



Embedded Packet Capture Configuration Guide, Cisco IOS XE Release 3S

Embedded Packet Capture	2
Finding Feature Information	2
Prerequisites for Embedded Packet Capture	2
Restrictions for Embedded Packet Capture	2
Information About Embedded Packet Capture	3
How to Implement Embedded Packet Capture	4
Configuration Examples for Embedded Packet Capture	6
Additional References	8
Feature Information for Embedded Packet Capture	9

Revised: March 17, 2016,

Embedded Packet Capture

Embedded Packet Capture (EPC) 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 a tool such as Wireshark. 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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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.

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 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 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) 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.

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

How to Implement Embedded Packet Capture

Managing Packet Data Capture

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	monitor capture <i>capture-name</i> access-list <i>access-list-name</i> 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 <i>capture-name</i> limit duration <i>seconds</i> Example: Device# monitor capture mycap limit duration 1000	Configures monitor capture limits.
Step 4	monitor capture <i>capture-name</i> interface <i>interface-name</i> 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 <ul style="list-style-type: none">• To change the traffic direction from both to in (ingress direction), enter the no monitor capture <i>capture-name</i> interface <i>interface-name</i> out command.• To change the traffic direction from both to out (egress direction), enter the no monitor capture <i>capture-name</i> interface <i>interface-name</i> in command.
Step 5	monitor capture <i>capture-name</i> buffer circular size <i>bytes</i> Example: Device# monitor capture mycap buffer circular size 10	Configures a buffer to capture packet data.
Step 6	monitor capture <i>capture-name</i> 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 <i>capture-name</i> export <i>file-location/file-name</i>	Exports captured data for analysis.

	Command or Action	Purpose
	Example: Device# monitor capture mycap export tftp://10.1.88.9/mycap.pcap	
Step 8	monitor capture <i>capture-name</i> 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.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show monitor capture <i>capture-buffer-name</i> 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 <i>capture-buffer-name</i> 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.

	Command or Action	Purpose
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 start
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 provisioning 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.
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
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

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 2: Feature Information for Embedded Packet Capture

Feature Name	Releases	Feature Information
Embedded Packet Capture	Cisco IOS XE Release 3.7S	<p>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. This feature facilitates better troubleshooting by gathering information about packet format. It also facilitates application analysis and security.</p> <p>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 .</p>

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <http://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

© 2016 Cisco Systems, Inc. All rights reserved.

**Americas Headquarters**

Cisco Systems, Inc.
San Jose, CA 95134-1706
USA

Asia Pacific Headquarters

Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters

Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.