Configure and Capture Embedded Packet on Software

Contents

[Introduction](#)
[Prerequisites](#)
[Requirements](#)
[Components Used](#)
[Background Information](#)
[Cisco IOS Configuration Example](#)
[Basic EPC Configuration](#)
[Additional Cisco IOS Configuration Information](#)
[Basic IP Traffic-Export Configuration](#)
[IP Traffic Export Disadvantages](#)
[Cisco IOS-XE Configuration Example](#)
[Basic EPC Configuration](#)
[Additional Information](#)
[Verify](#)
[Troubleshoot](#)
[Related Information](#)

Introduction

This document describes the Embedded Packet Capture (EPC) feature in Cisco IOS® software.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

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

- Cisco IOS Release 12.4(20)T or later
- Cisco IOS XE® Release 15.2(4)S - 3.7.0 or later

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.

Background Information

When enabled, the router captures the sent and received packets. The packets are stored within a buffer in DRAM and do not persist through a reload. Once the data is captured, it can be examined in a summary or detailed view on the router.

In addition, the data can be exported as a packet capture (PCAP) file to allow for further examination. The tool is configured in exec mode and is considered a temporary assistance tool. As a result, the tool configuration is not stored within the router configuration and does not remain in place after a system reload.

The Packet Capture Config Generator and Analyzer tool is available for Cisco Customers to aid in the configuration, capture, and extraction of packet captures.

Cisco IOS Configuration Example

Basic EPC Configuration

1. Define a 'capture buffer', which is a temporary buffer where the captured packets are stored.
2. There are various options that can be selected when the buffer is defined; such as size, maxium packet size, and circular/linear:

```
monitor capture buffer BUF size 2048 max-size 1518 linear
```

3. A filter is applicable to limit the capture to desired traffic. Define an Access Control List (ACL) within config mode and apply the filter to the buffer:

```
ip access-list extended BUF-FILTER
permit ip host 192.168.1.1 host 172.16.1.1
permit ip host 172.16.1.1 host 192.168.1.1

monitor capture buffer BUF filter access-list BUF-FILTER
```

4. Define a capture point which defines the location where the capture occurs.
5. The capture point also defines whether the capture occurs for IPv4 or IPv6 and in which switching path (process versus cef):

```
monitor capture point ip cef POINT fastEthernet 0 both
```

6. Attach the buffer to the capture point:

```
monitor capture point associate POINT BUF
```

7. Start the capture:

```
monitor capture point start POINT
```

8. The capture is now active. Allow collection of the necessary data.

9. Stop the capture:

```
monitor capture point stop POINT
```

10. Examine the buffer on the unit:

```
show monitor capture buffer BUF dump
```

Note: This output only shows the hex dump of the packets captures. In order to see them in human readable there are two ways.Export the buffer from the router for further analysis:

```
monitor capture buffer BUF export tftp://10.1.1.1/BUF.pcap
```

The previous method is not always practical as it required T/FTP access to the router. In such situations, take a copy of the hex dump and use any online hex-pcap convertor in order to view the files.

11. Once the necessary data has been collected, delete the 'capture point' and 'capture buffer':

```
no monitor capture point ip cef POINT fastEthernet 0 both
no monitor capture buffer BUF
```

Additional Cisco IOS Configuration Information

- In releases earlier than Cisco IOS® Release 15.0(1)M, the buffer size was limited to 512K.
- In releases earlier than Cisco IOS® Release 15.0(1)M, the captured packet size was limited to 1024 bytes.
- The packet buffer is stored in DRAM and does not persist through reloads.
- The capture configuration is not stored in NVRAM and does not persist through reloads.
- The capture point can be defined to capture in the cef or process switching paths.
- The capture point can be defined to capture only on an interface or globally.
- When the capture buffer is exported in PCAP format, L2 information (such as Ethernet encapsulation) is not preserved.
- See Best Practices for search Commands for more information on the commands used in this section.

Basic IP Traffic-Export Configuration

The IP Traffic Export is a diferent method to export IP packets that are received on multiple, simultaneous WAN or LAN interfaces.

1. In configuration mode define an IP traffic export profile.

```
Device(config)# ip traffic-export profile mypcap mode capture
```

2. Configure bidirectional traffic in the profile.

```
Device(config-rite)# bidirectional
```

3. Exit

4. Specify the interface for exported traffic.

```
Device(config-if)# interface GigabitEthernet 0/1
```

5. Enable IP traffic export on the interface.

```
Device(config-if)# ip traffic-export apply mypcap size 10000000
```

6. Exit

7. Start the capture. The capture is now active. Allow collection of the necessary data.

```
Device# traffic-export interface GigabitEthernet 0/1 start
```

8. Stop the capture.

```
Device# traffic-export interface GigabitEthernet 0/1 stop
```

9. Export the capture to an external TFTP server.

```
Device# traffic-export interface GigabitEthernet 0/1 copy tftp://<TFTP_Address>/mypcap.pcap
```

10. Once the necessary data has been collected, delete the profile.

```
Device(config)# no ip traffic-export profile mypcap
```

IP Traffic Export Disadvantages

IP Traffic Export has these disadvantages in comparison with EPC method:

- The interface where captured traffic is exported must be an ethernet interface.
- No IPv6 support.
- No layer 2 information, only layer 3 and higher.

Cisco IOS-XE Configuration Example

The Embedded Packet Capture feature was introduced in Cisco IOS-XE® Release 3.7 - 15.2(4)S. The configuration of the capture is different than Cisco IOS® because it adds more features.

Basic EPC Configuration

1. Define the location where the capture occurs:

```
monitor capture CAP interface GigabitEthernet0/0/1 both
```

2. Associate a filter. The filter is either specified inline, or an ACL or class-map can be referenced:

```
monitor capture CAP match ipv4 protocol tcp any any limit pps 1000000
```

3. Start the capture:

```
monitor capture CAP start
```

4. The capture is now active. Allow it to collect the necessary data.

5. Stop the capture:

```
monitor capture CAP stop
```

6. Examine the capture in a summary view:

```
show monitor capture CAP buffer brief
```

7. Examine the capture in a detailed view:

```
show monitor capture CAP buffer detailed
```

8. In addition, export the capture in PCAP format for further analysis:

```
monitor capture CAP export tftp://10.0.0.1/CAP.pcap
```

9. Once the necessary data has been collected, remove the capture:

```
no monitor capture CAP
```

Additional Information

- The capture is performed on physical interfaces, sub-interfaces, and tunnel interfaces.
- Network Based Application Recognition (NBAR) based filters (that use the **match protocol** command under the class-map) are currently not supported.
- See Best Practices for search Commands for more information on the commands used in this section.

Verify

There is currently no verification procedure available for this configuration.

Troubleshoot

For EPC that runs on Cisco IOS-XE®, this debug command is used to ensure EPC is set up properly:

```
debug epc provision
debug epc capture-point
```

Related Information

- **Embedded Packet Capture - Cisco IOS-XE**
- **Embedded Packet Capture - Cisco IOS**
- **Technical Support & Documentation - Cisco Systems**