Configuring Packet Capture

Prerequisites for Configuring Packet Capture

Packet capture is supported on Cisco Catalyst 9200 Series Switches

The following section provides information about the prerequisites for configuring packet capture.

Prerequisites for Configuring 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 1: 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 required.

Restrictions for Configuring Embedded Packet Capture

- Layer 2 EtherChannels are not supported.

- Neither VRFs, management ports, nor private VLANs can be used as attachment points.

- Embedded Packet Capture (EPC) is not supported on logical ports, which includes port channels, switch virtual interfaces (SVIs), and subinterfaces. It is supported only on physical ports.

- EPC is not supported on a VLAN interface that is in shutdown state.

- If the user changes interface from switch port to routed port (Layer 2 to Layer 3) or vice versa, they must delete the capture point and create a new one, once the interface comes back up. Stop/start the capture point will not work.

- Packets captured in the output direction of an interface might not reflect the changes made by the device rewrite (includes TTL, VLAN tag, CoS, checksum, MAC addresses, DSCP, precedent, UP, etc.).

- Even though the minimum configurable duration for packet capture is 1 second, packet capture works for a minimum of 2 seconds.

- It is not possible to modify a capture point parameter when a capture is already active or has started.

- EPC captures multicast packets only on ingress and does not capture the replicated packets on egress.

- The Rewrite information of both ingress and egress packets are not captured.

- CPU-injected packets are considered control plane packets. Therefore, these types of packets will not be captured on an interface egress capture.

- Control plane packets are not rate limited and performance impacting. Please use filters to limit control plane packet capture.

- Decoding of protocols such as Control and Provisioning of Wireless Access Points (CAPWAP) is supported in DNA Advantage.

- Up to 8 capture points can be defined, but only one can be active at a time. You need to stop one before you can start the other.

- MAC filter will not capture IP packets even if it matches the MAC address. This applies to all interfaces (Layer 2 switch port, Layer 3 routed port)

- MAC ACL is only used for non-IP packets such as ARP. It will not be supported on a Layer 3 port or SVI.

- MAC filter cannot capture Layer 2 packets (ARP) on Layer 3 interfaces.

- IPv6-based ACLs are not supported in VACL.

- EPC cannot capture based on the underlying routing protocols in MPLS packets.

Information About Packet Capture

The Packet Capture feature is an onboard packet capture facility that allows network administrators to capture packets flowing to, through, and from the device and to analyze them locally or save and export them for offline analysis by using Embedded Packet Capture (EPC). This feature simplifies network operations by allowing devices to become active participants in the management and operation of the network. This feature facilitates troubleshooting by gathering information about the packet format. This feature also facilitates application analysis and security.

About 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) and the maximum number of bytes of each packet to capture. The packet capture rate can be throttled using further administrative controls. For example, options allow 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.

Prior to Cisco IOS XE Amsterdam 17.2.1 , EPC is not supported on an interface in shutdown state. Starting from Cisco IOS XE Amsterdam 17.2.1 , EPC is supported on an interface in shutdown state. This is useful in capturing packets on an interface as it is being brought up.

Benefits of Embedded Packet Capture

- Ability to capture IPv4 and IPv6 packets in the device, and also capture non-IP packets with MAC filter or match any MAC address.

- 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 option for the buffer is linear and the other option for the buffer is circular.

- Specify match criteria that includes information about the protocol, IP address or port address.

How to Implement Embedded Packet Capture

The following sections provide information on how to implement EPC.

Managing Packet Data Capture

To manage Packet Data Capture in the buffer mode, perform the following 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.
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* **stop** **Example:** Device# **monitor capture mycap stop**	Stops the capture of packet data at a traffic trace point.

	Command or Action	**Purpose**
Step 8	**monitor capture** *capture-name* **export** *file-location/file-name* **Example:** Device# **monitor capture mycap export tftp://10.1.88.9/mycap.pcap**	Exports captured data for analysis.
Step 9	**end** **Example:** Device# **end**	Returns to 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.

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.
Step 4	**debug epc capture-point** **Example:** Device# **debug epc capture-point**	(Optional) Enables packet capture point debugging.
Step 5	**debug epc provision** **Example:**	(Optional) Enables packet capture provisioning debugging.

	Command or Action	Purpose
	Device# **debug epc provision**	
Step 6	**end** **Example:** Device(config)# **end**	Returns to 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 stop
Device# monitor capture mycap export tftp://10.1.88.9/mycap.pcap
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
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

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.
```

Feature History for Configuring Packet Capture

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Table 2: Feature History for Configuring Packet Capture

Release	Feature Name	Feature Information
Cisco IOS XE Gibraltar 16.10.1	Configuring Packet Capture	This feature was introduced.
Cisco IOS XE Amsterdam 17.2.1	Configuring EPC on an interface either in down state or admin state.	Configuring EPC on an interface that is either in down state or admin down state does not affect packet capture once the interface changes to up state.