

# Nexus 9000 Cloud Scale ASIC(Tahoe)NX-OS ELAM

## 内容

### [概要](#)

### [該当ハードウェア](#)

### [Nexus Tahoe ASIC ELAM手順](#)

### [トポロジ](#)

### [手順1：ポートのASIC、スライス、およびSrcIdの確認](#)

### [ステップ2：モジュールへの接続](#)

### [ステップ3:ELAMコンフィギュレーションモードに入り、ステップ1で取得した適切なASICを指定します](#)

### [手順4:ELAMトリガーの設定](#)

### [ステップ5:SRCおよびDEST IPを使用してELAMトリガーを設定します。](#)

### [ステップ6:ELAMを開始します。](#)

### [ステップ7:ELAMがトリガーされたかどうかを確認し、結果を確認します](#)

### [NX-OSバージョン7.0\(3\)I5\(2\)より前のELAM結果の表示](#)

### [VXLANカプセル化ELAM:](#)

### [NX-OS 7.0\(3\)I7\(2\)以降のELAM設定](#)

### 概要

m ELAM(Embedded Logic Analyzer Module)Cisco Nexus 9000 CloudScale ASIC

ヒント：ELAMの概要については、「[ELAMの概要](#)」を参照してください。

### 該当ハードウェア

このドキュメントで説明する手順は、次のハードウェアにのみ適用できます。

N9K-C93180YC-EX	N9K-C92304QC
N9K-X9736C-EX	N9K-C92300YC
N9K-C93108TC-EX	N9K-X9788TC-FX
N9K-X9732C-EX	N9K-X97284YC-FX
N9K-X97160YC-EX	N9K-C93180YC-FX
N9K-C93180LC-	N9K-C93108TC-

EX	FX
N9K-C92160YC-X	N9K-C9348GC-FXP
N9K-C9272Q	N9K-X9732C-FX
N9K-C9236C	N9K-C9336C-FX2
N9K-C93240YC-FX2	N9K-C93300YC-FX2
N9K-C9364C	N9K-C9332C

## Nexus Tahoe ASIC ELAM手順

### トポロジ



### 手順1: ポートのASIC、スライス、およびSrcIdの確認

```
N9K-C92160YC-X-2# show hardware internal tah interface e1/49
IfIndex: 436232192
DstIndex: 5952
IfType: 26
Asic: 0 <<<<<<<<<< Asic: 0
AsicPort: 56
SrcId: 48 <<<<<<<<<< Slice: 1 <<<<<<<<<< PortOnSlice: 24
```

**注意:** 各スライス、lu-a2dなどのグローバルコンテンツを保持するため、ELAMは1つのターミナルウィンドウでのみ使用してください。

たとえば、ポートチャネル(PO)にスライス0に対応するEth 1/53とスライス1に対応するEth 1/54の2つのリンクがあるとします。異なるスライスに対して2つの異なるターミナルウィンドウで同時にELAMを設定しても、後のスライス(たとえばスライス1)が最初のスライス(スライス0)を上書きするため、役に立たないため、両方のターミナルウィンドウで同じ結果になります。

この情報は、次の方法で再確認できます。

```
N9K-C92160YC-X-2# show system internal ethpm info interface e1/49 | i i src
  IF_STATIC_INFO: port_name=Ethernet1/49,if_index:0x1a006000,ltl=5952,slot=0,
nxos_port=192,dmod=1,dpid=56,
  unit=0,queue=65535,xbar_unitbmp=0x0,ns_pid=255,slice_num=1,port_on_slice=24,src_id=48
```

## ステップ2: モジュールへの接続

```
N9K-C92160YC-X-2# attach mod 1
```

ステップ3:ELAMコンフィギュレーションモードに入り、ステップ1で取得した適切なASICを指定します

```
module-1# debug platform internal tah elam asic 0
```

## 手順4:ELAMトリガーの設定

注: キャプチャされるパケット/フローに応じて、ここで指定できるオプションは多数あります

```
module-1(TAH-elam)# trigger init asic 0 slice 1 lu-a2d 1 in-select 6 out-select 0 use-src-id 48
```

### ヒント:

- 入力ポートと出力ポートが同じASIC上の異なるスライス上にある場合、出力スライス上のELAMは発信パケットをキャプチャしません。これは、パケットが出力スライス上のLUXブロックを通過せず、ELAMをバイパスするためです。
- 結果に基づくトリガーのリバースELAMにはlu-a2d 0、パケット属性に基づくトリガーのリバースELAMにはlu-a2d 1が使用されます
- in-selectにはalways 6を、out-selectには0を使用します

警告: lu-a2dの後に0を使用すると、スイッチがクラッシュする可能性があるため、使用しないでください。詳細については、[CSCvd64106](#)を参照してください

ステップ5:SRCおよびDEST IPを使用してELAMトリガーを設定します。

次に示すように、この例では、送信元IPは192.0.2.2、宛先IPは192.0.2.1です。

```
module-1(TAH-elam-insel6)# reset
```

```
module-1(TAH-elam-insel6)# set outer ipv4 dst_ip 192.0.2.1 src_ip 192.0.2.2
```

注: 「set」コマンドはELAM全体で使用され、予期しないフィールドでトリガーまたはトリガーされない可能性があるため、「reset」を実行してください。

ステップ6:ELAMを開始します。



1.パケットが実際に廃棄されているかどうかを確認する場合、[Final Drops] フィールドは**考慮すべき唯一の**フィールドです。つまり、LUA/B/C/Dなどの他のフィールドで例外がスローされる場合がありますが、これは必ずしもパケットがドロップされていることを意味するわけではありません。この出力を注意深く確認してください ( 必要に応じてTACに相談してください )。

2. CPUにパントされたトラフィックには、sup\_hitフラグが設定されます(詳細を報告する) | `grep sup_hit`。

- 理由をデコードするには、「`show system internal access-list sup-redirect-stats all`」を使用して、supインデックスを照合します
- 正しい[system routing mode] が設定されていることを確認します(show system routing mode)。 ガイドラインおよび制限事項については、『[VXLAN導入の考慮事項](#)』を参照してください。「System Routing Mode:template-vxlan-scale」は、Cisco NX-OSリリース7.0(3)I5(2)以降には適用されません。Cisco NX-OSリリース7.0(3)I4(x)またはNX-OSリリース7.0(3)I5(1)と組み合わせてVXLAN BGP EVPNを使用する場合、「System Routing Mode:template-vxlan-scale」が必要です。Cisco Nexus 9300-EXスイッチX9700-EXラインカードを搭載したCisco Nexus 9500スイッチ「システムルーティングモード」を変更するには、スイッチをリロードする必要があります。

CPUパントが発生しているトラフィックの例：

```
module-1(TAH-elam-insel6)# report
```

```
SUGARBOWL ELAM REPORT SUMMARY  
=====
```

```
Incoming Interface: Eth1/3  
Src Idx : 0x9, Src BD : 23  
Outgoing Interface Info: dmod 1, dpid 72  
Dst Idx : 0x601, Dst BD : 802
```

```
Packet Type: IPv4
```

```
Dst MAC address: B0:8B:CF:A3:D0:4B  
Src MAC address: 00:10:DB:FF:10:00  
.1q Tag0 VLAN: 23, cos = 0x0
```

```
Dst IPv4 address: 192.0.2.1  
Src IPv4 address: 192.0.2.2  
Ver      = 4, DSCP      = 2, Don't Fragment = 1  
Proto    = 6, TTL       = 49, More Fragments = 0  
Hdr len  = 20, Pkt len  = 60, Checksum      = 0x63c3
```

```
L4 Protocol : 6  
TCP Dst Port : 80  
TCP Src Port : 46340
```

```
Sup hit: 1, Sup Idx : 2720 <<---- CPU punt, use below CLI to resolve the meaning of Sup Idx
```

```
Drop Info:  
-----
```

```
LUA:  
LUB:  
LUC:  
LUD:  
Final Drops:
```

```
# show system internal access-list sup-redirect-stats all | grep 2720 2720 copp-system-p-acl-  
http 63
```

**NX-OSバージョン7.0(3)I5(2)より前のELAM結果の表示**

+これにはdot1qヘッダーがありますか。

```
module-1(TAH-elam-insel6)# report | grep pr_lu_vec_l2v.qtag0 GBL_C++: [MSG] pr_lu_vec_l2v.qtag0_vld:
0x1 << dot1q yes? 0x1 GBL_C++: [MSG] pr_lu_vec_l2v.qtag0_cos: 0x0 GBL_C++: [MSG]
pr_lu_vec_l2v.qtag0_de: 0x0 GBL_C++: [MSG] pr_lu_vec_l2v.qtag0_vlan: 0xA << VL 10
+ VLANのチェック:
```

```
module-1(TAH-elam-insel6)# report | grep -1 fpx_lookup_vec.lkup.macsakey.key.fid
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.vld: 0x1
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.fid_type: 0x0
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.fid_vld: 0x0
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.fid: 0xA << dec 0xa = VL 10
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715
+ SRC MACを確認します ( 前の手順でも実際に確認できます )。
```

```
module-1(TAH-elam-insel6)# report | grep -i fpx_lookup_vec.lkup.macsakey.key.mac
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715 << 00fe.c80e.2715
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715
+これは新しい学習ですか？
```

```
module-1(TAH-elam-insel6)# report | grep -1 fpx_lookup_vec.sa_notify_info
GBL_C++: [MSG] fpx_lookup_vec.lkup.ptvec.misc1.tcp_flags: 0x0
GBL_C++: [MSG] fpx_lookup_vec.sa_notify_info: 0x5200000C060
GBL_C++: [MSG] fpx_lookup_vec.sa_notify_info.enable: 0x0 << This will be set to 0x1 for learning
to happen
GBL_C++: [MSG] fpx_lookup_vec.sa_notify_info.conv_learn_only: 0x0
```

+ SRCとDSTのIPを確認します。

```
module-1(TAH-elam-insel6)# report | grep vec_l3v.ip.*a
GBL_C++: [MSG] pr_lu_vec_l3v.ip.da: 0x0000000000000000c0000201 << DST IP: 192.0.2.1
GBL_C++: [MSG] pr_lu_vec_l3v.ip.sa: 0x0000000000000000c0000202 << SRC IP: 192.0.2.2
```

+入力SRC\_IDを確認します。

```
module-1(TAH-elam-insel6)# report | egrep SRC
GBL_C++: [MSG] SRCID: 0x30
```

```
module-1(TAH-elam-insel6)# report | grep vec.ihdr.ieth.hdr.src_idx
GBL_C++: [MSG] lur_vec.ihdr.ieth.hdr.src_idx: 0xA9 << sh hardware internal tah int e1/49 | i i niv_idx
```

+ ELAMがトリガーされない場合は、次のようになります。

```
module-1(TAH-elam-insel6)# report  
GBL_C++: [MSG] tahusd_elam_wrapper_report:27d:asic type 5 inst 0 slice 1 a_to_d 1 insel 6  
outsel 0  
GBL_C++: [MSG] Inside tahusd_elam_wrapper_dav_report  
GBL_C++: [MSG] ELAM not yet triggered <<<<<<
```

### VXLANカプセル化ELAM:

VXLANパケットはカプセル化されるため、ELAMはOUTERヘッダーではなくINNERヘッダーでトリガーされる必要があります。ARPフレームについては、次の例を参照してください。

```
module-1# debug platform internal tah elam asic 0  
module-1(TAH-elam)# trigger init asic 0 slice 1 in-select 7 out-select 0 use-src-id 48  
module-1(TAH-elam-insel7)# reset  
module-1(TAH-elam-insel7)# set inner arp source-ip-addr 192.0.2.2 target-ip-addr 192.0.2.1  
module-1(TAH-elam-insel7)# start  
module-1(TAH-elam-insel7)# report
```

### NX-OS 7.0(3)I7(2)以降のELAM設定

NX-OS 7.0(3)I7(2)以降では、ASICまたはスライス番号を簡単に指定しなくても、ELAMをグローバルにトリガーできるようになりました。次の例を参照してください。

```
Nexus-9K# debug platform internal tah elam  
Nexus-9K(TAH-elam)# trigger init  
Nexus-9K(TAH-elam-insel6)# reset  
Nexus-9K(TAH-elam-insel6)# set outer ipv4 dst_ip 192.0.2.1 src_ip 192.0.2.2  
Nexus-9K(TAH-elam-insel6)# start  
Nexus-9K(TAH-elam-insel6)# report
```

## 翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。