

Embedded Packet Capture Overview

Embedded Packet Capture (EPC) provides an embedded systems management facility that helps in tracing and troubleshooting packets. This feature allows network administrators to capture data packets flowing through, to, and from a Cisco device. The network administrator may define the capture buffer size and type (circular, or linear), the maximum number of bytes of each packet to capture, and the direction of the traffic flow - ingress or egress, or both. The packet capture rate can be throttled using further administrative controls. For example, you can use the available options for filtering the packets to be captured using an Access Control List; and, optionally, further defined by specifying a maximum packet capture rate or by specifying a sampling interval.

- Feature Information for Embedded Packet Capture, on page 1
- Prerequisites for Embedded Packet Capture, on page 2
- Restrictions for Embedded Packet Capture, on page 3
- Information About Embedded Packet Capture, on page 3
- How to Implement Embedded Packet Capture, on page 4
- Configuration Examples for Embedded Packet Capture, on page 7
- Additional References, on page 11

Feature Information for Embedded Packet Capture

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for Embedded Packet Capture

Feature Name	Releases	Feature Information
Embedded Packet Capture support on LTE interface and FlexVPN interface	Cisco IOS XE Release Bengaluru 17.4.1a	Embedded Packet Capture (EPC) is an onboard packet capture facility that allows network administrators to capture packets flowing to, through, and from a device. With this feature, EPC is supported on LTE and FlexVPN interfaces. The following commands were introduced or modified: **show ip interface brief**, **show monitor capture epc**
Embedded Packet Capture	Cisco IOS XE Release 3.7S	Embedded Packet Capture (EPC) is an onboard packet capture facility that allows network administrators to capture packets flowing to, through, and from a device and to analyze them locally or save and export them for offline analysis using a tool such as Wireshark. This feature simplifies operations by allowing the devices to become active participants in the management and operation of the network. The following commands were introduced or modified: **debug epc**, **monitor capture (access list/class map)**, **monitor capture (interface/control plane)**, **monitor capture export**, **monitor capture limit**, **monitor capture start**, **monitor capture stop**, and **show monitor capture**.

Prerequisites for Embedded Packet Capture

The Embedded Packet Capture (EPC) software subsystem consumes CPU and memory resources during its operation. You must have adequate system resources for different types of operations. Some guidelines for using the system resources are provided in the table below.

Table 2: System Requirements for the EPC Subsystem

System Resources	Requirements
Hardware	CPU utilization requirements are platform dependent.
Memory	The packet buffer is stored in DRAM. The size of the packet buffer is user specified.
Diskspace	Packets can be exported to external devices. No intermediate storage on flash disk is requ

Restrictions for Embedded Packet Capture

- Embedded Packet Capture (EPC) captures multicast packets only on ingress and does not capture the replicated packets on egress.
- From Cisco IOS XE Release 3.7S, Embedded Packet Capture is only supported on Advance Enterprise Krypto (K9) images.
- From Cisco IOS XE Release 3.9S, Embedded Packet Capture is available on the following images:
  - IP Base Images
  - Special Services Images
  - Advance Security Images
  - Advance IP Services Images
  - Advance Enterprise Images

Information About Embedded Packet Capture

Embedded Packet Capture Overview

Embedded Packet Capture (EPC) provides an embedded systems management facility that helps in tracing and troubleshooting packets. This feature allows network administrators to capture data packets flowing through, to, and from a Cisco device. The network administrator may define the capture buffer size and type (circular, or linear), the maximum number of bytes of each packet to capture, and the direction of the traffic flow - ingress or egress, or both. The packet capture rate can be throttled using further administrative controls. For example, you can use the available options for filtering the packets to be captured using an Access Control List; and, optionally, further defined by specifying a maximum packet capture rate or by specifying a sampling interval.

Benefits of Embedded Packet Capture

- Ability to capture IPv4 and IPv6 packets in the device.
- Extensible infrastructure for enabling packet capture points. A capture point is a traffic transit point where a packet is captured and associated with a buffer.
- Facility to export the packet capture in packet capture file (PCAP) format suitable for analysis using any external tool.

- Methods to decode data packets captured with varying degrees of detail.

Packet Data Capture

Packet data capture is the capture of data packets that are then stored in a buffer. You can define packet data captures by providing unique names and parameters.

You can perform the following actions on the capture:

- Activate captures at any interface.
- Apply access control lists (ACLs) or class maps to capture points.

Note Network Based Application Recognition (NBAR) and MAC-style class map is not supported.

- Destroy captures.
- Specify buffer storage parameters such as size and type. The size ranges from 1 MB to 100 MB. The default buffer is linear; the other option for the buffer is circular.
- Specify any of the following limit options:
  - **duration** - limit total duration of capture in seconds.
  - **every** - limit capture to one in every nth packet.
  - **packet-len** - limit the packet length to capture.
  - **packets** - limit number of packets to capture.
  - **pps** - limit number of packets per second to capture.
- Specify match criteria that includes information about the protocol, IP address or port address.

How to Implement Embedded Packet Capture

Managing Packet Data Capture

SUMMARY STEPS

1. **enable**
2. **monitor capture** *capture-name* **access-list** *access-list-name*
3. **monitor capture** *capture-name* **limit duration** *seconds*
4. **monitor capture** *capture-name* **interface** *interface-name* **both**
5. **monitor capture** *capture-name* **buffer circular size** *bytes*
6. **monitor capture** *capture-name* **start**
7. **monitor capture** *capture-name* **export** *file-location/file-name*

8. **monitor capture** *capture-name* **stop**
9. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**monitor capture** *capture-name* **access-list** *access-list-name* **Example:** `Device# monitor capture mycap access-list v4acl`	Configures a monitor capture specifying an access list as the core filter for the packet capture.
Step 3	**monitor capture** *capture-name* **limit duration** *seconds* **Example:** `Device# monitor capture mycap limit duration 1000`	Configures monitor capture limits.
Step 4	**monitor capture** *capture-name* **interface** *interface-name* **both** **Example:** `Device# monitor capture mycap interface GigabitEthernet 0/0/1 both`	Configures monitor capture specifying an attachment point and the packet flow direction. **Note** • To change the traffic direction from **both** to **in** (ingress direction), enter the **no monitor capture** *capture-name* **interface** *interface-name* **out** command. • To change the traffic direction from **both** to **out** (egress direction), enter the **no monitor capture** *capture-name* **interface** *interface-name* **in** command.
Step 5	**monitor capture** *capture-name* **buffer circular size** *bytes* **Example:** `Device# monitor capture mycap buffer circular size 10`	Configures a buffer to capture packet data.
Step 6	**monitor capture** *capture-name* **start** **Example:** `Device# monitor capture mycap start`	Starts the capture of packet data at a traffic trace point into a buffer.
Step 7	**monitor capture** *capture-name* **export** *file-location/file-name* **Example:**	Exports captured data for analysis.

	Command or Action	Purpose
	`Device# monitor capture mycap export tftp://10.1.88.9/mycap.pcap`	
Step 8	**monitor capture** *capture-name* **stop** **Example:** `Device# monitor capture mycap stop`	Stops the capture of packet data at a traffic trace point.
Step 9	**end** **Example:** `Device# end`	Exits privileged EXEC mode.

Monitoring and Maintaining Captured Data

Perform this task to monitor and maintain the packet data captured. Capture buffer details and capture point details are displayed.

SUMMARY STEPS

1. **enable**
2. **show monitor capture** *capture-buffer-name* **buffer dump**
3. **show monitor capture** *capture-buffer-name* **parameter**
4. **debug epc capture-point**
5. **debug epc provision**
6. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show monitor capture** *capture-buffer-name* **buffer dump** **Example:** `Device# show monitor capture mycap buffer dump`	(Optional) Displays a hexadecimal dump of captured packet and its metadata.
Step 3	**show monitor capture** *capture-buffer-name* **parameter** **Example:** `Device# show monitor capture mycap parameter`	(Optional) Displays a list of commands that were used to specify the capture.

	Command or Action	Purpose
Step 4	**debug epc capture-point** **Example:** `Device# debug epc capture-point`	(Optional) Enables packet capture point debugging.
Step 5	**debug epc provision** **Example:** `Device# debug epc provision`	(Optional) Enables packet capture provisioning debugging.
Step 6	**exit** **Example:** `Device# exit`	Exits privileged EXEC mode.

Configuration Examples for Embedded Packet Capture

Example: Managing Packet Data Capture

The following example shows how to manage packet data capture:

```
Device> enable
Device# monitor capture mycap access-list v4acl
Device# monitor capture mycap limit duration 1000
Device# monitor capture mycap interface GigabitEthernet 0/0/1 both
Device# monitor capture mycap buffer circular size 10
Device# monitor capture mycap start
Device# monitor capture mycap export tftp://10.1.88.9/mycap.pcap
Device# monitor capture mycap stop
Device# end
```

Example: Monitoring and Maintaining Captured Data

The following example shows how to dump packets in ASCII format:

```
Device# show monitor capture mycap buffer dump

0
 0000:  01005E00 00020000 0C07AC1D 080045C0   ..^...........E.
 0010:  00300000 00000111 CFDC091D 0002E000   .0..............
 0020:  000207C1 07C1001C 802A0000 10030AFA   .........*......
 0030:  1D006369 73636F00 0000091D 0001       ..example.......

1
 0000:  01005E00 0002001B 2BF69280 080046C0   ..^.....+.....F.
 0010:  00200000 00000102 44170000 0000E000   . ......D.......
 0020:  00019404 00001700 E8FF0000 0000       ..............

2
```

```
 0000:  01005E00 0002001B 2BF68680 080045C0   ..^.....+.....E.
 0010:  00300000 00000111 CFDB091D 0003E000   .0..............
 0020:  000207C1 07C1001C 88B50000 08030A6E   ...............n
 0030:  1D006369 73636F00 0000091D 0001       ..example.......

3
 0000:  01005E00 000A001C 0F2EDC00 080045C0   ..^...........E.
 0010:  003C0000 00000258 CE7F091D 0004E000   .<.....X........
 0020:  000A0205 F3000000 00000000 00000000   ................
 0030:  00000000 00D10001 000C0100 01000000   ................
 0040:  000F0004 00080501 0300                ................
```

The following example shows how to display the list of commands used to configure the capture named mycap:

```
Device# show monitor capture mycap parameter

  monitor capture mycap interface GigabitEthernet 1/0/1 both
  monitor capture mycap match any
  monitor capture mycap buffer size 10
  monitor capture mycap limit pps 1000
```

The following example shows how to debug the capture point:

```
Device# debug epc capture-point

EPC capture point operations debugging is on
Device# monitor capture mycap start

*Jun  4 14:17:15.463: EPC CP:  Starting the capture cap1
*Jun  4 14:17:15.463: EPC CP:  (brief=3, detailed=4, dump=5) = 0
*Jun  4 14:17:15.463: EPC CP:  final check before activation
*Jun  4 14:17:15.463: EPC CP:  setting up c3pl infra
*Jun  4 14:17:15.463: EPC CP:  Setup c3pl acl-class-policy
*Jun  4 14:17:15.463: EPC CP:  Creating a class
*Jun  4 14:17:15.464: EPC CP:  Creating a class : Successful
*Jun  4 14:17:15.464: EPC CP:  class-map Created
*Jun  4 14:17:15.464: EPC CP: creating  policy-name epc_policy_cap1
*Jun  4 14:17:15.464: EPC CP:  Creating Policy epc_policy_cap1 of type 49 and client type
21
*Jun  4 14:17:15.464: EPC CP:  Storing a Policy
*Jun  4 14:17:15.464: EPC CP:  calling ppm_store_policy with epc_policy
*Jun  4 14:17:15.464: EPC CP:  Creating Policy : Successful
*Jun  4 14:17:15.464: EPC CP:  policy-map created
*Jun  4 14:17:15.464: EPC CP:  creating filter for ANY
*Jun  4 14:17:15.464: EPC CP:  Adding acl to class : Successful
*Jun  4 14:17:15.464: EPC CP:  Setup c3pl class to policy
*Jun  4 14:17:15.464: EPC CP:  Attaching Class to Policy
*Jun  4 14:17:15.464: EPC CP:  Attaching epc_class_cap1 to epc_policy_cap1
*Jun  4 14:17:15.464: EPC CP:  Attaching Class to Policy : Successful
*Jun  4 14:17:15.464: EPC CP:  setting up c3pl qos
*Jun  4 14:17:15.464: EPC CP:  DBG> Set packet rate limit to 1000
*Jun  4 14:17:15.464: EPC CP: creating action for policy_map epc_policy_cap1 class_map
epc_class_cap1
*Jun  4 14:17:15.464: EPC CP:  DBG> Set packet rate limit to 1000
*Jun  4 14:17:15.464: EPC CP:  Activating Interface GigabitEthernet1/0/1  direction both
*Jun  4 14:17:15.464: EPC CP: Id attached 0
*Jun  4 14:17:15.464: EPC CP:  inserting into active lists
*Jun  4 14:17:15.464: EPC CP: Id attached 0
*Jun  4 14:17:15.465: EPC CP:  inserting into active lists
*Jun  4 14:17:15.465: EPC CP:  Activating Vlan
*Jun  4 14:17:15.465: EPC CP:  Deleting all temp interfaces
```

```
*Jun  4 14:17:15.465: %BUFCAP-6-ENABLE: Capture Point cap1 enabled.
*Jun  4 14:17:15.465: EPC CP:  Active Capture 1

Device# monitor capture mycap1 stop

*Jun  4 14:17:31.963: EPC CP:  Stopping the capture cap1
*Jun  4 14:17:31.963: EPC CP: Warning: unable to unbind capture cap1
*Jun  4 14:17:31.963: EPC CP:  Deactivating policy-map
*Jun  4 14:17:31.963: EPC CP:  Policy epc_policy_cap1
*Jun  4 14:17:31.964: EPC CP:  Deactivating policy-map Successful
*Jun  4 14:17:31.964: EPC CP:  removing povision feature
*Jun  4 14:17:31.964: EPC CP:  Found action for policy-map epc_policy_cap1 class-map
epc_class_cap1
*Jun  4 14:17:31.964: EPC CP:  cleanning up c3pl infra
*Jun  4 14:17:31.964: EPC CP:  Removing Class epc_class_cap1 from Policy
*Jun  4 14:17:31.964: EPC CP:  Removing Class from epc_policy_cap1
*Jun  4 14:17:31.964: EPC CP:  Successfully removed
*Jun  4 14:17:31.964: EPC CP:  Removing acl mac from class
*Jun  4 14:17:31.964: EPC CP:  Removing acl from class : Successful
*Jun  4 14:17:31.964: EPC CP:  Removing all policies
*Jun  4 14:17:31.964: EPC CP:  Removing Policy  epc_policy_cap1
*Jun  4 14:17:31.964: EPC CP:  Removing Policy : Successful
*Jun  4 14:17:31.964: EPC CP:  Removing class epc_class_cap1
*Jun  4 14:17:31.965: EPC CP:  Removing class : Successful
*Jun  4 14:17:31.965: %BUFCAP-6-DISABLE: Capture Point cap1 disabled.
*Jun  4 14:17:31.965: EPC CP:  Active Capture 0
```

The following example shows how to debug the Embedded Packet Capture (EPC) provisioning:

```
Device# debug epc provision

EPC provisionioning debugging is on

Device# monitor capture mycap start

*Jun  4 14:17:54.991: EPC PROV: No action found for policy-map epc_policy_cap1 class-map
epc_class_cap1
*Jun  4 14:17:54.991: EPC PROV:
*Jun  4 14:17:54.991: Attempting to install service policy epc_policy_cap1

*Jun  4 14:17:54.992: EPC PROV: Attached service policy to epc idb subblock
*Jun  4 14:17:54.992: EPC PROV: Successful. Create feature object
*Jun  4 14:17:54.992: EPC PROV:
*Jun  4 14:17:54.992: Attempting to install service policy epc_policy_cap1

*Jun  4 14:17:54.992: EPC PROV: Successful. Create feature object
*Jun  4 14:17:54.992: %BUFCAP-6-ENABLE: Capture Point cap1 enabled.

Device# monitor capture mycap stop

*Jun  4 14:18:02.503: EPC PROV: Successful. Remove feature object
*Jun  4 14:18:02.504: EPC PROV: Successful. Remove feature object
*Jun  4 14:18:02.504: EPC PROV: Destroyed epc idb subblock
*Jun  4 14:18:02.504: EPC PROV: Found action for policy-map epc_policy_cap1 class-map
epc_class_cap1
*Jun  4 14:18:02.504: EPC PROV: Deleting EPC action
*Jun  4 14:18:02.504: EPC PROV: Successful. CLASS_REMOVE, policy-map epc_policy_cap1, class
 epc_class_cap1
*Jun  4 14:18:02.504: %BUFCAP-6-DISABLE: Capture Point cap1 disabled.
```

The following example shows the interfaces that are available during a crypto flexvpn session.

Note If the device supports an LTE module, the cellular interface is displayed under the **show ip interface brief**

```
Device# show ip interface brief

Interface              IP-Address      OK? Method Status                Protocol
Cellular                               YES        up                    up
GigabitEthernet1       9.45.7.78       YES NVRAM  up                    up
GigabitEthernet2       1.1.1.2         YES NVRAM  up                    up
GigabitEthernet3       unassigned      YES NVRAM  administratively down down
GigabitEthernet4       unassigned      YES NVRAM  administratively down down
GigabitEthernet5       unassigned      YES NVRAM  administratively down down
Loopback100            7.1.1.100       YES manual up                    up
Loopback200            8.8.8.100       YES manual up                    up
Loopback300            18.18.18.100    YES manual up                    up
Virtual-Access1        7.1.1.100       YES unset  up                    up
Virtual-Access2        7.1.1.100       YES unset  up                    up
Virtual-Template1      7.1.1.100       YES unset  up                    down
```

Note A virtual-access is available when the interface is dynamically created through a crypto (flexvpn) session using the **monitor capture***capture-name***interface** command.

The following example shows the cellular device operating in the controller mode:

```
Device# show monitor capture EPC

Status Information for Capture EPC
  Target Type: Interface: Cellular0/2/0, Direction: BOTH
Status : Active
  Filter Details: IPv4
    Source IP:  any
    Destination IP:  any
   Protocol: any
  Buffer Details:
   Buffer Type: LINEAR (default)
   Buffer Size (in MB): 10
  Limit Details:
   Number of Packets to capture: 0 (no limit)
   Packet Capture duration: 0 (no limit)
   Packet Size to capture: 0 (no limit)
   Maximum number of packets to capture per second: 1000
   Packet sampling rate: 0 (no sampling)
```

Additional References

Related Documents

Related Topic	Document Title
Embedded Packet Capture commands	Cisco IOS Embedded Packet Capture Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html