Catalyst 6500 Series Switches with Supervisor Engine 2T ELAM Procedure

Contents
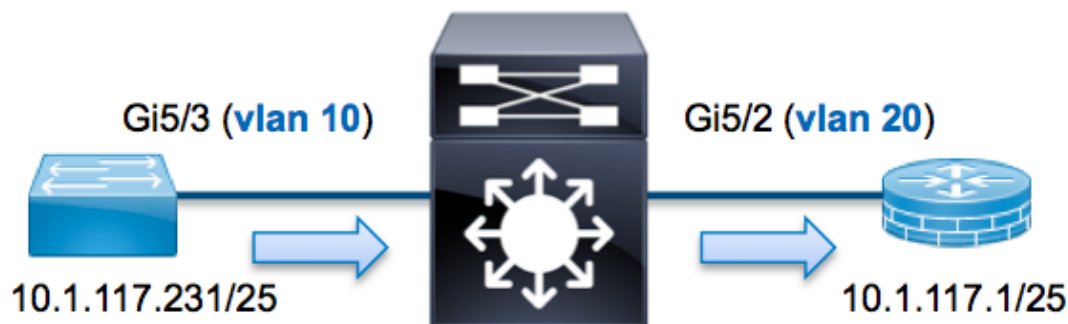
Introduction

This document describes the steps used in order to perform an ELAM on Cisco Catalyst 6500 Series switches that run Supervisor Engine 2T (Sup2T), explains the most relevant outputs, and describes how to interpret the results. This example also applies to DFC4–enabled linecards.

Tip: Refer to the ELAM Overview document for an overview on ELAM.

Topology

In this example, a host on VLAN 10 (*10.1.117.231*), port *G5/3* sends an Internet Control Message Protocol (ICMP) request to a host on VLAN 20 (*10.1.117.1*), port *G5/2*. ELAM is used in order to capture this single packet from *10.1.117.231* to *10.1.117.1*. It is important to remember that ELAM allows you to capture a single frame.

Note: For Sup2T, each ELAM command begins with this syntax: *show platform capture elam*.

Determine the Ingress Forwarding Engine

Traffic is expected to ingress the switch on port *G5/3*. When you check the modules in the system, you see that *module 5* is the *Active* supervisor. Therefore, you should configure the ELAM on *module 5*.

```
Sup2T#show module 5
Mod Ports Card Type                                    Model              Serial No.
--- ----- -------------------------------------- ------------------ -----------
  5    5  Supervisor Engine 2T 10GE w/ CTS (Active)VS-SUP2T-10G       SAL15056BKR
```

For the Sup2T, perform the ELAM on the Layer 2 (L2) Forwarding Engine (FE) with internal codename *Eureka*. Note that the L2 FE Data Bus (DBUS) contains original header information before the L2 and Layer 3 (L3) lookups, and the Result Bus (RBUS) contains the results after both L3 and L2 lookups. The L3 lookup is performed by the L3/Layer 4 (L4) FE with internal codename *Lamira*.

```
Sup2T(config)#service internal
Sup2T# show platform capture elam asic eureka slot 5
Assigned asic_desc=eu50
```

Note: The *service internal* command is required in order to run an ELAM on Sup2T. This configuration simply unlocks the hidden commands.

Configure the Trigger

The *Eureka* ASIC supports ELAM triggers for IPv4, IPv6, and others. The ELAM trigger must align with the frame type. If the frame is an IPv4 frame, then the trigger must also be IPv4. An IPv4 frame is not captured with an *other* trigger. The same logic applies to IPv6. The most commonly used triggers according to the frame−type are shown in this table:

IPv4	*IPv6*	*All Frame Types*

• SMAC
• DMAC
• IP_SA
• IP_DA
• IP_TTL
• IP_TOS
• L3_PT (ICMP,IGMP,TCP,UDP)
   ♦ TCP_SPORT, TCP_DPORT
   ♦ UDP_DPORT, UDP_SPORT
   ♦ ICMP_TYPE
	• SMAC
	• DMAC
	• IP6_SA
	• IP6_DA
	• IP6_TTL
	• IP6_CLASS
	• L3_PT (ICMP, IGMP, TCP, UDP)
	   ♦ IP6_L4DATA
		• VLAN
		• SRC_INDEX
		• DST_INDEX

Most of these fields should be self−explanatory. For example, *SMAC* and *DMAC* refer to the Source MAC address and the Destination MAC address, *IP_SA* and *IP_DA* refer to the Source IPv4 address and the Destination IPv4 address, and *L3_PT* refers to the L3 Protocol, which can be Internet Control Message Protocol (ICMP), Internet Group Management Protocol (IGMP), TCP, or UDP.

Note: An *other* trigger requires the user to provide the exact hex data and mask for the frame in question, and is outside of the scope of this document.

For this example, the frame is captured according to the source and destination IPv4 address. Remember that ELAM triggers allow various levels of specificity. Therefore, you can use additional fields, such as Time To Live (TTL), Type of Service (TOS), and Layer3 Protocol Type (L3_PT), if needed.

Eureka requires that triggers are set for the DBUS and the RBUS. There are two different Packet Buffers (PB) in which the RBUS data can reside. Determination of the correct PB instance is dependent upon the exact module type and ingress port. Typically, it is recommended that you configure PB1, and if the RBUS does not trigger, then repeat the configuration with PB2. If no RBUS trigger is provided, Cisco IOS® automatically creates a trigger on PB1.

Here is the DBUS trigger:

```
Sup2T# show platform capture elam trigger master eu50 dbus
  dbi ingress ipv4 if ip_sa=10.1.117.231 ip_da=10.1.117.1
```

Here is the RBUS trigger:

```
Sup2T#show platform capture elam trigger slave eu50 rbus rbi pb2
  New eu50 slave ELAM is RBI_PB2
```

In this example, *eu50* is used as the ELAM ASIC. This is because ASIC *Eureka* was selected on slot 5, instance zero.

Also, RBUS *PB2* was selected because, internally, you know that the RBUS for this particular example is in PB2. If the incorrect instance is chosen, then Cisco IOS provides this error message when you attempt to view the ELAM:

```
No SOP found or invalid Seq_Num. Pls try other PB interface:
sh pla cap elam tri s eu50 r r pb2
```

Start the Capture

Now that the ingress FE is selected and you configured the trigger, you can start the capture:

```
Sup2T#show platform capture elam start
```

In order to check the status of the ELAM, enter the *status* command:

```
Sup2T#show platform capture elam status
ID#    Role  ASIC     Slot  Inst  Ver  ELAM       Status
-----  ----  -------  ----  ----  ---  ---------  ------
eu50   M     EUREKA   5     0     1.3  DBI_ING    In Progress
eu50   s     EUREKA   5     0     1.3  RBI_PB2    In Progress
ID#    ELAM      Trigger
-----  --------- ----------
eu50   DBI_ING   FORMAT=IP L3_PROTOCOL=IPV4 IP_SA=10.1.117.231 IP_DA=10.1.117.1
eu50   RBI_PB2   TRIG=1
```

Once the frame that matches the trigger is received by the FE, the ELAM status shows as *completed*:

```
Sup2T#show platform capture elam status
ID#    Role  ASIC     Slot  Inst  Ver  ELAM       Status
-----  ----  -------  ----  ----  ---  ---------  ------
eu50   M     EUREKA   5     0     1.3  DBI_ING    Capture Completed
eu50   s     EUREKA   5     0     1.3  RBI_PB2    Capture Completed
ID#    ELAM      Trigger
-----  --------- ----------
eu50   DBI_ING   FORMAT=IP L3_PROTOCOL=IPV4 IP_SA=10.1.117.231 IP_DA=10.1.117.1
```

Interpret the Results

In order to display the ELAM results, enter the *data* command. Here is an excerpt of the ELAM data output that is most relevant to this example:

```
Sup2T#show platform capture elam data
(some output omitted)

DBUS:
VLAN .......................... [12] = 10
SRC_INDEX ..................... [19] = 0x102
DMAC .......................... = b414.8961.3780
SMAC .......................... = 0025.84e6.8dc1
L3_PROTOCOL ................... [4] = 0 [IPV4]
L3_PT ......................... [8] = 1 [ICMP]
IP_TTL ........................ [8] = 255
IP_SA ......................... = 10.1.117.231
IP_DA ......................... = 10.1.117.1

RBUS:
FLOOD ......................... [1] = 0
DEST_INDEX .................... [19] = 0x101
VLAN .......................... [12] = 20
IP_TTL ........................ [8] = 254
REWRITE_INFO
 i0  - replace bytes from ofs 0 to ofs 11 with seq
  '00 00 0C 07 AC CA B4 14 89 61 37 80'.
```

With the *DBUS* data, you can verify that the frame is received on VLAN 10 with a source MAC address of *0025.84e6.8dc1* and a destination MAC address of *b414.8961.3780*. You can also see that this is an IPv4 frame that is sourced from *10.1.117.231*, and is destined to *10.1.117.1*.

Tip: There are several other useful fields that are not included in this output, such as TOS value, IP flags, IP length, and L2 frame length.

In order to verify on which port the frame is received, enter the *SRC_INDEX* command (the source Local Target Logic (LTL)). Enter this command in order to map an LTL to a port or group of ports for Sup2T:

```
Sup2T#show platform hardware ltl index 0x102
LTL index 0x102 contain ports :
========================================
Gi5/3
```

The output shows that the *SRC_INDEX* of *0x102* maps to port *G5/3*. This confirms that the frame is received on port *G5/3*.

With the *RBUS* data, you can verify that the frame is routed to VLAN 20, and that the TTL is decremented from *255* in the *DBUS* data to *254* in the *RBUS*. The *REWRITE_INFO* from the output shows that the FE replaces bytes 0 through 11 (the first 12 bytes) that represent the MAC address rewrite for the destination and source MAC addresses. Additionally, you can verify from the *DEST_INDEX* (destination LTL) information where the frame is sent.

```
Sup2T#show platform hardware ltl index 0x101
LTL index 0x101 contain ports :
========================================
Gi5/2
```

The output shows that the *DEST_INDEX* of *0x101* maps to port *G5/2*. This confirms that the frame is sent to port *G5/2*.