

Configure and Verify Secure Firewall and Firepower Internal Switch Captures

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[High-Level Overview of the System Architecture](#)

[High-Level Overview of the Internal Switch Operations](#)

[Packet Flow and Capture Points](#)

[Configuration and Verification on Firepower 4100/9300](#)

[Packet Capture on a Physical or Port-channel Interface](#)

[Packet Captures on Backplane Interfaces](#)

[Packet Captures on Application and Application Ports](#)

[Packet Capture on a Subinterface of a Physical or Port-channel Interface](#)

[Packet Capture Filters](#)

[Collect Firepower 4100/9300 Internal Switch Capture Files](#)

[Guidelines, Limitations, and Best Practices for Internal Switch Packet Capture](#)

[Configuration and Verification on Secure Firewall 3100](#)

[Packet Capture on a Physical or Port-channel Interface](#)

[Packet Capture on a Subinterface of a Physical or Port-channel Interface](#)

[Packet Capture on Internal Interfaces](#)

[Packet Capture Filters](#)

[Collect Secure Firewall 3100 Internal Switch Capture Files](#)

[Guidelines, Limitations, and Best Practices for Internal Switch Packet Capture](#)

[Related Information](#)

Introduction

This document describes the configuration and verification of the Firepower, and the Secure Firewall internal switch captures.

Prerequisites

Requirements

Basic product knowledge, capture analysis.

Components Used

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

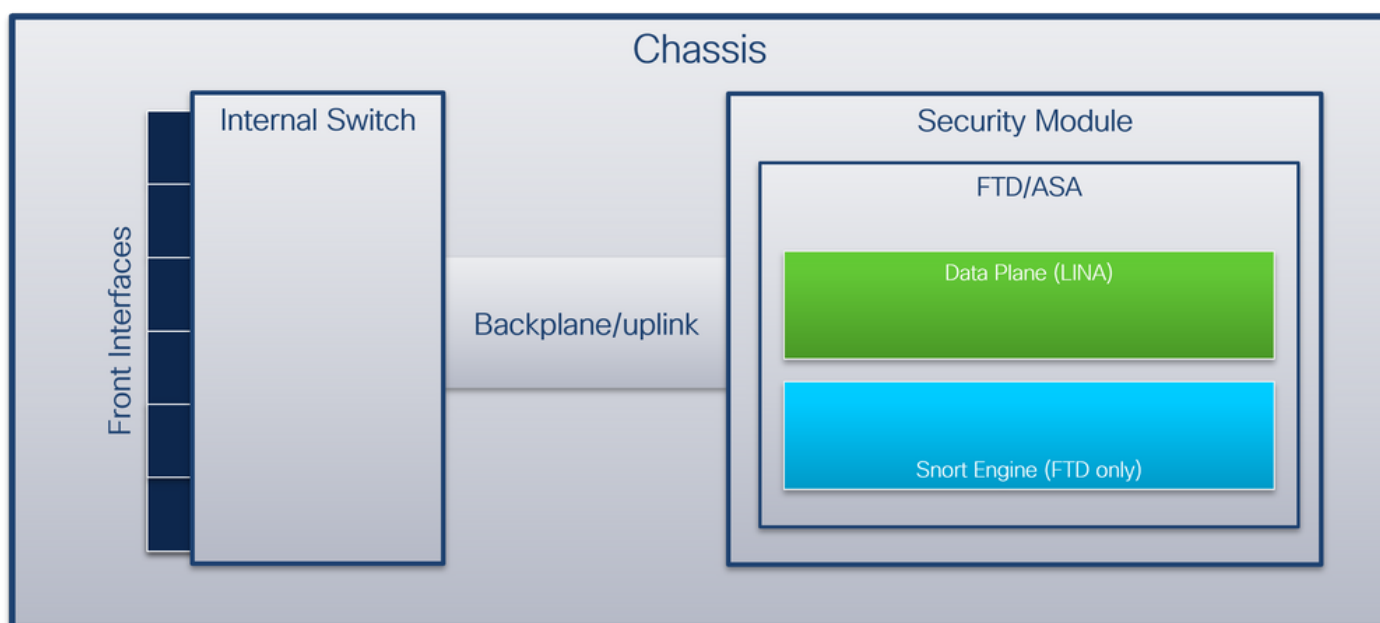
The information in this document is based on these software and hardware versions:

- Secure Firewall 31xx
- Firepower 41xx
- Firepower 93xx
- Cisco Secure eXtensible Operating System (FXOS) 2.12.0.x
- Cisco Secure Firewall Threat Defense (FTD) 7.2.0.x
- Cisco Secure Firewall Management Center (FMC) 7.2.0.x
- Cisco Secure Firewall Device Manager (FDM) 7.2.0.x
- Adaptive Security Appliance (ASA) 9.18(1)x
- Adaptive Security Appliance Device Manager (ASDM) 7.18.1.x
- Wireshark 3.6.7 (<https://www.wireshark.org/download.html>)

Background Information

High-Level Overview of the System Architecture

From the packet flow perspective, the architecture of the Firepower 4100/9300 and Secure Firewall 3100 can be visualized as shown in this figure:



The chassis includes these components:

- **Internal switch** – forwards packet from the network to the application and vice versa. The internal switch is connected to the **front interfaces** that reside on the built-in interface module or external network modules and connect to external devices, for example, switches. Examples of front interfaces are Ethernet 1/1, Ethernet 2/4, and so on. The “front” is not a strong technical definition. In this document, it is used to distinguish interfaces connected to external devices from the backplane or uplink interfaces.

- **Backplane or uplink** – an internal interface that connects the security module (SM) to the internal switch. This table shows backplane interfaces on Firepower 4100/9300 and uplink interface on Secure Firewall 3100:

Platform	Number of supported security modules	Backplane/uplink interfaces	Mapped application interfaces
Firepower 4100 (except Firepower 4110/4112)	1	SM1: Ethernet1/9 Ethernet1/10	Internal-Data0/0 Internal-Data0/1
Firepower 4110/4112	1	Ethernet1/9	Internal-Data0/0 Internal-Data0/0 Internal-Data0/1
Firepower 9300	3	SM1: Ethernet1/9 Ethernet1/10 SM2: Ethernet1/11 Ethernet1/12 SM3: Ethernet1/13 Ethernet1/14	Internal-Data0/0 Internal-Data0/1 Internal-Data0/0 Internal-Data0/1
Secure Firewall 3100	1	SM1: in_data_uplink1	Internal-Data0/1

In the case of 2 backplane interfaces per module, the internal switch and the applications on the modules perform traffic load-balancing over the 2 interfaces.

- **Security module, security engine, or blade** – the module where applications such as FTD or ASA are installed. Firepower 9300 supports up to 3 security modules.
- **Mapped application interface** - applications, such as FTD or ASA, map the backplane or uplink interfaces to internal interfaces. In other words, the backplane or uplink interfaces are visible as internal interfaces in applications.

Use the **show interface detail** command to verify internal interfaces:

```
> show interface detail | grep Interface
Interface Internal-Control0/0 "ha_ctl_nlp_int_tap", is up, line protocol is up
  Control Point Interface States:
    Interface number is 6
    Interface config status is active
    Interface state is active
Interface Internal-Data0/0 "", is up, line protocol is up
  Control Point Interface States:
    Interface number is 2
    Interface config status is active
    Interface state is active
Interface Internal-Data0/1 "", is up, line protocol is up
  Control Point Interface States:
    Interface number is 3
    Interface config status is active
    Interface state is active
Interface Internal-Data0/2 "nlp_int_tap", is up, line protocol is up
  Control Point Interface States:
    Interface number is 4
    Interface config status is active
```

```
Interface state is active
Interface Internal-Data0/3 "ccl_ha_nlp_int_tap", is up, line protocol is up
Control Point Interface States:
  Interface number is 5
  Interface config status is active
  Interface state is active
Interface Internal-Data0/4 "cmi_mgmt_int_tap", is up, line protocol is up
Control Point Interface States:
  Interface number is 7
  Interface config status is active
  Interface state is active
Interface Port-channel6.666 "", is up, line protocol is up
Interface Ethernet1/1 "diagnostic", is up, line protocol is up
Control Point Interface States:
  Interface number is 8
  Interface config status is active
  Interface state is active
```

High-Level Overview of the Internal Switch Operations

Firepower 4100/9300

To make a forwarding decision the internal switch uses an **interface VLAN tag**, or **port VLAN tag**, and a **virtual network tag (VN-tag)**.

The port VLAN tag is used by the internal switch to identify an interface. The switch inserts the port VLAN tag into each ingress packet that came on front interfaces. The VLAN tag is automatically configured by the system and cannot be manually changed. The tag value can be checked in the **fxos** command shell:

```
firepower# connect fxos
...
firepower(fxos)# show run int e1/2
!Command: show running-config interface Ethernet1/2
!Time: Tue Jul 12 22:32:11 2022

version 5.0(3)N2(4.120)

interface Ethernet1/2
  description U: Uplink
  no lldp transmit
  no lldp receive
  no cdp enable
  switchport mode dot1q-tunnel
  switchport trunk native vlan 102
  speed 1000
  duplex full
  uddl disable
  no shutdown
```

The VN-tag is also inserted by the internal switch and used to forward the packets to the application. It is automatically configured by the system and cannot be manually changed.

The port VLAN tag and the VN-tag are shared with the application. The application inserts the respective egress interface VLAN tags and the VN-tags into each packet. When a packet from the application is received by the internal switch on the backplane interfaces, the switch reads the egress interface VLAN tag and the VN-tag, identifies the application and the egress interface, strips the port VLAN tag and the VN-tag, and forwards the packet to the network.

Secure Firewall 3100

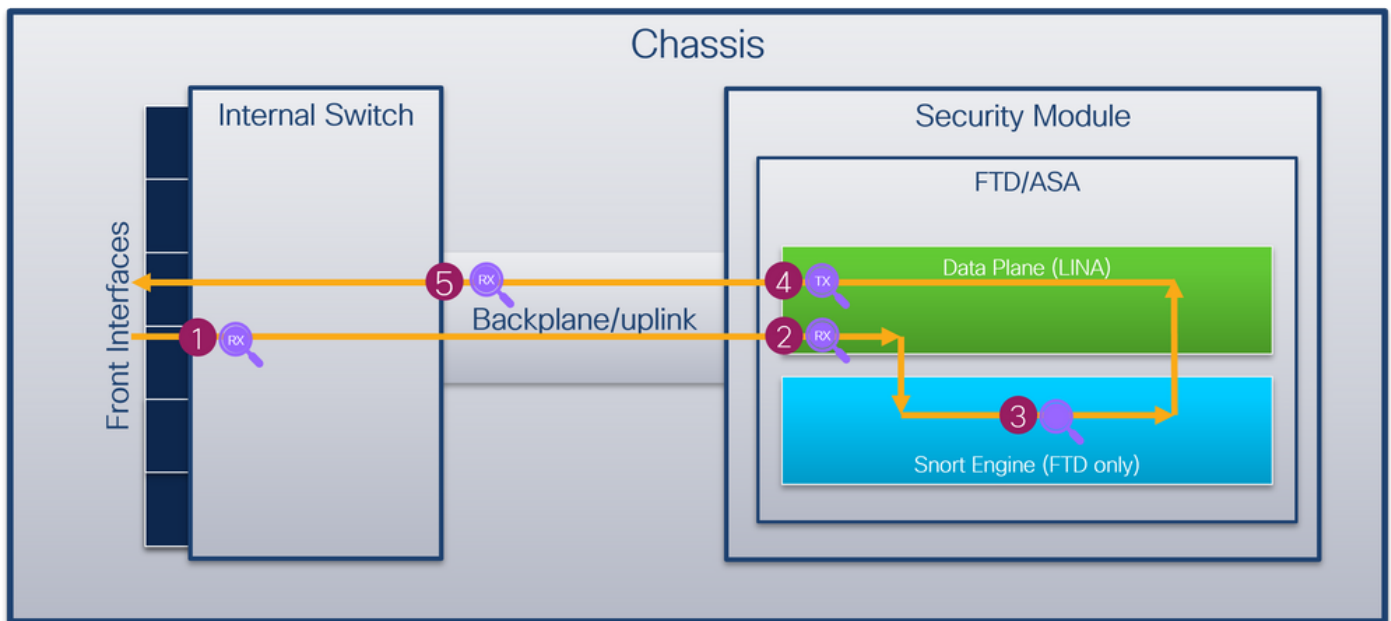
Like in Firepower 4100/9300, the port VLAN tag is used by the internal switch to identify an interface.

The port VLAN tag is shared with the application. The application inserts the respective egress interface VLAN tags into each packet. When a packet from the application is received by the internal switch on the uplink interface, the switch reads the egress interface VLAN tag, identifies the egress interface, strips the port VLAN tag, and forwards the packet to the network.

Packet Flow and Capture Points

The Firepower 4100/9300 and the Secure Firewall 3100 firewalls support packet captures on the interfaces of the internal switch.

This figure shows the packet capture points along the packet path within the chassis and the application:



The capture points are:

1. Internal switch front interface ingress capture point. A front interface is any interface connected to the peer devices such as switches.
2. Data plane interface ingress capture point
3. Snort capture point
4. Data plane interface egress capture point
5. Internal switch backplane or uplink ingress capture point. A backplane or uplink interface connects the internal switch to the application.

The internal switch supports only ingress interface captures. That is only the packets received from the network or from the ASA/FTD application can be captured. **Egress packet captures are not supported.**

Configuration and Verification on Firepower 4100/9300

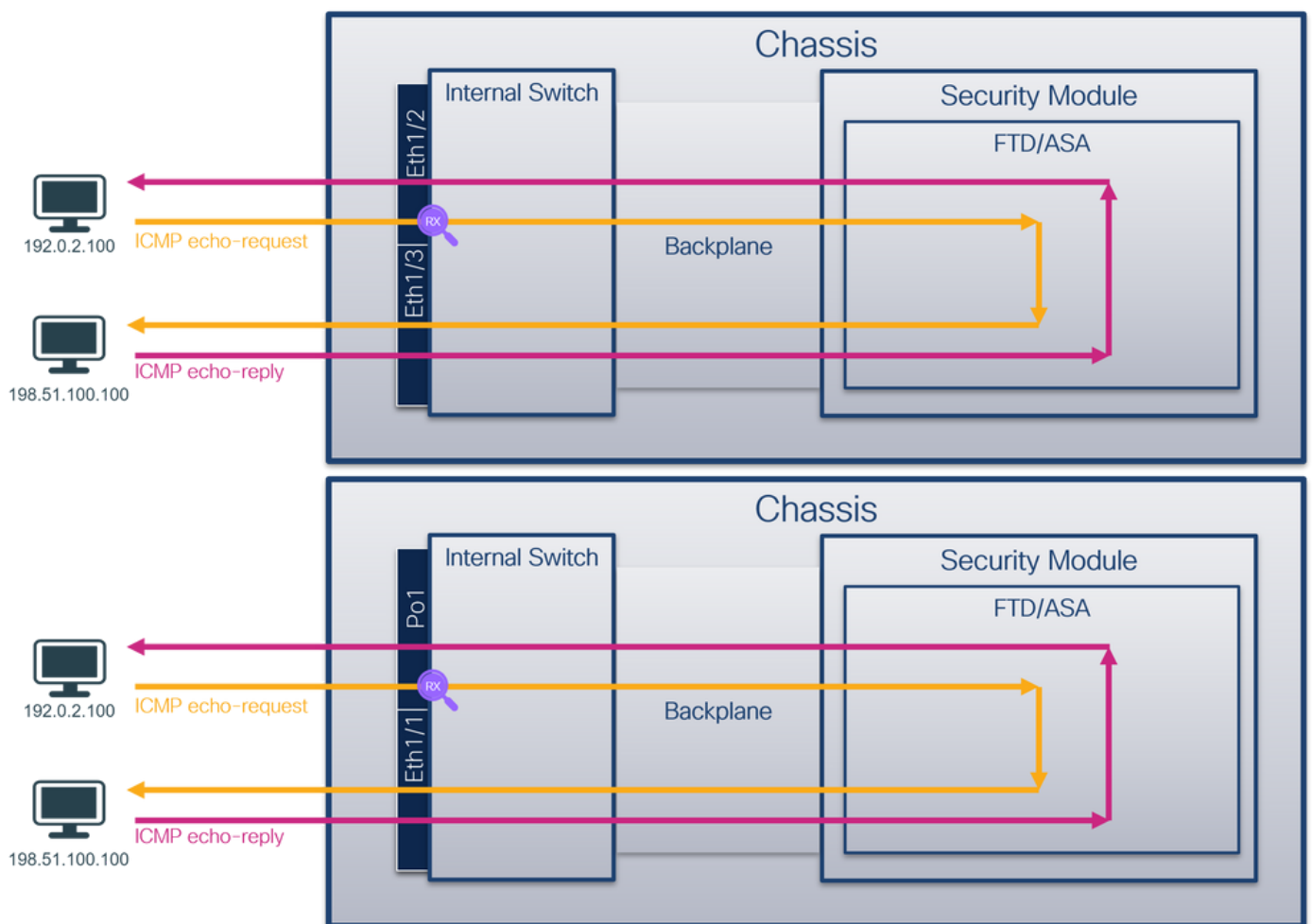
The Firepower 4100/9300 internal switch captures can be configured in **Tools > Packet Capture** on FCM or in **scope packet-capture** in FXOS CLI. For the description of the packet capture options refer to the *Cisco Firepower 4100/9300 FXOS Chassis Manager Configuration Guide* or *Cisco Firepower 4100/9300 FXOS CLI Configuration Guide*, chapter **Troubleshooting**, section **Packet Capture**.

These scenarios cover common use cases of Firepower 4100/9300 internal switch captures.

Packet Capture on a Physical or Port-channel Interface

Use the FCM and CLI to configure and verify a packet capture on interface Ethernet1/2 or Portchannel1 interface. In the case of a port-channel interface, ensure to select all physical member interfaces.

Topology, packet flow, and the capture points

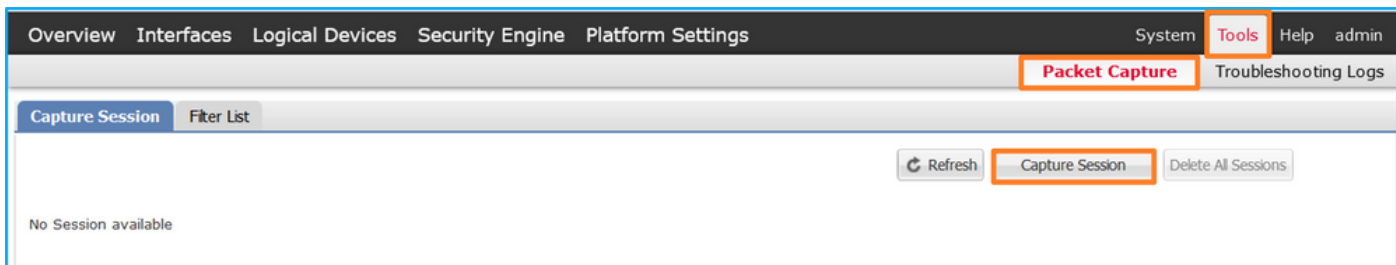


Configuration

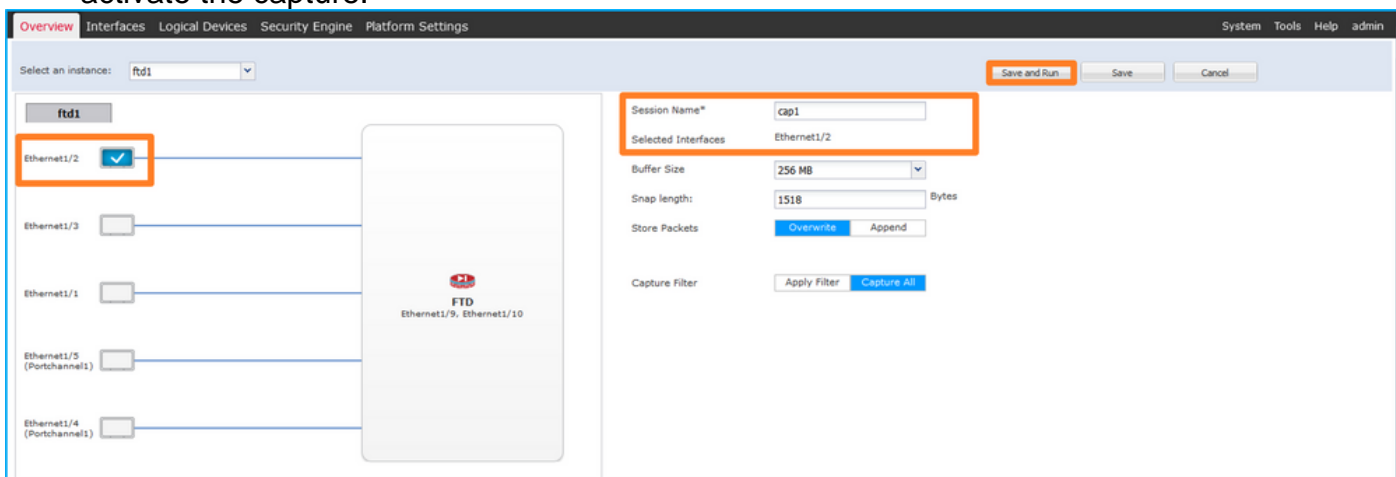
FCM

Follow these steps on FCM to configure a packet capture on interfaces Ethernet1/2 or Portchannel1:

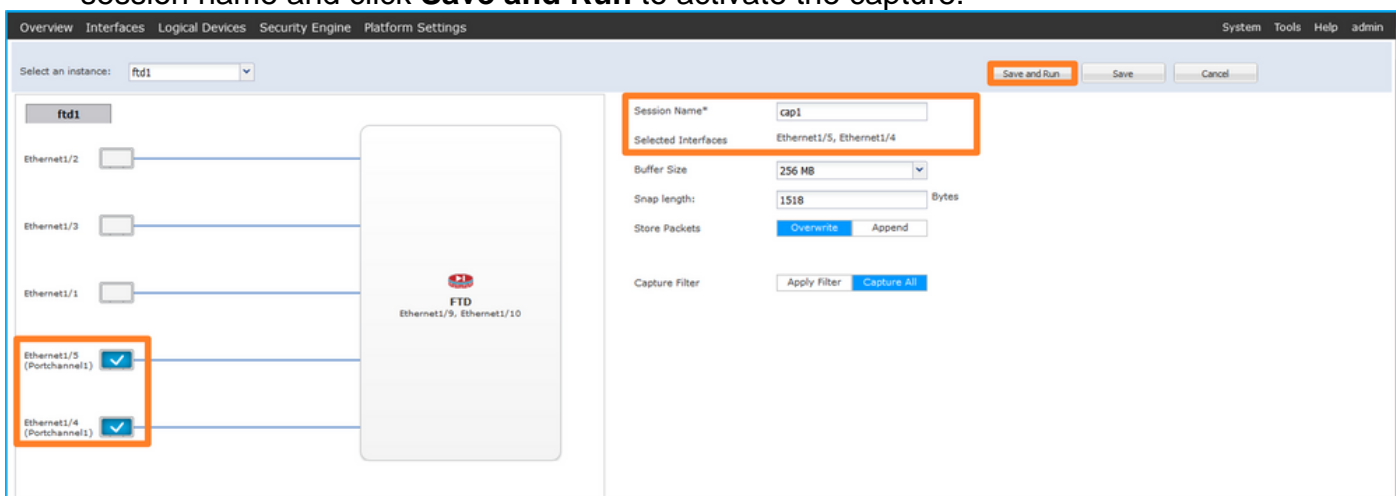
1. Use **Tools > Packet Capture > Capture Session** to create a new capture session:



2. Select the interface **Ethernet1/2**, provide the session name and click **Save and Run** to activate the capture:



3. In the case of a port-channel interface, select all physical member interfaces, provide the session name and click **Save and Run** to activate the capture:



FXOS CLI

Follow these steps on FXOS CLI to configure a packet capture on interfaces Ethernet1/2 or Portchannel1:

1. Identify the application type and identifier:

```
firepower# scope ssa
firepower /ssa # show app-instance
App Name  Identifier Slot ID  Admin State Oper State  Running Version Startup Version
Deploy Type Turbo Mode Profile Name Cluster State  Cluster Role
-----
ftd      ftd1      1      Enabled  Online      7.2.0.82      7.2.0.82
```

Native No Not Applicable None

2. In the case of a port-channel interface, identify its member interfaces:

```
firepower# connect fxos
<output skipped>
firepower(fxos)# show port-channel summary
Flags: D - Down          P - Up in port-channel (members)
       I - Individual    H - Hot-standby (LACP only)
       s - Suspended     r - Module-removed
       S - Switched     R - Routed
       U - Up (port-channel)
       M - Not in use. Min-links not met
```

```
-----
Group Port-      Type      Protocol  Member Ports
Channel
-----
1    Po1(SU)     Eth      LACP      Eth1/4(P)  Eth1/5(P)
```

3. Create a capture session:

```
firepower# scope packet-capture
firepower /packet-capture # create session cap1
firepower /packet-capture/session* # create phy-port Eth1/2
firepower /packet-capture/session/phy-port* # set app ftd
firepower /packet-capture/session/phy-port* # set app-identifier ftd1
firepower /packet-capture/session/phy-port* # up
firepower /packet-capture/session* # enable
firepower /packet-capture/session* # commit
firepower /packet-capture/session #
```

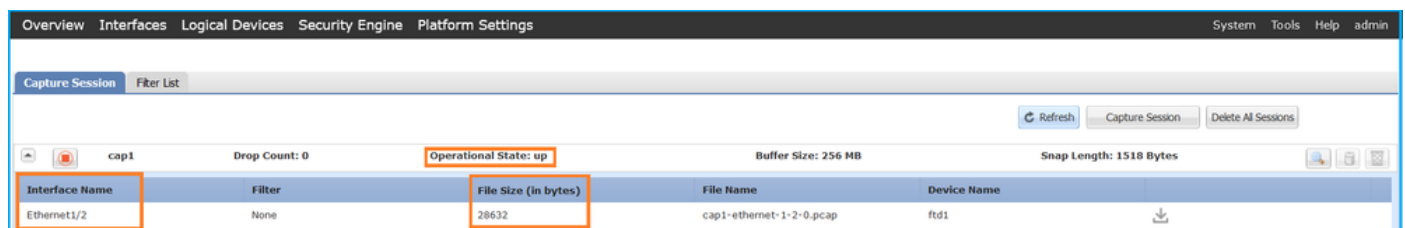
For port-channel interfaces, a separate capture for each member interface is configured:

```
firepower# scope packet-capture
firepower /packet-capture # create session cap1
firepower /packet-capture/session* # create phy-port Eth1/4
firepower /packet-capture/session/phy-port* # set app ftd
firepower /packet-capture/session/phy-port* # set app-identifier ftd1
firepower /packet-capture/session/phy-port* # up
firepower /packet-capture/session* # create phy-port Eth1/5
firepower /packet-capture/session/phy-port* # set app ftd
firepower /packet-capture/session/phy-port* # set app-identifier ftd1
firepower /packet-capture/session/phy-port* # up
firepower /packet-capture/session* # enable
firepower /packet-capture/session* # commit
firepower /packet-capture/session #
```

Verification

FCM

Verify the **Interface Name**, ensure that the **Operational Status** is up and that the **File Size (in bytes)** increases:



The screenshot shows the FCM interface with a table of capture sessions. The table has columns for Interface Name, Filter, File Size (in bytes), File Name, and Device Name. The first row shows a session named 'cap1' on 'Ethernet1/2' with a file size of 28632 bytes. The 'Operational State' is 'up'.

Interface Name	Filter	File Size (in bytes)	File Name	Device Name
Ethernet1/2	None	28632	cap1-ethernet-1-2-0.pcap	ftd1

Portchannel1 with member interfaces Ethernet1/4 and Ethernet1/5:

Interface Name	Filter	Operational State	File Size (in bytes)	File Name	Device Name
Ethernet1/5	None	up	160	cap1-ethernet-1-5-0.pcap	fd1
Ethernet1/4	None	up	85000	cap1-ethernet-1-4-0.pcap	fd1

FXOS CLI

Verify the capture details in **scope packet-capture**:

```
firepower# scope packet-capture  
firepower /packet-capture # show session cap1
```

Traffic Monitoring Session:

```
Packet Capture Session Name: cap1  
Session: 1  
Admin State: Enabled  
Oper State: Up  
Oper State Reason: Active  
Config Success: Yes  
Config Fail Reason:  
Append Flag: Overwrite  
Session Mem Usage: 256 MB  
Session Pcap Snap Len: 1518 Bytes  
Error Code: 0  
Drop Count: 0
```

Physical ports involved in Packet Capture:

```
Slot Id: 1  
Port Id: 2  
Pcapfile: /workspace/packet-capture/session-1/cap1-ethernet-1-2-0.pcap  
Pcapsize: 75136 bytes  
Filter:  
Sub Interface: 0  
Application Instance Identifier: ftd1  
Application Name: ftd
```

Port-channel 1 with member interfaces Ethernet1/4 and Ethernet1/5:

```
firepower# scope packet-capture  
firepower /packet-capture # show session cap1
```

Traffic Monitoring Session:

```
Packet Capture Session Name: cap1  
Session: 1  
Admin State: Enabled  
Oper State: Up  
Oper State Reason: Active  
Config Success: Yes  
Config Fail Reason:  
Append Flag: Overwrite  
Session Mem Usage: 256 MB  
Session Pcap Snap Len: 1518 Bytes  
Error Code: 0  
Drop Count: 0
```

Physical ports involved in Packet Capture:

Slot Id: 1
 Port Id: 4
 Pcapfile: /workspace/packet-capture/session-1/cap1-ethernet-1-4-0.pcap
 Pcapsize: 310276 bytes
 Filter:
 Sub Interface: 0
 Application Instance Identifier: ftd1
 Application Name: ftd

Slot Id: 1
 Port Id: 5
 Pcapfile: /workspace/packet-capture/session-1/cap1-ethernet-1-5-0.pcap
 Pcapsize: 160 bytes
 Filter:
 Sub Interface: 0
 Application Instance Identifier: ftd1
 Application Name: ftd

Collect capture files

Follow the steps in the section **Collect Firepower 4100/9300 Internal Switch Capture Files**.

Capture file analysis

Use a packet capture file reader application to open the capture file for Ethernet1/2. Select the first packet and check the key points:

1. Only ICMP echo-request packets are captured. Each packet is captured and shown 2 times.
2. The original packet header is without the VLAN tag.
3. The internal switch inserts additional port VLAN tag **102** that identifies the ingress interface Ethernet1/2.
4. The internal switch inserts an additional VN tag.

The screenshot shows a network traffic capture analysis. The top part is a table of captured packets:

No.	Time	Source	Destination	Protocol	Length	IP ID	IP TTL	Info
1	2022-07-13 06:23:58.285080930	192.0.2.100	198.51.100.100	ICMP	108	0x9dec (40428)	64	Echo (ping) request id=0x001a, seq=7/1792, ttl=64 (no response found!)
2	2022-07-13 06:23:58.285082858	192.0.2.100	198.51.100.100	ICMP	102	0x9dec (40428)	64	Echo (ping) request id=0x001a, seq=7/1792, ttl=64 (no response found!)
3	2022-07-13 06:23:59.309048886	192.0.2.100	198.51.100.100	ICMP	108	0x9ed0 (40656)	64	Echo (ping) request id=0x001a, seq=8/2048, ttl=64 (no response found!)
4	2022-07-13 06:23:59.309193731	192.0.2.100	198.51.100.100	ICMP	102	0x9ed0 (40656)	64	Echo (ping) request id=0x001a, seq=8/2048, ttl=64 (no response found!)
5	2022-07-13 06:24:00.333054190	192.0.2.100	198.51.100.100	ICMP	108	0xc9f20 (40736)	64	Echo (ping) request id=0x001a, seq=9/2304, ttl=64 (no response found!)
6	2022-07-13 06:24:00.333056034	192.0.2.100	198.51.100.100	ICMP	102	0xc9f20 (40736)	64	Echo (ping) request id=0x001a, seq=9/2304, ttl=64 (no response found!)
7	2022-07-13 06:24:01.357173530	192.0.2.100	198.51.100.100	ICMP	108	0xf0f2d (40749)	64	Echo (ping) request id=0x001a, seq=10/2560, ttl=64 (no response found!)
8	2022-07-13 06:24:01.357174708	192.0.2.100	198.51.100.100	ICMP	102	0xf0f2d (40749)	64	Echo (ping) request id=0x001a, seq=10/2560, ttl=64 (no response found!)
9	2022-07-13 06:24:02.381073741	192.0.2.100	198.51.100.100	ICMP	108	0xf0f88 (40840)	64	Echo (ping) request id=0x001a, seq=11/2816, ttl=64 (no response found!)
10	2022-07-13 06:24:02.381074999	192.0.2.100	198.51.100.100	ICMP	102	0xf0f88 (40840)	64	Echo (ping) request id=0x001a, seq=11/2816, ttl=64 (no response found!)
11	2022-07-13 06:24:03.405199041	192.0.2.100	198.51.100.100	ICMP	108	0xa077 (41079)	64	Echo (ping) request id=0x001a, seq=12/3072, ttl=64 (no response found!)
12	2022-07-13 06:24:03.405200261	192.0.2.100	198.51.100.100	ICMP	102	0xa077 (41079)	64	Echo (ping) request id=0x001a, seq=12/3072, ttl=64 (no response found!)
13	2022-07-13 06:24:04.429156631	192.0.2.100	198.51.100.100	ICMP	108	0xa10f (41231)	64	Echo (ping) request id=0x001a, seq=13/3328, ttl=64 (no response found!)
14	2022-07-13 06:24:04.429156831	192.0.2.100	198.51.100.100	ICMP	102	0xa10f (41231)	64	Echo (ping) request id=0x001a, seq=13/3328, ttl=64 (no response found!)
15	2022-07-13 06:24:05.453156612	192.0.2.100	198.51.100.100	ICMP	108	0xa16a (41322)	64	Echo (ping) request id=0x001a, seq=14/3584, ttl=64 (no response found!)
16	2022-07-13 06:24:05.453158052	192.0.2.100	198.51.100.100	ICMP	102	0xa16a (41322)	64	Echo (ping) request id=0x001a, seq=14/3584, ttl=64 (no response found!)
17	2022-07-13 06:24:06.477127687	192.0.2.100	198.51.100.100	ICMP	108	0xa1e9 (41449)	64	Echo (ping) request id=0x001a, seq=15/3840, ttl=64 (no response found!)
18	2022-07-13 06:24:06.477129899	192.0.2.100	198.51.100.100	ICMP	102	0xa1e9 (41449)	64	Echo (ping) request id=0x001a, seq=15/3840, ttl=64 (no response found!)
19	2022-07-13 06:24:07.501291314	192.0.2.100	198.51.100.100	ICMP	108	0xa1f6 (41462)	64	Echo (ping) request id=0x001a, seq=16/4096, ttl=64 (no response found!)
20	2022-07-13 06:24:07.501293041	192.0.2.100	198.51.100.100	ICMP	102	0xa1f6 (41462)	64	Echo (ping) request id=0x001a, seq=16/4096, ttl=64 (no response found!)
21	2022-07-13 06:24:08.525089956	192.0.2.100	198.51.100.100	ICMP	108	0xa257 (41559)	64	Echo (ping) request id=0x001a, seq=17/4352, ttl=64 (no response found!)
22	2022-07-13 06:24:08.525092088	192.0.2.100	198.51.100.100	ICMP	102	0xa257 (41559)	64	Echo (ping) request id=0x001a, seq=17/4352, ttl=64 (no response found!)
23	2022-07-13 06:24:09.549236500	192.0.2.100	198.51.100.100	ICMP	108	0xa2a9 (41641)	64	Echo (ping) request id=0x001a, seq=18/4608, ttl=64 (no response found!)
24	2022-07-13 06:24:09.549238564	192.0.2.100	198.51.100.100	ICMP	102	0xa2a9 (41641)	64	Echo (ping) request id=0x001a, seq=18/4608, ttl=64 (no response found!)
25	2022-07-13 06:24:10.573110146	192.0.2.100	198.51.100.100	ICMP	108	0xa345 (41797)	64	Echo (ping) request id=0x001a, seq=19/4864, ttl=64 (no response found!)
26	2022-07-13 06:24:10.573112504	192.0.2.100	198.51.100.100	ICMP	102	0xa345 (41797)	64	Echo (ping) request id=0x001a, seq=19/4864, ttl=64 (no response found!)
27	2022-07-13 06:24:11.597086027	192.0.2.100	198.51.100.100	ICMP	108	0xa349 (41801)	64	Echo (ping) request id=0x001a, seq=20/5120, ttl=64 (no response found!)
28	2022-07-13 06:24:11.597088170	192.0.2.100	198.51.100.100	ICMP	102	0xa349 (41801)	64	Echo (ping) request id=0x001a, seq=20/5120, ttl=64 (no response found!)
29	2022-07-13 06:24:12.621061022	192.0.2.100	198.51.100.100	ICMP	108	0xa3dc (41948)	64	Echo (ping) request id=0x001a, seq=21/5376, ttl=64 (no response found!)

The bottom part of the screenshot shows the detailed headers for the second packet (Frame 1):

```

> Ethernet II, Src: VMware 9d:e8:be (00:50:56:9d:e8:be), Dst: Cisco b9:77:0e (58:97:bd:b9:77:0e)
  > VN-Tag
    1... .. = Direction: From Bridge
    .0... .. = Pointer: vif1d
    ..00 0000 0000 1010 .. .. = Destination: 10
    .. .. = Looped: No
    .. .. = Reserved: 0
    .. .. = Version: 0
    .. .. = Source: 0
    Type: 802.1Q Virtual LAN (0x8100)
  > 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 102
    000... .. = Priority: Best Effort (default) (0)
    ..0... .. = DEI: Ineligible
    .. 0000 0110 0110 = ID: 102
    Type: IPv4 (0x8000)
  > Internet Protocol Version 4, Src: 192.0.2.100, Dst: 198.51.100.100
  > Internet Control Message Protocol
  
```

Select the second packet and check the key points:

1. Only ICMP echo-request packets are captured. Each packet is captured and shown 2 times.
2. The original packet header is without the VLAN tag.
3. The internal switch inserts additional port VLAN tag **102** that identifies the ingress interface Ethernet1/2.

No.	Time	Source	Destination	Protocol	Length	IP ID	TTL	Info
1	2022-07-13 06:23:58.285080930	192.0.2.100	198.51.100.100	ICMP	108	0x9d5c (40428)	64	Echo (ping) request id=0x001a, seq=7/1792, ttl=64 (no response found!)
2	2022-07-13 06:23:58.285082858	192.0.2.100	198.51.100.100	ICMP	102	0x9d5c (40428)	64	Echo (ping) request id=0x001a, seq=7/1792, ttl=64 (no response found!)


```

> Frame 2: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface capture_u0_1, id 0
  Ethernet II, Src: VMware 08:e:8e:be (00:50:56:9d:e8:be), Dst: Cisco 0a:77:72:0e (58:97:1d:b9:77:0e)
  802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 102
  0000..... = Priority: Best Effort (default) (0)
  ...0..... = DEI: Ineligible
  ....0000 0110 0110 = ID: 102
  Type: IPv4 (0x0800)
  Internet Protocol Version 4, Src: 192.0.2.100, Dst: 198.51.100.100
  Internet Control Message Protocol
  
```

Open the capture files for Portchannel1 member interfaces. Select the first packet and check the key points:

1. Only ICMP echo-request packets are captured. Each packet is captured and shown 2 times.
2. The original packet header is without the VLAN tag.
3. The internal switch inserts an additional port VLAN tag **1001** that identifies the ingress interface Portchannel1.
4. The internal switch inserts an additional VN tag.

No.	Time	Source	Destination	Protocol	Length	IP ID	TTL	Info
1	2022-08-05 23:07:31.865872877	192.0.2.100	198.51.100.100	ICMP	108	0x322e (12846)	64	Echo (ping) request id=0x002d, seq=245/62720, ttl=64 (no response found!)
2	2022-08-05 23:07:31.865875131	192.0.2.100	198.51.100.100	ICMP	102	0x322e (12846)	64	Echo (ping) request id=0x002d, seq=245/62720, ttl=64 (no response found!)


```

> Frame 1: 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interface capture_u0_3, id 0
  Ethernet II, Src: VMware 9d:e8:be (00:50:56:9d:e8:be), Dst: a2:76:f2:00:00:25 (a2:76:f2:00:00:25)
  VN-Tag
  1..... = Direction: From Bridge
  .0..... = Pointer: vif_id
  ..00 0000 0101 0100..... = Destination: 84
  ..... = Looped: No
  ..... = Reserved: 0
  .....00..... = Version: 0
  ....0000 0000 0000 = Source: 0
  Type: 802.1Q Virtual LAN (0x8100)
  802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 1001
  0000..... = Priority: Best Effort (default) (0)
  ...0..... = DEI: Ineligible
  ....0011 1110 1001 = ID: 1001
  Type: IPv4 (0x0800)
  Internet Protocol Version 4, Src: 192.0.2.100, Dst: 198.51.100.100
  Internet Control Message Protocol
  
```

Select the second packet and check the key points:

1. Only ICMP echo-request packets are captured. Each packet is captured and shown 2 times.
2. The original packet header is without the VLAN tag.
3. The internal switch inserts an additional port VLAN tag **1001** that identifies the ingress interface Portchannel1.

The image shows a Wireshark packet capture. The packet list pane displays 19 ICMP Echo (ping) request packets. Packet 1 and 2 are highlighted in blue and have their P ID field (0x322e) circled in orange. The details pane for packet 2 shows the Ethernet II header with a Virtual LAN tag (VLAN ID: 1001) and the Internet Protocol Version 4 header. The packet bytes pane shows the raw data of the packet.

Explanation

When a packet capture on a front interface is configured, the switch simultaneously captures each packet twice:

- After the insertion of the port VLAN tag.
- After the insertion of the VN tag.

In the order of operations, the VN tag is inserted at a later stage than the port VLAN tag insertion. However, in the capture file, the packet with the VN tag is shown earlier than the packet with the port VLAN tag.

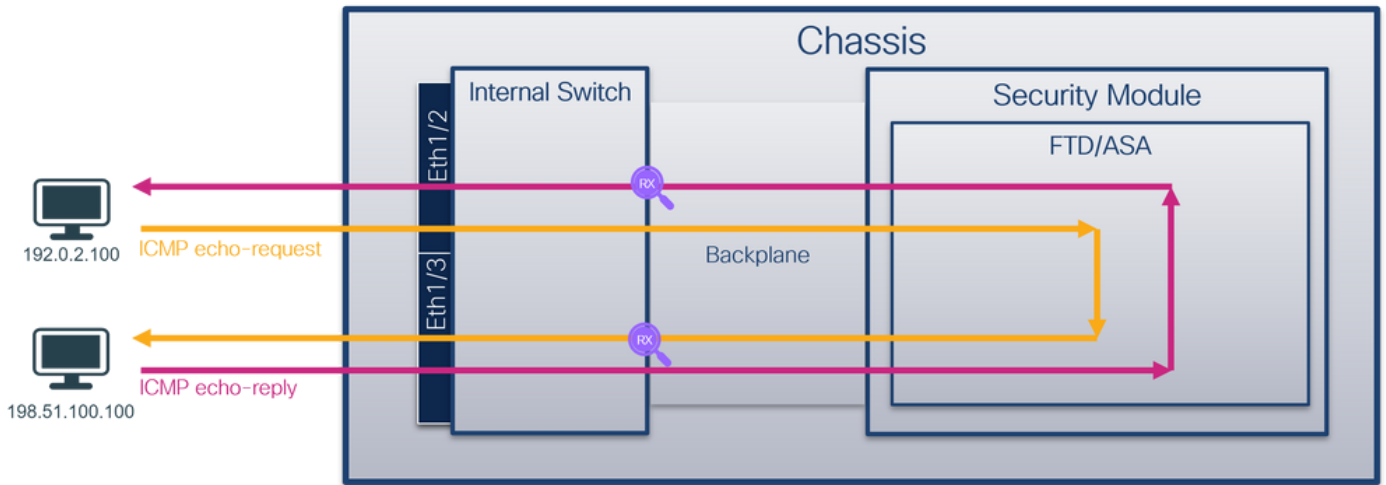
This table summarizes the task:

Task	Capture point	Internal port VLAN in captured packets	Direction	Captured traffic
Configure and verify a packet capture on interface Ethernet1/2	Ethernet1/2	102	Ingress only	ICMP echo requests from host 192.0.2.100 to host 198.51.100.1
Configure and verify a packet capture on interface Portchannel1 with member interfaces Ethernet1/4 and Ethernet1/5	Ethernet1/4 Ethernet1/5	1001	Ingress only	ICMP echo requests from host 192.0.2.100 to host 198.51.100.1

Packet Captures on Backplane Interfaces

Use the FCM and CLI to configure and verify a packet capture on backplane interfaces.

Topology, packet flow, and the capture points

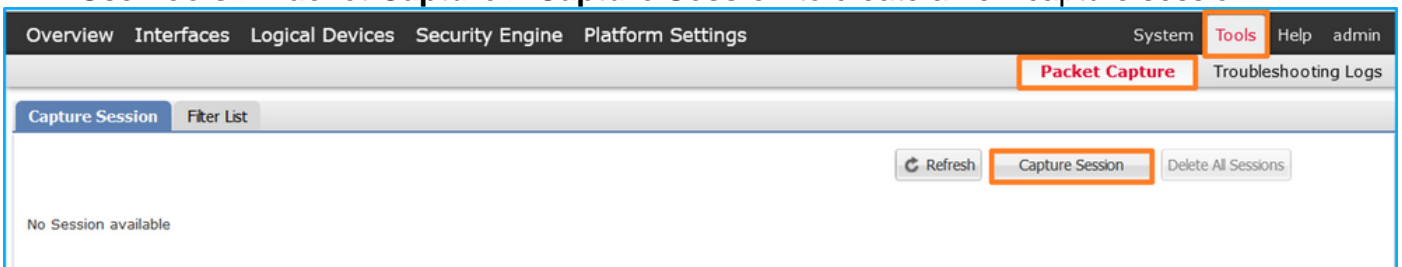


Configuration

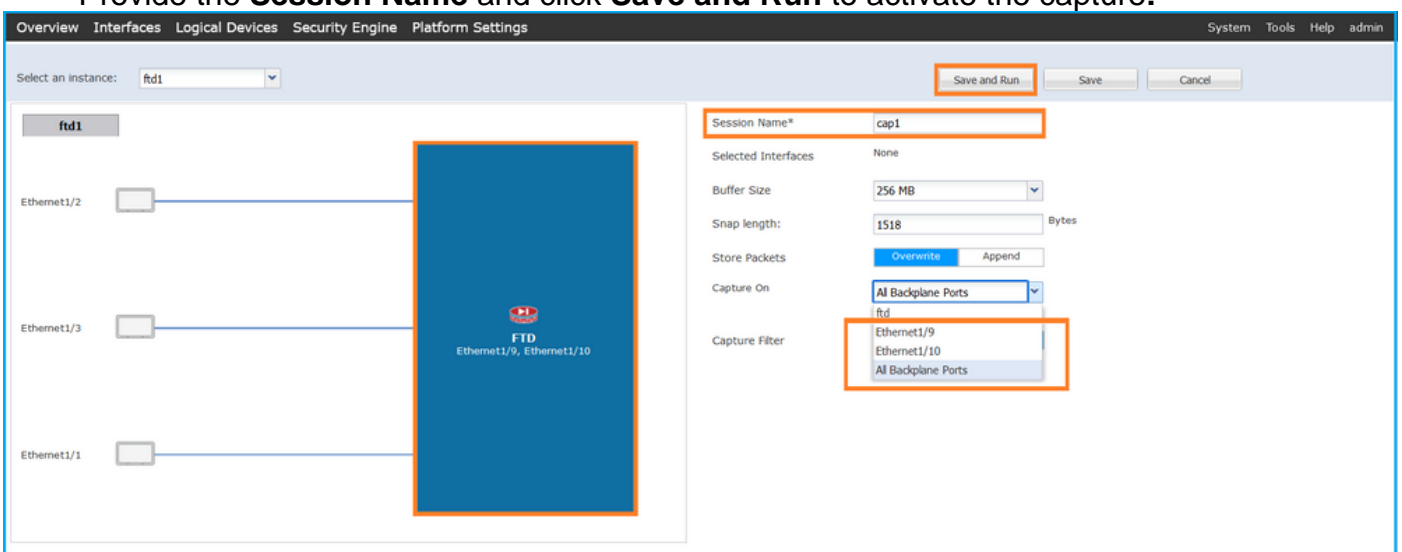
FCM

Follow these steps on FCM to configure packet captures on backplane interfaces:

1. Use **Tools > Packet Capture > Capture Session** to create a new capture session:



2. To capture packets on all backplane interfaces, select the application, then **All Backplane Ports** from the **Capture On** the dropdown list. Alternatively, choose the specific backplane interface. In this case, backplane interfaces Ethernet1/9 and Ethernet1/10 are available. Provide the **Session Name** and click **Save and Run** to activate the capture:



FXOS CLI

Follow these steps on FXOS CLI to configure packet captures on backplane interfaces:

1. Identify the application type and identifier:

```
firepower# scope ssa
firepower /ssa# show app-instance
App Name      Identifier Slot ID      Admin State Oper State      Running Version Startup Version
Deploy Type Turbo Mode Profile Name Cluster State   Cluster Role
-----
ftd           ftd1       1           Enabled      Online          7.2.0.82       7.2.0.82
Native       No                Not Applicable None
```

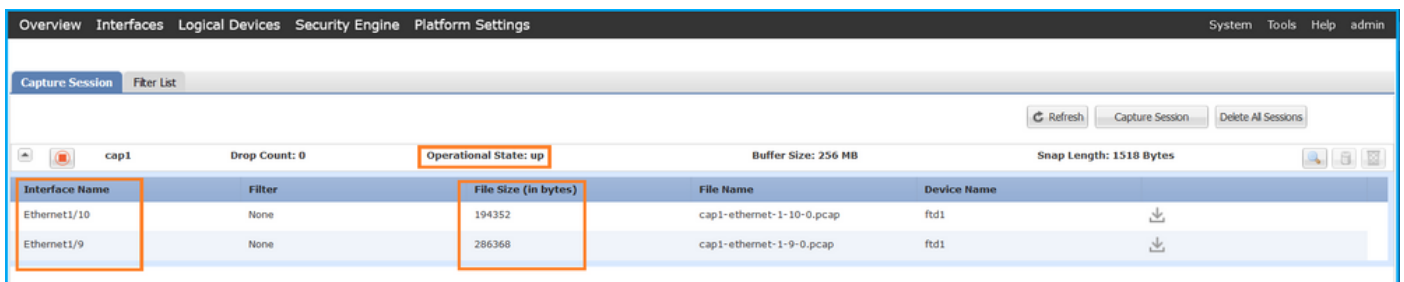
2. Create a capture session:

```
firepower# scope packet-capture
firepower /packet-capture # create session cap1
firepower /packet-capture/session* # create phy-port Eth1/9
firepower /packet-capture/session/phy-port* # set app ftd
firepower /packet-capture/session/phy-port* # set app-identifier ftd1
firepower /packet-capture/session/phy-port* # up
firepower /packet-capture/session* # create phy-port Eth1/10
firepower /packet-capture/session/phy-port* # set app ftd
firepower /packet-capture/session/phy-port* # set app-identifier ftd1
firepower /packet-capture/session/phy-port* # up
firepower /packet-capture/session* # enable
firepower /packet-capture/session* # commit
firepower /packet-capture/session #
```

Verification

FCM

Verify the **Interface Name**, ensure that the **Operational Status** is up and that the **File Size (in bytes)** increases:



The screenshot shows the Fortinet FCM interface with the 'Capture Session' tab selected. The session 'cap1' is active, with a drop count of 0 and an operational state of 'up'. The buffer size is 256 MB and the snap length is 1518 bytes. A table below lists the captured packets:

Interface Name	Filter	File Size (in bytes)	File Name	Device Name
Ethernet1/10	None	194352	cap1-ethernet-1-10-0.pcap	ftd1
Ethernet1/9	None	286368	cap1-ethernet-1-9-0.pcap	ftd1

FXOS CLI

Verify the capture details in **scope packet-capture**:

```
firepower# scope packet-capture
firepower /packet-capture # show session cap1
```

Traffic Monitoring Session:

```
Packet Capture Session Name: cap1
Session: 1
Admin State: Enabled
Oper State: Up
Oper State Reason: Active
Config Success: Yes
Config Fail Reason:
```

Append Flag: Overwrite
Session Mem Usage: 256 MB
Session Pcap Snap Len: 1518 Bytes
Error Code: 0
Drop Count: 0

Physical ports involved in Packet Capture:

Slot Id: 1
Port Id: 10
Pcapfile: /workspace/packet-capture/session-1/cap1-ethernet-1-10-0.pcap
Pcapsize: 1017424 bytes

Filter:

Sub Interface: 0

Application Instance Identifier: ftd1
Application Name: ftd

Slot Id: 1
Port Id: 9
Pcapfile: /workspace/packet-capture/session-1/cap1-ethernet-1-9-0.pcap
Pcapsize: 1557432 bytes

Filter:

Sub Interface: 0

Application Instance Identifier: ftd1
Application Name: ftd

Collect capture files

Follow the steps in the section **Collect Firepower 4100/9300 Internal Switch Capture Files**.

Capture file analysis

Use a packet capture file reader application to open the capture files. In the case of more than 1 backplane interface, ensure to open all capture files for each backplane interface. In this case, the packets are captured on the backplane interface Ethernet1/9.

Select the first and the second packets, and check the key points:

1. Each ICMP echo request packet is captured and shown 2 times.
2. The original packet header is without the VLAN tag.
3. The internal switch inserts additional port VLAN tag **103** that identifies the egress interface Ethernet1/3.
4. The internal switch inserts an additional VN tag.

Frame 1: 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interface capture_u0_8, id 0
 Ethernet II, Src: Cisco b9:77:2d (58:97:bd:b9:77:2d), Dst: Vmware 9d:e7:50 (00:50:56:9d:e7:50)

VN-Tag

```

0..... = Direction: To Bridge
..0..... = Pointer: vif_id
..00 0000 0000 0000 .. = Destination: 0
.....0..... = Looped: No
.....0..... = Reserved: 0
.....0..... = Version: 0
.....0000 0000 1010 = Source: 10
Type: 802.1Q Virtual LAN (0x8100)
802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 103
0000..... = Priority: Best Effort (default) (0)
...0..... = DEI: Ineligible
...0000 0110 0111 = ID: 103
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.0.2.100, Dst: 198.51.100.100
Internet Control Message Protocol
  
```

4
3
2

Frame 2: 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interface capture_u0_8, id 0
 Ethernet II, Src: Cisco b9:77:2d (58:97:bd:b9:77:2d), Dst: Vmware 9d:e7:50 (00:50:56:9d:e7:50)

VN-Tag

```

0..... = Direction: To Bridge
..0..... = Pointer: vif_id
..00 0000 0000 0000 .. = Destination: 0
.....0..... = Looped: No
.....0..... = Reserved: 0
.....0..... = Version: 0
.....0000 0000 1010 = Source: 10
Type: 802.1Q Virtual LAN (0x8100)
802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 103
0000..... = Priority: Best Effort (default) (0)
...0..... = DEI: Ineligible
...0000 0110 0111 = ID: 103
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.0.2.100, Dst: 198.51.100.100
Internet Control Message Protocol
  
```

4
3
2

Select the third and the fourth packets, and check the key points:

1. Each ICMP echo reply is captured and shown 2 times.
2. The original packet header is without the VLAN tag.
3. The internal switch inserts additional port VLAN tag 102 that identifies the egress interface Ethernet1/2.
4. The internal switch inserts an additional VN tag.

The screenshot displays a network traffic capture with the following details:

- Packet List:** Shows ICMP Echo (ping) requests and replies between source IP 192.0.2.100 and destination IP 198.51.100.100. The protocol is ICMP, and the length is 108 bytes.
- Packet Details:**
 - Ethernet II:** Src: Cisco b9:77:0e (58:97:bd:b9:77:0e), Dst: VMware 9d:e8:be (00:50:56:9d:e8:be)
 - VN-Tag:**
 - Direction: To Bridge
 - Pointer: vif_id
 - Destination: 0
 - Looped: No
 - Reserved: 0
 - Version: 0
 - Type: 802.1Q Virtual LAN (0x8100)
 - 802.1Q Virtual LAN:** PRI: 0, DEI: 0, ID: 102
 - Internet Protocol Version 4:** Src: 198.51.100.100, Dst: 192.0.2.100
 - Internet Control Message Protocol:**

Explanation

When a packet capture on a backplane interface is configured, the switch simultaneously captures each packet twice. In this case, the internal switch receives packets that are already tagged by the application on the security module with the port VLAN tag and the VN tag. The VLAN tag identifies the egress interface that the internal chassis uses to forward the packets to the network. The VLAN tag 103 in ICMP echo request packets identifies Ethernet1/3 as the egress interface, while VLAN tag 102 in ICMP echo reply packets identifies Ethernet1/2 as the egress interface. The internal switch removes the VN tag and the internal interface VLAN tag before the packets are forwarded to the network.

This table summarizes the task:

Task	Capture point	Internal port captured packets	Direction on	Captured traffic
Configure and verify packet captures on backplane interfaces	Backplane interface	102	Ingress only	ICMP echo requests from host 192.0.2.100 to host 198.51.100.100
	Switch	103	only	ICMP echo replies from host 198.51.100.100 to host 192.0.2.100

Packet Captures on Application and Application Ports

Application or application port packet captures are always configured on backplane interfaces and additionally on the front interfaces if the user specifies the application capture direction.

There are mainly 2 use cases:

- Configure packet captures on backplane interfaces for packets that leave a specific front interface. For example, configure packet captures on the backplane interface Ethernet1/9 for

packets that leave interface Ethernet1/2.

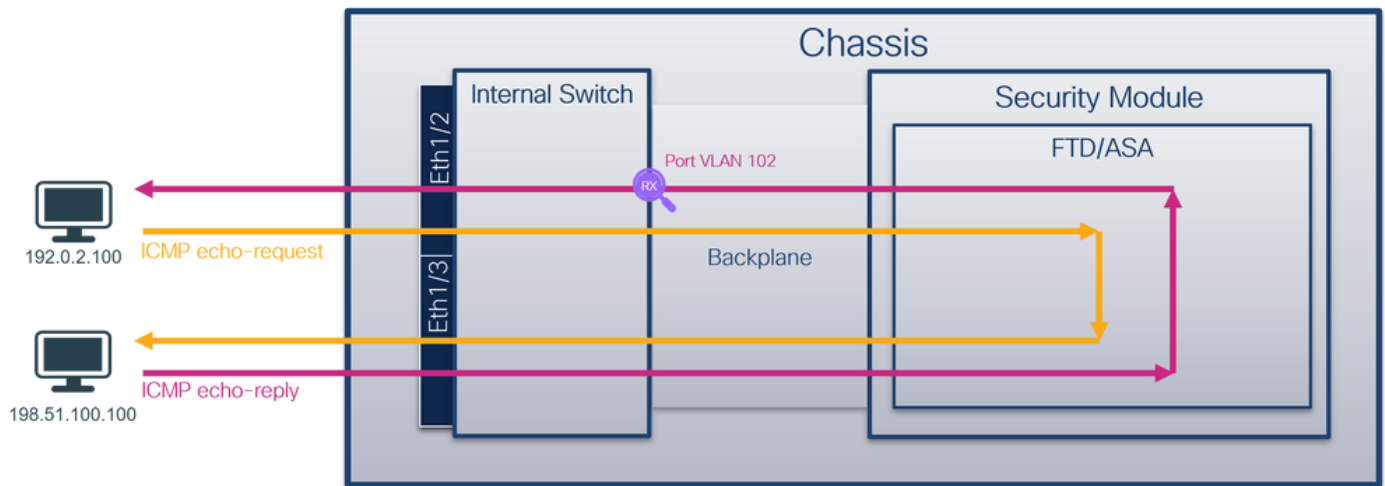
- Configure simultaneous packet captures on a specific front interface and the backplane interfaces. For example, configure simultaneous packet captures on interface Ethernet1/2 and on the backplane interface Ethernet1/9 for packets that leave interface Ethernet1/2.

This section covers both use cases.

Task 1

Use the FCM and CLI to configure and verify a packet capture on the backplane interface. Packets for which the application port Ethernet1/2 is identified as the egress interface are captured. In this case, ICMP replies are captured.

Topology, packet flow, and the capture points

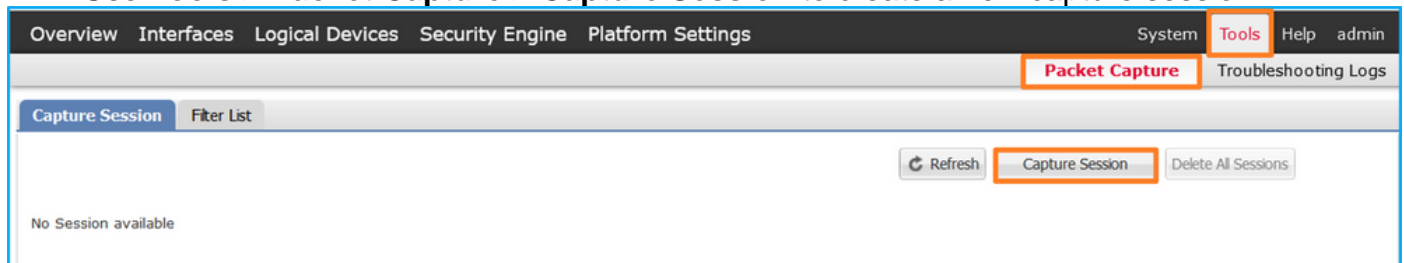


Configuration

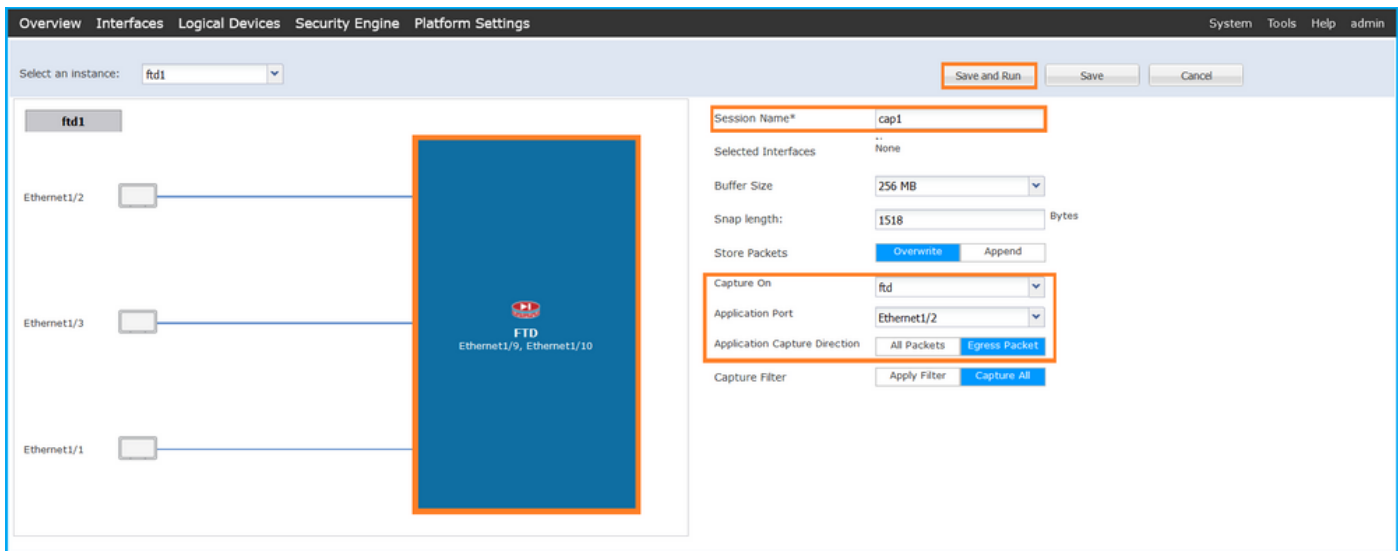
FCM

Follow these steps on FCM to configure a packet capture on the FTD application and the application port Ethernet1/2:

1. Use **Tools > Packet Capture > Capture Session** to create a new capture session:



2. Select the application, **Ethernet1/2** in the **Application Port** dropdown list and select **Egress Packet** in the **Application Capture Direction**. Provide the **Session Name** and click **Save and Run** to activate the capture:



FXOS CLI

Follow these steps on FXOS CLI to configure packet captures on backplane interfaces:

1. Identify the application type and identifier:

```
firepower# scope ssa
firepower /ssa# show app-instance
App Name  Identifier Slot ID  Admin State Oper State  Running Version Startup Version
Deploy Type Turbo Mode Profile Name Cluster State  Cluster Role
-----
ftd      ftd1      1           Enabled   Online    7.2.0.82      7.2.0.82
Native   No                Not Applicable None
```

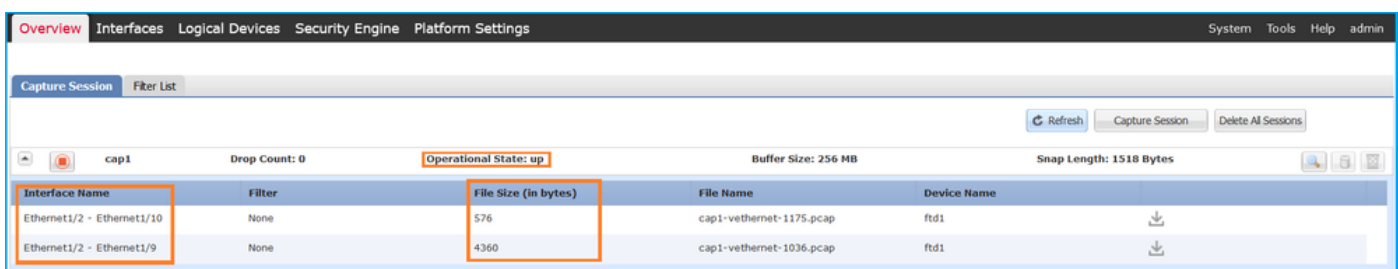
2. Create a capture session:

```
firepower# scope packet-capture
firepower /packet-capture # create session cap1
firepower /packet-capture/session* # create app-port 1 112 Ethernet1/2 ftd
firepower /packet-capture/session/app-port* # set app-identifier ftd1
firepower /packet-capture/session/app-port* # set filter ""
firepower /packet-capture/session/app-port* # set subinterface 0
firepower /packet-capture/session/app-port* # up
firepower /packet-capture/session* # commit
firepower /packet-capture/session #
```

Verification

FCM

Verify the **Interface Name**, ensure that the **Operational Status** is up and that the **File Size (in bytes)** increases:



FXOS CLI

Verify the capture details in **scope packet-capture**:

```
firepower# scope packet-capture  
firepower /packet-capture # show session cap1
```

Traffic Monitoring Session:

```
Packet Capture Session Name: cap1  
Session: 1  
Admin State: Enabled  
Oper State: Up  
Oper State Reason: Active  
Config Success: Yes  
Config Fail Reason:  
Append Flag: Overwrite  
Session Mem Usage: 256 MB  
Session Pcap Snap Len: 1518 Bytes  
Error Code: 0  
Drop Count: 0
```

Application ports involved in Packet Capture:

```
Slot Id: 1  
Link Name: 112  
Port Name: Ethernet1/2  
App Name: ftd  
Sub Interface: 0  
Application Instance Identifier: ftd1
```

Application ports resolved to:

```
Name: vnic1  
Eq Slot Id: 1  
Eq Port Id: 9  
Pcapfile: /workspace/packet-capture/session-1/cap1-vethernet-1036.pcap  
Pcapsize: 53640 bytes  
Vlan: 102  
Filter:
```

```
Name: vnic2  
Eq Slot Id: 1  
Eq Port Id: 10  
Pcapfile: /workspace/packet-capture/session-1/cap1-vethernet-1175.pcap  
Pcapsize: 1824 bytes  
Vlan: 102  
Filter:
```

Collect capture files

Follow the steps in the section **Collect Firepower 4100/9300 Internal Switch Capture Files**.

Capture file analysis

Use a packet capture file reader application to open the capture files. In the case of multiple backplane interfaces, ensure to open all capture files for each backplane interface. In this case, the packets are captured on the backplane interface Ethernet1/9.

Select the first and the second packets, and check the key points:

1. Each ICMP echo reply is captured and shown 2 times.

- The original packet header is without the VLAN tag.
- The internal switch inserts additional port VLAN tag **102** that identifies the egress interface Ethernet1/2.
- The internal switch inserts an additional VN tag.

No.	Time	Source	Destination	Protocol	Length	IP ID	IP TTL	Info
1	2022-08-01 10:03:22.231237959	198.51.100.100	192.0.2.100	ICMP	108	0x42f8 (17144)	64	Echo (ping) reply
2	2022-08-01 10:03:22.231239747	198.51.100.100	192.0.2.100	ICMP	108	0x42f8 (17144)	64	Echo (ping) reply
3	2022-08-01 10:03:23.232244769	198.51.100.100	192.0.2.100	ICMP	108	0x4303 (17331)	64	Echo (ping) reply
4	2022-08-01 10:03:23.232247753	198.51.100.100	192.0.2.100	ICMP	108	0x43b3 (17331)	64	Echo (ping) reply
5	2022-08-01 10:03:24.234703981	198.51.100.100	192.0.2.100	ICMP	108	0x445e (17502)	64	Echo (ping) reply
6	2022-08-01 10:03:24.234706751	198.51.100.100	192.0.2.100	ICMP	108	0x445e (17502)	64	Echo (ping) reply
7	2022-08-01 10:03:25.258672449	198.51.100.100	192.0.2.100	ICMP	108	0x4464 (17508)	64	Echo (ping) reply
8	2022-08-01 10:03:25.258674861	198.51.100.100	192.0.2.100	ICMP	108	0x4464 (17508)	64	Echo (ping) reply
9	2022-08-01 10:03:26.282663169	198.51.100.100	192.0.2.100	ICMP	108	0x44c3 (17603)	64	Echo (ping) reply
10	2022-08-01 10:03:26.282666183	198.51.100.100	192.0.2.100	ICMP	108	0x44c3 (17603)	64	Echo (ping) reply
11	2022-08-01 10:03:27.306671694	198.51.100.100	192.0.2.100	ICMP	108	0x44e7 (17639)	64	Echo (ping) reply
12	2022-08-01 10:03:27.306674378	198.51.100.100	192.0.2.100	ICMP	108	0x44e7 (17639)	64	Echo (ping) reply
13	2022-08-01 10:03:28.330664677	198.51.100.100	192.0.2.100	ICMP	108	0x4550 (17744)	64	Echo (ping) reply
14	2022-08-01 10:03:28.330667153	198.51.100.100	192.0.2.100	ICMP	108	0x4550 (17744)	64	Echo (ping) reply
15	2022-08-01 10:03:29.354795931	198.51.100.100	192.0.2.100	ICMP	108	0x4553 (17747)	64	Echo (ping) reply
16	2022-08-01 10:03:29.354936706	198.51.100.100	192.0.2.100	ICMP	108	0x4553 (17747)	64	Echo (ping) reply
17	2022-08-01 10:03:30.378795204	198.51.100.100	192.0.2.100	ICMP	108	0x4597 (17815)	64	Echo (ping) reply
18	2022-08-01 10:03:30.378798172	198.51.100.100	192.0.2.100	ICMP	108	0x4597 (17815)	64	Echo (ping) reply
19	2022-08-01 10:03:31.402772217	198.51.100.100	192.0.2.100	ICMP	108	0x467a (18042)	64	Echo (ping) reply
20	2022-08-01 10:03:31.402774775	198.51.100.100	192.0.2.100	ICMP	108	0x467a (18042)	64	Echo (ping) reply
21	2022-08-01 10:03:32.426693254	198.51.100.100	192.0.2.100	ICMP	108	0x468a (18058)	64	Echo (ping) reply
22	2022-08-01 10:03:32.426695691	198.51.100.100	192.0.2.100	ICMP	108	0x468a (18058)	64	Echo (ping) reply

```

> Frame 1: 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interface capture_u0_8, id 0
  Ethernet II, Src: Cisco b9:77:0e (58:97:bd:b9:77:0e), Dst: VMware 9d:e8:be (00:50:56:9d:e8:be)

  VN-Tag
  0... .. = Direction: To Bridge
  .0... .. = Pointer: vif_id
  ..00 0000 0000 0000 .. = Destination: 0
  .. .. .. .. = Looped: No
  .. .. .. .. = Reserved: 0
  .. .. .. .. = Version: 0
  .. .. .. .. 0000 0000 1010 = Source: 10
  Type: 802.1Q Virtual LAN (0x8100)
  802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 102
  000... .. = Priority: Best Effort (default) (0)
  ...0 .. .. = DEI: Ineligible
  ... 0000 0110 0110 = ID: 102
  Type: IPv4 (0x0800)
  Internet Protocol Version 4, Src: 198.51.100.100, Dst: 192.0.2.100
  Internet Control Message Protocol
  
```

No.	Time	Source	Destination	Protocol	Length	IP ID	IP TTL	Info
1	2022-08-01 10:03:22.231237959	198.51.100.100	192.0.2.100	ICMP	108	0x42f8 (17144)	64	Echo (ping) reply
2	2022-08-01 10:03:22.231239747	198.51.100.100	192.0.2.100	ICMP	108	0x42f8 (17144)	64	Echo (ping) reply
3	2022-08-01 10:03:23.232244769	198.51.100.100	192.0.2.100	ICMP	108	0x4303 (17331)	64	Echo (ping) reply
4	2022-08-01 10:03:23.232247753	198.51.100.100	192.0.2.100	ICMP	108	0x43b3 (17331)	64	Echo (ping) reply
5	2022-08-01 10:03:24.234703981	198.51.100.100	192.0.2.100	ICMP	108	0x445e (17502)	64	Echo (ping) reply
6	2022-08-01 10:03:24.234706751	198.51.100.100	192.0.2.100	ICMP	108	0x445e (17502)	64	Echo (ping) reply
7	2022-08-01 10:03:25.258672449	198.51.100.100	192.0.2.100	ICMP	108	0x4464 (17508)	64	Echo (ping) reply
8	2022-08-01 10:03:25.258674861	198.51.100.100	192.0.2.100	ICMP	108	0x4464 (17508)	64	Echo (ping) reply
9	2022-08-01 10:03:26.282663169	198.51.100.100	192.0.2.100	ICMP	108	0x44c3 (17603)	64	Echo (ping) reply
10	2022-08-01 10:03:26.282666183	198.51.100.100	192.0.2.100	ICMP	108	0x44c3 (17603)	64	Echo (ping) reply
11	2022-08-01 10:03:27.306671694	198.51.100.100	192.0.2.100	ICMP	108	0x44e7 (17639)	64	Echo (ping) reply
12	2022-08-01 10:03:27.306674378	198.51.100.100	192.0.2.100	ICMP	108	0x44e7 (17639)	64	Echo (ping) reply
13	2022-08-01 10:03:28.330664677	198.51.100.100	192.0.2.100	ICMP	108	0x4550 (17744)	64	Echo (ping) reply
14	2022-08-01 10:03:28.330667153	198.51.100.100	192.0.2.100	ICMP	108	0x4550 (17744)	64	Echo (ping) reply
15	2022-08-01 10:03:29.354795931	198.51.100.100	192.0.2.100	ICMP	108	0x4553 (17747)	64	Echo (ping) reply
16	2022-08-01 10:03:29.354936706	198.51.100.100	192.0.2.100	ICMP	108	0x4553 (17747)	64	Echo (ping) reply
17	2022-08-01 10:03:30.378795204	198.51.100.100	192.0.2.100	ICMP	108	0x4597 (17815)	64	Echo (ping) reply
18	2022-08-01 10:03:30.378798172	198.51.100.100	192.0.2.100	ICMP	108	0x4597 (17815)	64	Echo (ping) reply
19	2022-08-01 10:03:31.402772217	198.51.100.100	192.0.2.100	ICMP	108	0x467a (18042)	64	Echo (ping) reply
20	2022-08-01 10:03:31.402774775	198.51.100.100	192.0.2.100	ICMP	108	0x467a (18042)	64	Echo (ping) reply
21	2022-08-01 10:03:32.426693254	198.51.100.100	192.0.2.100	ICMP	108	0x468a (18058)	64	Echo (ping) reply
22	2022-08-01 10:03:32.426695691	198.51.100.100	192.0.2.100	ICMP	108	0x468a (18058)	64	Echo (ping) reply

```

> Frame 2: 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interface capture_u0_8, id 0
  Ethernet II, Src: Cisco b9:77:0e (58:97:bd:b9:77:0e), Dst: VMware 9d:e8:be (00:50:56:9d:e8:be)

  VN-Tag
  0... .. = Direction: To Bridge
  .0... .. = Pointer: vif_id
  ..00 0000 0000 0000 .. = Destination: 0
  .. .. .. .. = Looped: No
  .. .. .. .. = Reserved: 0
  .. .. .. .. = Version: 0
  .. .. .. .. 0000 0000 1010 = Source: 10
  Type: 802.1Q Virtual LAN (0x8100)
  802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 102
  000... .. = Priority: Best Effort (default) (0)
  ...0 .. .. = DEI: Ineligible
  ... 0000 0110 0110 = ID: 102
  Type: IPv4 (0x0800)
  Internet Protocol Version 4, Src: 198.51.100.100, Dst: 192.0.2.100
  Internet Control Message Protocol
  
```

Explanation

In this case, Ethernet1/2 with port VLAN tag 102 is the egress interface for the ICMP echo reply packets.

When the application capture direction is set to **Egress** in the capture options, packets with the

port VLAN tag 102 in the Ethernet header are captured on the backplane interfaces in the ingress direction.

This table summarizes the task:

Task	Capture point	Internal port VLAN in captured packets	Direction	Captured traffic
Configure and verify captures on application and application port Ethernet1/2	Backplane interfaces	102	Ingress only	ICMP echo replies from host 198.51.100.100 to host 192.0.2.100

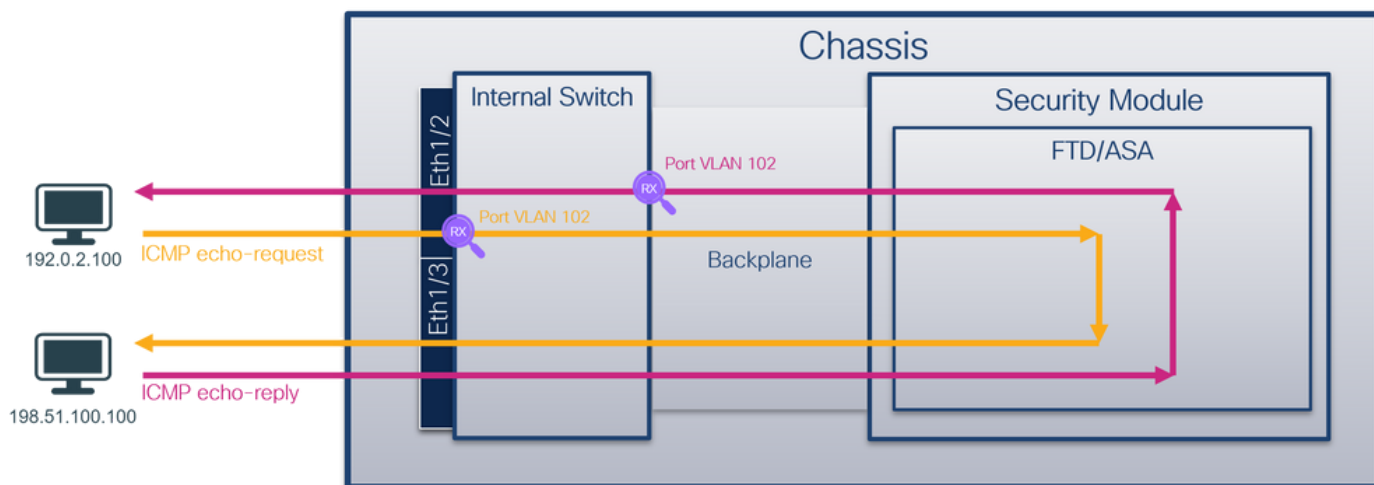
Task 2

Use the FCM and CLI to configure and verify a packet capture on the backplane interface and the front interface Ethernet1/2.

Simultaneous packet captures are configured on:

- Front interface – the packets with the port VLAN 102 on the interface Ethernet1/2 are captured. Captured packets are ICMP echo requests.
- Backplane interfaces – packets for which Ethernet1/2 is identified as the egress interface, or the packets with the port VLAN 102, are captured. Captured packets are ICMP echo replies.

Topology, packet flow, and the capture points

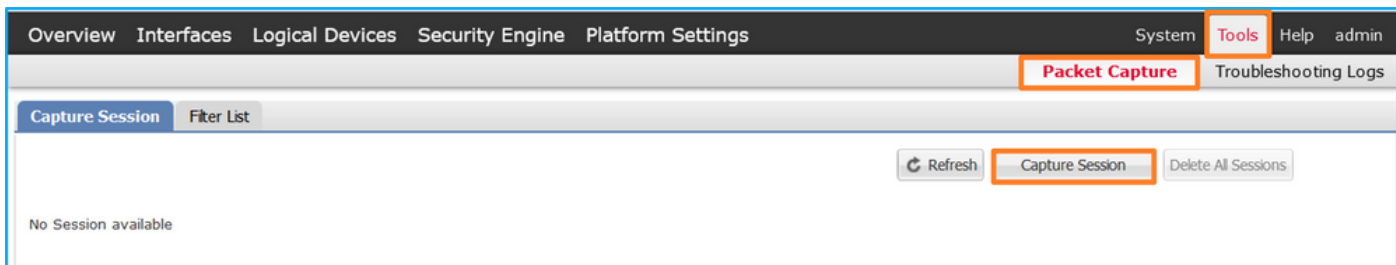


Configuration

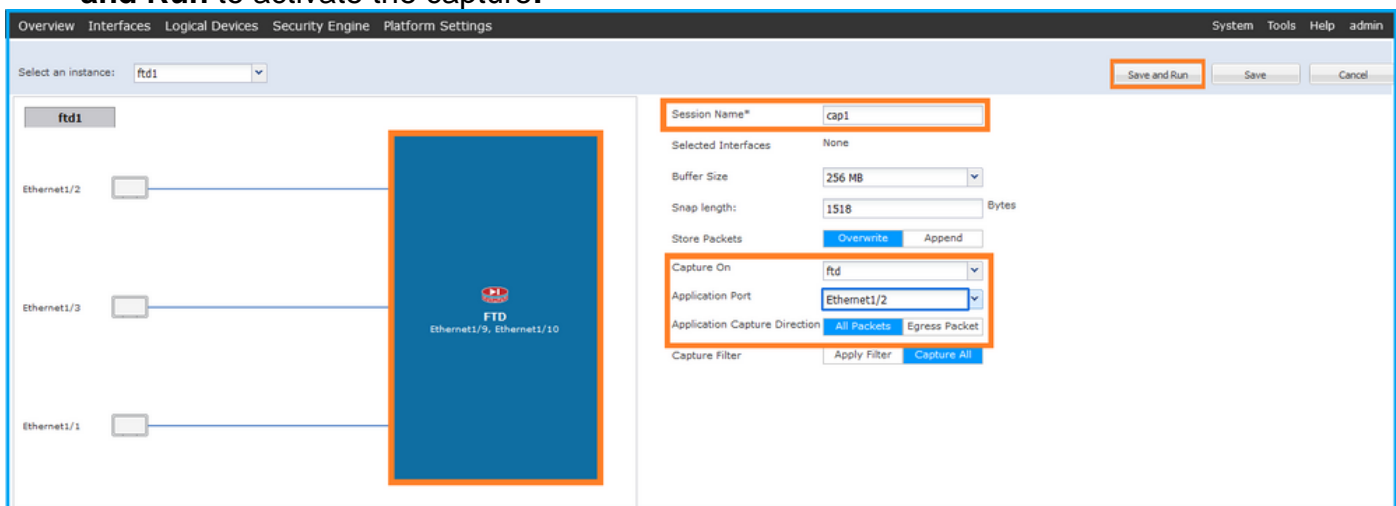
FCM

Follow these steps on FCM to configure a packet capture on the FTD application and the application port Ethernet1/2:

1. Use **Tools > Packet Capture > Capture Session** to create a new capture session:



2. Select the FTD application, **Ethernet1/2** in the **Application Port** dropdown list and select **All Packets** in the **Application Capture Direction**. Provide the **Session Name** and click **Save and Run** to activate the capture:



FXOS CLI

Follow these steps on FXOS CLI to configure packet captures on backplane interfaces:

1. Identify the application type and identifier:

```
firepower# scope ssa
firepower /ssa# show app-instance
App Name   Identifier Slot ID   Admin State Oper State   Running Version Startup Version
Deploy Type Turbo Mode Profile Name Cluster State Cluster Role
-----
ftd        ftd1          1           Enabled   Online       7.2.0.82      7.2.0.82
Native     No            Not Applicable None
```

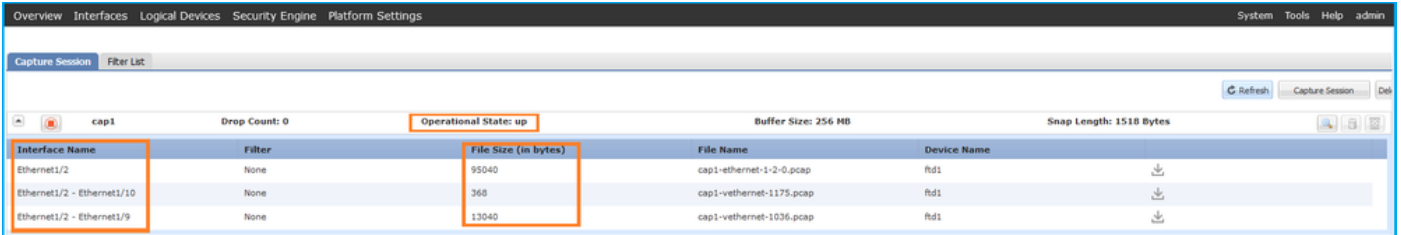
2. Create a capture session:

```
firepower# scope packet-capture
firepower /packet-capture # create session cap1
firepower /packet-capture/session* # create phy-port eth1/2
firepower /packet-capture/session/phy-port* # set app-identifier ftd1
firepower /packet-capture/session/phy-port* # exit
firepower /packet-capture/session* # create app-port 1 link12 Ethernet1/2 ftd
firepower /packet-capture/session/app-port* # set app-identifier ftd1
firepower /packet-capture/session* # enable
firepower /packet-capture/session* # commit
firepower /packet-capture/session # commit
```

Verification

FCM

Verify the **Interface Name**, ensure that the **Operational Status** is up and that the **File Size (in bytes)** increases:



Interface Name	Filter	File Size (in bytes)	File Name	Device Name
Ethernet1/2	None	95040	cap1-ethernet-1-2-0.pcap	ftd1
Ethernet1/2 - Ethernet1/10	None	368	cap1-vethernet-1175.pcap	ftd1
Ethernet1/2 - Ethernet1/9	None	13040	cap1-vethernet-1036.pcap	ftd1

FXOS CLI

Verify the capture details in **scope packet-capture**:

```
firepower# scope packet-capture
firepower /packet-capture # show session cap1
```

Traffic Monitoring Session:

```
Packet Capture Session Name: cap1
Session: 1
Admin State: Enabled
Oper State: Up
Oper State Reason: Active
Config Success: Yes
Config Fail Reason:
Append Flag: Overwrite
Session Mem Usage: 256 MB
Session Pcap Snap Len: 1518 Bytes
Error Code: 0
Drop Count: 0
```

Physical ports involved in Packet Capture:

```
Slot Id: 1
Port Id: 2
Pcapfile: /workspace/packet-capture/session-1/cap1-ethernet-1-2-0.pcap
Pcapsize: 410444 bytes
Filter:
Sub Interface: 0
Application Instance Identifier: ftd1
Application Name: ftd
```

Application ports involved in Packet Capture:

```
Slot Id: 1
Link Name: link12
Port Name: Ethernet1/2
App Name: ftd
Sub Interface: 0
Application Instance Identifier: ftd1
```

Application ports resolved to:

```
Name: vnic1
Eq Slot Id: 1
Eq Port Id: 9
Pcapfile: /workspace/packet-capture/session-1/cap1-vethernet-1036.pcap
Pcapsize: 128400 bytes
Vlan: 102
Filter:
```



```
Name: vnic2
Eq Slot Id: 1
Eq Port Id: 10
Pcapfile: /workspace/packet-capture/session-1/cap1-vethernet-1175.pcap
Pcapsize: 2656 bytes
Vlan: 102
Filter:
```

Collect capture files

Follow the steps in the section **Collect Firepower 4100/9300 Internal Switch Capture Files**.

Capture file analysis

Use a packet capture file reader application to open the capture files. In the case of multiple backplane interfaces, ensure to open all capture files for each backplane interface. In this case, the packets are captured on the backplane interface Ethernet1/9.

Open the capture file for the interface Ethernet1/2, select the first packet, and check the key points:

1. Only ICMP echo request packets are captured. Each packet is captured and shown 2 times.
2. The original packet header is without the VLAN tag.
3. The internal switch inserts additional port VLAN tag **102** that identifies the ingress interface Ethernet1/2.
4. The internal switch inserts an additional VN tag.

The screenshot displays a network packet capture analysis interface. At the top, a table lists 29 captured packets, all of which are ICMP Echo (ping) requests from 192.0.2.100 to 198.51.100.100. The second packet in the list is highlighted with a red box and labeled with a red '1'. Below the table, the detailed view of the second packet is shown. The packet is 108 bytes long and is an Internet Control Message Protocol (ICMP) Echo (ping) request. The source IP is 192.0.2.100 and the destination IP is 198.51.100.100. The packet is captured on interface capture_u0_1. The detailed view is annotated with red boxes and numbers:

- 1:** Points to the IP ID field (0xc009) in the IP header.
- 2:** Points to the Internet Protocol Version 4 (IPv4) header.
- 3:** Points to the 802.1Q Virtual LAN (VLAN) tag, specifically the priority field (000).
- 4:** Points to the Virtual Network (VN) tag, specifically the destination field (1010).

Select the second packet and check the key points:

1. Only ICMP echo request packets are captured. Each packet is captured and shown 2 times.
2. The original packet header is without the VLAN tag.
3. The internal switch inserts additional port VLAN tag **102** that identifies the ingress interface Ethernet1/2.

Network traffic capture showing ICMP Echo (ping) requests. The first and second packets are highlighted in orange. Below the traffic list, the details for the second packet (packet 2) are shown, including Ethernet II, 802.1Q Virtual LAN (VLAN 102), Internet Protocol Version 4, and Internet Control Message Protocol. The packet details are annotated with red boxes and numbers 2, 3, and 4.

Open the capture file for the interface Ethernet1/9, select the first and the second packets, and check the key points:

1. Each ICMP echo reply is captured and shown 2 times.
2. The original packet header is without the VLAN tag.
3. The internal switch inserts additional port VLAN tag **102** that identifies the egress interface Ethernet1/2.
4. The internal switch inserts an additional VN tag.

Network traffic capture showing ICMP Echo (ping) replies. The first and second packets are highlighted in orange. Below the traffic list, the details for the second packet (packet 2) are shown, including Ethernet II, 802.1Q Virtual LAN (VLAN 102), Internet Protocol Version 4, and Internet Control Message Protocol. The packet details are annotated with red boxes and numbers 2, 3, and 4.

No.	Time	Source	Destination	Protocol	Length	IP ID	IP TTL	Info
1	2022-08-01 11:33:19.071512698	198.51.100.100	192.0.2.100	ICMP	108	0xaf27 (20263)	64	Echo (ping) reply id=0x0013, seq=1/256, ttl=64
2	2022-08-01 11:33:19.071514882	198.51.100.100	192.0.2.100	ICMP	108	0xaf27 (20263)	64	Echo (ping) reply id=0x0013, seq=1/256, ttl=64
3	2022-08-01 11:33:20.072677302	198.51.100.100	192.0.2.100	ICMP	108	0xa170 (20475)	64	Echo (ping) reply id=0x0013, seq=2/512, ttl=64
4	2022-08-01 11:33:20.072679384	198.51.100.100	192.0.2.100	ICMP	108	0xaffb (20475)	64	Echo (ping) reply id=0x0013, seq=2/512, ttl=64
5	2022-08-01 11:33:21.073913640	198.51.100.100	192.0.2.100	ICMP	108	0x50ac (20652)	64	Echo (ping) reply id=0x0013, seq=3/768, ttl=64
6	2022-08-01 11:33:21.073915690	198.51.100.100	192.0.2.100	ICMP	108	0x50ac (20652)	64	Echo (ping) reply id=0x0013, seq=3/768, ttl=64
7	2022-08-01 11:33:22.075239381	198.51.100.100	192.0.2.100	ICMP	108	0x513e (20798)	64	Echo (ping) reply id=0x0013, seq=4/1024, ttl=64
8	2022-08-01 11:33:22.075241491	198.51.100.100	192.0.2.100	ICMP	108	0x513e (20798)	64	Echo (ping) reply id=0x0013, seq=4/1024, ttl=64
9	2022-08-01 11:33:23.076447152	198.51.100.100	192.0.2.100	ICMP	108	0x51c9 (20937)	64	Echo (ping) reply id=0x0013, seq=5/1280, ttl=64
10	2022-08-01 11:33:23.076449303	198.51.100.100	192.0.2.100	ICMP	108	0x51c9 (20937)	64	Echo (ping) reply id=0x0013, seq=5/1280, ttl=64
11	2022-08-01 11:33:24.082407896	198.51.100.100	192.0.2.100	ICMP	108	0x528e (21134)	64	Echo (ping) reply id=0x0013, seq=6/1536, ttl=64
12	2022-08-01 11:33:24.082410099	198.51.100.100	192.0.2.100	ICMP	108	0x528e (21134)	64	Echo (ping) reply id=0x0013, seq=6/1536, ttl=64
13	2022-08-01 11:33:25.106382424	198.51.100.100	192.0.2.100	ICMP	108	0x52af (21167)	64	Echo (ping) reply id=0x0013, seq=7/1792, ttl=64
14	2022-08-01 11:33:25.106384549	198.51.100.100	192.0.2.100	ICMP	108	0x52af (21167)	64	Echo (ping) reply id=0x0013, seq=7/1792, ttl=64
15	2022-08-01 11:33:26.130437851	198.51.100.100	192.0.2.100	ICMP	108	0x53a6 (21414)	64	Echo (ping) reply id=0x0013, seq=8/2048, ttl=64
16	2022-08-01 11:33:26.130440320	198.51.100.100	192.0.2.100	ICMP	108	0x53a6 (21414)	64	Echo (ping) reply id=0x0013, seq=8/2048, ttl=64
17	2022-08-01 11:33:27.154398212	198.51.100.100	192.0.2.100	ICMP	108	0x5446 (21574)	64	Echo (ping) reply id=0x0013, seq=9/2304, ttl=64
18	2022-08-01 11:33:27.154400198	198.51.100.100	192.0.2.100	ICMP	108	0x5446 (21574)	64	Echo (ping) reply id=0x0013, seq=9/2304, ttl=64
19	2022-08-01 11:33:28.178469866	198.51.100.100	192.0.2.100	ICMP	108	0x5493 (21651)	64	Echo (ping) reply id=0x0013, seq=10/2560, ttl=64
20	2022-08-01 11:33:28.178471810	198.51.100.100	192.0.2.100	ICMP	108	0x5493 (21651)	64	Echo (ping) reply id=0x0013, seq=10/2560, ttl=64
21	2022-08-01 11:33:29.202395869	198.51.100.100	192.0.2.100	ICMP	108	0x54f4 (21740)	64	Echo (ping) reply id=0x0013, seq=11/2816, ttl=64
22	2022-08-01 11:33:29.202398067	198.51.100.100	192.0.2.100	ICMP	108	0x54f4 (21740)	64	Echo (ping) reply id=0x0013, seq=11/2816, ttl=64
23	2022-08-01 11:33:30.226398735	198.51.100.100	192.0.2.100	ICMP	108	0x5526 (21798)	64	Echo (ping) reply id=0x0013, seq=12/3072, ttl=64
24	2022-08-01 11:33:30.226401017	198.51.100.100	192.0.2.100	ICMP	108	0x5526 (21798)	64	Echo (ping) reply id=0x0013, seq=12/3072, ttl=64
25	2022-08-01 11:33:31.250387808	198.51.100.100	192.0.2.100	ICMP	108	0x55f2 (22002)	64	Echo (ping) reply id=0x0013, seq=13/3328, ttl=64
26	2022-08-01 11:33:31.250389971	198.51.100.100	192.0.2.100	ICMP	108	0x55f2 (22002)	64	Echo (ping) reply id=0x0013, seq=13/3328, ttl=64
27	2022-08-01 11:33:32.274416011	198.51.100.100	192.0.2.100	ICMP	108	0x5660 (22112)	64	Echo (ping) reply id=0x0013, seq=14/3584, ttl=64
28	2022-08-01 11:33:32.274418229	198.51.100.100	192.0.2.100	ICMP	108	0x5660 (22112)	64	Echo (ping) reply id=0x0013, seq=14/3584, ttl=64
29	2022-08-01 11:33:33.298397657	198.51.100.100	192.0.2.100	ICMP	108	0x56e7 (22247)	64	Echo (ping) reply id=0x0013, seq=15/3840, ttl=64


```

> Frame 2: 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interface capture_u0_8, id 0
> Ethernet II, Src: Cisco b9:77:0e (58:97:bd:b9:77:0e), Dst: VMware 9d:e8:be (00:50:56:9d:e8:be)
  0000  00 50 56 9d e8 be 58 97 bd b9 77 0e 89 26 00 00  -PV...X:..m..8..
  0010  00 0a 81 00 00 66 08 00 45 00 00 54 4f 27 00 00  -.....F...E..TO...
  0020  40 01 3e 86 c6 33 64 64 c0 00 02 64 00 00 95 7c  -@->...3dd ...d...|
  0030  00 13 00 01 f2 b9 e7 62 00 00 00 00 cb 7f 06 00  -.....b.....
  0040  00 00 00 00 10 11 12 13 14 15 16 17 18 19 1a 1b  -.....
  0050  1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b  -.....l'm $5$'()*+
  0060  2c 2d 2e 2f 30 31 32 33 34 35 36 37              -./:0123 4567
  
```

```

VN-Tag
0... .. = Direction: To Bridge
.0... .. = Pointer: vif_id
..00 0000 0000 0000 .. = Destination: 0
... .. = Looped: No
... .. = Reserved: 0
... .. = Version: 0
... .. 0000 0000 1010 = Source: 10
Type: 802.1Q Virtual LAN (0x8100)

802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 102
000... .. = Priority: Best Effort (default) (0)
...0... .. = DEI: Ineligible
... 0000 0110 0110 = ID: 102
Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 198.51.100.100, Dst: 192.0.2.100
Internet Control Message Protocol
  
```

Explanation

If the option **All Packets** in the **Application Capture Direction** is selected, 2 simultaneous packet captures related to the selected application port Ethernet1/2 are configured: a capture on the front interface Ethernet1/2 and a capture on selected backplane interfaces.

When a packet capture on a front interface is configured, the switch simultaneously captures each packet twice:

- After the insertion of the port VLAN tag.
- After the insertion of the VN tag.

In the order of operations, the VN tag is inserted at a later stage than the port VLAN tag insertion. But in the capture file, the packet with the VN tag is shown earlier than the packet with the port VLAN tag. In this example, the VLAN tag 102 in ICMP echo request packets identifies Ethernet1/2 as the ingress interface.

When a packet capture on a backplane interface is configured, the switch simultaneously captures each packet twice. The internal switch receives packets that are already tagged by the application on the security module with the port VLAN tag and the VN tag. The port VLAN tag identifies the egress interface that the internal chassis uses to forward the packets to the network. In this example, the VLAN tag 102 in ICMP echo reply packets identifies Ethernet1/2 as the egress interface.

The internal switch removes the VN tag and the internal interface VLAN tag before the packets are forwarded to the network.

This table summarizes the task:

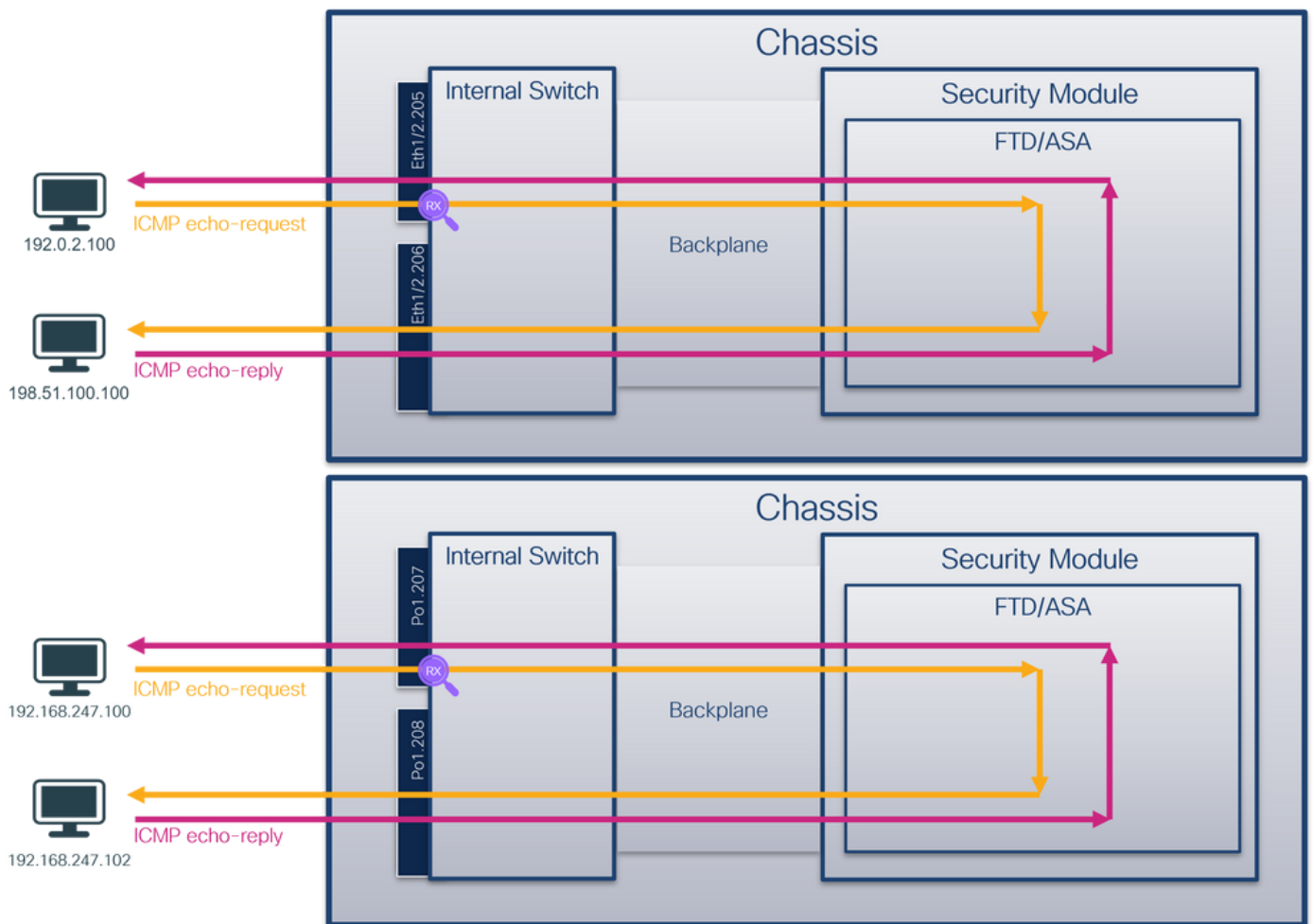
Task	Capture point	Internal port VLAN in captured packets	Direction	Captured traffic
Configure and verify captures	Backplane	102	Ingress	ICMP echo replies from host

on application and application port Ethernet1/2	interfaces		only	198.51.100.100 to host
	Interface Ethernet1/2	102	Ingress only	192.0.2.100
				ICMP echo requests from host
				192.0.2.100 to host
				198.51.100.100

Packet Capture on a Subinterface of a Physical or Port-channel Interface

Use the FCM and CLI to configure and verify a packet capture on subinterface Ethernet1/2.205 or port-channel subinterface Portchannel1.207. Subinterfaces and captures on subinterfaces are supported only for the FTD application in container mode. In this case, a packet capture on Ethernet1/2.205 and Portchannel1.207 are configured.

Topology, packet flow, and the capture points

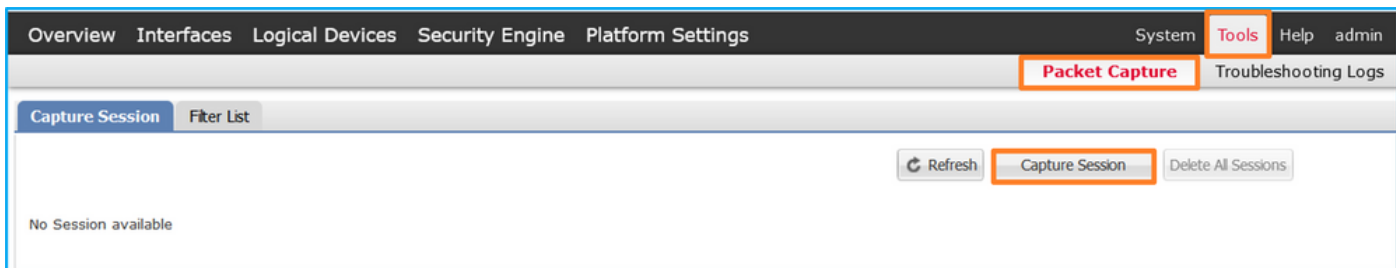


Configuration

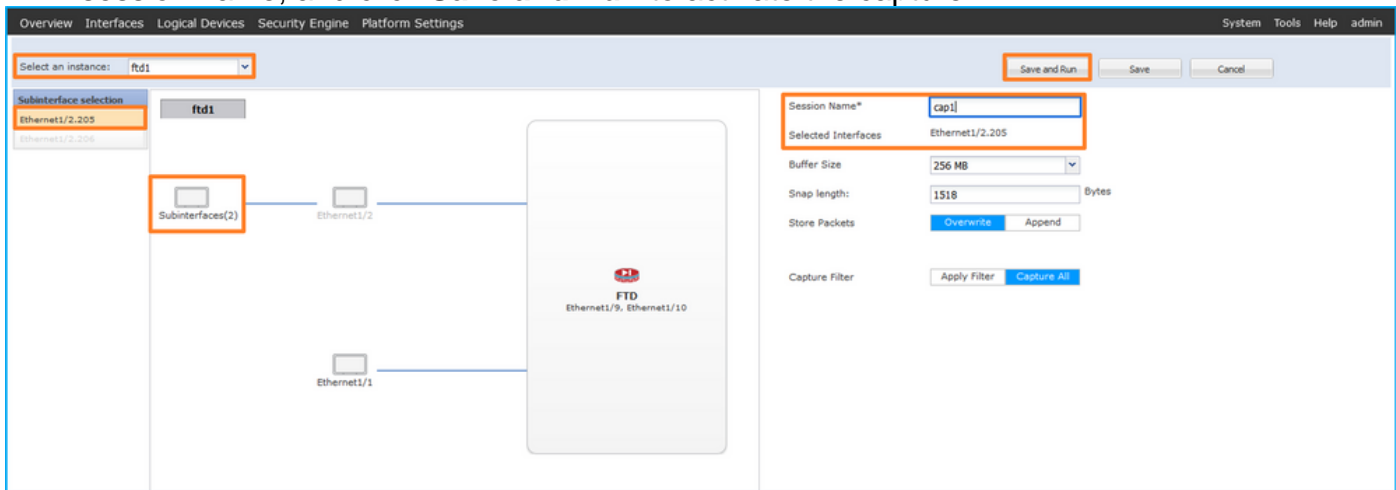
FCM

Follow these steps on FCM to configure a packet capture on the FTD application and the application port Ethernet1/2:

1. Use **Tools > Packet Capture > Capture Session** to create a new capture session:



2. Select the specific application instance ftd1, the subinterface Ethernet1/2.205, provide the session name, and click **Save and Run** to activate the capture:



3. In the case of a port-channel subinterface, due to the Cisco bug ID [CSCvq33119](https://www.cisco.com/cisco/webbugtool/show_bug.do?bugID=CSCvq33119) subinterfaces are not visible in the FCM. Use the FXOS CLI to configure captures on port-channel subinterfaces.

FXOS CLI

Follow these steps on FXOS CLI to configure a packet capture on subinterfaces Ethernet1/2.205 and Portchannel1.207:

1. Identify the application type and identifier:

```
firepower# scope ssa
firepower /ssa # show app-instance
```

App Name	Identifier	Slot ID	Admin State	Oper State	Running Version	Startup Version
ftd	ftd1	1	Enabled	Online	7.2.0.82	7.2.0.82
Container	No	RP20	Not Applicable	None		
ftd	ftd2	1	Enabled	Online	7.2.0.82	7.2.0.82
Container	No	RP20	Not Applicable	None		

2. In the case of a port-channel interface, identify its member interfaces:

```
firepower# connect fxos
<output skipped>
firepower(fxos)# show port-channel summary
```

Flags: D - Down P - Up in port-channel (members)
I - Individual H - Hot-standby (LACP only)
s - Suspended r - Module-removed
S - Switched R - Routed
U - Up (port-channel)
M - Not in use. Min-links not met

Group	Port-Channel	Type	Protocol	Member Ports
1	Po1(SU)	Eth	LACP	Eth1/3(P) Eth1/3(P)

3. Create a capture session:

```
firepower# scope packet-capture
firepower /packet-capture # create session cap1
firepower /packet-capture/session* # create phy-port Eth1/2
firepower /packet-capture/session/phy-port* # set app ftd
firepower /packet-capture/session/phy-port* # set app-identifier ftd1
firepower /packet-capture/session/phy-port* # set subinterface 205
firepower /packet-capture/session/phy-port* # up
firepower /packet-capture/session* # enable
firepower /packet-capture/session* # commit
firepower /packet-capture/session #
```

For port-channel subinterfaces, create a packet capture for each port-channel member interface:

```
firepower# scope packet-capture
firepower /packet-capture # create filter vlan207
firepower /packet-capture/filter* # set ovlan 207
firepower /packet-capture/filter* # up
firepower /packet-capture* # create session cap1
firepower /packet-capture/session* create phy-port Eth1/3
firepower /packet-capture/session/phy-port* # set app ftd
firepower /packet-capture/session/phy-port* # set app-identifier ftd1
firepower /packet-capture/session/phy-port* # set subinterface 207
firepower /packet-capture/session/phy-port* # up
firepower /packet-capture/session* # create phy-port Eth1/4
firepower /packet-capture/session/phy-port* # set app ftd
firepower /packet-capture/session/phy-port* # set app-identifier ftd1
firepower /packet-capture/session/phy-port* # set subinterface 207
firepower /packet-capture/session/phy-port* # up
firepower /packet-capture/session* # enable
firepower /packet-capture/session* # commit
firepower /packet-capture/session #
```

Verification

FCM

Verify the **Interface Name**, ensure that the **Operational Status** is up and that the **File Size (in bytes)** increases:



Interface Name	Filter	File Size (in bytes)	File Name	Device Name
Ethernet1/2.205	None	233992	cap1-ethernet-1-2-0.pcap	ftd1

Port-channel subinterface captures configured on FXOS CLI are also visible on FCM; however, they cannot be edited:

Interface Name	Filter	File Size (in bytes)	File Name	Device Name
Ethernet1/4-207	None	624160	cap1-ethernet-1-4-0.pcap	Not available
Ethernet1/3-207	None	160	cap1-ethernet-1-3-0.pcap	Not available

FXOS CLI

Verify the capture details in **scope packet-capture**:

```
firepower# scope packet-capture
firepower /packet-capture # show session cap1
```

Traffic Monitoring Session:

```
Packet Capture Session Name: cap1
Session: 1
Admin State: Enabled
Oper State: Up
Oper State Reason: Active
Config Success: Yes
Config Fail Reason:
Append Flag: Overwrite
Session Mem Usage: 256 MB
Session Pcap Snap Len: 1518 Bytes
Error Code: 0
Drop Count: 0
```

Physical ports involved in Packet Capture:

```
Slot Id: 1
Port Id: 2
Pcapfile: /workspace/packet-capture/session-1/cap1-ethernet-1-2-0.pcap
Pcapsize: 9324 bytes
Filter:
Sub Interface: 205
Application Instance Identifier: ftd1
Application Name: ftd
```

Port-channel 1 with member interfaces Ethernet1/3 and Ethernet1/4:

```
firepower# scope packet-capture
firepower /packet-capture # show session cap1
```

Traffic Monitoring Session:

```
Packet Capture Session Name: cap1
Session: 1
Admin State: Enabled
Oper State: Up
Oper State Reason: Active
Config Success: Yes
Config Fail Reason:
Append Flag: Overwrite
Session Mem Usage: 256 MB
Session Pcap Snap Len: 1518 Bytes
Error Code: 0
Drop Count: 0
```

Physical ports involved in Packet Capture:

```
Slot Id: 1
```

```

Port Id: 3
Pcapfile: /workspace/packet-capture/session-1/cap1-ethernet-1-3-0.pcap
Pcapsize: 160 bytes
Filter:
Sub Interface: 207
Application Instance Identifier: ftd1
Application Name: ftd
Slot Id: 1
Port Id: 4
Pcapfile: /workspace/packet-capture/session-1/cap1-ethernet-1-4-0.pcap
Pcapsize: 624160 bytes
Filter:
Sub Interface: 207
Application Instance Identifier: ftd1
Application Name: ftd

```

Collect capture files

Follow the steps in the section **Collect Firepower 4100/9300 Internal Switch Capture Files**.

Capture file analysis

Use a packet capture file reader application to open the capture file. Select the first packet and check the key points:

1. Only ICMP echo request packets are captured. Each packet is captured and shown 2 times.
2. The original packet header has the VLAN tag **205**.
3. The internal switch inserts additional port VLAN tag **102** that identifies the ingress interface Ethernet1/2.
4. The internal switch inserts an additional VN tag.

The screenshot displays a network traffic capture analysis. The top section shows a list of 27 ICMP echo request packets. The second packet is highlighted, and its details are expanded below. The expanded view shows the following key sections:

- VN-Tag:** Direction: From Bridge; Pointer: vif_id; Destination: 84; Looped: No; Reserved: 0; Version: 0; Source: 0.
- 802.1Q Virtual LAN:** PRI: 0, DEI: 0, ID: 102. Priority: Best Effort (default) (0); DEI: Ineligible; ID: 102.
- 802.1Q Virtual LAN:** PRI: 0, DEI: 0, ID: 205. Priority: Best Effort (default) (0); DEI: Ineligible; ID: 205.
- Internet Protocol Version 4:** Src: 192.0.2.100, Dst: 198.51.100.10.
- Internet Control Message Protocol:**

The packet details on the right show the raw data and hex representation of the packet, including the IP header and ICMP data.

Select the second packet and check the key points:

1. Only ICMP echo request packets are captured. Each packet is captured and shown 2 times.
2. The original packet header has the VLAN tag 205.

Wireshark capture showing ICMP echo request packets. The first packet (No. 1) is highlighted with a red box and a '1' next to it. The packet details pane shows '802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 205' highlighted with a red box and a '2' next to it.

Now open the capture files for Portchannel1.207. Select the first packet and check the key points

1. Only ICMP echo request packets are captured. Each packet is captured and shown 2 times.
2. The original packet header has the VLAN tag 207.
3. The internal switch inserts an additional port VLAN tag 1001 that identifies the ingress interface Portchannel1.
4. The internal switch inserts an additional VN tag.

Wireshark capture showing ICMP echo request packets. The first packet (No. 1) is highlighted with a red box. The packet details pane shows '802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 1001' highlighted with a red box and a '4' next to it, and '802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 207' highlighted with a red box and a '3' next to it.

Select the second packet and check the key points:

1. Only ICMP echo request packets are captured. Each packet is captured and shown 2 times.

2. The original packet header has the VLAN tag 207.

The image shows a Wireshark packet capture. The top pane displays a list of 27 ICMP Echo (ping) requests. A red box highlights the first few rows, with a '1' indicating the first packet. The bottom pane shows the details of the selected packet (Frame 2), with a blue box highlighting the Ethernet II header. The details show:

- Ethernet II, Src: Cisco d6:ec:00 (00:17:df:d6:ec:00), Dst: a2:76:f2:00:00:1c (a2:76:f2:00:00:1c)
- 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 207
- ...0 ... = Priority: Best Effort (default) (0)
- ...0 ... = DEI: Ineligible
- ... 0000 1100 1111 = ID: 207
- Type: IPv4 (0x0800)
- Internet Protocol Version 4, Src: 192.168.247.100, Dst: 192.168.247.102
- Internet Control Message Protocol

Explanation

When a packet capture on a front interface is configured, the switch simultaneously captures each packet twice:

- After the insertion of the port VLAN tag.
- After the insertion of the VN tag.

In the order of operations, the VN tag is inserted at a later stage than the port VLAN tag insertion. But in the capture file, the packet with the VN tag is shown earlier than the packet with the port VLAN tag. Additionally, in the case of subinterfaces, in the capture files, every second packet does not contain the port VLAN tag.

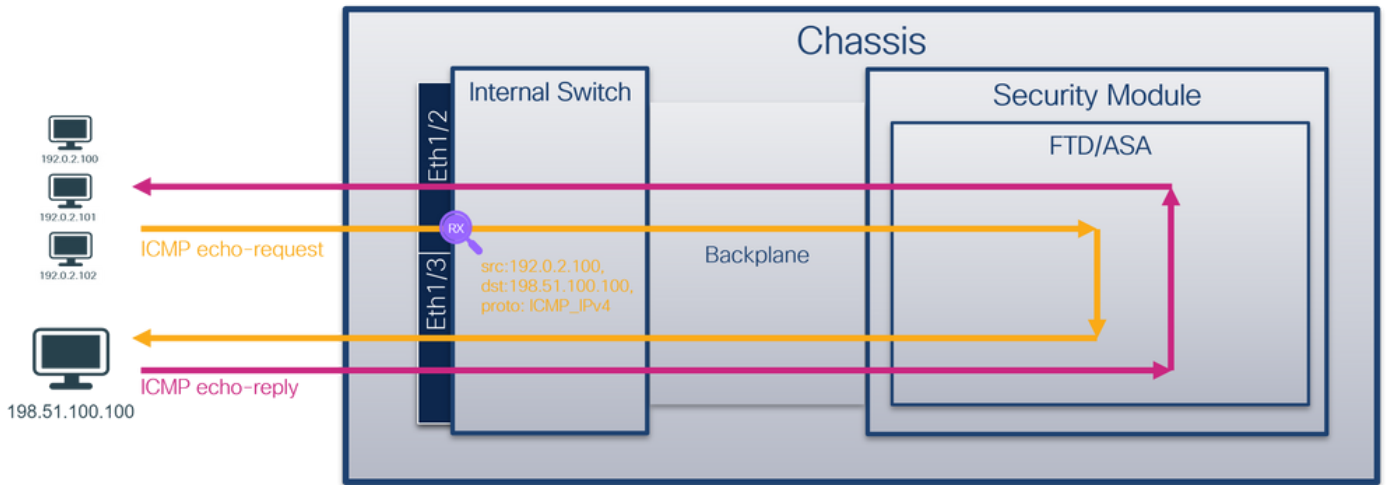
This table summarizes the task:

Task	Capture point	Internal port VLAN in captured packets	Direction	Captured traffic
Configure and verify a packet capture on subinterface Ethernet1/2.205	Ethernet1/2.205	102	Ingress only	ICMP echo requests from host 192.0.2.100 to host 198.51.100.
Configure and verify a packet capture on Portchannel1 subinterface with member interfaces Ethernet1/3 and Ethernet1/4	Ethernet1/3 Ethernet1/4	1001	Ingress only	ICMP echo requests from 192.168.207.100 to host 192.168.207.102

Packet Capture Filters

Use the FCM and CLI to configure and verify a packet capture on interface Ethernet1/2 with a filter.

Topology, packet flow, and the capture points



Configuration

FCM

Follow these steps on FCM to configure a capture filter for ICMP echo request packets from host 192.0.2.100 to host 198.51.100.100 and apply it to packet capture on interface Ethernet1/2:

1. Use **Tools > Packet Capture > Filter List > Add Filter** to create a capture filter.
2. Specify the **Filter Name, Protocol, Source IPv4, Destination IPv4** and click **Save**:

The screenshot shows the FCM interface for configuring a filter. The 'Filter List' table contains the following entry:

Filter Name	MAC	From IPv4	From IPv6	Port	MAC	To IPv4	To IPv6	Port	Protocol	Inner vlan	Outer vlan	EtherType
filter_icmp	00:00:00:00:00:00	192.0.2.100	::	0	00:00:00:00:00:00	198.51.100.100	::	0	1	0	0	0

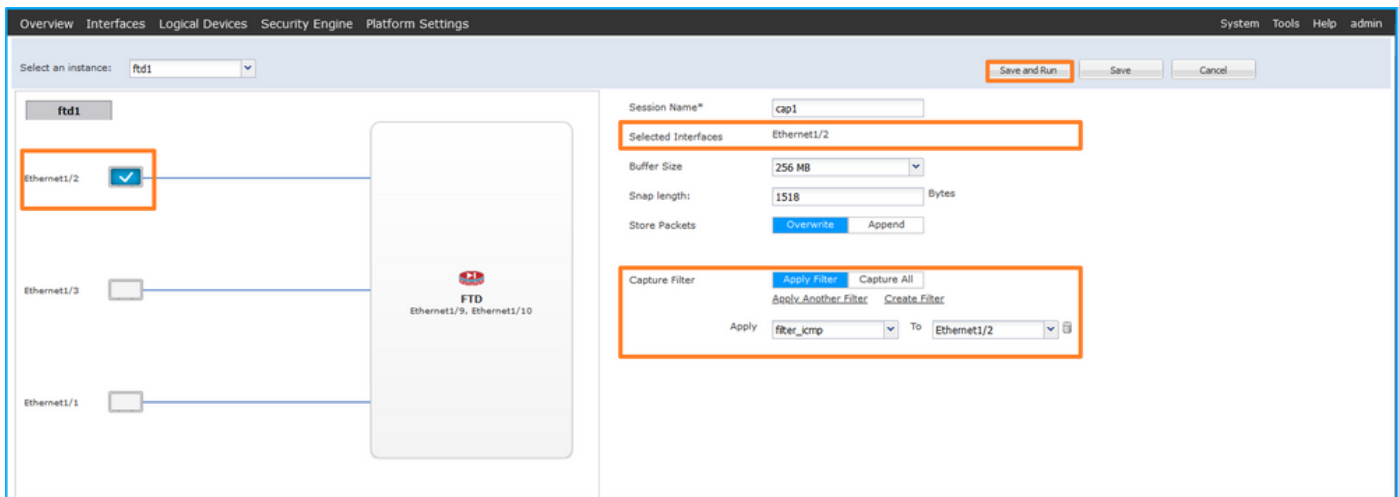
The 'Edit Packet Filter' dialog box is open, showing the configuration for 'filter_icmp':

- Filter Name*: filter_icmp
- Protocol: ICMP_IPV4
- EtherType: Any
- Inner vlan: 0
- Outer vlan: 0
- Source IPv4: 192.0.2.100
- Destination IPv4: 198.51.100.100
- IPv6: ::
- Port: 0
- MAC: 00:00:00:00:00:00

3. Use **Tools > Packet Capture > Capture Session** to create a new capture session:

The screenshot shows the FCM interface for creating a capture session. The 'Tools' menu is open, and 'Packet Capture' is selected. The 'Capture Session' button is highlighted.

4. Select Ethernet1/2, provide the **Session Name**, apply the capture filter and click **Save and Run** to activate the capture:



FXOS CLI

Follow these steps on FXOS CLI to configure packet captures on backplane interfaces:

1. Identify the application type and identifier:

```
firepower# scope ssa
firepower /ssa# show app-instance
App Name   Identifier Slot ID   Admin State Oper State   Running Version Startup Version
Deploy Type Turbo Mode Profile Name Cluster State   Cluster Role
-----
ftd        ftd1         1           Enabled    Online      7.2.0.82      7.2.0.82
Native     No           Not Applicable None
```

2. Identify the IP protocol number in <https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>. In this case, the ICMP protocol number is 1.

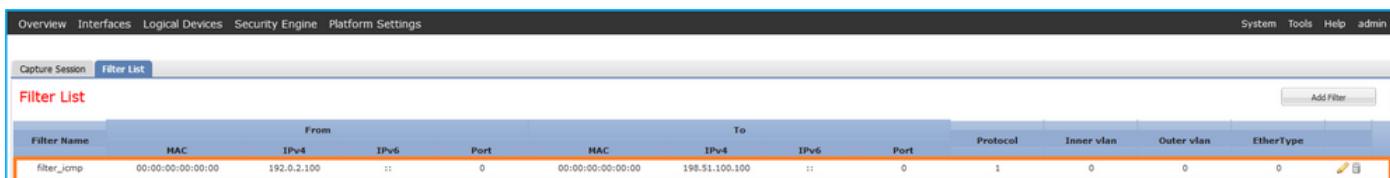
3. Create a capture session:

```
2.
firepower# scope packet-capture
firepower /packet-capture # create filter filter_icmp
firepower /packet-capture/filter* # set destip 198.51.100.100
firepower /packet-capture/filter* # set protocol 1
firepower /packet-capture/filter* # set srcip 192.0.2.100
firepower /packet-capture/filter* # exit
firepower /packet-capture* # create session cap1
firepower /packet-capture/session* # create phy-port Ethernet1/2
firepower /packet-capture/session/phy-port* # set app ftd
firepower /packet-capture/session/phy-port* # set app-identifier ftd1
firepower /packet-capture/session/phy-port* # set filter filter_icmp
firepower /packet-capture/session/phy-port* # exit
firepower /packet-capture/session* # enable
firepower /packet-capture/session* # commit
firepower /packet-capture/session #
```

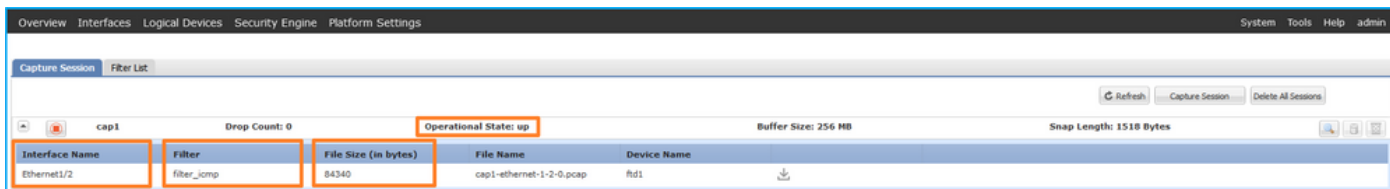
Verification

FCM

Verify the **Interface Name**, ensure that the **Operational Status** is up and that the **File Size (in bytes)** increases:



Verify the **Interface Name**, the **Filter**, ensure the **Operational Status** is up, and the **File Size (in bytes)** increases in **Tools > Packet Capture > Capture Session**:



FXOS CLI

Verify the capture details in **scope packet-capture**:

```
firepower# scope packet-capture
firepower /packet-capture # show filter detail
```

Configure a filter for packet capture:

```
Name: filter_icmp
Protocol: 1
Ivlan: 0
Ovlan: 0
Src Ip: 192.0.2.100
Dest Ip: 198.51.100.100
Src MAC: 00:00:00:00:00:00
Dest MAC: 00:00:00:00:00:00
Src Port: 0
Dest Port: 0
Ethertype: 0
Src Ipv6: ::
Dest Ipv6: ::
```

```
firepower /packet-capture # show session cap1
```

Traffic Monitoring Session:

```
Packet Capture Session Name: cap1
Session: 1
Admin State: Enabled
Oper State: Up
Oper State Reason: Active
Config Success: Yes
Config Fail Reason:
Append Flag: Overwrite
Session Mem Usage: 256 MB
Session Pcap Snap Len: 1518 Bytes
Error Code: 0
Drop Count: 0
```

Physical ports involved in Packet Capture:

```
Slot Id: 1
Port Id: 2
Pcapfile: /workspace/packet-capture/session-1/cap1-ethernet-1-2-0.pcap
Pcapsize: 213784 bytes
Filter: filter_icmp
Sub Interface: 0
Application Instance Identifier: ftd1
```

Application Name: ftd

Collect capture files

Follow the steps in the section **Collect Firepower 4100/9300 Internal Switch Capture Files**.

Capture file analysis

Use a packet capture file reader application to open the capture file. Select the first packet and check the key points

1. Only ICMP echo request packets are captured. Each packet is captured and shown 2 times.
2. The original packet header is without the VLAN tag.
3. The internal switch inserts additional port VLAN tag **102** that identifies the ingress interface Ethernet1/2.
4. The internal switch inserts an additional VN tag.

The screenshot displays a network traffic capture analysis. The top section is a table of captured packets:

No.	Time	Source	Destination	Protocol	Length	IP ID	TTL	Info
1	2022-08-02 15:46:55.603277760	192.0.2.100	198.51.100.100	ICMP	108	0x0012 (18)	64	Echo (ping) request id=0x0018, seq=349/23809, ttl=64 (no r...
2	2022-08-02 15:46:55.603279688	192.0.2.100	198.51.100.100	ICMP	102	0x0012 (18)	64	Echo (ping) request id=0x0018, seq=349/23809, ttl=64 (no r...
3	2022-08-02 15:46:56.627139252	192.0.2.100	198.51.100.100	ICMP	108	0x00db (219)	64	Echo (ping) request id=0x0018, seq=350/24065, ttl=64 (no r...
4	2022-08-02 15:46:56.627140919	192.0.2.100	198.51.100.100	ICMP	102	0x00db (219)	64	Echo (ping) request id=0x0018, seq=350/24065, ttl=64 (no r...
5	2022-08-02 15:46:57.651185193	192.0.2.100	198.51.100.100	ICMP	108	0x01cb (459)	64	Echo (ping) request id=0x0018, seq=351/24321, ttl=64 (no r...
6	2022-08-02 15:46:57.651186787	192.0.2.100	198.51.100.100	ICMP	102	0x01cb (459)	64	Echo (ping) request id=0x0018, seq=351/24321, ttl=64 (no r...
7	2022-08-02 15:46:58.675153317	192.0.2.100	198.51.100.100	ICMP	108	0x01d6 (470)	64	Echo (ping) request id=0x0018, seq=352/24577, ttl=64 (no r...
8	2022-08-02 15:46:58.675154503	192.0.2.100	198.51.100.100	ICMP	102	0x01d6 (470)	64	Echo (ping) request id=0x0018, seq=352/24577, ttl=64 (no r...
9	2022-08-02 15:46:59.699152639	192.0.2.100	198.51.100.100	ICMP	108	0x01f4 (500)	64	Echo (ping) request id=0x0018, seq=353/24833, ttl=64 (no r...
10	2022-08-02 15:46:59.699153835	192.0.2.100	198.51.100.100	ICMP	102	0x01f4 (500)	64	Echo (ping) request id=0x0018, seq=353/24833, ttl=64 (no r...
11	2022-08-02 15:47:00.723142641	192.0.2.100	198.51.100.100	ICMP	108	0x01f9 (505)	64	Echo (ping) request id=0x0018, seq=354/25089, ttl=64 (no r...
12	2022-08-02 15:47:00.723144643	192.0.2.100	198.51.100.100	ICMP	102	0x01f9 (505)	64	Echo (ping) request id=0x0018, seq=354/25089, ttl=64 (no r...
13	2022-08-02 15:47:01.747162204	192.0.2.100	198.51.100.100	ICMP	108	0x026e (622)	64	Echo (ping) request id=0x0018, seq=355/25345, ttl=64 (no r...
14	2022-08-02 15:47:01.747163783	192.0.2.100	198.51.100.100	ICMP	102	0x026e (622)	64	Echo (ping) request id=0x0018, seq=355/25345, ttl=64 (no r...
15	2022-08-02 15:47:02.771209952	192.0.2.100	198.51.100.100	ICMP	108	0x02bc (700)	64	Echo (ping) request id=0x0018, seq=356/25601, ttl=64 (no r...
16	2022-08-02 15:47:02.771211062	192.0.2.100	198.51.100.100	ICMP	102	0x02bc (700)	64	Echo (ping) request id=0x0018, seq=356/25601, ttl=64 (no r...
17	2022-08-02 15:47:03.772258550	192.0.2.100	198.51.100.100	ICMP	108	0x032f (815)	64	Echo (ping) request id=0x0018, seq=357/25857, ttl=64 (no r...
18	2022-08-02 15:47:03.772259724	192.0.2.100	198.51.100.100	ICMP	102	0x032f (815)	64	Echo (ping) request id=0x0018, seq=357/25857, ttl=64 (no r...
19	2022-08-02 15:47:04.791118519	192.0.2.100	198.51.100.100	ICMP	108	0x040f (1039)	64	Echo (ping) request id=0x0018, seq=358/26113, ttl=64 (no r...
20	2022-08-02 15:47:04.791119721	192.0.2.100	198.51.100.100	ICMP	102	0x040f (1039)	64	Echo (ping) request id=0x0018, seq=358/26113, ttl=64 (no r...

The bottom section shows a detailed view of the first packet (Frame 1) with the following headers highlighted:

- VN-Tag**: Direction: From Bridge, Pointer: vif_id, Destination: 10, Looped: No, Reserved: 0, Version: 0, Source: 0. Type: 802.1Q Virtual LAN (0x8100).
- 802.1Q Virtual LAN**: Priority: Best Effort (default) (0), DEI: Ineligible, ID: 102. Type: IPv4 (0x0800).
- Internet Protocol Version 4**: Src: 192.0.2.100, Dst: 198.51.100.100.
- Internet Control Message Protocol**.

Select the second packet, and check the key points:

1. Only ICMP echo request packets are captured. Each packet is captured and shown 2 times.
2. The original packet header is without the VLAN tag.
3. The internal switch inserts additional port VLAN tag **102** that identifies the ingress interface Ethernet1/2.

No.	Time	Source	Destination	Protocol	Length	ID	TTL	Info
1	2022-08-02 15:46:55.603277760	192.0.2.100	198.51.100.100	ICMP	108	0x0012 (18)	64	Echo (ping) request id=0x0018, seq=349/23809, ttl=64 (no r
2	2022-08-02 15:46:55.603279688	192.0.2.100	198.51.100.100	ICMP	102	0x0012 (18)	64	Echo (ping) request id=0x0018, seq=349/23809, ttl=64 (no r
3	2022-08-02 15:46:56.627139252	192.0.2.100	198.51.100.100	ICMP	108	0x00db (219)	64	Echo (ping) request id=0x0018, seq=350/24065, ttl=64 (no r
4	2022-08-02 15:46:56.627140919	192.0.2.100	198.51.100.100	ICMP	102	0x00db (219)	64	Echo (ping) request id=0x0018, seq=350/24065, ttl=64 (no r
5	2022-08-02 15:46:57.651185193	192.0.2.100	198.51.100.100	ICMP	108	0x01cb (459)	64	Echo (ping) request id=0x0018, seq=351/24321, ttl=64 (no r
6	2022-08-02 15:46:57.651186787	192.0.2.100	198.51.100.100	ICMP	102	0x01cb (459)	64	Echo (ping) request id=0x0018, seq=351/24321, ttl=64 (no r
7	2022-08-02 15:46:58.675153317	192.0.2.100	198.51.100.100	ICMP	108	0x01d6 (470)	64	Echo (ping) request id=0x0018, seq=352/24577, ttl=64 (no r
8	2022-08-02 15:46:58.675154503	192.0.2.100	198.51.100.100	ICMP	102	0x01d6 (470)	64	Echo (ping) request id=0x0018, seq=352/24577, ttl=64 (no r
9	2022-08-02 15:46:59.699152639	192.0.2.100	198.51.100.100	ICMP	108	0x01f4 (500)	64	Echo (ping) request id=0x0018, seq=353/24833, ttl=64 (no r
10	2022-08-02 15:46:59.699153835	192.0.2.100	198.51.100.100	ICMP	102	0x01f4 (500)	64	Echo (ping) request id=0x0018, seq=353/24833, ttl=64 (no r
11	2022-08-02 15:47:00.723142641	192.0.2.100	198.51.100.100	ICMP	108	0x01f9 (505)	64	Echo (ping) request id=0x0018, seq=354/25089, ttl=64 (no r
12	2022-08-02 15:47:00.723144643	192.0.2.100	198.51.100.100	ICMP	102	0x01f9 (505)	64	Echo (ping) request id=0x0018, seq=354/25089, ttl=64 (no r
13	2022-08-02 15:47:01.747162204	192.0.2.100	198.51.100.100	ICMP	108	0x026e (622)	64	Echo (ping) request id=0x0018, seq=355/25345, ttl=64 (no r
14	2022-08-02 15:47:01.747163783	192.0.2.100	198.51.100.100	ICMP	102	0x026e (622)	64	Echo (ping) request id=0x0018, seq=355/25345, ttl=64 (no r
15	2022-08-02 15:47:02.771209952	192.0.2.100	198.51.100.100	ICMP	108	0x02bc (700)	64	Echo (ping) request id=0x0018, seq=356/25601, ttl=64 (no r
16	2022-08-02 15:47:02.771211062	192.0.2.100	198.51.100.100	ICMP	102	0x02bc (700)	64	Echo (ping) request id=0x0018, seq=356/25601, ttl=64 (no r
17	2022-08-02 15:47:03.772258550	192.0.2.100	198.51.100.100	ICMP	108	0x032f (815)	64	Echo (ping) request id=0x0018, seq=357/25857, ttl=64 (no r
18	2022-08-02 15:47:03.772259724	192.0.2.100	198.51.100.100	ICMP	102	0x032f (815)	64	Echo (ping) request id=0x0018, seq=357/25857, ttl=64 (no r
19	2022-08-02 15:47:04.791118519	192.0.2.100	198.51.100.100	ICMP	108	0x040f (1039)	64	Echo (ping) request id=0x0018, seq=358/26113, ttl=64 (no r
20	2022-08-02 15:47:04.791119721	192.0.2.100	198.51.100.100	ICMP	102	0x040f (1039)	64	Echo (ping) request id=0x0018, seq=358/26113, ttl=64 (no r

```

> Frame 2: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface capture_u0_1, i
> Ethernet II, Src: VMware 9d:e8:be (00:50:56:9d:e8:be), Dst: Cisco b9:77:0e (58:97:bd:b9:77:0e)
  802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 102
    000. .... = Priority: Best Effort (default) (0)
    ...0 .... = DEI: Ineligible
    .... 0000 0110 0110 = ID: 102
    Type: IPv4 (0x0800)
  > Internet Protocol Version 4, Src: 192.0.2.100, Dst: 198.51.100.100
  > Internet Control Message Protocol
  
```

Explanation

When a packet capture on a front interface is configured, the switch simultaneously captures each packet twice:

- After the insertion of the port VLAN tag.
- After the insertion of the VN tag.

In the order of operations, the VN tag is inserted at a later stage than the port VLAN tag insertion. But in the capture file, the packet with the VN tag is shown earlier than the packet with the port VLAN tag.

When a capture filter is applied only the packets that match the filter in the ingress direction are captured.

This table summarizes the task:

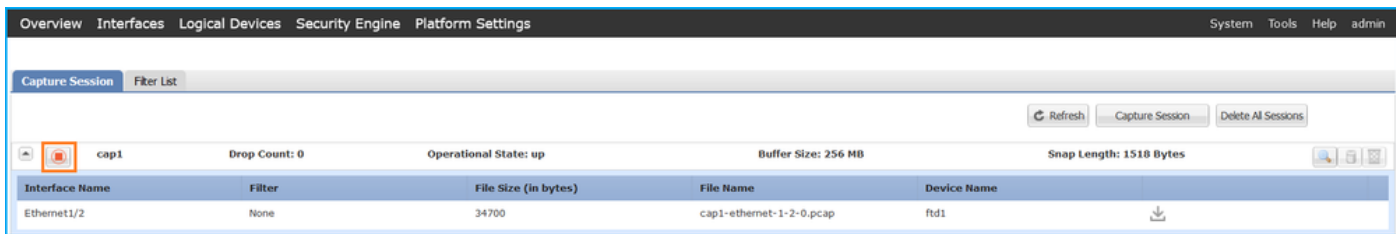
Task	Capture point	Internal port	VLAN in captured packets	Direction	User filter	Captured traffic
Configure and verify a packet capture with a filter on the front interface Ethernet1/2	Ethernet1/2		102	Ingress only	Protocol: ICMP Source:192.0.2.100 Destination: 198.51.100.100	ICMP echo requests from host 192.0.2.100 to 198.51.100.100

Collect Firepower 4100/9300 Internal Switch Capture Files

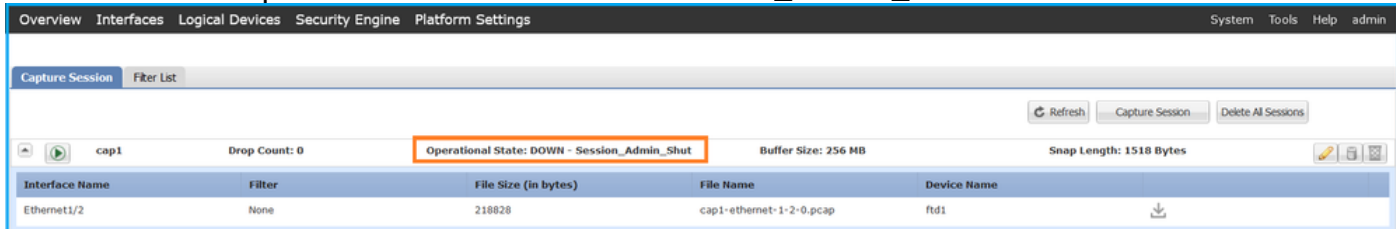
FCM

Follow these steps on FCM to collect internal switch capture files:

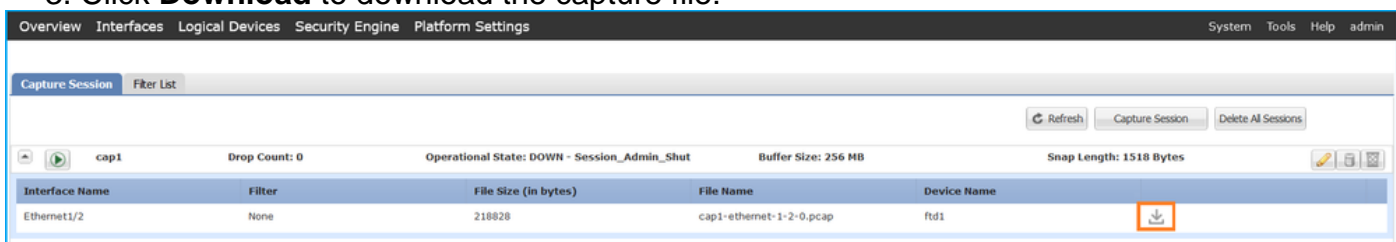
1. Click the **Disable Session** button to stop the active capture:



2. Ensure the operational state is **DOWN - Session_Admin_Shut**:



3. Click **Download** to download the capture file:



In the case of port-channel interfaces, repeat this step for each member interface.

FXOS CLI

Follow these steps on the FXOS CLI to collect capture files:

1. Stop the active capture:

```
firepower# scope packet-capture
firepower /packet-capture # scope session cap1
firepower /packet-capture/session # disable
firepower /packet-capture/session* # commit
firepower /packet-capture/session # up
firepower /packet-capture # show session cap1 detail
```

Traffic Monitoring Session:

```
Packet Capture Session Name: cap1
Session: 1
Admin State: Disabled
Oper State: Down
Oper State Reason: Admin Disable
Config Success: Yes
Config Fail Reason:
Append Flag: Overwrite
Session Mem Usage: 256 MB
Session Pcap Snap Len: 1518 Bytes
Error Code: 0
Drop Count: 0
```

Physical ports involved in Packet Capture:

```
Slot Id: 1
Port Id: 2
Pcapfile: /workspace/packet-capture/session-1/cap1-ethernet-1-2-0.pcap
```



```
Pcapsize: 115744 bytes
Filter:
Sub Interface: 0
Application Instance Identifier: ftd1
Application Name: ftd
```

2. Upload the capture file from the **local-mgmt** command scope:

```
firepower# connect local-mgmt
firepower(local-mgmt)# copy /packet-capture/session-1/cap1-ethernet-1-2-0.pcap ?
ftp:          Dest File URI
http:         Dest File URI
https:        Dest File URI
scp:          Dest File URI
sftp:         Dest File URI
tftp:         Dest File URI
usbdrive:     Dest File URI
volatile:     Dest File URI
workspace:    Dest File URI
```

```
firepower(local-mgmt)# copy /packet-capture/session-1/cap1-ethernet-1-2-0.pcap
ftp://ftpuser@10.10.10.1/cap1-ethernet-1-2-0.pcap
Password:
```

In the case of port-channel interfaces, copy the capture file for each member interface.

Guidelines, Limitations, and Best Practices for Internal Switch Packet Capture

For the guidelines and limitations related to Firepower 4100/9300 internal switch capture refer to the *Cisco Firepower 4100/9300 FXOS Chassis Manager Configuration Guide* or *Cisco Firepower 4100/9300 FXOS CLI Configuration Guide*, chapter **Troubleshooting**, section **Packet Capture**.

This is the list of best practices based on the usage of packet capture in TAC cases:

- Be aware of guidelines and limitations.
- Capture packets on all port-channel member interfaces and analyze all capture files.
- Use capture filters.
- Consider the impact of NAT on packet IP addresses when a capture filter is configured.
- Increase or decrease the **Snap Len** that specifies frame size in case it differs from the default value of 1518 bytes. Shorter size results in an increased number of captured packets and vice versa.
- Adjust the **Buffer Size** as needed.
- Be aware of the **Drop Count** on FCM or FXOS CLI. Once the buffer size limit is reached, the drop count counter increases.
- Use the filter **!vntag** on Wireshark to display only packets without the VN-tag. This is useful to hide VN-tagged packets in the front interface packet capture files.
- Use the filter **frame.number&1** on Wireshark to display only odd frames. This is useful to hide duplicate packets in the backplane interface packet capture files.
- In the case of protocols like TCP, Wireshark by default applies colorization rules that display packets with specific conditions in different colors. In the case of internal switch captures due to duplicate packets in capture files, the packet can be colored and marked in a false-positive way. If you analyze packet capture files and apply any filter, then export the displayed packets to a new file and open the new file instead.

Configuration and Verification on Secure Firewall 3100

Unlike Firepower 4100/9300, the internal switch captures on the Secure Firewall 3100 are configured on the application command line interface via the **capture <name> switch** command, where the **switch** option specifies that the captures are configured on the internal switch.

This is the **capture** command with the **switch** option:

```
> capture cap_sw switch ?
buffer          Configure size of capture buffer, default is 256MB
ethernet-type   Capture Ethernet packets of a particular type, default is IP
interface       Capture packets on a specific interface
ivlan           Inner Vlan
match           Capture packets based on match criteria
ovlan           Outer Vlan
packet-length   Configure maximum length to save from each packet, default is
                64 bytes
real-time       Display captured packets in real-time. Warning: using this
                option with a slow console connection may result in an
                excessive amount of non-displayed packets due to performance
                limitations.
stop            Stop packet capture
trace           Trace the captured packets
type            Capture packets based on a particular type
<cr>
```

General steps for packet capture configuration are as follows:

1. Specify an ingress interface:

Switch capture configuration accepts the ingress interface **nameif**. The user can specify data interfaces names, internal uplink, or the management interfaces:

```
> capture capsw switch interface ?
Available interfaces to listen:
in_data_uplink1  Capture packets on internal data uplink1 interface
in_mgmt_uplink1  Capture packets on internal mgmt uplink1 interface
inside           Name of interface Ethernet1/1.205

management       Name of interface Management1/1
```

2. Specify the ethernet frame EtherType. The default EtherType is IP. The **ethernet-type** option values specify the EtherType:

```
> capture capsw switch interface inside ethernet-type ?
802.1Q
<0-65535> Ethernet type
arp
ip
ip6
pppoed
pppoes
rarp
sgt
vlan
```

3. Specify the match conditions. The capture **match** option specifies the match criteria:

```
> capture capsw switch interface inside match ?
```

```
<0-255> Enter protocol number (0 - 255)
```

```
ah
eigrp
esp
gre
icmp
icmp6
igmp
igrp
ip
ipinip
ipsec
mac      Mac-address filter
nos
ospf
pcp
pim
pptp
sctp
snp
spi      SPI value
tcp
udp
<cr>
```

4. Specify other optional parameters such as the buffer size, the packet length, and so on.

5. Enable the capture. The command **no capture <name> switch stop** activates the capture:

```
> capture capsw switch interface inside match ip
```

```
>no capture capsw switch stop
```

6. Verify the capture details:

- Administrative status is **enabled**, and operational status is **up** and active.
- Packet capture file size **Pcapsize** increases.
- The number of captured packets in the output of the **show capture <cap_name>** is non-zero.
- Capture path **Pcapfile**. The captured packets are automatically saved in the **/mnt/disk0/packet-capture/** folder.
- Capture conditions. The software automatically creates capture filters based on capture conditions.

```
> show capture capsw
```

```
27 packet captured on disk using switch capture
```

```
Reading of capture file from disk is not supported
```

```
>show capture capsw detail
```

```
Packet Capture info
```

```
  Name:          capsw
  Session:       1
  Admin State:   enabled
  Oper State:    up
  Oper State Reason: Active
  Config Success: yes
  Config Fail Reason:
  Append Flag:   overwrite
  Session Mem Usage: 256
  Session Pcap Snap Len: 1518
  Error Code:    0
```

Drop Count: 0

Total Physical ports involved in Packet Capture: 1

Physical port:

Slot Id: 1
Port Id: 1
Pcapfile: /mnt/disk0/packet-capture/sess-1-capsw-ethernet-1-1-0.pcap
Pcapsize: 18838
Filter: capsw-1-1

Packet Capture Filter Info

Name: capsw-1-1
Protocol: 0
Ivlan: 0
Ovlan: 205
Src Ip: 0.0.0.0
Dest Ip: 0.0.0.0
Src Ipv6: ::
Dest Ipv6: ::
Src MAC: 00:00:00:00:00:00
Dest MAC: 00:00:00:00:00:00
Src Port: 0
Dest Port: 0
Ethertype: 0

Total Physical breakout ports involved in Packet Capture: 0

0 packet captured on disk using switch capture

Reading of capture file from disk is not supported

7. Stop the captures when needed:

```
> capture capsw switch stop
```

```
>show capture capsw detail
```

Packet Capture info

Name: capsw
Session: 1
Admin State: disabled
Oper State: down
Oper State Reason: Session_Admin_Shut
Config Success: yes
Config Fail Reason:
Append Flag: overwrite
Session Mem Usage: 256
Session Pcap Snap Len: 1518
Error Code: 0
Drop Count: 0

Total Physical ports involved in Packet Capture: 1

Physical port:

Slot Id: 1
Port Id: 1
Pcapfile: /mnt/disk0/packet-capture/sess-1-capsw-ethernet-1-1-0.pcap
Pcapsize: 24
Filter: capsw-1-1

Packet Capture Filter Info

Name: capsw-1-1
Protocol: 0
Ivlan: 0
Ovlan: 205
Src Ip: 0.0.0.0
Dest Ip: 0.0.0.0
Src Ipv6: ::

```

Dest Ipv6:      ::
Src MAC:       00:00:00:00:00:00
Dest MAC:      00:00:00:00:00:00
Src Port:      0
Dest Port:     0
Ethertype:     0

```

Total Physical breakout ports involved in Packet Capture: 0
 0 packet captured on disk using switch capture
 Reading of capture file from disk is not supported

8. Collect the capture files. Follow the steps in the section **Collect Secure Firewall 3100 Internal Switch Capture Files**.

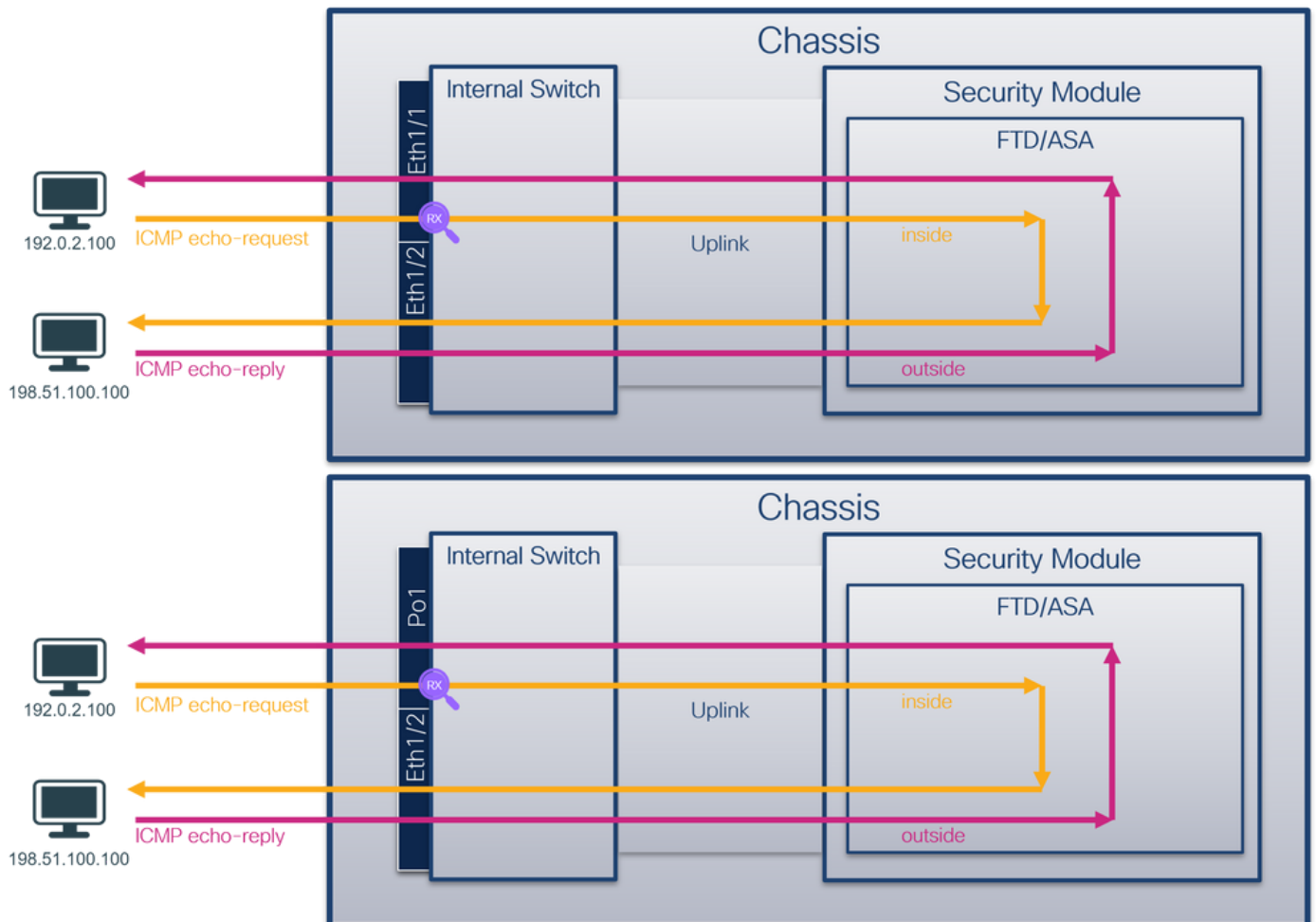
In version 7.2, the internal switch capture configuration is not supported on the FMC or FDM. In the case of ASA software version 9.18(1) and later, internal switch captures can be configured in ASDM versions 7.18.1.x and later.

These scenarios cover common use cases of Secure Firewall 3100 internal switch captures.

Packet Capture on a Physical or Port-channel Interface

Use the FTD or ASA CLI to configure and verify a packet capture on interface Ethernet1/1 or Portchannel1 interface. Both interfaces have the nameif **inside**.

Topology, packet flow, and the capture points



Configuration

Follow these steps on ASA or FTD CLI to configure a packet capture on interface Ethernet1/1 or Port-channel1:

1. Verify the nameif:

```
> show nameif
Interface          Name          Security
Ethernet1/1       inside       0
Ethernet1/2       outside      0
Management1/1    diagnostic   0
```

```
> show nameif
Interface          Name          Security
Port-channel1     inside       0
Ethernet1/2       outside      0
Management1/1    diagnostic   0
```

2. Create a capture session:

```
> capture capsw switch interface inside
```

3. Enable the capture session:

```
> no capture capsw switch stop
```

Verification

Verify the capture session name, administrative and operational state, interface slot, and identifier. Ensure the **Pcapsize** value in bytes increases and the number of captured packets is non-zero:

```
> show capture capsw detail
Packet Capture info
  Name:          capsw
  Session:       1
  Admin State:   enabled
  Oper State:    up
  Oper State Reason: Active
  Config Success: yes
  Config Fail Reason:
  Append Flag:   overwrite
  Session Mem Usage: 256
  Session Pcap Snap Len: 1518
  Error Code:    0
  Drop Count:    0
```

```
Total Physical ports involved in Packet Capture: 1
```

```
Physical port:
```

```
  Slot Id:      1
  Port Id:      1
  Pcapfile:     /mnt/disk0/packet-capture/sess-1-capsw-ethernet-1-1-0.pcap
  Pcapsize:     12653
  Filter:       capsw-1-1
```

```
Packet Capture Filter Info
```

```
  Name:         capsw-1-1
  Protocol:     0
  Ivlan:        0
  Ovlan:        0
```

Src Ip: 0.0.0.0
Dest Ip: 0.0.0.0
Src Ipv6: ::
Dest Ipv6: ::
Src MAC: 00:00:00:00:00:00
Dest MAC: 00:00:00:00:00:00
Src Port: 0
Dest Port: 0
Ethertype: 0

Total Physical breakout ports involved in Packet Capture: 0

79 packets captured on disk using switch capture

Reading of capture file from disk is not supported

In the case of Port-channel1 the capture is configured on all member interfaces:

> **show capture capsw detail**

Packet Capture info

Name: capsw
Session: 1
Admin State: enabled
Oper State: up
Oper State Reason: Active
Config Success: yes
Config Fail Reason:
Append Flag: overwrite
Session Mem Usage: 256
Session Pcap Snap Len: 1518
Error Code: 0
Drop Count: 0

Total Physical ports involved in Packet Capture: 2

Physical port:

Slot Id: 1
Port Id: 4
Pcapfile: /mnt/disk0/packet-capture/sess-1-capsw-ethernet-1-4-0.pcap
Pcapsize: 28824
Filter: capsw-1-4

Packet Capture Filter Info

Name: capsw-1-4
Protocol: 0
Ivlan: 0
Ovlan: 0
Src Ip: 0.0.0.0
Dest Ip: 0.0.0.0
Src Ipv6: ::
Dest Ipv6: ::
Src MAC: 00:00:00:00:00:00
Dest MAC: 00:00:00:00:00:00
Src Port: 0
Dest Port: 0
Ethertype: 0

Physical port:

Slot Id: 1
Port Id: 3
Pcapfile: /mnt/disk0/packet-capture/sess-1-capsw-ethernet-1-3-0.pcap
Pcapsize: 18399
Filter: capsw-1-3

Packet Capture Filter Info

```
Name:          capsw-1-3
Protocol:      0
Ivlan:        0
Ovlan:        0
Src Ip:        0.0.0.0
Dest Ip:       0.0.0.0
Src Ipv6:      ::
Dest Ipv6:     ::
Src MAC:       00:00:00:00:00:00
Dest MAC:      00:00:00:00:00:00
Src Port:      0
Dest Port:     0
Ethertype:    0
```

Total Physical breakout ports involved in Packet Capture: 0

56 packet captured on disk using switch capture

Reading of capture file from disk is not supported

The port-channel member interfaces can be verified in the FXOS **local-mgmt** command shell via the **show portchannel summary** command:

```
> connect fxos
```

```
...
```

```
KSEC-FPR3100-1 connect local-mgmt
```

```
KSEC-FPR3100-1(local-mgmt) show portchannel summary
```

```
Flags:  D - Down          P - Up in port-channel (members)
```

```
I - Individual  H - Hot-standby (LACP only)
```

```
s - Suspended   r - Module-removed
```

```
S - Switched   R - Routed
```

```
U - Up (port-channel)
```

```
M - Not in use. Min-links not met
```

```
-----
Group Port-      Type      Protocol  Member Ports
  Channel
-----
1    Po1(U)      Eth       LACP      Eth1/3(P)  Eth1/4(P)
```

```
LACP KeepAlive Timer:
```

```
-----
Channel  PeerKeepAliveTimerFast
-----
```

```
1    Po1(U)      False
```

```
Cluster LACP Status:
```

```
-----
Channel  ClusterSpanned  ClusterDetach  ClusterUnitID  ClusterSysID
-----
```

```
1    Po1(U)      False          False          0              clust
```

To access the FXOS on ASA, run the **connect fxos admin** command. In the case of multi-context, run the command in the admin context.

Collect capture files

Follow the steps in the section **Collect Secure Firewall 3100 Internal Switch Capture Files**.

Capture file analysis

Use a packet capture file reader application to open the capture files for Ethernet1/1. Select the first packet and check the key points:

1. Only ICMP echo request packets are captured.
2. The original packet header is without the VLAN tag.

The screenshot shows a packet capture tool interface. The top pane displays a list of 18 captured packets, all of which are ICMP Echo (ping) requests from source 192.0.2.100 to destination 198.51.100.100. The second pane shows the details of the first packet (No. 1), highlighting the Ethernet II, Internet Protocol Version 4, and Internet Control Message Protocol headers. The IP ID is 0x9a10 (39440) and the TTL is 64. The packet length is 102 bytes.

Open the capture files for Portchannel1 member interfaces. Select the first packet and check the key points:

1. Only ICMP echo request packets are captured.
2. The original packet header is without the VLAN tag.

The screenshot shows a packet capture tool interface. The top pane displays a list of 18 captured packets, all of which are ICMP Echo (ping) requests from source 192.0.2.100 to destination 198.51.100.100. The second pane shows the details of the first packet (No. 1), highlighting the Ethernet II, Internet Protocol Version 4, and Internet Control Message Protocol headers. The IP ID is 0x9296 (37526) and the TTL is 64. The packet length is 102 bytes.

Explanation

The switch captures are configured on interfaces Ethernet1/1 or Portchannel1.

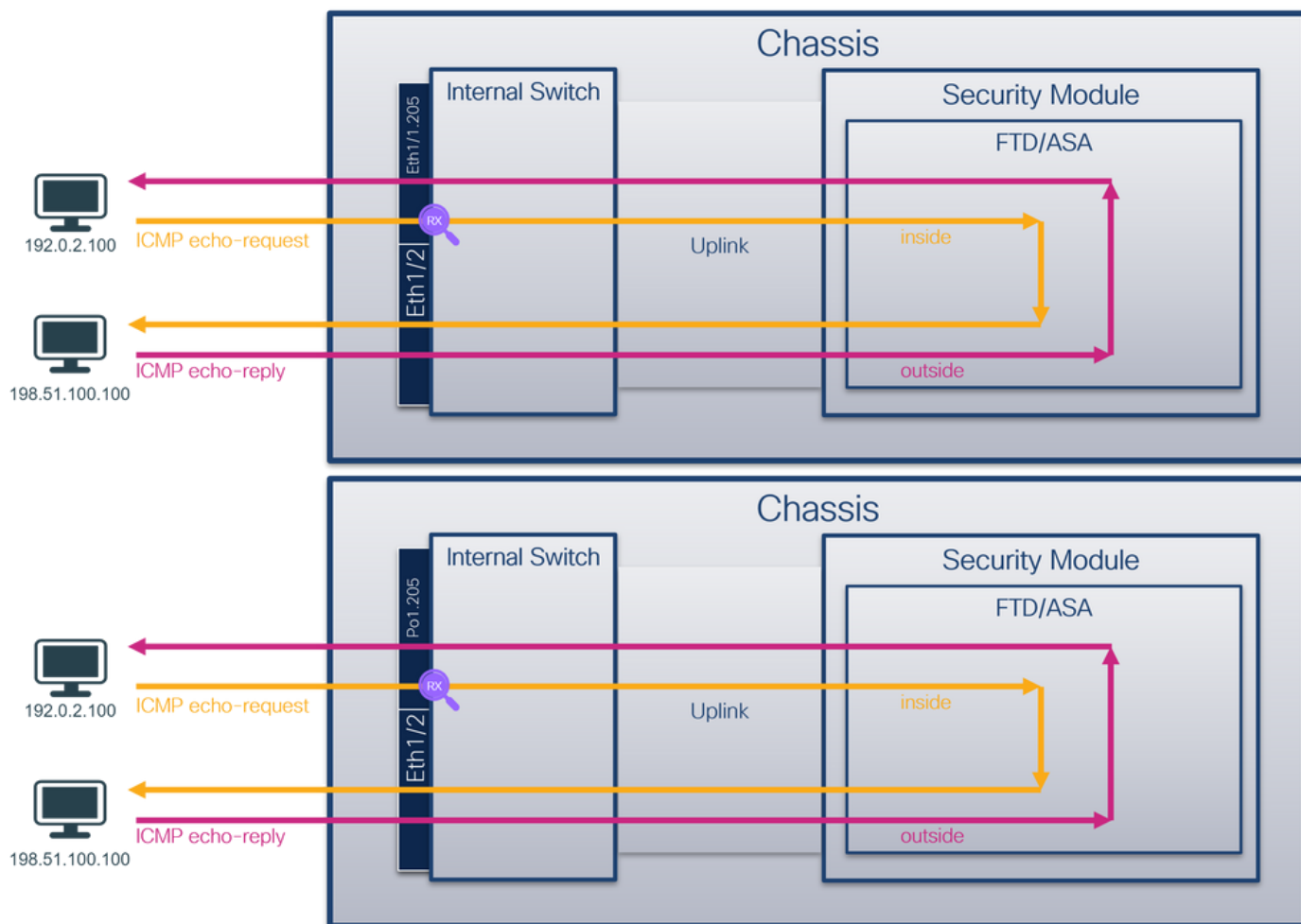
This table summarizes the task:

Task	Capture point	Internal filter	Direction	Captured traffic
Configure and verify a packet capture on interface Ethernet1/1	Ethernet1/1	None	Ingress only	ICMP echo requests from host 192.0.2.100 to host 198.51.100.100
Configure and verify a packet capture on interface Portchannel1 with member interfaces Ethernet1/3 and Ethernet1/4	Ethernet1/3 Ethernet1/4	None	Ingress only	ICMP echo requests from host 192.0.2.100 to host 198.51.100.100

Packet Capture on a Subinterface of a Physical or Port-channel Interface

Use the FTD or ASA CLI to configure and verify a packet capture on subinterfaces Ethernet1/1.205 or Portchannel1.205. Both subinterfaces have the nameif **inside**.

Topology, packet flow, and the capture points



Configuration

Follow these steps on ASA or FTD CLI to configure a packet capture on interface Ethernet1/1 or Port-channel1:

1. Verify the nameif:

```
> show nameif
Interface          Name          Security
Ethernet1/1.205   inside        0
Ethernet1/2       outside       0
Management1/1     diagnostic    0
```

```
> show nameif
Interface          Name          Security
Port-channel1.205 inside        0
Ethernet1/2       outside       0
Management1/1     diagnostic    0
```

2. Create a capture session:

```
> capture capsw switch interface inside
```

3. Enable the capture session:

```
> no capture capsw switch stop
```

Verification

Verify the capture session name, administrative and operational state, interface slot, and identifier. Ensure the **Pcapsize** value in bytes increases and the number of captured packets is non-zero:

```
> show capture capsw detail
```

Packet Capture info

```
Name: capsw
Session: 1
Admin State: enabled
Oper State: up
Oper State Reason: Active
Config Success: yes
Config Fail Reason:
Append Flag: overwrite
Session Mem Usage: 256
Session Pcap Snap Len: 1518
Error Code: 0
Drop Count: 0
```

Total Physical ports involved in Packet Capture: 1

Physical port:

```
Slot Id: 1
Port Id: 1
Pcapfile: /mnt/disk0/packet-capture/sess-1-capsw-ethernet-1-1-0.pcap
Pcapsize: 6360
Filter: capsw-1-1
```

Packet Capture Filter Info

```
Name: capsw-1-1
Protocol: 0
Ivlan: 0
Ovlan: 205
Src Ip: 0.0.0.0
Dest Ip: 0.0.0.0
Src Ipv6: ::
Dest Ipv6: ::
Src MAC: 00:00:00:00:00:00
Dest MAC: 00:00:00:00:00:00
Src Port: 0
Dest Port: 0
Ethertype: 0
```

Total Physical breakout ports involved in Packet Capture: 0

46 packets captured on disk using switch capture

Reading of capture file from disk is not supported

In this case, a filter with outer VLAN **Ovlan=205** is created and applied to the interface.

In the case of Port-channel1 the capture with a filter **Ovlan=205** is configured on all member interfaces:

> show capture capsw detail

Packet Capture info

Name: capsw
Session: 1
Admin State: enabled
Oper State: up
Oper State Reason: Active
Config Success: yes
Config Fail Reason:
Append Flag: overwrite
Session Mem Usage: 256
Session Pcap Snap Len: 1518
Error Code: 0
Drop Count: 0

Total Physical ports involved in Packet Capture: 2

Physical port:

Slot Id: 1
Port Id: 4
Pcapfile: /mnt/disk0/packet-capture/sess-1-capsw-ethernet-1-4-0.pcap
Pcapsize: 23442
Filter: capsw-1-4

Packet Capture Filter Info

Name: capsw-1-4
Protocol: 0
Ivlan: 0
Ovlan: 205
Src Ip: 0.0.0.0
Dest Ip: 0.0.0.0
Src Ipv6: ::
Dest Ipv6: ::
Src MAC: 00:00:00:00:00:00
Dest MAC: 00:00:00:00:00:00
Src Port: 0
Dest Port: 0
Ethertype: 0

Physical port:

Slot Id: 1
Port Id: 3
Pcapfile: /mnt/disk0/packet-capture/sess-1-capsw-ethernet-1-3-0.pcap
Pcapsize: 5600
Filter: capsw-1-3

Packet Capture Filter Info

Name: capsw-1-3
Protocol: 0
Ivlan: 0
Ovlan: 205
Src Ip: 0.0.0.0
Dest Ip: 0.0.0.0
Src Ipv6: ::
Dest Ipv6: ::
Src MAC: 00:00:00:00:00:00
Dest MAC: 00:00:00:00:00:00
Src Port: 0
Dest Port: 0
Ethertype: 0

Total Physical breakout ports involved in Packet Capture: 0

49 packet captured on disk using switch capture

Reading of capture file from disk is not supported

The port-channel member interfaces can be verified in the FXOS **local-mgmt** command shell via the **show portchannel summary** command:

```
> connect fxos
...
KSEC-FPR3100-1 connect local-mgmt
KSEC-FPR3100-1(local-mgmt) show portchannel summary
Flags:  D - Down          P - Up in port-channel (members)
I - Individual  H - Hot-standby (LACP only)
s - Suspended   r - Module-removed
S - Switched    R - Routed
U - Up (port-channel)
M - Not in use. Min-links not met
-----
Group Port-      Type      Protocol  Member Ports
  Channel
-----
1    Po1(U)      Eth       LACP      Eth1/3(P)  Eth1/4(P)

LACP KeepAlive Timer:
-----
Channel  PeerKeepAliveTimerFast
-----
1    Po1(U)      False

Cluster LACP Status:
-----
Channel  ClusterSpanned  ClusterDetach  ClusterUnitID  ClusterSysID
-----
1    Po1(U)      False          False           0              clust
```

To access the FXOS on ASA, run the **connect fxos admin** command. In the case of multi-context, run this command in the admin context.

Collect capture files

Follow the steps in the section **Collect Secure Firewall 3100 Internal Switch Capture Files**.

Capture file analysis

Use a packet capture file reader application to open the capture files for Ethernet1/1.205. Select the first packet and check the key points:

1. Only ICMP echo request packets are captured.
2. The original packet header has VLAN tag **205**.

No.	Time	Source	Destination	Protocol	Length	IP ID	IP TTL	Info
1	2022-08-07 21:21:01.607187	192.0.2.100	198.51.100.100	ICMP	106	0x411f (16671)	64	Echo (ping) request id=0x0037, seq=1/256, ttl=64 (no res
2	2022-08-07 21:21:02.609418	192.0.2.100	198.51.100.100	ICMP	106	0x413a (16698)	64	Echo (ping) request id=0x0037, seq=2/512, ttl=64 (no res
3	2022-08-07 21:21:03.610671	192.0.2.100	198.51.100.100	ICMP	106	0x421a (16922)	64	Echo (ping) request id=0x0037, seq=3/768, ttl=64 (no res
4	2022-08-07 21:21:04.609160	192.0.2.100	198.51.100.100	ICMP	106	0x426c (17004)	64	Echo (ping) request id=0x0037, seq=4/1024, ttl=64 (no res
5	2022-08-07 21:21:05.609409	192.0.2.100	198.51.100.100	ICMP	106	0x4310 (17168)	64	Echo (ping) request id=0x0037, seq=5/1280, ttl=64 (no res
6	2022-08-07 21:21:06.611847	192.0.2.100	198.51.100.100	ICMP	106	0x43df (17375)	64	Echo (ping) request id=0x0037, seq=6/1536, ttl=64 (no res
7	2022-08-07 21:21:07.616688	192.0.2.100	198.51.100.100	ICMP	106	0x44d3 (17619)	64	Echo (ping) request id=0x0037, seq=7/1792, ttl=64 (no res
8	2022-08-07 21:21:08.618023	192.0.2.100	198.51.100.100	ICMP	106	0x4518 (17688)	64	Echo (ping) request id=0x0037, seq=8/2048, ttl=64 (no res
9	2022-08-07 21:21:09.619326	192.0.2.100	198.51.100.100	ICMP	106	0x453d (17725)	64	Echo (ping) request id=0x0037, seq=9/2304, ttl=64 (no res
10	2022-08-07 21:21:10.616696	192.0.2.100	198.51.100.100	ICMP	106	0x462b (17963)	64	Echo (ping) request id=0x0037, seq=10/2560, ttl=64 (no res
11	2022-08-07 21:21:11.621629	192.0.2.100	198.51.100.100	ICMP	106	0x4707 (18183)	64	Echo (ping) request id=0x0037, seq=11/2816, ttl=64 (no res
12	2022-08-07 21:21:12.619309	192.0.2.100	198.51.100.100	ICMP	106	0x474b (18251)	64	Echo (ping) request id=0x0037, seq=12/3072, ttl=64 (no res
13	2022-08-07 21:21:13.620168	192.0.2.100	198.51.100.100	ICMP	106	0x4781 (18305)	64	Echo (ping) request id=0x0037, seq=13/3328, ttl=64 (no res
14	2022-08-07 21:21:14.623169	192.0.2.100	198.51.100.100	ICMP	106	0x4858 (18520)	64	Echo (ping) request id=0x0037, seq=14/3584, ttl=64 (no res
15	2022-08-07 21:21:15.622497	192.0.2.100	198.51.100.100	ICMP	106	0x4909 (18697)	64	Echo (ping) request id=0x0037, seq=15/3840, ttl=64 (no res
16	2022-08-07 21:21:16.626226	192.0.2.100	198.51.100.100	ICMP	106	0x490b (18699)	64	Echo (ping) request id=0x0037, seq=16/4096, ttl=64 (no res
17	2022-08-07 21:21:17.629363	192.0.2.100	198.51.100.100	ICMP	106	0x4932 (18738)	64	Echo (ping) request id=0x0037, seq=17/4352, ttl=64 (no res
18	2022-08-07 21:21:18.626651	192.0.2.100	198.51.100.100	ICMP	106	0x4a05 (18949)	64	Echo (ping) request id=0x0037, seq=18/4608, ttl=64 (no res

Frame 1: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) Ethernet II, Src: VMware 9d:e8:be (00:50:56:9d:e8:be), Dst: Cisco_34:9a:14 (bc:e7:12:34:9a:14) 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 205 000. = Priority: Best Effort (default) (0) ...0 = DEI: Ineligible ... 0000 1100 1101 = ID: 205 Type: IPv4 (0x0800) Trailer: 55555555 Internet Protocol Version 4, Src: 192.0.2.100, Dst: 198.51.100.100 Internet Control Message Protocol		0000 bc e7 12 34 9a 14 00 50 56 9d e8 be 81 00 00 cd ...4...P V..... 0010 08 00 45 00 00 54 41 1f 40 00 40 01 0c 8e c0 00 ..E..TA. @ @..... 0020 02 64 c6 33 64 64 08 00 06 67 00 37 01 b0 2c ..d:3dd...g:7...., 0030 f0 62 00 00 00 00 8e fe 03 00 00 00 00 10 11 ..b..... 0040 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 ..b..... 0050 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 "##%&'()*+,-./01 0060 32 33 34 35 36 37 55 55 55 55 234567UU UU
---	--	--

Open the capture files for Portchannel1 member interfaces. Select the first packet and check the key points:

1. Only ICMP echo request packets are captured.
2. The original packet header has VLAN tag 205.

No.	Time	Source	Destination	Protocol	Length	IP ID	IP TTL	Info
1	2022-08-07 21:21:01.607187	192.0.2.100	198.51.100.100	ICMP	106	0x411f (16671)	64	Echo (ping) request id=0x0037, seq=1/256, ttl=64 (no res
2	2022-08-07 21:21:02.609418	192.0.2.100	198.51.100.100	ICMP	106	0x413a (16698)	64	Echo (ping) request id=0x0037, seq=2/512, ttl=64 (no res
3	2022-08-07 21:21:03.610671	192.0.2.100	198.51.100.100	ICMP	106	0x421a (16922)	64	Echo (ping) request id=0x0037, seq=3/768, ttl=64 (no res
4	2022-08-07 21:21:04.609160	192.0.2.100	198.51.100.100	ICMP	106	0x426c (17004)	64	Echo (ping) request id=0x0037, seq=4/1024, ttl=64 (no res
5	2022-08-07 21:21:05.609409	192.0.2.100	198.51.100.100	ICMP	106	0x4310 (17168)	64	Echo (ping) request id=0x0037, seq=5/1280, ttl=64 (no res
6	2022-08-07 21:21:06.611847	192.0.2.100	198.51.100.100	ICMP	106	0x43df (17375)	64	Echo (ping) request id=0x0037, seq=6/1536, ttl=64 (no res
7	2022-08-07 21:21:07.616688	192.0.2.100	198.51.100.100	ICMP	106	0x44d3 (17619)	64	Echo (ping) request id=0x0037, seq=7/1792, ttl=64 (no res
8	2022-08-07 21:21:08.618023	192.0.2.100	198.51.100.100	ICMP	106	0x4518 (17688)	64	Echo (ping) request id=0x0037, seq=8/2048, ttl=64 (no res
9	2022-08-07 21:21:09.619326	192.0.2.100	198.51.100.100	ICMP	106	0x453d (17725)	64	Echo (ping) request id=0x0037, seq=9/2304, ttl=64 (no res
10	2022-08-07 21:21:10.616696	192.0.2.100	198.51.100.100	ICMP	106	0x462b (17963)	64	Echo (ping) request id=0x0037, seq=10/2560, ttl=64 (no res
11	2022-08-07 21:21:11.621629	192.0.2.100	198.51.100.100	ICMP	106	0x4707 (18183)	64	Echo (ping) request id=0x0037, seq=11/2816, ttl=64 (no res
12	2022-08-07 21:21:12.619309	192.0.2.100	198.51.100.100	ICMP	106	0x474b (18251)	64	Echo (ping) request id=0x0037, seq=12/3072, ttl=64 (no res
13	2022-08-07 21:21:13.620168	192.0.2.100	198.51.100.100	ICMP	106	0x4781 (18305)	64	Echo (ping) request id=0x0037, seq=13/3328, ttl=64 (no res
14	2022-08-07 21:21:14.623169	192.0.2.100	198.51.100.100	ICMP	106	0x4858 (18520)	64	Echo (ping) request id=0x0037, seq=14/3584, ttl=64 (no res
15	2022-08-07 21:21:15.622497	192.0.2.100	198.51.100.100	ICMP	106	0x4909 (18697)	64	Echo (ping) request id=0x0037, seq=15/3840, ttl=64 (no res
16	2022-08-07 21:21:16.626226	192.0.2.100	198.51.100.100	ICMP	106	0x490b (18699)	64	Echo (ping) request id=0x0037, seq=16/4096, ttl=64 (no res
17	2022-08-07 21:21:17.629363	192.0.2.100	198.51.100.100	ICMP	106	0x4932 (18738)	64	Echo (ping) request id=0x0037, seq=17/4352, ttl=64 (no res
18	2022-08-07 21:21:18.626651	192.0.2.100	198.51.100.100	ICMP	106	0x4a05 (18949)	64	Echo (ping) request id=0x0037, seq=18/4608, ttl=64 (no res

Frame 1: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) Ethernet II, Src: VMware 9d:e8:be (00:50:56:9d:e8:be), Dst: Cisco_34:9a:14 (bc:e7:12:34:9a:14) 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 205 000. = Priority: Best Effort (default) (0) ...0 = DEI: Ineligible ... 0000 1100 1101 = ID: 205 Type: IPv4 (0x0800) Trailer: 55555555 Internet Protocol Version 4, Src: 192.0.2.100, Dst: 198.51.100.100 Internet Control Message Protocol		0000 bc e7 12 34 9a 14 00 50 56 9d e8 be 81 00 00 cd ...4...P V..... 0010 08 00 45 00 00 54 41 1f 40 00 40 01 0c 8e c0 00 ..E..TA. @ @..... 0020 02 64 c6 33 64 64 08 00 06 67 00 37 01 b0 2c ..d:3dd...g:7...., 0030 f0 62 00 00 00 00 8e fe 03 00 00 00 00 10 11 ..b..... 0040 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 ..b..... 0050 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 "##%&'()*+,-./01 0060 32 33 34 35 36 37 55 55 55 55 234567UU UU
---	--	--

Explanation

The switch captures are configured on subinterfaces Ethernet1/1.205 or Portchannel1.205 with a filter that matches outer VLAN 205.

This table summarizes the task:

Task	Capture point	Internal filter	Direction	Captured traffic
Configure and verify a packet capture on subinterface Ethernet1/1.205	Ethernet 1/1	Outer VLAN 205 only	Ingress	ICMP echo requests from host 192.0.2.100 to host 198.51.100.1
Configure and verify a packet capture on subinterface Portchannel1.205 with member interfaces Ethernet1/3 and Ethernet1/4	Ethernet 1/3 Ethernet 1/4	Outer VLAN 205 only	Ingress	ICMP echo requests from host 192.0.2.100 to host 198.51.100.1

Packet Capture on Internal Interfaces

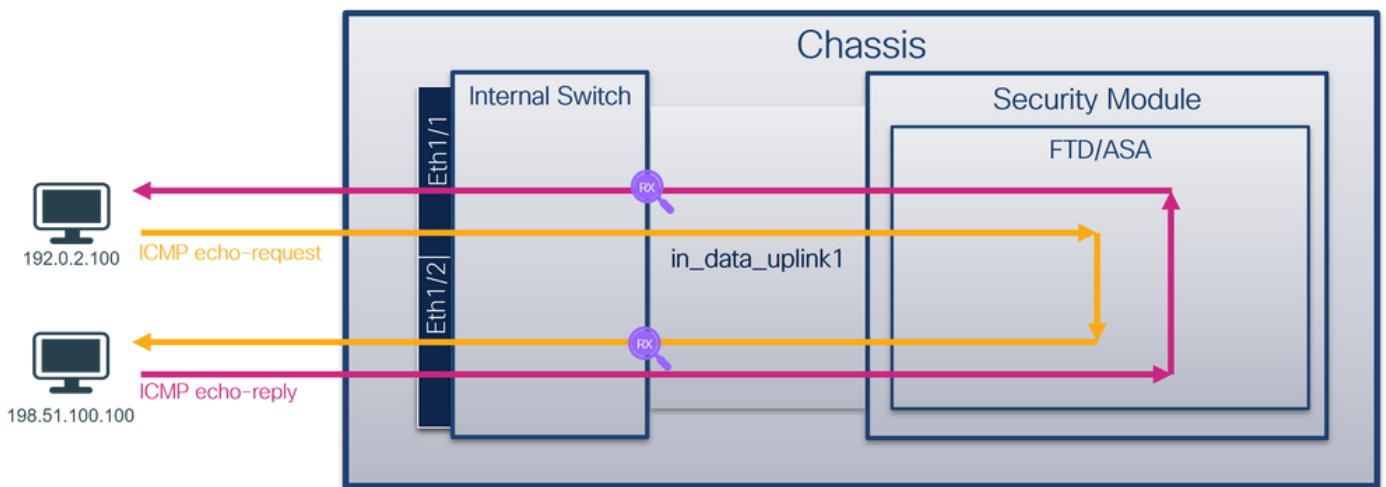
The Secure Firewall has 2 internal interfaces:

- **in_data_uplink1** - connects the application to the internal switch.
- **in_mgmt_uplink1** - provides a dedicated packet path for management connections, such as SSH to the management interface, or the management connection, also known as the sftunnel, between the FMC and the FTD.

Task 1

Use the FTD or ASA CLI to configure and verify a packet capture on the uplink interface **in_data_uplink1**.

Topology, packet flow, and the capture points



Configuration

Follow these steps on ASA or FTD CLI to configure a packet capture on interface **in_data_uplink1**:

1. Create a capture session:

```
> capture capsw switch interface in_data_uplink1
```

2. Enable the capture session:

```
> no capture capsw switch stop
```

Verification

Verify the capture session name, administrative and operational state, interface slot, and identifier. Ensure the **Pcapsize** value in bytes increases and the number of captured packets is non-zero:

```
> show capture capsw detail
Packet Capture info
  Name:          capsw
  Session:       1
  Admin State:   enabled
  Oper State:    up
```

Oper State Reason: Active

Config Success: yes
Config Fail Reason:
Append Flag: overwrite
Session Mem Usage: 256
Session Pcap Snap Len: 1518
Error Code: 0
Drop Count: 0

Total Physical ports involved in Packet Capture: 1

Physical port:

Slot Id: 1
Port Id: 18
Pcapfile: /mnt/disk0/packet-capture/sess-1-capsw-data-uplink1.pcap
Pcapsize: 7704
Filter: capsw-1-18

Packet Capture Filter Info

Name: capsw-1-18
Protocol: 0
Ivlan: 0
Ovlan: 0
Src Ip: 0.0.0.0
Dest Ip: 0.0.0.0
Src Ipv6: ::
Dest Ipv6: ::
Src MAC: 00:00:00:00:00:00
Dest MAC: 00:00:00:00:00:00
Src Port: 0
Dest Port: 0
Ethertype: 0

Total Physical breakout ports involved in Packet Capture: 0

66 packets captured on disk using switch capture

Reading of capture file from disk is not supported

In this case, a capture is created on the interface with an internal ID **18** which is the `in_data_uplink1` interface on the Secure Firewall 3130. The **show portmanager switch status** command in the FXOS `local-mgmt` command shell shows the interface IDs:

> **connect fxos**

...

KSEC-FPR3100-1 **connect local-mgmt**

KSEC-FPR3100-1(local-mgmt) **show portmanager switch status**

Dev/Port	Mode	Link	Speed	Duplex	Loopback Mode	Port Manager
0/1	SGMII	Up	1G	Full	None	Link-Up
0/2	SGMII	Up	1G	Full	None	Link-Up
0/3	SGMII	Up	1G	Full	None	Link-Up
0/4	SGMII	Up	1G	Full	None	Link-Up
0/5	SGMII	Down	1G	Half	None	Mac-Link-Down
0/6	SGMII	Down	1G	Half	None	Mac-Link-Down
0/7	SGMII	Down	1G	Half	None	Mac-Link-Down
0/8	SGMII	Down	1G	Half	None	Mac-Link-Down
0/9	1000_BaseX	Down	1G	Full	None	Link-Down
0/10	1000_BaseX	Down	1G	Full	None	Link-Down
0/11	1000_BaseX	Down	1G	Full	None	Link-Down
0/12	1000_BaseX	Down	1G	Full	None	Link-Down
0/13	1000_BaseX	Down	1G	Full	None	Link-Down
0/14	1000_BaseX	Down	1G	Full	None	Link-Down

0/15	1000_BaseX	Down	1G	Full	None	Link-Down
0/16	1000_BaseX	Down	1G	Full	None	Link-Down
0/17	1000_BaseX	Up	1G	Full	None	Link-Up
0/18	KR2	Up	50G	Full	None	Link-Up
0/19	KR	Up	25G	Full	None	Link-Up
0/20	KR	Up	25G	Full	None	Link-Up
0/21	KR4	Down	40G	Full	None	Link-Down
0/22	n/a	Down	n/a	Full	N/A	Reset
0/23	n/a	Down	n/a	Full	N/A	Reset
0/24	n/a	Down	n/a	Full	N/A	Reset
0/25	1000_BaseX	Down	1G	Full	None	Link-Down
0/26	n/a	Down	n/a	Full	N/A	Reset
0/27	n/a	Down	n/a	Full	N/A	Reset
0/28	n/a	Down	n/a	Full	N/A	Reset
0/29	1000_BaseX	Down	1G	Full	None	Link-Down
0/30	n/a	Down	n/a	Full	N/A	Reset
0/31	n/a	Down	n/a	Full	N/A	Reset
0/32	n/a	Down	n/a	Full	N/A	Reset
0/33	1000_BaseX	Down	1G	Full	None	Link-Down
0/34	n/a	Down	n/a	Full	N/A	Reset
0/35	n/a	Down	n/a	Full	N/A	Reset
0/36	n/a	Down	n/a	Full	N/A	Reset

To access the FXOS on ASA, run the **connect fxos admin** command. In the case of multi-context, run this command in the admin context.

Collect capture files

Follow the steps in the section **Collect Secure Firewall 3100 Internal Switch Capture Files**.

Capture file analysis

Use a packet capture file reader application to open the capture files for interface in_data_uplink1. Check the key point - in this case, ICMP echo request and echo reply packets are captured. These are the packets sent from the application to the internal switch.

No.	Time	Source	Destination	Protocol	Length	IP ID	IP TTL	Info
1	2022-08-07 22:40:06.685606	192.0.2.100	198.51.100.100	ICMP	102	0x4d93 (19859)	64	Echo (ping) request id=0x003a, seq=33/8448, ttl=64 (repl
2	2022-08-07 22:40:06.685615	198.51.100.100	192.0.2.100	ICMP	102	0x6cdc (27868)	64	Echo (ping) reply id=0x003a, seq=33/8448, ttl=64 (requ
3	2022-08-07 22:40:07.684219	192.0.2.100	198.51.100.100	ICMP	102	0x40e8 (19944)	64	Echo (ping) request id=0x003a, seq=34/8704, ttl=64 (repl
4	2022-08-07 22:40:07.689300	198.51.100.100	192.0.2.100	ICMP	102	0x6db2 (28082)	64	Echo (ping) reply id=0x003a, seq=34/8704, ttl=64 (requ
5	2022-08-07 22:40:08.685736	192.0.2.100	198.51.100.100	ICMP	102	0x4edc (20188)	64	Echo (ping) request id=0x003a, seq=35/8960, ttl=64 (repl
6	2022-08-07 22:40:08.690806	198.51.100.100	192.0.2.100	ICMP	102	0x6dbf (28095)	64	Echo (ping) reply id=0x003a, seq=35/8960, ttl=64 (requ
7	2022-08-07 22:40:09.690737	192.0.2.100	198.51.100.100	ICMP	102	0x4f2d (20269)	64	Echo (ping) request id=0x003a, seq=36/9216, ttl=64 (repl
8	2022-08-07 22:40:09.690744	198.51.100.100	192.0.2.100	ICMP	102	0x6e80 (28288)	64	Echo (ping) reply id=0x003a, seq=36/9216, ttl=64 (requ
9	2022-08-07 22:40:10.692266	192.0.2.100	198.51.100.100	ICMP	102	0x4fb1 (20401)	64	Echo (ping) request id=0x003a, seq=37/9472, ttl=64 (repl
10	2022-08-07 22:40:10.692272	198.51.100.100	192.0.2.100	ICMP	102	0x6ed5 (28373)	64	Echo (ping) reply id=0x003a, seq=37/9472, ttl=64 (requ
11	2022-08-07 22:40:11.691159	192.0.2.100	198.51.100.100	ICMP	102	0x5008 (20488)	64	Echo (ping) request id=0x003a, seq=38/9728, ttl=64 (repl
12	2022-08-07 22:40:11.691166	198.51.100.100	192.0.2.100	ICMP	102	0x6f3b (28475)	64	Echo (ping) reply id=0x003a, seq=38/9728, ttl=64 (requ
13	2022-08-07 22:40:12.692135	192.0.2.100	198.51.100.100	ICMP	102	0x50b8 (20664)	64	Echo (ping) request id=0x003a, seq=39/9984, ttl=64 (repl
14	2022-08-07 22:40:12.697209	198.51.100.100	192.0.2.100	ICMP	102	0x6fd7 (28631)	64	Echo (ping) reply id=0x003a, seq=39/9984, ttl=64 (requ
15	2022-08-07 22:40:13.697320	192.0.2.100	198.51.100.100	ICMP	102	0x5184 (20868)	64	Echo (ping) request id=0x003a, seq=40/10240, ttl=64 (rep
16	2022-08-07 22:40:13.697327	198.51.100.100	192.0.2.100	ICMP	102	0x703e (28734)	64	Echo (ping) reply id=0x003a, seq=40/10240, ttl=64 (rec
17	2022-08-07 22:40:14.698512	192.0.2.100	198.51.100.100	ICMP	102	0x51d8 (20952)	64	Echo (ping) request id=0x003a, seq=41/10496, ttl=64 (rep
18	2022-08-07 22:40:14.698518	198.51.100.100	192.0.2.100	ICMP	102	0x70dd (28893)	64	Echo (ping) reply id=0x003a, seq=41/10496, ttl=64 (rec

> Frame 1: 102 bytes on wire (816 bits), 102 bytes captured (816 bits)
 > Ethernet II, Src: Cisco_34:9a:15 (bc:e7:12:34:9a:15), Dst: VMware_9d:e7:50 (00:50:56:9d:e7:50)
 > Internet Protocol Version 4, Src: 192.0.2.100, Dst: 198.51.100.100
 > Internet Control Message Protocol

```

0000  00 50 56 9d e7 50 bc e7 12 34 9a 15 08 00 45 00  PV..P..4...E
0010  00 54 4d 93 40 00 40 01 00 1a c0 00 02 64 c6 33  TH.@.@....d.3
0020  64 64 08 00 7f 15 00 00 00 21 39 3f f0 62 00 00  dd....197.b...
0030  00 00 8b 1a 05 00 00 00 00 00 10 11 12 13 14 15  ....
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25  ....!%$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35  &'()*+,-./012345
0060  36 37 55 55 55 55 67UUUU

```

Explanation

When a switch capture on the uplink interface is configured, only packets sent from the application to the internal switch are captured. Packets sent to the application are not captured.

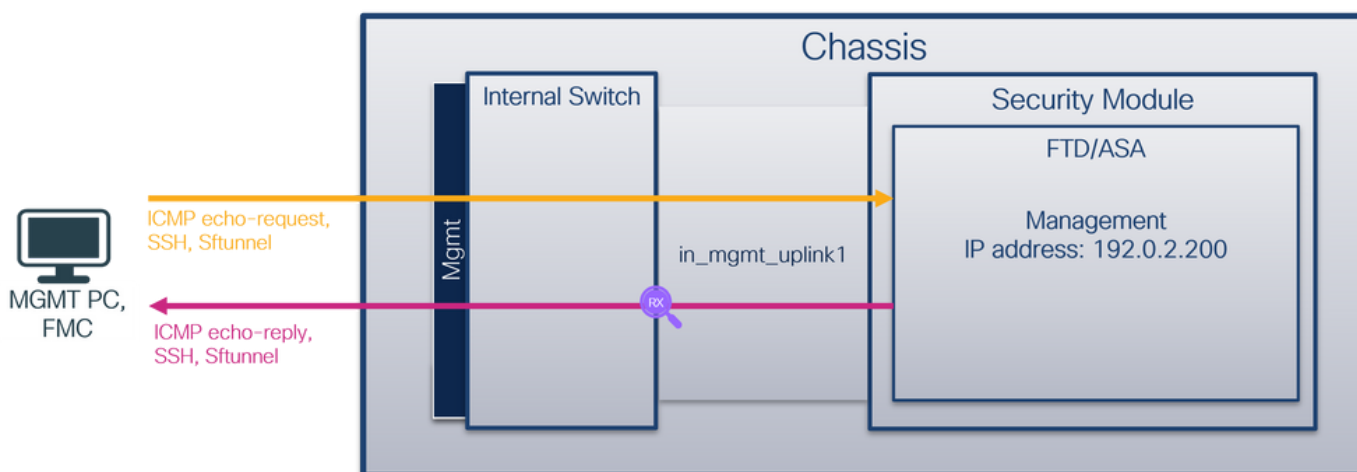
This table summarizes the task:

Task	Capture point	Internal filter	Direction	Captured traffic
Configure and verify a packet capture on the uplink interface <code>in_data_uplink1</code>	<code>in_data_uplink1</code>	None	Ingress only	ICMP echo requests from host 192.0.2.100 to host 198.51.100.100 ICMP echo replies from host 198.51.100.100 to host 192.0.2.100

Task 2

Use the FTD or ASA CLI to configure and verify a packet capture on the uplink interface `in_mgmt_uplink1`. Only the packets of management plane connections are captured.

Topology, packet flow, and the capture points



Configuration

Follow these steps on ASA or FTD CLI to configure a packet capture on interface `in_mgmt_uplink1`:

1. Create a capture session:

```
> capture capsw switch interface in_mgmt_uplink1
```

2. Enable the capture session:

```
> no capture capsw switch stop
```

Verification

Verify the capture session name, administrative and operational state, interface slot, and identifier. Ensure the **Pcapsize** value in bytes increases and the number of captured packets is non-zero:

```
> show capture capsw detail
```

Packet Capture info

```
Name:          capsw
Session:       1
Admin State:   enabled
Oper State:    up
Oper State Reason: Active
Config Success:  yes
Config Fail Reason:
```

Append Flag: overwrite
Session Mem Usage: 256
Session Pcap Snap Len: 1518
Error Code: 0
Drop Count: 0

Total Physical ports involved in Packet Capture: 1

Physical port:

Slot Id: 1
Port Id: 19
Pcapfile: /mnt/disk0/packet-capture/sess-1-capsw-mgmt-uplink1.pcap
Pcapsize: 137248
Filter: capsw-1-19

Packet Capture Filter Info

Name: capsw-1-19
Protocol: 0
Ivlan: 0
Ovlan: 0
Src Ip: 0.0.0.0
Dest Ip: 0.0.0.0
Src Ipv6: ::
Dest Ipv6: ::
Src MAC: 00:00:00:00:00:00
Dest MAC: 00:00:00:00:00:00
Src Port: 0
Dest Port: 0
Ethertype: 0

Total Physical breakout ports involved in Packet Capture: 0

281 packets captured on disk using switch capture

Reading of capture file from disk is not supported

In this case, a capture is created on the interface with an internal ID 19 which is the **in_mgmt_uplink1** interface on the Secure Firewall 3130. The **show portmanager switch status** command in the FXOS **local-mgmt** command shell shows the interface IDs:

> **connect fxos**

...

KSEC-FPR3100-1 **connect local-mgmt**

KSEC-FPR3100-1(local-mgmt) **show portmanager switch status**

Dev/Port	Mode	Link	Speed	Duplex	Loopback Mode	Port Manager
0/1	SGMII	Up	1G	Full	None	Link-Up
0/2	SGMII	Up	1G	Full	None	Link-Up
0/3	SGMII	Up	1G	Full	None	Link-Up
0/4	SGMII	Up	1G	Full	None	Link-Up
0/5	SGMII	Down	1G	Half	None	Mac-Link-Down
0/6	SGMII	Down	1G	Half	None	Mac-Link-Down
0/7	SGMII	Down	1G	Half	None	Mac-Link-Down
0/8	SGMII	Down	1G	Half	None	Mac-Link-Down
0/9	1000_BaseX	Down	1G	Full	None	Link-Down
0/10	1000_BaseX	Down	1G	Full	None	Link-Down
0/11	1000_BaseX	Down	1G	Full	None	Link-Down
0/12	1000_BaseX	Down	1G	Full	None	Link-Down
0/13	1000_BaseX	Down	1G	Full	None	Link-Down
0/14	1000_BaseX	Down	1G	Full	None	Link-Down
0/15	1000_BaseX	Down	1G	Full	None	Link-Down
0/16	1000_BaseX	Down	1G	Full	None	Link-Down
0/17	1000_BaseX	Up	1G	Full	None	Link-Up

0/18	KR2	Up	50G	Full	None	Link-Up
0/19	KR	Up	25G	Full	None	Link-Up
0/20	KR	Up	25G	Full	None	Link-Up
0/21	KR4	Down	40G	Full	None	Link-Down
0/22	n/a	Down	n/a	Full	N/A	Reset
0/23	n/a	Down	n/a	Full	N/A	Reset
0/24	n/a	Down	n/a	Full	N/A	Reset
0/25	1000_BaseX	Down	1G	Full	None	Link-Down
0/26	n/a	Down	n/a	Full	N/A	Reset
0/27	n/a	Down	n/a	Full	N/A	Reset
0/28	n/a	Down	n/a	Full	N/A	Reset
0/29	1000_BaseX	Down	1G	Full	None	Link-Down
0/30	n/a	Down	n/a	Full	N/A	Reset
0/31	n/a	Down	n/a	Full	N/A	Reset
0/32	n/a	Down	n/a	Full	N/A	Reset
0/33	1000_BaseX	Down	1G	Full	None	Link-Down
0/34	n/a	Down	n/a	Full	N/A	Reset
0/35	n/a	Down	n/a	Full	N/A	Reset
0/36	n/a	Down	n/a	Full	N/A	Reset

To access the FXOS on ASA, run the **connect fxos admin** command. In the case of multi-context, run this command in the admin context.

Collect capture files

Follow the steps in the section **Collect Secure Firewall 3100 Internal Switch Capture Files**.

Capture file analysis

Use a packet capture file reader application to open the capture files for interface **in_mgmt_uplink1**. Check the key point - in this case only the packets from the management IP address 192.0.2.200 are shown. Examples are SSH, Sftunnel or ICMP echo reply packets. These are the packets sent from the application management interface to the network through the internal switch.

The screenshot displays a network traffic capture analysis interface. The top section is a table of captured packets with columns for No., Time, Source, Destination, Protocol, Length, IP ID, and IP TTL. A red box highlights the Source and Destination columns, showing traffic from 192.0.2.200 to 192.0.2.101. The bottom section shows a detailed view of a selected packet (Frame 1: 747 bytes on wire), including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol (TCP) details.

No.	Time	Source	Destination	Protocol	Length	IP ID	IP TTL	Info
196	2022-08-07 23:21:45.133362	192.0.2.200	192.0.2.101	TCP	1518	0xb7d0 (47056)	64	39181 → 8305 [ACK] Seq=61372 Ack=875 Win=1384 Len=1448 TS
197	2022-08-07 23:21:45.133385	192.0.2.200	192.0.2.101	TCP	1518	0xb7d1 (47057)	64	39181 → 8305 [ACK] Seq=62820 Ack=875 Win=1384 Len=1448 TS
198	2022-08-07 23:21:45.133388	192.0.2.200	192.0.2.101	TLSv1.2	990	0xb7d2 (47058)	64	Application Data
199	2022-08-07 23:21:45.928772	192.0.2.200	192.0.2.100	ICMP	78	0xbd48 (48456)	64	Echo (ping) reply id=0x0001, seq=4539/47889, ttl=64
200	2022-08-07 23:21:45.949024	192.0.2.200	192.0.2.101	TLSv1.2	128	0x4a97 (19095)	64	Application Data
201	2022-08-07 23:21:45.949027	192.0.2.200	192.0.2.101	TCP	70	0x4a98 (19096)	64	8305 → 58885 [ACK] Seq=21997 Ack=26244 Win=4116 Len=0 TSv
202	2022-08-07 23:21:46.019895	192.0.2.200	192.0.2.101	TLSv1.2	100	0x4a99 (19097)	64	Application Data
203	2022-08-07 23:21:46.019899	192.0.2.200	192.0.2.101	TLSv1.2	96	0x4a9a (19098)	64	Application Data
204	2022-08-07 23:21:46.019903	192.0.2.200	192.0.2.101	TCP	70	0x4a9b (19099)	64	8305 → 58885 [ACK] Seq=22053 Ack=26274 Win=4116 Len=0 TSv
205	2022-08-07 23:21:46.019906	192.0.2.200	192.0.2.101	TCP	70	0x4a9c (19100)	64	8305 → 58885 [ACK] Seq=22053 Ack=26300 Win=4116 Len=0 TSv
206	2022-08-07 23:21:46.136415	192.0.2.200	192.0.2.101	TCP	70	0xb7d3 (47059)	64	39181 → 8305 [ACK] Seq=65188 Ack=921 Win=1384 Len=0 TSval
207	2022-08-07 23:21:46.958148	192.0.2.200	192.0.2.100	ICMP	78	0xbd9e (48542)	64	Echo (ping) reply id=0x0001, seq=4540/48145, ttl=64
208	2022-08-07 23:21:47.980409	192.0.2.200	192.0.2.100	ICMP	78	0xbdf2 (48626)	64	Echo (ping) reply id=0x0001, seq=4541/48401, ttl=64
209	2022-08-07 23:21:48.406312	192.0.2.200	192.0.2.101	TCP	70	0x4a9d (19101)	64	8305 → 58885 [ACK] Seq=22053 Ack=26366 Win=4116 Len=0 TSv
210	2022-08-07 23:21:48.903236	192.0.2.200	192.0.2.101	TLSv1.2	747	0x4a9e (19102)	64	Application Data
211	2022-08-07 23:21:48.994386	192.0.2.200	192.0.2.100	ICMP	78	0xbe48 (48712)	64	Echo (ping) reply id=0x0001, seq=4542/48657, ttl=64
212	2022-08-07 23:21:50.008576	192.0.2.200	192.0.2.100	ICMP	78	0xbe4e (48806)	64	Echo (ping) reply id=0x0001, seq=4543/48913, ttl=64
213	2022-08-07 23:21:50.140167	192.0.2.200	192.0.2.101	TCP	1518	0xb7d4 (47060)	64	39181 → 8305 [ACK] Seq=65188 Ack=921 Win=1384 Len=1448 TS
214	2022-08-07 23:21:50.140171	192.0.2.200	192.0.2.101	TCP	1518	0xb7d5 (47061)	64	39181 → 8305 [ACK] Seq=66636 Ack=921 Win=1384 Len=1448 TS
215	2022-08-07 23:21:50.140175	192.0.2.200	192.0.2.101	TLSv1.2	990	0xb7d6 (47062)	64	Application Data
216	2022-08-07 23:21:51.015884	192.0.2.200	192.0.2.100	ICMP	78	0xbec1 (48833)	64	Echo (ping) reply id=0x0001, seq=4544/49169, ttl=64
217	2022-08-07 23:21:51.142842	192.0.2.200	192.0.2.101	TCP	70	0xb7d7 (47063)	64	39181 → 8305 [ACK] Seq=69004 Ack=967 Win=1384 Len=0 TSval
218	2022-08-07 23:21:52.030118	192.0.2.200	192.0.2.100	ICMP	78	0xbf02 (48898)	64	Echo (ping) reply id=0x0001, seq=4545/49425, ttl=64
219	2022-08-07 23:21:53.042744	192.0.2.200	192.0.2.100	ICMP	78	0xbf59 (48985)	64	Echo (ping) reply id=0x0001, seq=4546/49681, ttl=64
220	2022-08-07 23:21:53.073144	192.0.2.200	192.0.2.100	SSH	170	0xad34 (44340)	64	Server: Encrypted packet (len=112)
221	2022-08-07 23:21:53.194906	192.0.2.200	192.0.2.100	TCP	64	0xad35 (44341)	64	22 → 53249 [ACK] Seq=1025 Ack=881 Win=946 Len=0
222	2022-08-07 23:21:53.905480	192.0.2.200	192.0.2.101	TLSv1.2	747	0x4a9f (19103)	64	Application Data
223	2022-08-07 23:21:54.102899	192.0.2.200	192.0.2.100	ICMP	78	0xbf63 (48995)	64	Echo (ping) reply id=0x0001, seq=4547/49937, ttl=64
224	2022-08-07 23:21:54.903675	192.0.2.200	192.0.2.101	TCP	70	0x4aa0 (19104)	64	8305 → 58885 [ACK] Seq=23407 Ack=26424 Win=4116 Len=0 TSv
225	2022-08-07 23:21:55.136700	192.0.2.200	192.0.2.100	TCP	70	0xbf64 (49000)	64	Echo (ping) reply id=0x0001, seq=4548/50102, ttl=64

Frame 1: 747 bytes on wire (5976 bits), 747 bytes captured (5976 bits)
 > Ethernet II, Src: Cisco_34:9a:00 (bc:e7:12:34:9a:00), Dst: Cisco_11:38:2a (a4:53:0e:11:38:2a)
 > Internet Protocol Version 4, Src: 192.0.2.200, Dst: 192.0.2.101
 > Transmission Control Protocol, Src Port: 8305, Dst Port: 58885, Seq: 1, Ack: 1, Len: 677
 > Transport Layer Security

Explanation

When a switch capture on the management uplink interface is configured, only ingress packets sent from the application management interface are captured. Packets destined for the application management interface are not captured.

This table summarizes the task:

Task	Capture point	Internal filter	Direction	Captured traffic
Configure and verify a packet capture on the management uplink interface	in_mgmt_uplink1	None	Ingress only (from the management interface to the network through the internal switch)	ICMP echo replies from FTD management address 192.0.2.200 to host 192.0.2.100 Sftunnel from FTD management IP address 192.0.2.200 to FMC IP address 192.0.2.1 SSH from FTD management IP address 192.0.2.200 to host 192.0.2.100

Packet Capture Filters

Internal switch packet capture filters are configured the same way as the data plane captures. Use the **ethernet-type** and **match** options to configure filters.

Configuration

Follow these steps on ASA or FTD CLI to configure a packet capture with a filter that matches ARP frames or ICMP packets from host 198.51.100.100 on interface Ethernet1/1:

1. Verify the nameif:

```
> show nameif
Interface          Name          Security
Ethernet1/1       inside        0
Ethernet1/2       outside       0
Management1/1     diagnostic    0
```

2. Create a capture session for ARP or ICMP:

```
> capture capsw switch interface inside ethernet-type arp
> capture capsw switch interface inside match icmp 198.51.100.100
```

Verification

Verify the capture session name and the filter. The Ethertype value is **2054** in decimal and **0x0806** in hexadecimal:

```
> show capture capsw detail
Packet Capture info
Name:          capsw
Session:       1
Admin State:   disabled
Oper State:    down
Oper State Reason: Session_Admin_Shut
Config Success: yes
```

Config Fail Reason:
Append Flag: overwrite
Session Mem Usage: 256
Session Pcap Snap Len: 1518
Error Code: 0
Drop Count: 0

Total Physical ports involved in Packet Capture: 1

Physical port:

Slot Id: 1
Port Id: 1
Pcapfile: /mnt/disk0/packet-capture/sess-1-capsw-ethernet-1-1-0.pcap
Pcapsize: 0
Filter: caps-1-1

Packet Capture Filter Info

Name: caps-1-1
Protocol: 0
Ivlan: 0
Ovlan: 0
Src Ip: 0.0.0.0
Dest Ip: 0.0.0.0
Src Ipv6: ::
Dest Ipv6: ::
Src MAC: 00:00:00:00:00:00
Dest MAC: 00:00:00:00:00:00
Src Port: 0
Dest Port: 0
Ethertype: 2054

Total Physical breakout ports involved in Packet Capture: 0

0 packet captured on disk using switch capture

Reading of capture file from disk is not supported

This is the verification of the filter for ICMP. IP protocol 1 is the ICMP:

> **show capture caps detail**

Packet Capture info

Name: caps
Session: 1
Admin State: disabled
Oper State: down
Oper State Reason: Session_Admin_Shut
Config Success: yes
Config Fail Reason:
Append Flag: overwrite
Session Mem Usage: 256
Session Pcap Snap Len: 1518
Error Code: 0
Drop Count: 0

Total Physical ports involved in Packet Capture: 1

Physical port:

Slot Id: 1
Port Id: 1
Pcapfile: /mnt/disk0/packet-capture/sess-1-capsw-ethernet-1-1-0.pcap
Pcapsize: 0
Filter: caps-1-1

Packet Capture Filter Info

Name: caps-1-1
Protocol: 1
Ivlan: 0
Ovlan: 0
Src Ip: 198.51.100.100
Dest Ip: 0.0.0.0
Src Ipv6: ::
Dest Ipv6: ::
Src MAC: 00:00:00:00:00:00
Dest MAC: 00:00:00:00:00:00
Src Port: 0
Dest Port: 0
Ethertype: 0

Total Physical breakout ports involved in Packet Capture: 0

0 packets captured on disk using switch capture

Reading of capture file from disk is not supported

Collect Secure Firewall 3100 Internal Switch Capture Files

Use ASA or FTD CLI to collect internal switch capture files. On FTD, the capture file can also be exported via the CLI **copy** command to destinations reachable via the data or diagnostic interfaces.

Alternatively, the file can be copied to **/ngfw/var/common** in expert mode and downloaded from FMC via the **File Download** option.

In the case of port-channel interfaces ensure to collect packet capture files from all member interfaces.

ASA

Follow these steps on to collect internal switch capture files on ASA CLI:

1. Stop the capture:

```
asa# capture caps switch stop
```

2. Verify the capture session is stopped and note the capture file name.

```
asa# show capture caps detail
```

```
Packet Capture info  
Name: caps  
Session: 1  
Admin State: disabled  
Oper State: down  
Oper State Reason: Session_Admin_Shut  
Config Success: yes  
Config Fail Reason:  
Append Flag: overwrite  
Session Mem Usage: 256  
Session Pcap Snap Len: 1518  
Error Code: 0  
Drop Count: 0
```

Total Physical ports involved in Packet Capture: 1

Physical port:

Slot Id: 1
Port Id: 1
Pcapfile: /mnt/disk0/packet-capture/**sess-1-capsw-ethernet-1-1-0.pcap**
Pcapsize: 139826
Filter: capsw-1-1

Packet Capture Filter Info

Name: capsw-1-1
Protocol: 0
Ivlan: 0
Ovlan: 0
Src Ip: 0.0.0.0
Dest Ip: 0.0.0.0
Src Ipv6: ::
Dest Ipv6: ::
Src MAC: 00:00:00:00:00:00
Dest MAC: 00:00:00:00:00:00
Src Port: 0
Dest Port: 0
Ethertype: 0

Total Physical breakout ports involved in Packet Capture: 0

886 packets captured on disk using switch capture

Reading of capture file from disk is not supported

3. Use the CLI **copy** command to export the file to remote destinations:

```
asa# copy flash:/packet-capture/sess-1-capsw-ethernet-1-1-0.pcap ?
```

```
cluster: Copy to cluster: file system  
disk0: Copy to disk0: file system  
disk1: Copy to disk1: file system  
flash: Copy to flash: file system  
ftp: Copy to ftp: file system  
running-config Update (merge with) current system configuration  
scp: Copy to scp: file system  
smb: Copy to smb: file system  
startup-config Copy to startup configuration  
system: Copy to system: file system  
tftp: Copy to tftp: file system
```

```
asa# copy flash:/packet-capture/sess-1-capsw-ethernet-1-1-0.pcap tftp://198.51.100.10/
```

```
Source filename [/packet-capture/sess-1-capsw-ethernet-1-1-0.pcap]?
```

```
Destination filename [sess-1-capsw-ethernet-1-1-0.pcap]?
```

```
Copy in progress...C
```

```
139826 bytes copied in 0.532 secs
```

FTD

Follow these steps to collect internal switch capture files on FTD CLI and copy them to servers reachable via data or diagnostic interfaces:

1. Go to diagnostic CLI:

```
> system support diagnostic-cli
```

```
Attaching to Diagnostic CLI ... Click 'Ctrl+a then d' to detach.
```

```
Type help or '?' for a list of available commands.
```



```
firepower> enable
Password: <-- Enter
firepower#
```

2. Stop the capture:

```
firepower# capture capi switch stop
```

3. Verify the capture session is stopped and note the capture file name:

```
firepower# show capture capsw detail
```

Packet Capture info

```
Name:                capsw
Session:                1
Admin State:        disabled
Oper State:         down
Oper State Reason: Session_Admin_Shut
Config Success:        yes
Config Fail Reason:
Append Flag:           overwrite
Session Mem Usage:     256
Session Pcap Snap Len: 1518
Error Code:            0
Drop Count:            0
```

Total Physical ports involved in Packet Capture: 1

Physical port:

```
Slot Id:                1
Port Id:                1
Pcapfile:           /mnt/disk0/packet-capture/sess-1-capsw-ethernet-1-1-0.pcap
Pcapsize:              139826
Filter:                capsw-1-1
```

Packet Capture Filter Info

```
Name:                  capsw-1-1
Protocol:              0
Ivlan:                0
Ovlan:                0
Src Ip:                0.0.0.0
Dest Ip:               0.0.0.0
Src Ipv6:              ::
Dest Ipv6:             ::
Src MAC:               00:00:00:00:00:00
Dest MAC:              00:00:00:00:00:00
Src Port:              0
Dest Port:             0
Ethertype:            0
```

Total Physical breakout ports involved in Packet Capture: 0

886 packets captured on disk using switch capture

Reading of capture file from disk is not supported

4. Use the CLI **copy** command to export the file to remote destinations.

```
firepower# copy flash:/packet-capture/sess-1-capsw-ethernet-1-1-0.pcap ?
cluster:              Copy to cluster: file system
disk0:                Copy to disk0: file system
disk1:                Copy to disk1: file system
flash:                Copy to flash: file system
```

```
ftp:          Copy to ftp: file system
running-config Update (merge with) current system configuration
scp:         Copy to scp: file system
smb:        Copy to smb: file system
startup-config Copy to startup configuration
system:     Copy to system: file system
tftp:       Copy to tftp: file system
```

```
firepower# copy flash:/packet-capture/sess-1-capsw-ethernet-1-1-0.pcap tftp://198.51.100.10/
Source filename [/packet-capture/sess-1-capsw-ethernet-1-1-0.pcap]?
Destination filename [sess-1-capsw-ethernet-1-1-0.pcap]?
Copy in progress...C
139826 bytes copied in 0.532 secs
```

Follow these steps on to collect capture files from FMC via the **File Download** option:

1. Stop the capture:

```
> capture capsw switch stop
```

2. Verify the capture session is stopped and note the file name and full capture file path:

```
> show capture capsw detail
```

```
Packet Capture info
Name:          capsw
Session:       1
Admin State:   disabled
Oper State:    down
Oper State Reason: Session_Admin_Shut
Config Success: yes
Config Fail Reason:
Append Flag:   overwrite
Session Mem Usage: 256
Session Pcap Snap Len: 1518
Error Code:    0
Drop Count:    0
```

Total Physical ports involved in Packet Capture: 1

Physical port:

```
Slot Id:      1
Port Id:      1
Pcapfile:     /mnt/disk0/packet-capture/sess-1-capsw-ethernet-1-1-0.pcap
Pcapsize:    139826
Filter:       capsw-1-1
```

Packet Capture Filter Info

```
Name:         capsw-1-1
Protocol:     0
Ivlan:       0
Ovlan:       0
Src Ip:      0.0.0.0
Dest Ip:     0.0.0.0
Src Ipv6:    ::
Dest Ipv6:   ::
Src MAC:     00:00:00:00:00:00
Dest MAC:    00:00:00:00:00:00
Src Port:    0
Dest Port:   0
Ethertype:   0
```

Total Physical breakout ports involved in Packet Capture: 0

886 packets captured on disk using switch capture
Reading of capture file from disk is not supported

3. Go to expert mode and switch to root mode:

> **expert**

```
admin@firepower:~$ sudo su
```

```
root@firepower:/home/admin
```

4. Copy the capture file to **/ngfw/var/common/**:

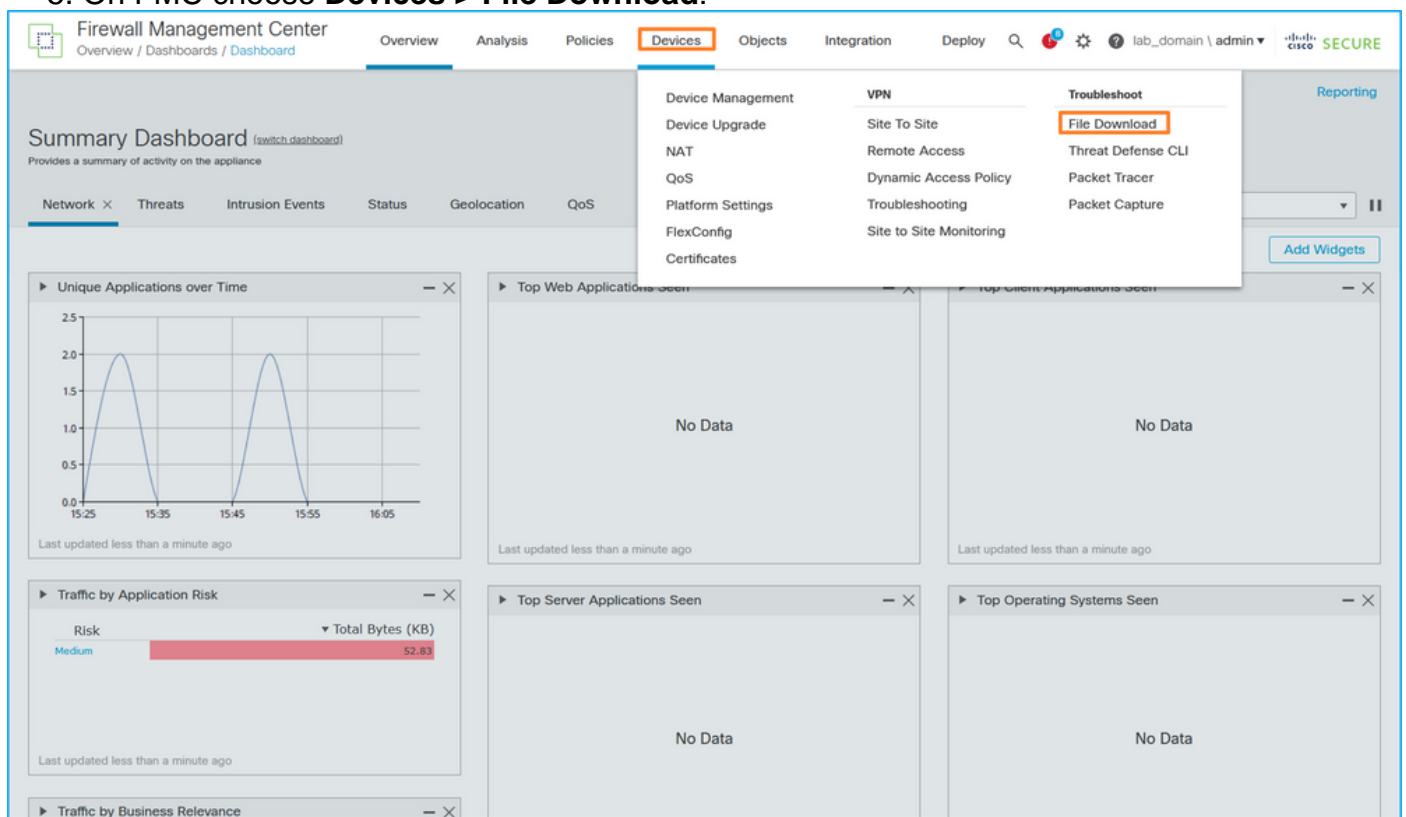
```
root@KSEC-FPR3100-1:/home/admin cp /mnt/disk0/packet-capture/sess-1-capsw-ethernet-1-1-0.pcap /ngfw/var/common/
```

```
root@KSEC-FPR3100-1:/home/admin ls -l /ngfw/var/common/sess*
```

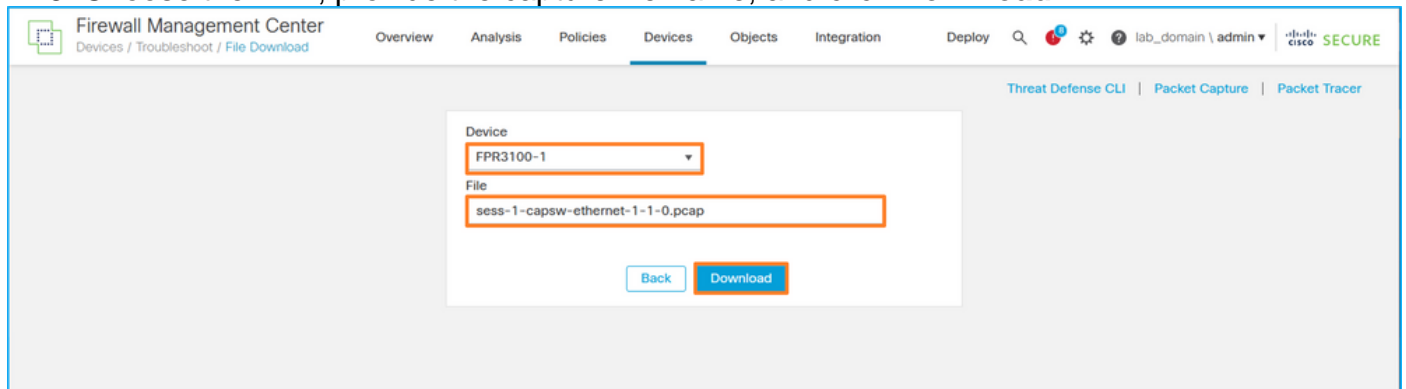
```
-rwxr-xr-x 1 root admin 139826 Aug  7 20:14 /ngfw/var/common/sess-1-capsw-ethernet-1-1-0.pcap
```

```
-rwxr-xr-x 1 root admin    24 Aug  6 21:58 /ngfw/var/common/sess-1-capsw-ethernet-1-3-0.pcap
```

5. On FMC choose **Devices > File Download**:



6. Choose the FTD, provide the capture file name, and click **Download**:



Guidelines, Limitations, and Best Practices for Internal Switch Packet Capture

Guidelines and limitations:

- Multiple switch capture configuration sessions are supported, but only 1 switch capture session can be active at a time. An attempt to enable 2 or more capture sessions results in an error "**ERROR: Failed to enable session, as limit of maximum 1 active packet capture sessions reached**".
- An active switch capture cannot be deleted.
- Switch captures cannot be read on the application. The user must export the files.
- Certain data plane capture options such as **dump**, **decode**, **packet-number**, **trace**, and others are not supported for switch captures.
- In the case of multi-context ASA, the switch captures on data interfaces are configured in user contexts. The switch captures on interfaces in_data_uplink1, and in_mgmt_uplink1 are supported only in the admin context.

This is the list of best practices based on the usage of packet capture in TAC cases:

- Be aware of guidelines and limitations.
- Use capture filters.
- Consider the impact of NAT on packet IP addresses when a capture filter is configured.
- Increase or decrease the **packet-length** that specifies frame size, in case it differs from the default value of 1518 bytes. Shorter size results in an increased number of captured packets and vice versa.
- Adjust the **buffer** size as needed.
- Be aware of the **Drop Count** in the output of the **show cap <cap_name> detail** command. Once the buffer size limit is reached, the drop count counter increases.

Related Information

- [Firepower 4100/9300 Chassis Manager and FXOS CLI Configuration Guides](#)
- [Cisco Secure Firewall 3100 Getting Started Guide](#)
- [Cisco Firepower 4100/9300 FXOS Command Reference](#)