

Déployer ELAM pour capturer les paquets d'encapsulation VXLAN sur les commutateurs Nexus 7000

Table des matières

[Introduction](#)

[Informations générales](#)

[Topologie](#)

[Configuration du déclencheur](#)

[Interprétation des résultats](#)

[Informations connexes](#)

Introduction

Ce document décrit comment déployer le module ELAM (Embedded Logic Analyzer Module) pour capturer les paquets d'encapsulation VXLAN sur les commutateurs de la gamme Nexus 7000.

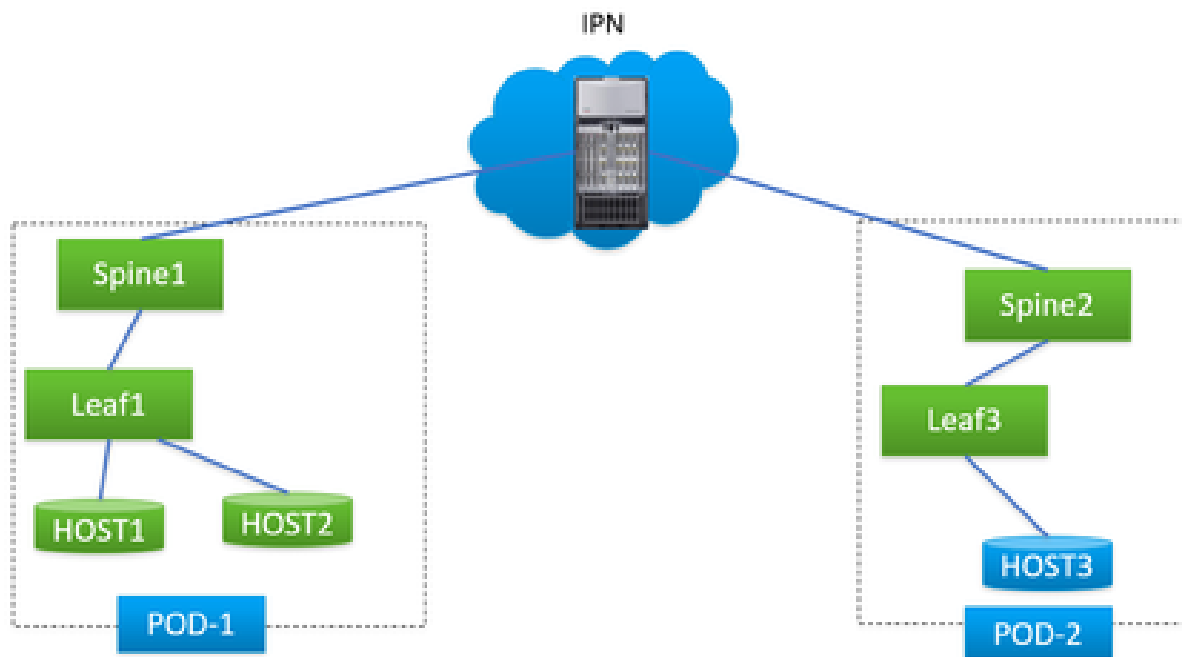


Conseil : reportez-vous au document [Présentation d'ELAM](#) pour obtenir une présentation d'ELAM.

Informations générales

De nombreux utilisateurs utilisent actuellement N7K comme périphérique de transit IPN/ISN pour leur déploiement MPOD/MSITE ACI. Cependant, par rapport à N9K, N7K n'a pas la capacité robuste de définir le déclencheur ELAM basé sur une combinaison externe (I2(vntag)|I3|I4)-interne (I2|I3|I4)-ieth riche. Par conséquent, il devient difficile de déterminer si un paquet spécifique encapsulé par VXLAN atteint le N7K au niveau de la périphérie IPN du point de vue d'ELAM. Ce document présente une méthode pour relever ce défi.

Topologie



Dans ce scénario, une topologie MPOD ACI simple est illustrée, où IPN est un N7K avec une carte F3. HOST1 et HOST2 sont dans pod1, HOST3 est dans pod2. HOST1 peut communiquer avec HOST3, mais HOST2 ne le peut pas. À la suite du dépannage effectué par un ingénieur ACI, il a été déterminé que les paquets de HOST2 à HOST3 ont été envoyés à N7K à partir de spine1 dans pod1, mais n'ont jamais été reçus par spine2 dans pod2. Cela a été vérifié par ELAM sur les spines ACI, ce qui a conduit à la suspicion que les paquets étaient abandonnés sur N7K.

Est-il possible d'attribuer définitivement le problème au N7K en se basant uniquement sur les résultats ELAM sur les spines ACI ? Certainement pas. L'ELAM sur le spine de sortie 1 a indiqué qu'il avait envoyé le paquet à N7K, mais cela ne garantit pas que le paquet a physiquement atteint N7K, car les paquets peuvent toujours être abandonnés après le cycle ELAM en raison de problèmes de couche inférieure. Cependant, lorsque vous effectuez un ELAM sur ces paquets spécifiques du côté N7K, cela peut nous aider à identifier avec précision le périphérique correct impliqué dans le problème.

Configuration du déclencheur

'l3-packet-length' est un déclencheur ELAM valide pour presque toutes les différentes générations de LC sur la plate-forme N7K. Par conséquent, utilisons-le pour établir la condition ELAM sur l'IPN N7K. La tâche implique de contrôler l'HÔTE pour transmettre des paquets de test avec une longueur de paquet spécifiée, comme illustré :

```
<#root>
```

```
#
```

```
ping 172.28.1.20 packet-size 777
```

```
PING 172.28.1.20 (172.28.1.20): 777 data bytes
785 bytes from 172.28.1.20: icmp_seq=0 ttl=252 time=1.246 ms
785 bytes from 172.28.1.20: icmp_seq=1 ttl=252 time=0.846 ms
785 bytes from 172.28.1.20: icmp_seq=2 ttl=252 time=0.84 ms
785 bytes from 172.28.1.20: icmp_seq=3 ttl=252 time=0.814 ms
785 bytes from 172.28.1.20: icmp_seq=4 ttl=252 time=0.817 ms
```

```
--- 172.28.1.20 ping statistics ---
```

```
5 packets transmitted, 5 packets received, 0.00% packet loss
```

```
round-trip min/avg/max = 0.814/0.912/1.246 ms
```

L'utilitaire PING est intégré à n'importe quel type de système d'exploitation, avec de légères variations dans les paramètres en fonction du système d'exploitation que vous utilisez. Il est essentiel de souligner la taille de paquet que vous spécifiez lorsque vous lancez la commande PING dans votre système d'exploitation. Dans cet exemple, le 777B représente la longueur des données pures, nécessitant une longueur supplémentaire de 8B (en-tête ICMP) et de 20B (en-tête IP) pour obtenir la longueur IP finale de 805B. Après l'encapsulation VXLAN (en ajoutant une surcharge de 50 milliards), vous pouvez anticiper que le paquet atteigne le N7K à 855 milliards. Configurons-le dans ELAM.

Dans cet exemple, l'interface qui se connecte à spine1 est E7/1 et E7/4 se connecte à spine2.

```
<#root>
```

```
#
```

```
show module 7
```

Mod	Ports	Module-Type	Model	Status
7	12	10/40 Gbps Ethernet Module	N7K-F312FQ-25	ok

```
module-7#
```

```
show hardware internal dev-port-map
```

```
-----
```

CARD_TYPE: 12 port 40G

>Front Panel ports:12

```
-----  
Device name           Dev role           Abbr num_inst:  
-----  
> Flanker Eth Mac Driver DEV_ETHERNET_MAC   MAC_0  6  
> Flanker Fwd Driver    DEV_LAYER_2_LOOKUP L2LKP  6  
> Flanker Xbar Driver   DEV_XBAR_INTF      XBAR_INTF 6  
> Flanker Queue Driver  DEV_QUEUEING       QUEUE  6  
> Sacramento Xbar ASIC  DEV_SWITCH_FABRIC  SWICHF  1  
> Flanker L3 Driver     DEV_LAYER_3_LOOKUP L3LKP  6  
> EDC                   DEV_PHY            PHYS  2
```

```
+-----+  
+-----+++FRONT PANEL PORT TO ASIC INSTANCE MAP+++-----+  
+-----+
```

FP port | PHYS | MAC_0 | L2LKP | L3LKP | QUEUE |SWICHF

1		0	0	0	0	0
2		0	0	0	0	0
3		1	1	1	1	0
4		1	1	1	1	0
5	0	2	2	2	2	0
6	0	2	2	2	2	0
7	1	3	3	3	3	0
8	1	3	3	3	3	0
9		4	4	4	4	0
10		4	4	4	4	0
11		5	5	5	5	0
12		5	5	5	5	0

```
+-----+  
+-----+
```

Vous devez donc le configurer dans l'instance 0.

```
<#root>
```

```
module-7# elam asic flanker instance 0
```

```
module-7(fln-elam)# layer2
```

```
module-7(fln-l2-elam)#
```

```
trigger dbus ipv4 ingress if l3-packet-length 855
```

```
module-7(fln-l2-elam)#
```

```
trigger rbus ingress if trig
```

```
module-7(fln-l2-elam)# start
```

```
module-7(fln-l2-elam)# status
```

```
ELAM Slot 7 instance 0: L2 DBUS Configuration: trigger dbus ipv4 ingress if l3-packet-length 855
```

```
L2 DBUS: Triggered
```

```
ELAM Slot 7 instance 0: L2 RBUS Configuration: trigger rbus ingress if trig
```

```
L2 RBUS: Triggered
```

Interprétation des résultats

```
<#root>
```

```
module-7(fln-l2-elam)#
```

```
show dbus
```

```
cp = 0x10084d00, buf = 0x10084d00, end = 0x10091050
```

```
-----  
Flanker Instance 00 - Capture Buffer On L2 DBUS:
```

```
Status(0x1102), TriggerWord(0x000), SampleStored(0x008), CaptureBufferPointer(0x000)
```

is_l2_egress: 0x0000, data_size: 0x023

[000]: 14f4a000 08010000 00000000 6d200800 00006000 00000000 01800100 00000000 00000000 00000000 000030

0590 00990000 00000000 00000000 00000005 88405000 00000000 00000000 00000000 00000000 00000000 00000000

a4 2dbeef00

Printing packet 0

L2 DBUS PRS MLH IPV4

label-count	: 0x0	mc	: 0x0
null-label-valid	: 0x0	null-label-exp	: 0x0
null-label-ttl	: 0x0	1b10-vld	: 0x0
1b10-eos	: 0x0	1b10-1b1	: 0x0
1b10-exp	: 0x0	1b10-ttl	: 0x0
1b11-exp	: 0x0	1b11-ttl	: 0x0
ipv4	: 0x0	ipv6	: 0x0
14-protocol	: 0x11		
df	: 0x0		
mf	: 0x0	frag	: 0x0
t11	: 0x1f	13-packet-length	: 0x357
option	: 0x0	tos	: 0x0
sup-eid	: 0x0	header-type	: 0x1
error	: 0x0	redirect	: 0x0
port-id	: 0x0	last-ethertype	: 0x800
l2-frame-type	: 0x0	da-type	: 0x0
packet-type	: 0x0	l2-length-check	: 0x0
ip-da-multicast	: 0x0	ip-multicast	: 0x0
ip-multicast-control	: 0x0	ids-check-fail	: 0x0
tr	: 0x0	outer-cos	: 0x0

inner-cos	: 0x0	vqi-valid	: 0x0
vqi	: 0x0	packet-length	: 0x369
vlan	: 0x4	destination-index	: 0x0
source-index	: 0x30		
bundle-port	: 0x0		
acos	: 0x0	outer-drop-eligibility	: 0x0
inner-drop-eligibility	: 0x0	sg-tag	: 0x0
rbh	: 0x0	vs1-num	: 0x0
inband-flow-creation-deletion	: 0x0	ignore-qoso	: 0x0
ignore-qosi	: 0x0	ignore-aclo	: 0x0
ignore-acli	: 0x0	index-direct	: 0x0
no-stats	: 0x0	dont-forward	: 0x0
notify-index-learn	: 0x1	notify-new-learn	: 0x1
disable-new-learn	: 0x0	disable-index-learn	: 0x0
dont-learn	: 0x0	bpdu	: 0x0
ff	: 0x0	rf	: 0x0
ccc	: 0x0	l2	: 0x0
rdt	: 0x0	dft	: 0x0
dfst	: 0x0	status-ce-1q	: 0x0
status-is-1q	: 0x1	trill-encap	: 0x0
mim-valid	: 0x0	dtag-ttl	: 0x0
dtag-ftag	: 0x0	valid	: 0x1
erspan-kpa-valid	: 0x0	recir-shim-vxlan-src-peer-id	: 0x0
vn-valid	: 0x0	source-vif	: 0x0
destination-vif	: 0x0	vn-p	: 0x0
sequence-number	: 0x60	v1	: 0x0
inner-de-valid	: 0x0	de-cfi	: 0x0
second-inner-cos	: 0x0	tunnel-type	: 0x2

UDP OTV/LISP TUNNEL BNDL

```
vlan-tag-valid: 0x0          segment-id-valid: 0x0
v1: 0x0                    de: 0x0
sgt-valid: 0x0             inner-ip-ttl: 0x0
ip-da-multicast: 0x0
lisp-inst-id: 0x2c8004

lisp-flags: 0xc8          isis-mac-da-valid: 0x0
type: 0x0
shim-valid                : 0x0
segment-id-valid         : 0x0          copp                : 0x0
dti-type-vpnid          : 0x0          segment-id            : 0x0
ib-length-bundle        : 0x58840      m1h-type              : 0x5
ulh-type                 : 0x4

source-ipv4-address: 10.0.200.64

destination-ipv4-address: 10.1.224.67

mim-destination-mac-address : 0000.0000.0000
mim-source-mac-address      : 0000.0000.0000
destination-mac-address     : 00c1.b1c9.c2c4
source-mac-address          : 000d.0d0d.0d0d
```

Étant donné que la longueur de paquet de couche 3 est utilisée comme déclencheur, il est possible que l'ELAM soit déclenché par des paquets d'arrière-plan non destinés à la capture. Par conséquent, il est impératif d'utiliser d'autres champs dans la capture pour effectuer une double vérification croisée des résultats de la capture. Cela garantit que le paquet capturé correspond à nos critères prévus, y compris les champs tels que l'adresse IP source (sip), l'adresse IP de destination (dip), la durée de vie (ttl), l'index source, etc. Une observation intéressante est que, bien que N7K ne supporte pas l'utilisation de VNID VXLAN comme déclencheur, dans l'interpréteur de sortie, le champ 'lisp-inst-id: 0x2c8004' correspond au VNID dans l'en-tête VXLAN.

<#root>

module-7(fln-12-elam)# dec

0x2c8004

2916356

Leaf3#

show system internal epm endpoint ip 172.28.1.20

MAC : 0000.2222.1202 ::: Num IPs : 1

IP# 0 : 172.28.1.20 ::: IP# 0 flags : host-tracked| ::: 13-sw-hit: Yes ::: flags2 :

Vlan id : 186 ::: Vlan vnid : 11494 ::: VRF name : zixu:vrf

BD vnid : 16482209 :::

VRF vnid : 2916356

/* Confirming the VNID from ACI LEAF side */

Phy If : 0x1a00b000 ::: Tunnel If : 0

Interface : Ethernet1/12

Flags : 0x80005c04 ::: sclass : 16388 ::: Ref count : 5

EP Create Timestamp : 01/22/2021 15:42:49.243582

EP Update Timestamp : 02/08/2021 11:26:52.882308

EP Flags : local|IP|MAC|host-tracked|sclass|timer|

module-7(fln-12-elam)#

show rbus

cp = 0x100a96fc, buf = 0x100a96fc, end = 0x100b5a4c

Flanker Instance 00 - Capture Buffer On L2 RBUS:

Status(0x1102), TriggerWord(0x000), SampleStored(0x008),CaptureBufferPointer(0x000)

is_l2_egress: 0x0000, data_size: 0x018

[000]: 0015cb30 0000006d 20000000 03000000 00000000 00000000 00000014 2d8000a0 3c3c0000 00000000 02000000
0000 00000400 00008000 005d0000 001e0002 2bd7c0cf f96002a0 000000ba

Printing packet 0

L2 RBUS INGRESS CONTENT

pad	: 0x572c	valid	: 0x1
l2-rbus-trigger	: 0x1	sequence-number	: 0x60
rit-ipv4-id	: 0x0	ipv4-tunnel-encap	: 0x0
rit-mp1s-rw	: 0x0	m12-ptr	: 0x0
m13-ptr	: 0x0	mark	: 0x0
result-cap3	: 0x0	di1-v5-delta-length	: 0x0
di1-v5-delta-length-plus	: 0x0	di1-v4-delta-length	: 0x0
di1-v4-delta-length-plus	: 0x0	di2-delta-length	: 0x0
di2-delta-length-plus	: 0x0	m12-delta-length	: 0x0
m12-delta-length-plus	: 0x0	m13-delta-length	: 0x0
m13-delta-length-plus	: 0x0	s-vector	: 0x0
lcpu-ff-valid	: 0x0	sup-di-vqi	: 0x0
erspan-term-index-dir	: 0x0	erspan-buffer-check	: 0x0
l2-tunnel-decapped	: 0x0	l3-delta-length	: 0x0
rit-crc16-valid	: 0x1	rit-crc16	: 0x42d8
vntag-p	: 0x0	frr-recirc	: 0x0
ingress-lif	: 0x5	ear1-proxy-vld	: 0x0
md-di-vld	: 0x0	rc	: 0x0
segment-id-valid	: 0x0	t11-out	: 0x1e
t11-mid	: 0x1e	tos-out	: 0x0
tos-in	: 0x0	orig-vlan1	: 0x0
vlan1	: 0x0	source-peer-id	: 0x0
final-ignore-qoso	: 0x0	port-id	: 0x0
cr-type	: 0x1	pup-packet	: 0x0

bpdu	: 0x0	vdc	: 0x0
tr	: 0x0	de	: 0x0
cos	: 0x0	inner-drop-eligibility:	0x0
inner-cos	: 0x0	acos	: 0x0
di-1tl-index	: 0x3c		
13-multicast-di	: 0x3c		
source-index	: 0x30	vlan	: 0x4
index-direct	: 0x0	di1-valid	: 0x1
vqi	: 0x4a	di2-valid	: 0x0
v5-fpoe-idx	: 0x0	di2-fpoe-idx	: 0x0
13-multicast-v5	: 0x0	dft	: 0x0
dfst	: 0x0	13-learning-ff	: 0x0
result-rbh	: 0x40	di2-cr-type	: 0x0
result-2	: 0x1	dtag-ftag	: 0x0
dtag-ttl	: 0x20	mac-in-mac-op	: 0x0
dvif	: 0x0	result-cap1	: 0x0
result-cap2	: 0x0	erspan-term	: 0x0
erspan-decap	: 0x0	dont-learn	: 0x0
routed-frame	: 0x1	copy-cause	: 0x0
12-copy-cause	: 0x0	13-rit-ptr	: 0x5d
sg-tag	: 0x0	trill-nh-id	: 0x0
t1-in	: 0x1e	fc-up	: 0x0
up-did	: 0x0	did	: 0x22bd
up-sid	: 0x0	sid	: 0xf819ff
shim-12-tunnel-encap:	0x0	shim-ls-hash	: 0xb
shim-rc	: 0x0	shim-lif	: 0x5
shim-replication-pkt:	0x0	shim-router-mac	: 0x1
shim-mark-enable	: 0x0	shim-qos-group-id	: 0x0
shim-destination-table-index:	0x5d	shim-acos-preserve	: 0x0
mim-destination-mac-address	: 0000.0000.0000		
mim-source-mac-address	: 0000.0000.0000		

```
module-7(fln-12-elam)#
```

```
show system internal pixmc info lt1-cb lt1 0x30
```

```
lt1 | lt1_type | if_index | lc_type | vdc | v4_fpoE | v5_fpoE | base_fpoE_idx | flag  
0x0030 | 5 |  
Eth7/1  
| 2 | 4 | 0x00 | 0x00 | 0x0000 | 0x0
```

```
module-7(fln-12-elam)#
```

```
show system internal pixmc info lt1-cb lt1 0x3c
```

```
lt1 | lt1_type | if_index | lc_type | vdc | v4_fpoE | v5_fpoE | base_fpoE_idx | flag  
0x003c | 5 |  
Eth7/4  
| 2 | 4 | 0x00 | 0x00 | 0x0000 | 0x0
```

La même méthodologie fonctionne également si vous essayez de capturer des paquets de diffusion, de monodiffusion inconnue et de multidiffusion (BUM) dans un BD d'inondation ACI, il vous suffit de définir un ARP statique et de pointer vers un MAC inexistant sur votre HÔTE, puis de lancer la commande PING de la même manière.

Informations connexes

Vous pouvez consulter ces liens pour plus de détails sur la façon d'utiliser ELAM sur différents LC N7K :

- [Présentation d'ELAM](#)
- [Procédure ELAM du module N7K série M](#)
- [Procédure ELAM du module F1 N7K](#)
- [Procédure ELAM du module F2 N7K](#)
- [Procédure ELAM du module M3 N7K](#)
- [Assistance technique de Cisco et téléchargements](#)

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.