

# Vérification des contrats et des règles dans le fabric ACI

## Table des matières

---

[Introduction](#)

[Topologie](#)

[Aperçu du processus](#)

[Identifier la règle de contrat/de zonage utilisée](#)

[Vérification de la programmation matérielle](#)

[Dépannage des problèmes de programmation matérielle](#)

[Commandes de dépannage utiles](#)

[Conseils de dépannage](#)

[Dériver le nom du contrat de l'ID règle](#)

---

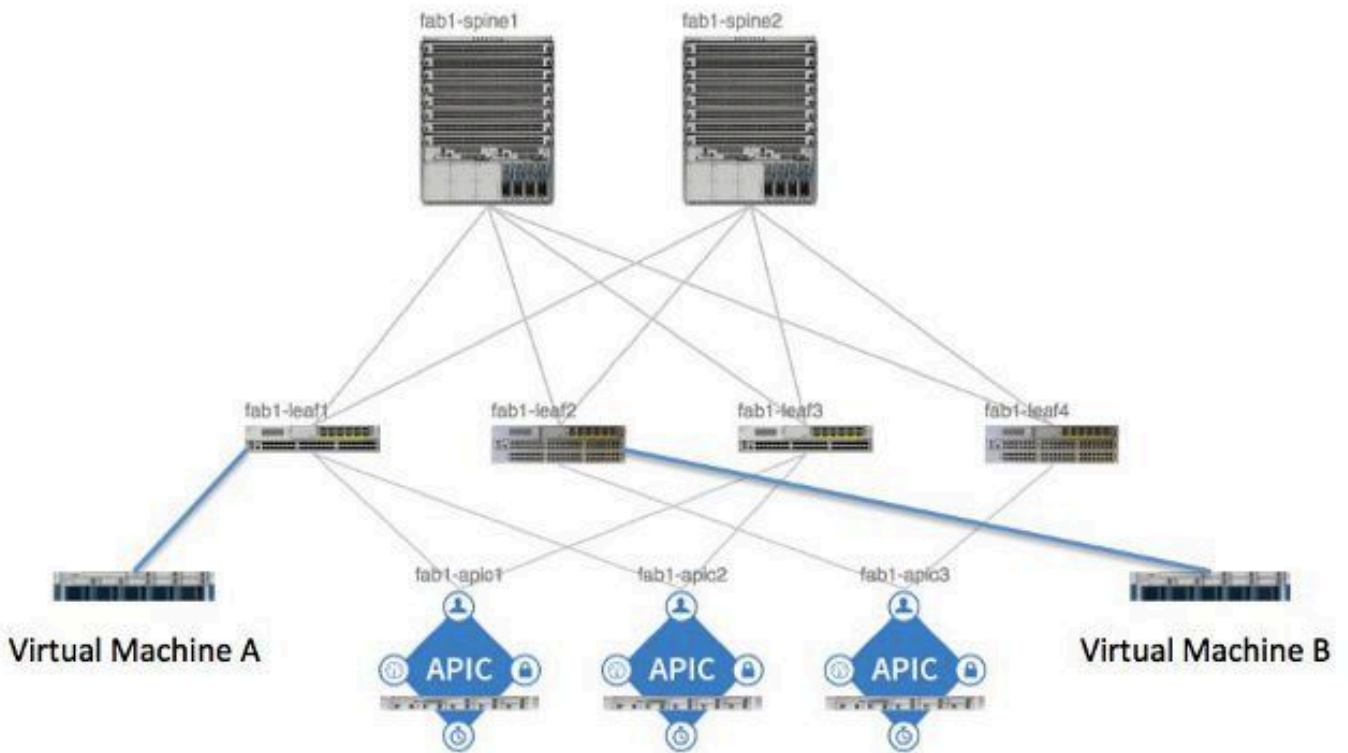
## Introduction

Ce document décrit comment vérifier que les contrats sont configurés et se comportent correctement dans le fabric ACI (infrastructure axée sur les applications).

## Topologie

Dans l'exemple utilisé tout au long de ce document, la machine virtuelle A (VM) est attachée à Leaf1 et un contrat est en place qui lui permet de communiquer avec la machine virtuelle B, qui est attachée à Leaf2. Le contrat autorise les protocoles ICMP (Internet Control Message Protocol) et HTTP.

Cette image illustre la topologie :



## Aperçu du processus

Il s'agit de l'interaction et du flux des politiques pour les contrats et les règles :

1. Le Policy Manager sur le contrôleur APIC (Application Policy Infrastructure Controller) communique avec le Policy Element Manager sur le commutateur.
2. Le Gestionnaire d'éléments de stratégie du commutateur programme le magasin d'objets sur le commutateur.
3. Le Policy Manager sur le commutateur communique avec le client ACLQOS (Access Control List Quality of Service) sur le commutateur.
4. Le client ACLQOS programme le matériel.

## Identifier la règle de contrat/de zonage utilisée

Voici un exemple de la sortie de la commande show zoning-rule du leaf, avant que le contrat ne soit ajouté pour les deux groupes de points d'extrémité (EPG).

```
<#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]

Il s'agit de la même sortie de commande après l'ajout du contrat afin que les deux groupes de terminaux puissent communiquer entre eux :

```
<#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]



Remarque : notez les nouveaux ID de règle (4131 et 4132) qui ont été ajoutés, les ID de filtre 7 et 6 et l'étendue 2686976.



Attention : cette sortie de commande vous permet de localiser facilement les règles que vous devez examiner dans un système de travaux pratiques. Toutefois, cela peut être fastidieux dans un environnement de production avec les modifications dynamiques qui se produisent.

---

Une autre méthode que vous pouvez employer afin de localiser les règles d'intérêt est d'utiliser Visore. Effectuez une recherche sur l'objet géré de contexte (MO) pour fvCtx. Vous pouvez alors rechercher votre nom distinctif (DN) de contexte spécifique sur cet écran, comme indiqué ici :

APIC Object Store Browser

Filter  0 of 0

Class or DN: fvCtx

Property: Op: Val1: Val2:

Run Query

Display URI of last query

Display last response

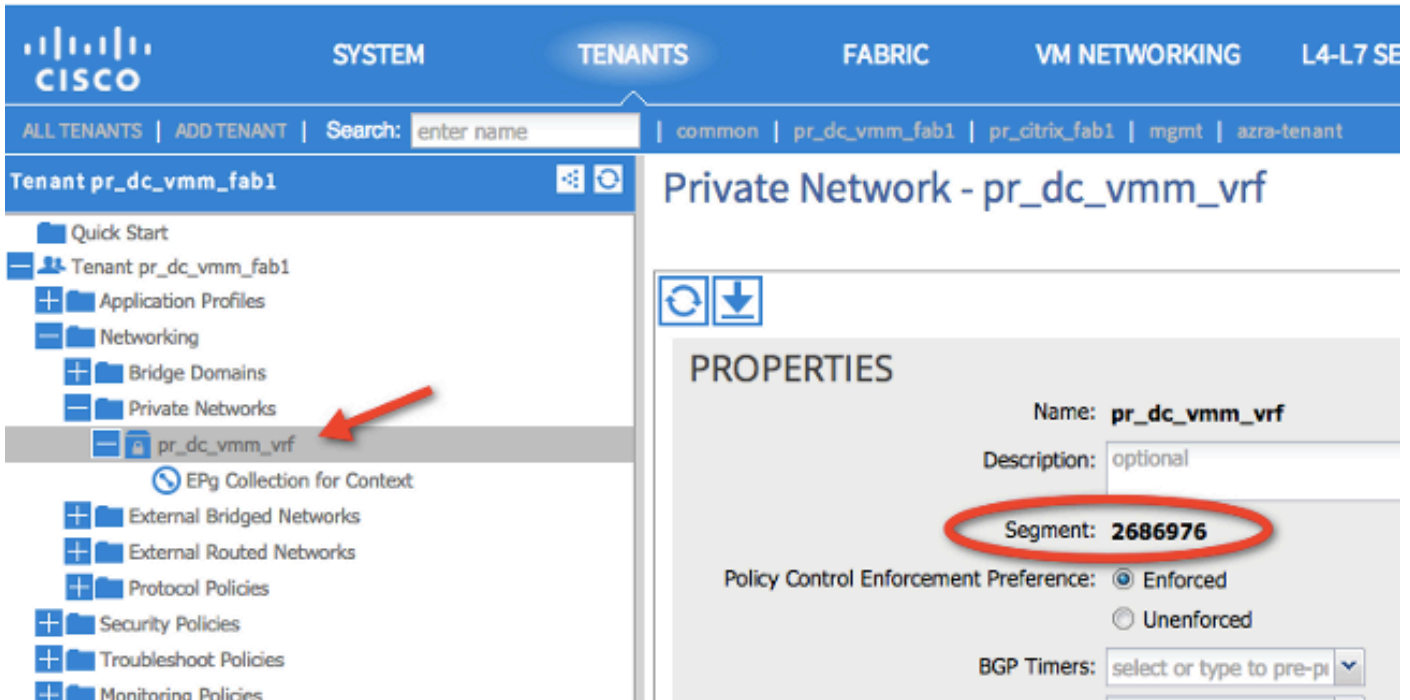
Total objects shown: 8

fvCtx	
childAction	
descr	
dn	<a href="#">uni/tn-infra/ctx-overlay-1</a> < >

Prenez note de la portée de ce contexte. Vous pouvez utiliser ceci afin de mapper à la sortie de commande show-zoning-rule afin que vous puissiez localiser les règles que vous devez interroger :

fvCtx	
childAction	
descr	
dn	<a href="#">uni/tn-pr_dc_vmm_fab1/ctx-pr_dc_vmm_vrf</a> < >
knwMcicastAct	permit
lcOwn	local
modTs	2014-09-03T09:32:36.625-04:00
monPolDn	<a href="#">uni/tn-common/monepg-default</a> < >
name	pr_dc_vmm_vrf
ownerKey	
ownerTag	
pcEntPref	enforced
pcTag	32770
scope	2686976
seg	2686976
status	
uid	15374

Vous pouvez également identifier l'ID/l'étendue du segment pour le contexte à partir de l'interface utilisateur, comme indiqué ici :



Cette étendue correspond à celle affichée dans le résultat de la commande show zoning-rules :

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

Une fois que vous disposez des informations d'ID d'étendue et que vous identifiez les ID de filtre et de règle, vous pouvez utiliser la commande suivante afin de vérifier que vous avez atteint les nouveaux filtres (et non les messages de refus implicites entre les groupes de terminaux). Le message de refus implicite est inclus afin que, par défaut, les groupes de terminaux ne puissent pas communiquer.

Notez dans le résultat de cette commande que Leaf1, Filter-6 (f-6) s'incrémente :

```
<#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
```

Notez dans le résultat de cette commande que Leaf2, Filter-7 (f-7) s'incrémente :

<#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

---

 Conseil : il est important de connaître l'étendue, l'ID de règle, la destination, le pcTags source et le filtre pour tenter de résoudre ce problème. Il est également utile de connaître les groupes de terminaux entre lesquels se trouve l'ID de règle.

---

Vous pouvez effectuer une recherche sur le MO avec le nom DN fvAEPg et grep pour le pcTag particulier via la commande moquery, comme montré ici :

```
<#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
```

Vous pouvez également utiliser l'option filter avec la commande moquery, comme indiqué ici :

```
<#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-commerceworkspace/

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

## Vérification de la programmation matérielle

Vous pouvez maintenant vérifier l'entrée matérielle de la règle. Afin d'afficher les informations matérielles, entrez la commande `show platform internal ns table mth_lux_slvz_DHS_SecurityGroupStatTable_memif_data ingress` (il s'agit d'une commande `vsh_lc`) :

```
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
```

Dans cet exemple, l'entrée matérielle 41 (ENTRY [000041]) est incrémentée.



Remarque : la commande précédente est utilisée pour l'ASIC Northstar. La commande utilisée pour Donner ou Donner+ est show platform internal ns table mth\_luxh\_slvy\_DHS\_SecurityGroupStatTable\_memif\_data.



Remarque : l'utilisation de cette commande n'est pas pratique dans un environnement de

 production, mais vous pouvez utiliser les autres commandes décrites dans cette section à la place.

Rappelez-vous la règle (4132) et la portée (268976).

4098	0	0	rule (4098) DN	implicit	enabled	2686976	deny, log
4099	0	49154	s: 81553	implicit	enabled	2686976	permit
4131	49155	32771	(4099) DN	(sys/actn1/scop	enabled	2686976	permit
4132	32771	49155	ress: 0	6	enabled	2686976	permit

Entrez cette commande afin de déterminer l'ID de règle pour le mappage d'entrée d'index matériel TCAM (Ternary Content-Addressable Memory), et filtrez en fonction de l'ID de règle et/ou de l'ID de filtre :

<#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
```

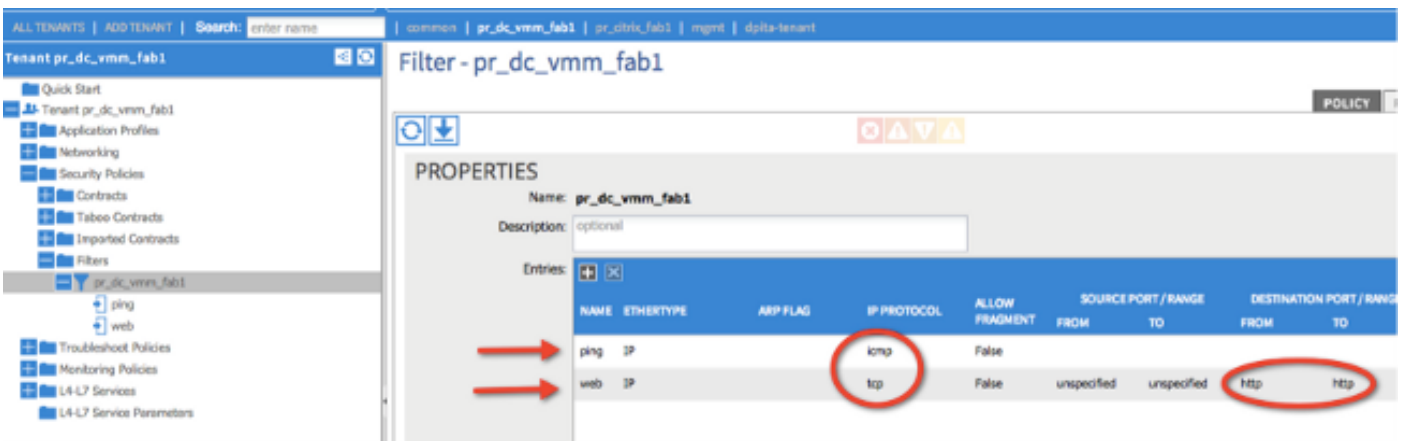
```
=== Region priority: 771 (rule prio: 3 entry: 3)===  
sw_index = 67 |
```

```
hw_index = 42
```

[snip]

Pour cet exemple, la combinaison EPG source et de destination intéressante est 32771=0x8003, 49155=0xC003. Par conséquent, vous pouvez considérer toutes les entrées TCAM pour ces classes source et de destination qui correspondent aux ID de règle (4131 et 4132) et aux ID de filtre (6 et 7).

Dans cet exemple, certaines de ces entrées TCAM sont vidées. Pour référence, voici la configuration du contrat qui autorise les requêtes ping et le trafic Web pour ces groupes de terminaux :



<#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)
```

 Remarque : la commande précédente est utilisée pour l'ASIC Northstar. La commande utilisée pour Donner ou Donner+ est 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	<a href="#">[RFC2460]</a>
1	ICMP	Internet Control Message		<a href="#">[RFC792]</a>
2	IGMP	Internet Group Management		<a href="#">[RFC1112]</a>

<#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	<a href="#">[RFC2460]</a>
1	ICMP	Internet Control Message		<a href="#">[RFC792]</a>
2	IGMP	Internet Group Management		<a href="#">[RFC1112]</a>
3	GGP	Gateway-to-Gateway		<a href="#">[RFC823]</a>
4	IPv4	IPv4 encapsulation		<a href="#">[RFC2003]</a>
5	ST	Stream		<a href="#">[RFC1190]</a> <a href="#">[RFC1819]</a>
6	TCP	Transmission Control		<a href="#">[RFC793]</a>
7	CBT	CBT		<a href="#">[Tony Ballardie]</a>

Port ↕	TCP ↕	UDP ▲	Description
0	TCP		Programming technique for specifying system-allocated (dynamic) ports <sup>[3]</sup>
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 <a href="#">↗</a> )
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) <sup>[12]</sup>
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

---

 Conseil : vous pouvez vérifier chacune des entrées TCAM avec la même méthode.

---

## Dépannage des problèmes de programmation matérielle

Cette section fournit des commandes et des conseils de dépannage utiles.

### Commandes de dépannage utiles

Voici quelques commandes utiles que vous pouvez utiliser afin de localiser les erreurs Leaf Policy Manager lorsque des problèmes sont rencontrés :

```
<#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]

---

 Conseil : certains fichiers sont volumineux, il est donc plus facile de les envoyer au bootflash et de les examiner dans un éditeur.

---

<#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

## Conseils de dépannage

Voici quelques conseils de dépannage utiles :

- Si vous rencontrez un problème d'épuisement de la TCAM, recherchez les erreurs associées à la règle en question dans l'interface utilisateur ou l'interface de ligne de commande. Cette erreur peut être signalée :

```
<#root>
```

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

Une règle peut prendre plusieurs entrées TCAM dans le circuit intégré spécifique à l'application (ASIC). Afin d'afficher le nombre d'entrées sur l'ASIC, entrez ces commandes :

```
<#root>
```

```
fab1-leaf1#
```

```
vsh_1c
```

```
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
```



Remarque : dans cet exemple, 43 entrées sont présentes. Cette utilisation est également rapportée au contrôleur APIC dans la classe eqptCapacity.

---

- Lorsqu'il y a plusieurs correspondances, la recherche TCAM renvoie l'index hw le plus bas. Afin de vérifier l'index, entrez cette commande :

```
<#root>
```

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

Lors du dépannage, vous pouvez observer la perte causée par la règle any-any-implicit. Cette règle est toujours en bas, ce qui signifie que le paquet est abandonné parce qu'une règle n'existe pas. Cela est dû à une configuration incorrecte ou le Gestionnaire d'éléments de stratégie ne le programme pas comme prévu.

- Les pcTags peuvent avoir une étendue locale ou globale :

System Reserved pcTag - Ce pcTag est utilisé pour les règles internes du système (1-15).

pcTag d'étendue globale : ce pcTag est utilisé pour le service partagé (16-16385).

pcTag à portée locale : ce pcTag est utilisé localement par VRF (plage de 16386 à 65535).

Lorsque vous effectuez un dépannage, la longueur de la valeur indique son étendue.

## Dériver le nom du contrat de l'ID règle

Souvent, lors d'un dépannage, un ingénieur examine des règles de zonage. Dans certains cas, un EPG/pcTag a de nombreux contrats et il peut être difficile à résoudre. Cette section décrit un moyen de déterminer le nom du contrat utilisé entre les EPG/pcTags à partir de l'ID de règle affiché sur l'interface de ligne de commande du commutateur.

Pour commencer, recherchez l'objet de contrat/règle concret `actrlRule` si vous le souhaitez, limitez la recherche par propriété : `id valeur : rule-d`

Une fois la règle trouvée, cliquez sur la flèche verte sur le DN pour afficher les enfants des objets `actrlRule`. Les enfants sont là où se trouve notre réponse.

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

L'objet enfant ici est actrlRsToEpgConn. En général, il peut y en avoir deux, un pour chaque EPG. Le DN de cet objet indique les deux EPG entre lesquels le contrat est appliqué, ainsi que la direction (fournisseur ou consommateur) et, plus important, le nom de l'objet du contrat.

actrlRsToEpgConn	
childAction	
dn	<a href="#">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]]</a> < >   
forceResolve	no
leOwn	local
modTs	2016-01-08T19:44:02.267+00:00
rType	mo
state	unformed
stateQual	none
status	
tCl	vzToEPg
tDn	<a href="#">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]]</a> < >   
tType	mo

Comme souligné, le nom du contrat dans ce cas est brc-dpita-ssh.

Si nécessaire, recherchez vzBrCP pour trouver le bon contrat.

## vzBrCP

?

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

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.