

部署ELAM以捕获Nexus 7000系列交换机上的VXLAN封装数据包

目录

[简介](#)

[背景信息](#)

[拓扑](#)

[配置触发器](#)

[解释结果](#)

[相关信息](#)

简介

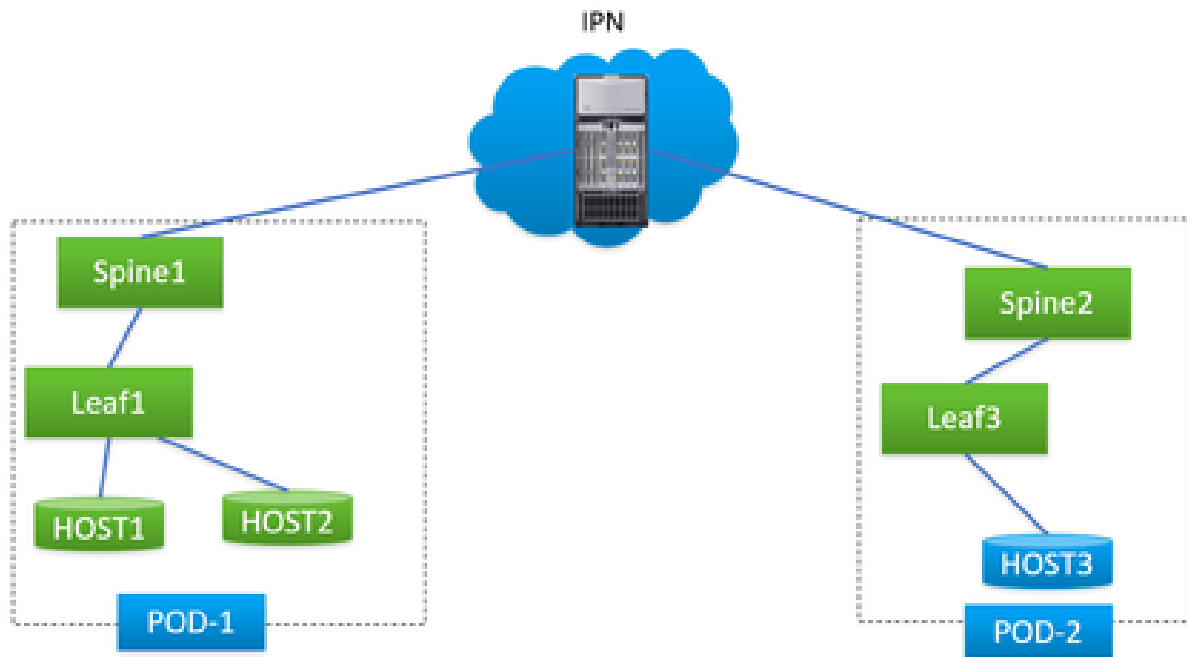
本文档介绍如何部署嵌入式逻辑分析器模块(ELAM)以捕获Nexus 7000系列交换机上的VXLAN封装数据包。

 提示：有关ELAM的[概述](#)，请参阅ELAM概述文档。

背景信息

许多用户目前将N7K用作ACI MPOD/MSITE部署的IPN/ISN传输设备。然而，与N9K相比，N7K缺乏基于丰富的Outer(l2(vntag)|l3|l4)-inner(l2|l3|l4)-eight组合设置ELAM触发器的稳健能力。因此，从ELAM的角度来看，很难确定特定VXLAN封装的数据包是否到达IPN边缘的N7K。本文档概述了解决此挑战的方法。

拓扑



在此场景中，显示了简单的ACI MPOD拓扑，其中IPN是带F3卡的N7K。HOST1和HOST2在Pod1中，HOST3在Pod2中。HOST1可以与HOST3通信，但HOST2无法通信。ACI工程师进行故障排除后，确定从HOST2到HOST3的数据包从pod1中的spine1发送到N7K，但pod2中的spine2从未收到这些数据包。这通过ACI主干上的ELAM进行了验证，导致怀疑数据包被丢弃在N7K上。

是否可以仅根据ACI主干上的ELAM结果确定问题归因于N7K?当然不会。出口主干1上的ELAM表示它已将数据包发送到N7K，但这并不能保证数据包实际到达N7K，因为数据包在ELAM周期后由于较低层问题仍可能被丢弃。但是，当您在N7K端对这些特定数据包执行ELAM时，它可以帮助我们准确识别问题所涉及的正确设备。

配置触发器

“l3-packet-length”是N7K平台上几乎所有不同层代LC的有效ELAM触发器。因此，让我们利用它来建立IPN N7K上的ELAM条件。此任务涉及控制HOST传输具有指定数据包长度的测试数据包，如下所示：

```
<#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
```

PING实用程序已集成到任何类型的操作系统中，但参数仅根据您使用的操作系统略有变化。要强调的关键一点是，注意在操作系统中启动PING时指定的数据包大小。在本示例中，777B表示纯数据长度，需要额外8B（ICMP报头）和20B（IP报头）才能获得最终IP长度805B。在VXLAN封装后（增加额外50B开销），您可以预计数据包到达N7K的855B。让我们在ELAM中配置它。

在本示例中，连接到spine1的接口是E7/1，而E7/4连接到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      0
6          0      2      2      2      0
7          1      3      3      3      0
8          1      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

```

```

+-----+
+-----+

```

因此，您需要在实例0中设置它。

```
<#root>
```

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

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

```
module-7(f1n-12-elam)#
trigger dbus ipv4 ingress if 13-packet-length 855

module-7(f1n-12-elam)#
trigger rbus ingress if trig

module-7(f1n-12-elam)# start
module-7(f1n-12-elam)# status

ELAM Slot 7 instance 0: L2 DBUS Configuration: trigger dbus ipv4 ingress if 13-packet-length 855
L2 DBUS: Triggered

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

解释结果

```
<#root>
```

```
module-7(f1n-12-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 00003000
```

```
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
12-frame-type	: 0x0	da-type	: 0x0
packet-type	: 0x0	12-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
```

由于I3数据包长度用作触发器，ELAM可能会被无意捕获的后台数据包触发。因此，必须利用捕获中的其他字段对捕获结果进行双重交叉检查。这样可以确保捕获的数据包与我们的预期条件一致，包括源IP(sip)、目标IP(dip)、生存时间(ttl)、源索引等字段。有趣的是，虽然N7K不支持使用VXLAN VNID作为触发器，但在输出解释器中，字段“lisp-inst-id: 0x2c8004”对应于VXLAN报头中的VNID。

```
<#root>
```

```
module-7(f1n-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| ::: l3-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|sc|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 020000

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
1cpu-ff-valid	: 0x0	sup-di-vqi	: 0x0
erspan-term-index-dir	: 0x0	erspan-buffer-check	: 0x0
12-tunnel-decapped	: 0x0	13-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-1t1-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

```

l3-multicast-v5      : 0x0          dft                : 0x0
dfst                 : 0x0          l3-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
l2-copy-cause       : 0x0          l3-rit-ptr         : 0x5d
sg-tag              : 0x0          trill-nh-id        : 0x0
ttl-in              : 0x1e          fc-up              : 0x0
up-did              : 0x0          did                : 0x22bd
up-sid              : 0x0          sid                : 0xf819ff
shim-l2-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-l2-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-l2-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
```

如果您尝试在ACI泛洪BD内捕获广播、未知单播和组播(BUM)数据包，则此方法同样有效，您只需设置静态ARP并指向主机上不存在的MAC，然后以同样的方式启动PING。

相关信息

有关如何在不同N7K LC上使用ELAM的详细信息，请参阅以下链接：

- [ELAM概述](#)
- [N7K M系列模块ELAM流程](#)
- [N7K F1模块ELAM过程](#)
- [N7K F2模块ELAM过程](#)
- [N7K M3模块ELAM流程](#)
- [思科技术支持和下载](#)

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。