

Überprüfen von Verträgen und Regeln in der ACI-Fabric

Inhalt

[Einführung](#)

[Topologie](#)

[Prozessübersicht](#)

[Identifizieren der verwendeten Vertrags-/Zoning-Regel](#)

[Überprüfen der Hardwareprogrammierung](#)

[Fehlerbehebung bei Problemen mit der Hardwareprogrammierung](#)

[Nützliche Befehle zur Fehlerbehebung](#)

[Tipps zur Fehlerbehebung](#)

[Ableiten des Vertragsnamens von Regel-ID](#)

Einführung

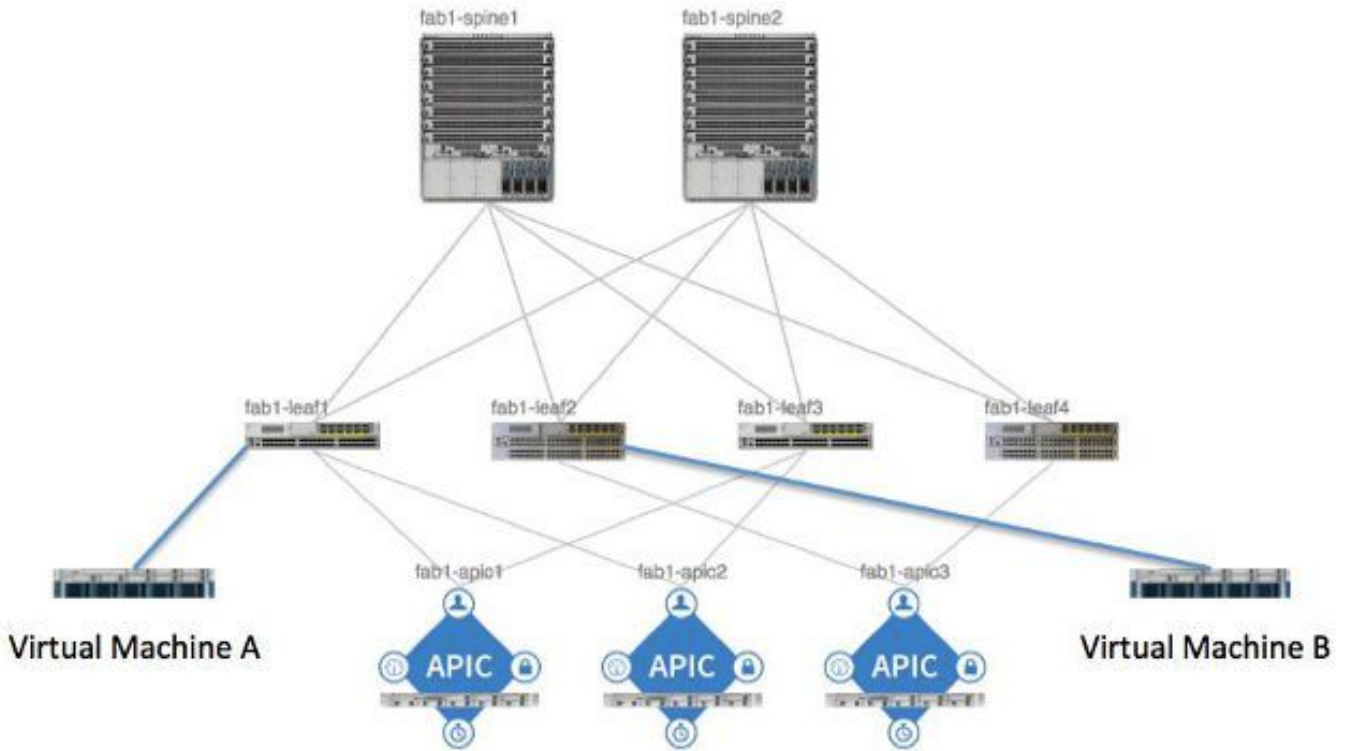
In diesem Dokument wird beschrieben, wie Sie überprüfen können, ob Verträge in der ACI-Fabric (Application Centric Infrastructure) konfiguriert wurden und ordnungsgemäß funktionieren.

Hinweis: Die Verifizierung der logischen und konkreten Modelle sowie die Hardwareprogrammierung wird in diesem Dokument beschrieben.

Topologie

In dem in diesem Dokument verwendeten Beispiel ist Virtual Machine-A (VM) an Leaf1 angeschlossen, und es ist ein Vertrag vorhanden, der die Kommunikation mit VM-B ermöglicht, das an Leaf2 angeschlossen ist. Der Vertrag ermöglicht sowohl Internet Control Message Protocol (ICMP) als auch HTTP.

Dieses Bild zeigt die Topologie:



Prozessübersicht

Dies ist der Richtlinien austausch und -fluss für Verträge und Regeln:

1. Der Policy Manager des Application Policy Infrastructure Controller (APIC) kommuniziert mit dem Policy Element Manager auf dem Switch.
2. Der Policy Element Manager auf dem Switch programmiert den Objektspeicher auf dem Switch.
3. Der Policy Manager des Switches kommuniziert mit dem ACLQOS-Client (Access Control List Quality of Service) auf dem Switch.
4. Der ACLQOS-Client programmiert die Hardware.

Identifizieren der verwendeten Vertrags-/Zoning-Regel

Im folgenden Beispiel wird die Ausgabe des Befehls für **Zoning-Regeln** aus dem Leaf **angezeigt**, bevor der Vertrag für die beiden Endpunktgruppen (EPGs) hinzugefügt wird.

```
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]

Dies ist die gleiche Befehlsausgabe nach dem Hinzufügen des Vertrags, sodass die beiden EPGs miteinander kommunizieren können:

```
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]

Hinweis: Beachten Sie die neuen hinzugefügten Regel-IDs (**4131** und **4132**), die Filter-IDs von **7** und **6** und den Umfang von **2686976**.

Vorsicht: Mit dieser Befehlsausgabe können Sie die Regeln, die Sie in einem Übungssystem überprüfen müssen, auf einfache Weise finden. Dies kann jedoch in einer Produktionsumgebung mit dynamischen Änderungen umständlich sein.

Eine weitere Methode, die Sie zum Auffinden der Interessenregeln verwenden können, ist die Verwendung von *Visore*. Durchsuchen Sie den Kontext Managed Object (MO) für **fvCtx**. Sie können dann auf diesem Bildschirm nach Ihrem spezifischen Kontextnamen (DN) suchen, wie hier gezeigt:



Notieren Sie den Umfang dieses Kontexts. Sie können diese verwenden, um der Befehlsausgabe **der** Befehlszeilenausgabe **für** die **Anzeigezonenregel** zuzuordnen, sodass Sie die Regeln suchen können, die Sie abfragen müssen:

fvCtx	
childAction	
deser	
dn	uni/tn-pr_dc_vmm_fab1/ctx-pr_dc_vmm_vrf < > M I 96
knwMcastAct	permit
lcOwn	local
modTs	2014-09-03T09:32:36.625-04:00
monPolDn	uni/tn-common/monepg-default < > M I 96
name	pr_dc_vmm_vrf
ownerKey	
ownerTag	
pcEnfPref	enforced
pcTag	32770
scope	2686976
seg	2686976
status	
uid	15374

Sie können die Segment-ID/den Bereich für den Kontext auch über die Benutzeroberfläche (UI) identifizieren, wie hier gezeigt:

The screenshot shows the Cisco CMN interface. The top navigation bar includes 'SYSTEM', 'TENANTS', 'FABRIC', 'VM NETWORKING', and 'L4-L7 SE'. Below the navigation, there's a search bar and a list of tenants: 'common', 'pr_dc_vmm_fab1', 'pr_citrix_fab1', 'mgmt', and 'azra-tenant'. The main content area is titled 'Private Network - pr_dc_vmm_vrf'. On the left, a tree view shows the tenant structure, with 'pr_dc_vmm_vrf' highlighted under 'Private Networks'. The right panel displays the 'PROPERTIES' for this network, including 'Name: pr_dc_vmm_vrf', 'Description: optional', 'Segment: 2686976', and 'Policy Control Enforcement Preference: Enforced'.

Dieser Bereich entspricht dem Bereich, der in der Befehlsausgabe **show zoning-rules** angezeigt wird:

```

4098      0      0      4098 DN (4098) /scop implicit enabled 2686976 deny,log
4099      0      49154 49154 s: 81553 implicit enabled 2686976 permit
4131     49155 32771 (4099) DN (7) /actrl/scop enabled 2686976 permit
4132     32771 49155 49155 press: 0 6 enabled 2686976 permit

```

Sobald Sie über die Bereich-ID-Informationen verfügen und die Regel und Filter-IDs identifizieren,

können Sie mit dem nächsten Befehl überprüfen, ob Sie die neuen Filter (und nicht die *impliziten Deny*-Nachrichten zwischen den EPGs) drücken. Die implizite Deny-Nachricht ist enthalten, sodass die EPGs standardmäßig nicht kommunizieren können.

Beachten Sie in dieser Befehlsausgabe, dass Leaf1, Filter-6 (**f-6**) inkrementiert:

```
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
```

Beachten Sie in dieser Befehlsausgabe, dass Leaf2, Filter-7 (**f-7**) inkrementiert:

```
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

Tipp: Kenntnisse des Bereichs, der Regel-ID, des Ziels und der Quell-PCs sowie des Filters sind wichtig, um dieses Problem weiter zu beheben. Es ist auch nützlich, sich mit den EPGs vertraut zu machen, zwischen denen die Regel-ID existiert.

Sie können eine Suche auf dem MO mit dem DN-Namen **fvAEPg** und *grep* für das jeweilige pcTag über den Befehl **moquery** durchführen, wie hier gezeigt:

```
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
```

Sie können die *Filteroption* auch mit dem Befehl **moquery** verwenden, wie hier gezeigt:

```
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
```

Überprüfen der Hardwareprogrammierung

Jetzt können Sie den Hardwareeintrag für die Regel überprüfen. Um die Hardwareinformationen anzuzeigen, geben Sie den Befehl **show platform internal ns table mth_lux_slvz_DHS_SecurityGroupStatTable_memif_data ingress** (dies ist ein Befehl **vsh_lc**) ein:

```
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 diesem Beispiel wird der Hardwareeintrag 41 (ENTRY [00041]) erhöht.

Hinweis: Der vorhergehende blaue Befehl wird für den Northstar ASIC verwendet. Der für Donner oder Donner+ verwendete Befehl ist `show platform internal ns table mth_luxh_slvy_DHS_SecurityGroupStatTable_memif_data`.

Hinweis: Die Verwendung dieses Befehls ist in einer Produktionsumgebung nicht praktikabel, Sie können jedoch die anderen Befehle verwenden, die in diesem Abschnitt beschrieben werden.

Denken Sie an die Regel (4132) und den Anwendungsbereich (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

Geben Sie diesen Befehl ein, um die Regel-ID für die Ternary Content-Addressable Memory (TCAM)-Hardwareindex-Zuordnung zu ermitteln und anhand der Regel-ID und/oder Filter-ID zu filtern:

```
module-1# show system internal acls qos 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]

In diesem Beispiel ist die Quelle-Ziel-EPG-Kombination von Interesse **32771=0x8003**, **49155=0xC003**. Daher sollten Sie alle TCAM-Einträge für diese Quell- und Zielklassen berücksichtigen, die mit den Regel-IDs (4131 und 4132) und Filter-IDs (6 und 7) übereinstimmen.

In diesem Beispiel werden einige dieser TCAM-Einträge verworfen. Als Referenz dient die Vertragskonfiguration, die Pings und Webdatenverkehr für diese EPGs zulässt:



```
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)
```

Hinweis: Der vorhergehende blaue Befehl wird für den Northstar ASIC verwendet. Der für Donner oder Donner+ verwendete Befehl ist `show platform internal ns table 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: Sie können alle TCAM-Einträge mit derselben Methode überprüfen.

Fehlerbehebung bei Problemen mit der

Hardwareprogrammierung

Dieser Abschnitt enthält einige nützliche Befehle und Tipps zur Fehlerbehebung.

Nützliche Befehle zur Fehlerbehebung

Im Folgenden finden Sie einige hilfreiche Befehle, mit denen Sie die Leaf-Policy-Manager-Fehler finden können, wenn Probleme auftreten:

```
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-histor 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]

Tipp: Einige der Dateien sind groß, daher ist es einfacher, sie in den Bootflash zu senden und sie in einem Editor zu untersuchen.

```
module-1# show system internal aclqos ?
asic          Asic information
brcm          Broadcom 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

```

Tipps zur Fehlerbehebung

Hier einige hilfreiche Tipps zur Fehlerbehebung:

- Wenn Sie ein Problem mit der Erschöpfung des TCAM haben, überprüfen Sie die Benutzeroberfläche oder die CLI auf Fehler, die der betreffenden Regel zugeordnet sind. Dieser Fehler kann gemeldet werden:

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

Eine Regel kann mehr als einen TCAM-Eintrag im Application-Specific Integrated Circuit (ASIC) erfordern. Geben Sie folgende Befehle ein, um die Anzahl der Einträge im ASIC anzuzeigen:

```

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

```

Hinweis: In diesem Beispiel sind **43** Einträge vorhanden. Diese Verwendung wird dem APIC auch in der *eqptCapacity*-Klasse gemeldet.

- Bei mehreren Übereinstimmungen gibt die TCAM-Suche den **unteren hw-index** zurück. Geben Sie den folgenden Befehl ein, um den Index zu überprüfen:

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

Bei der Fehlerbehebung können Sie den durch die *Any-Implicit*-Regel verursachten Abbruch beobachten. Diese Regel befindet sich immer unten, d. h. das Paket wird verworfen, weil keine Regel existiert. Dies ist entweder auf eine Fehlkonfiguration zurückzuführen, oder der Policy Element Manager programmiert diese nicht wie erwartet.

- Die PC-Tags können entweder *lokalen* oder *globalen* Umfang haben:
- **System Reserved pcTag** - Dieses pcTag wird für interne Systemregeln (1-15) verwendet.**Globally scoped pcTag** - Dieses pcTag wird für gemeinsam genutzten Service (16-16385) verwendet.**Lokales pcTag** - Dieses pcTag wird lokal pro VRF verwendet (Bereich von 16386 bis 65535).

Ableiten des Vertragsnamens von Regel-ID





Häufig werden Zoning-Regeln von einem Techniker bei der Fehlerbehebung in einem Fall geprüft. In einigen Fällen verfügt ein EPG/pcTag über viele Verträge und die Fehlerbehebung kann umständlich sein. In diesem Abschnitt wird erläutert, wie der Name des Vertrags, der zwischen den EPGs/pcTags verwendet wird, anhand der Regel-ID, die in der Switch-CLI angezeigt wird, ermittelt wird.

Um zu beginnen, sollte die Abfrage für das konkrete Vertrags-/Regelobjekt **actrlRule** bei Bedarf die Suche nach der **Eigenschaft** eingrenzen: **ID-Wert: Regel-d**

Wenn die richtige Regel gefunden wurde, klicken Sie auf den grünen Pfeil auf der DN, um die untergeordneten StrgRule-Objekte anzuzeigen. Die Kinder sind, wo unsere Antwort liegt.





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	

Das untergeordnete Objekt hier ist **actrlRsToEpgConn**. In der Regel sollte für jede EPG jeweils zwei EPGs vorhanden sein. Diese Objekt-DN zeigt die beiden EPGs, zwischen denen der Vertrag angewendet wird, sowie die Richtung (Anbieter oder Consumer) und vor allem den Vertragsobjektnamen.

<u>actlRsToEpgConn</u>	
childAction	
dn	topology/pod-1/node-101/sys/actl/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]] < >  
rType	mo

Wie hervorgehoben, lautet der Vertragsname in diesem Fall **brc-dpita-ssh**.

Wenn erforderlich, Abfrage für vzBrCP zum Auffinden des richtigen Vertrags

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