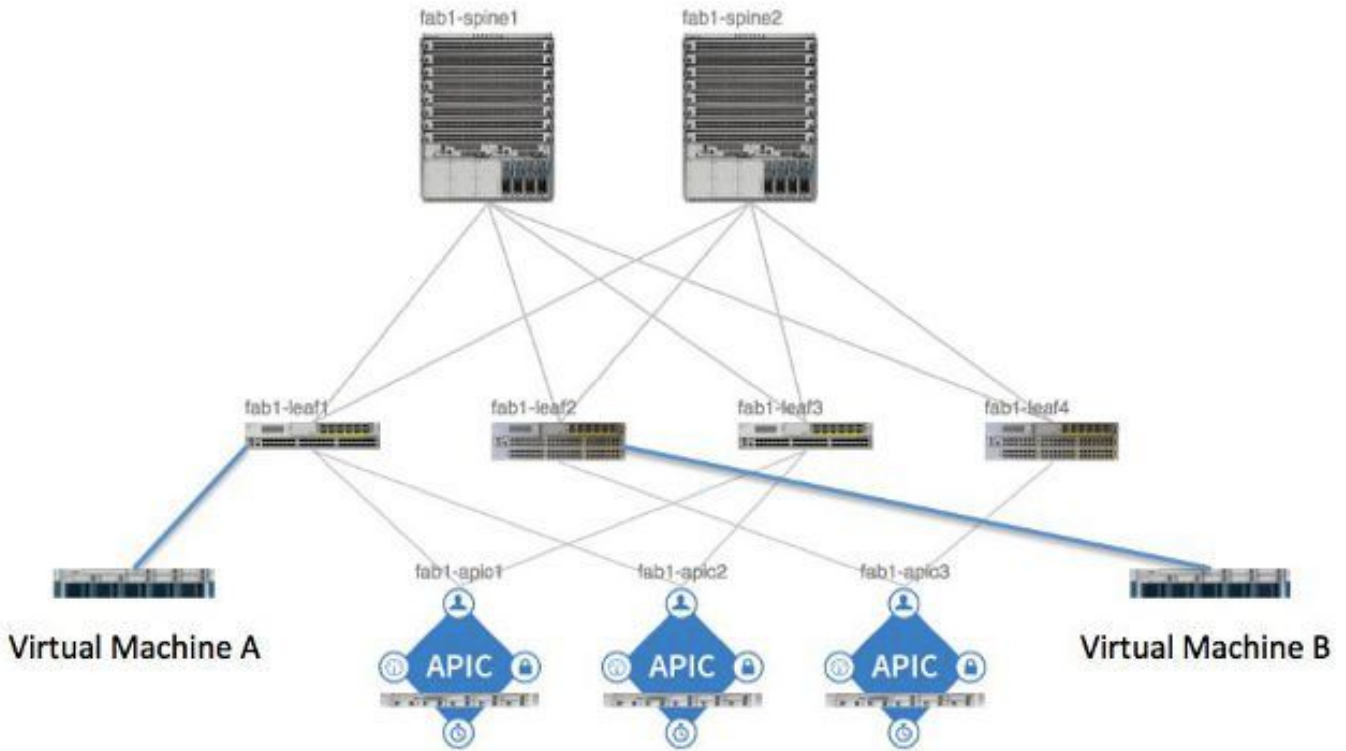
In dit document wordt beschreven hoe u kunt controleren of contracten zijn geconfigureerd en hoe u zich naar behoren gedraagt in het weefsel van de Application Centric Infrastructure (ACI).

Opmerking: De verificatie van de logische en concrete modellen en de hardwareprogrammering wordt in dit document beschreven.

Topologie

In het voorbeeld dat in dit document wordt gebruikt, is Virtual Machine-A (VM) aan Leaf1 gekoppeld en is er een contract dat het in staat stelt te communiceren met VM-B, dat aan Leaf2 is gekoppeld. Het contract staat zowel Internet Control Message Protocol (ICMP) als HTTP toe.

Dit beeld illustreert de topologie:



Overzicht van processen

Dit is de beleidswisselwerking en -stroom voor contracten en regels:

1. De Policy Manager op de Application Policy Infrastructure Controller (APIC) communiceert met de Policy Element Manager op de switch.
2. De manager van het Gebiedsdeel op de switch programma's de Opslag van Objecten op de switch.
3. Policy Manager op de switch communiceert met de ACLQOS-client (Access Control List Quality of Service) op de switch.
4. De ACLQOS-client programmeert de hardware.

Identificeer de gebruikte contract/Zoningregel

Hier is een voorbeeld van de opdrachtoutput van de **zoning-regel** van het blad, voordat het contract wordt toegevoegd voor de twee End Point Group (EPG's).

```
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]

Dit is dezelfde opdrachtoutput nadat het contract is toegevoegd, zodat de twee EPG's met elkaar kunnen communiceren:

```
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]

Opmerking: Let op de nieuwe regel-ID's (**4131** en **4132**) die zijn toegevoegd, de filter-ID's van **7** en **6**, en het toepassingsgebied van **2686976**.

Voorzichtig: Deze opdrachtoutput stelt u in staat om gemakkelijk de regels te vinden die u in een labsysteem moet onderzoeken; dit kan echter omslachtig zijn in een productieomgeving met de dynamische veranderingen die zich voordoen .

Een andere methode die je kunt gebruiken om de regels van belang te vinden, is *Visore* gebruiken. Voer een zoekopdracht uit op de context Managed Object (MO) voor **fvCtx**. U kunt dan op dat scherm zoeken naar uw specifieke context Naam (DN), zoals hier getoond:



Houd rekening met de reikwijdte van deze context. U kunt dit gebruiken om in kaart te brengen naar de opdrachtoutput **van de show-zoning-regel**, zodat u de regels kunt vinden die u moet vragen:

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

U kunt het segment ID/scope voor de context ook identificeren vanuit de gebruikersinterface (UI), zoals hier wordt getoond:

The screenshot shows the Cisco GUI for configuring a Private Network. The breadcrumb navigation indicates the path: SYSTEM > TENANTS > FABRIC > VM NETWORKING > L4-L7 SERVICES. The current page is titled "Private Network - pr_dc_vmm_vrf". In the left-hand navigation pane, the "Private Networks" folder is expanded, and the specific network "pr_dc_vmm_vrf" is selected, indicated by a red arrow. The main content area displays the "PROPERTIES" for this network, including fields for Name, Description, Segment, Policy Control Enforcement Preference, and BGP Timers. The "Segment" field is highlighted with a red circle and contains the value "2686976".

Dit bereik komt overeen met dat in de opdrachtoutput van **showzoning-regels** wordt getoond:

4098	0	0	rule (4098) DN (4098) /scope	implicit	enabled	2686976	deny, log
4099	0	49154	rule (4099) DN (4099) /scope	implicit	enabled	2686976	permit
4131	49155	32771	rule (4131) DN (4131) /scope	implicit	enabled	2686976	permit
4132	32771	49155	rule (4132) DN (4132) /scope	implicit	enabled	2686976	permit

Zodra u de grootte-ID-informatie hebt en u de regel- en filterID's identificeert, kunt u de volgende opdracht gebruiken om te controleren of u de nieuwe filters aanslaat (en niet de *impliciete ontkent*

berichten tussen de EPG's). Het impliciete ontkeningsbericht is opgenomen zodat de EPG's per default niet kunnen communiceren.

Opmerking in deze opdrachtoutput dat Leaf1, Filter-6 (f-6) groter is:

```
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
```

```
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
```

Opmerking in deze opdrachtoutput dat Leaf2, Filter-7 (f-7) toeneemt:

```
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
```

```
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

Tip: Kennis van het bereik, regel-ID, bestemming en bron-pc-tags, en filter is belangrijk met pogingen om deze kwestie verder op te lossen. Het is ook nuttig kennis te hebben van de EPG's waartussen de regel-ID bestaat.

U kunt een zoekopdracht op de MO uitvoeren met de DN-naam **fvAEPg** en *grep* voor de specifieke pcTag via de **moquery** opdracht, zoals hieronder wordt getoond:

```
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
```

U kunt de filteroptie ook gebruiken in combinatie met de opdracht **Moquery**, zoals hieronder wordt getoond:

```
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
```

Controleer de hardwareprogramma's

U kunt nu de hardwareinvoer voor de regel controleren. Om de hardwareinformatie te bekijken, voer de [show platform interne ns tabel mth_lux_slvz_DHS_SecurityGroupStatTable_memif_data ingress](#) opdracht in (dit is een **vsh_lc** opdracht):

```
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
```

In dit voorbeeld stijgt hardwareinvoer 41 (INTRY [000041]).

Opmerking: De vorige opdracht in blauw wordt gebruikt voor de Northstar ASIC. De opdracht die wordt gebruikt voor Donner of Donner+ is tonen platform interne ns tabel mth_luxh_slvy_DHS_SecurityGroupStatTable_memif_data.

Opmerking: Het gebruik van deze opdracht is niet praktisch in een productieomgeving, maar u kunt de andere opdrachten gebruiken die in deze sectie worden beschreven.

Denk aan de regel (4132) en het toepassingsgebied (268976):

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

Voer deze opdracht in om de regelid-ID te bepalen voor de Ternary Content-Adressable Memory (TCAM) hardware-index mapping, en een filter op basis van de regel-ID en/of filter-ID:

```
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
```

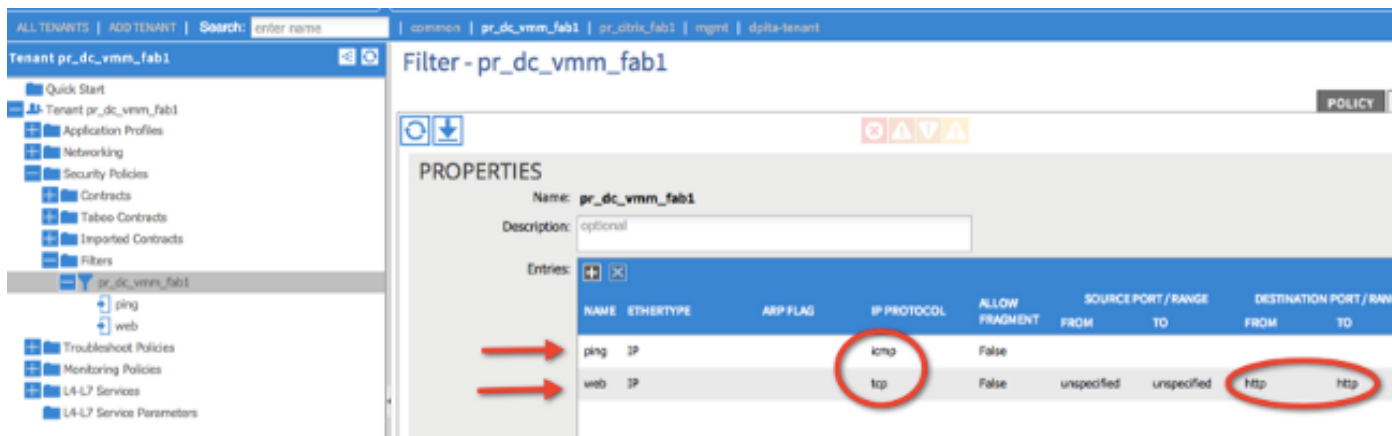
Curr TCAM resource:

```
=====  
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]

Bijvoorbeeld, de bron en bestemming EPG-combinatie van rente is **32771=0x8003**, **49155=0xC003**. Daarom dient u alle TCAM-items voor deze bron- en doelklassen in aanmerking te nemen die overeenkomen met de regel-IDs (41331 en 4132) en filter-ID's (6 en 7).

In dit voorbeeld worden een aantal van deze TCAM-posten gedumpt. Ter referentie, hier is de contractconfiguratie die pings en web verkeer voor deze EPG's toestaat:



```
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)
```

Opmerking: de vorige opdracht in blauw wordt gebruikt voor de ASIC van Northstar. De opdracht die wordt gebruikt voor Donner of Donner+ is **tonen platform interne ns tabel mth_luxh_slvq_DHS_SecurityGroupKeyTable0_memif_data**.

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]

```
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

    l4_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 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

Tip: U kunt alle TCAM-items met dezelfde methode controleren.

Probleemoplossing voor hardwareprogrammeerproblemen

Deze sectie verschaft een aantal nuttige opdrachten en tips voor het oplossen van problemen.

Opdrachten voor nuttige probleemoplossing

Hier zijn een aantal behulpzame opdrachten die u kunt gebruiken om de fouten van de manager van het bladbeleid te vinden wanneer er problemen worden aangetroffen:

```
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_mrulen_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]

```

Tip: Sommige bestanden zijn groot, dus het is gemakkelijker om ze naar de flitser te sturen en ze in een editor te onderzoeken.

```

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

```

Tips bij het oplossen van problemen

Hier zijn een paar nuttige tips voor probleemoplossing:

- Als u een TCAM-uitputtingsprobleem lijkt te ervaren, controleer dan de UI of CLI op fouten die met de betreffende regel zijn geassocieerd. Mogelijk wordt deze fout gemeld:

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

Eén regel kan meerdere TCAM-toetsen in het Application-Specific Integrated Circuit (ASIC) gebruiken. Voer deze opdrachten in om het aantal items op de ASIC te weergeven:

```

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

```

Opmerking: In dit voorbeeld zijn er **43** inzendingen aanwezig. Dit gebruik wordt ook aan APIC gerapporteerd in de klasse *QptCapacity*.

- Wanneer er meerdere overeenkomsten zijn, geeft de TCAM lookup terug naar de **lagere hw-index**. Om de index te controleren voert u deze opdracht in:

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

Wanneer u problemen oplost, kunt u de daling waarnemen die door de *om het even welke-om-even* regel wordt veroorzaakt. Deze regel staat altijd onderaan, wat betekent dat het pakje komt neer omdat een regel niet bestaat. Dit is te wijten aan een verkeerde configuratie of de beheerder van het beleidselement programmeert het niet zoals verwacht.

- De pcTags kunnen een *lokale* of *mondiale* reikwijdte hebben:
- **Systeem gereserveerd pcTag** - Dit pcTag wordt gebruikt voor systeem interne regels (1-15).**Gelijktijdig opgezette pcTag** - Deze pcTag wordt gebruikt voor gedeelde service (16-16385).**Lokaal bereik pcTag** - Deze pcTag wordt lokaal gebruikt per VRF (variërend van 16386-65535).

Contractnaam aan regel-id afleiden

Vaak wanneer een ingenieur in een zaak het oplossen van problemen heeft gekeken naar zoning regels. In sommige gevallen heeft een EPG/pcTag veel contracten en het kan omslachtig zijn om problemen op te lossen. Deze sectie zal een manier gaan om de naam van het contract te bepalen dat gebruikt wordt tussen de EPG's/pcTags van de regel ID die op de schakelaar CLI wordt gezien

Om te beginnen, onderzoek naar het concrete contract/heerobject **actrlRule** indien gewenst, versmal de zoekopdracht door **eigenschap: id-waarde: regel**

Als de juiste regel is gevonden, klikt u op de groene pijl op de DN om de kinderen van de actrlRule te bekijken. De kinderen zijn waar ons antwoord ligt.

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	

Het kindobject hier is **actrlRsToEpgConn**. Meestal moeten er twee zijn, één voor elke EPG. Deze objecten DN tonen de twee EPG's waartussen de overeenkomst wordt toegepast, evenals de richting (aanbieder of consument) en, nog belangrijker, de naam van het contractobject.

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	
lcOwn	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	

Zoals reeds is opgemerkt, is de contractnaam in dit geval **brc-dpita-ssh**

Indien nodig vraagt u om vzBrCP om het juiste contract te vinden

<u>vzBrCP</u>		?
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	