Embedded Packet Capture Configuration Guide, Cisco IOS Release 15M&T

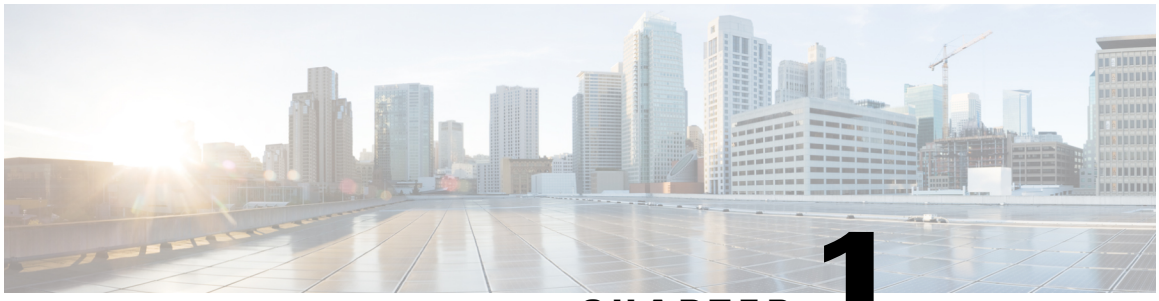
First Published: 2012-11-29

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

CHAPTER 1 **Embedded Packet Capture** 1

- Finding Feature Information 1
- Prerequisites for Embedded Packet Capture 2
- Restrictions for Embedded Packet Capture 2
- Information About Embedded Packet Capture 2
  - Embedded Packet Capture Overview 2
  - Benefits of EPC 2
  - Capture Buffer 3
  - Capture Point 3
- How to Implement Embedded Packet Capture 4
  - Starting Packet Data Capture 4
  - Stopping Packet Data Capture 5
  - Exporting Packet Data for Analysis 6
  - Monitoring and Maintaining Captured Data 7
- Configuration Examples for Embedded Packet Capture 7
  - Starting Packet Data Capture Example 7
  - Stopping Packet Data Capture Example 8
  - Exporting Packet Data Example 8
  - Monitoring and Maintaining Captured Data Example 8
- Additional References 9
- Feature Information for Embedded Packet Capture 10



CHAPTER 1

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, page 1
- Prerequisites for Embedded Packet Capture, page 2
- Restrictions for Embedded Packet Capture, page 2
- Information About Embedded Packet Capture, page 2
- How to Implement Embedded Packet Capture, page 4
- Configuration Examples for Embedded Packet Capture, page 7
- Additional References, page 9
- Feature Information for Embedded Packet Capture, page 10

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

- In Cisco IOS Release 12.2(33)SRE, Embedded Packet Capture is supported only on 7200 platform.
- Embedded Packet Capture only captures multicast packets on ingress and does not capture the replicated packets on egress.
- Currently, the capture file can only be exported off the device; for example, TFTP or FTP servers and local disk.

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 EPC

Some of the benefits of this feature include:

- Ability to capture IPv4 and IPv6 packets in the Cisco Express Forwarding (CEF) path.
- A flexible method for specifying the capture buffer parameters.
- Filter captured packets.
- Methods to decode data packets captured with varying degree of detail.
- Facility to export the packet capture in PCAP format suitable for analysis using an external tool.
- Extensible infrastructure for enabling packet capture points.

Capture Buffer

The capture buffer is an area in memory for holding the packet data. You can specify unique names, size and type of the buffer, and configure the buffer to handle incoming data as required.

The following types of data are stored in a capture buffer:

- Packet data
- Metadata

The packet data starts from datagramstart and copies a minimum of the per-packet-capture size or datagramsize to the capture buffer.

The metadata contains descriptive information about a set of packet data. It contains:

- A timestamp of when it is added to a buffer.
- The direction in which the packet data is transmitted--egress or ingress.
- The switch path captured.
- Encapsulation type corresponding to input or output interface to allow the decoding of L2 decoders.

The following actions can be performed on capture buffers:

- Define a capture buffer and associate it with a capture point.
- Clear capture buffers.
- Export capture buffers for offline analysis. Export writes off the file using one of the supported file transfer options: FTP, HTTP, HTTPS, PRAM, RCP, SCP, and TFTP.
- Display content of the capture buffers.

Capture Point

The capture point is a traffic transit point where a packet is captured and associated with a buffer. You can define capture points by providing unique names and different parameters.

The following capture points are available:

- IPv4 CEF/interrupt switching path with interface input and output
- IPv6 CEF/interrupt switching path with interface input and output

You can perform the following actions on the capture point:

- Associate or disassociate capture points with capture buffers. Each capture point can be associated with only one capture buffer.
- Destroy capture points.
- Activate packet capture points on a given interface. Multiple packet capture points can be made active on a given interface. For example, Border Gateway Protocol (BGP) packets can be captured into one capture buffer and Open Shortest Path First (OSPF) packets can captured into another capture buffer.
- Access Control Lists (ACLs) can be applied to capture points.

How to Implement Embedded Packet Capture

Starting Packet Data Capture

Perform this task to start capturing packet data for analysis and troubleshooting. To capture packet data, a capture buffer and a capture point need to be defined. The capture point should then be associated with the capture buffer. Enabling the capture point will start the process of capturing packet data.

SUMMARY STEPS

1. **enable**
2. **monitor capture buffer** *buffer-name* [**clear** | **export** *export-location* | **filter access-list** {*ip-access-list* | *ip-expanded-list* | *access-list-name*} | **limit** {**allow-nth-pak** *nth-packet* | **duration** *seconds* | **packet-count** *total-packets* | **packets-per-sec** *packets*} | [**max-size** *element-size*] [**size** *buffer-size*] [**circular**| **linear**]]
3. **monitor capture point** {**ip**| **ipv6**}{**cef** *capture-point-name interface-name interface-type*{**both** | **in** | **out**}| **process-switched** *capture-point-name* {**both**| **from-us**| **in** | **out**}}
4. **monitor capture point associate** *capture-point-name capture-buffer-name*
5. **monitor capture point start** {*capture-point-name* | **all**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**monitor capture buffer** *buffer-name* [**clear**	**export** *export-location*	**filter access-list** {*ip-access-list*	*ip-expanded-list*	*access-list-name*}	**limit** {**allow-nth-pak**	Defines a capture buffer with the specified name and parameters. • In this example, a circular capture buffer by name pktrace1 with a size of 256 bytes and a maximum buffer elemment size of 100 bytes is defined.

	Command or Action	Purpose
	nth-packet | **duration** *seconds* | **packet-count** *total-packets* | **packets-per-sec** *packets*} | [**max-size** *element-size*] [**size** *buffer-size*] [**circular**| **linear**]] **Example:** `Router# monitor capture buffer pktrace1 size 256 max-size 100 circular`	
Step 3	**monitor capture point** {**ip**| **ipv6**}{**cef** *capture-point-name interface-name interface-type*{**both** | **in** | **out**}| **process-switched** *capture-point-name* {**both**| **from-us**| **in** | **out**}} **Example:** `Router# monitor capture point ip cef ipceffa0/1 fastEthernet 0/1 both`	Defines a capture point with the specified parameters. • In this example, a capture point by name ipceffa0/1 with the Fast Ethernet 0/1 interface in both directions is defined.
Step 4	**monitor capture point associate** *capture-point-name capture-buffer-name* **Example:** `Router# monitor capture point associate ipceffa0/1 pktrace1`	Associates the capture point with the capture buffer specified. • Associating a capture point with a capture buffer results in all packets captured from the specified capture point to be dumped to the associated capture buffer. • In this example, the capture point ipceffa0/1 is associated with the capture buffer pktrace1.
Step 5	**monitor capture point start** {*capture-point-name* | **all**} **Example:** `Router# monitor capture point start ipceffa0/1`	Enables the capture point to start capturing packet data. • In this example, the capture point ipceffa0/1 is enabled.

Stopping Packet Data Capture

Perform this task to stop capturing packet data.

SUMMARY STEPS

1. **enable**
2. **monitor capture point stop** {*capture-point-name* \| **all**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**monitor capture point stop** {*capture-point-name* | **all**} **Example:** `Router# monitor capture point stop ipceffa0/1`	Disables the capture point and stops the packet data capture process. • In this example, the capture point ipceffa0/1 is disabled.

Exporting Packet Data for Analysis

Perform this task to export the packet data for analysis using an external tool.

SUMMARY STEPS

1. **enable**
2. **monitor capture buffer** *buffer-name* **export** *export-location*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**monitor capture buffer** *buffer-name* **export** *export-location* **Example:** `Router# monitor capture buffer pktrace1 export tftp://10.1.88.9/pktrace1`	Exports the data for analysis. • In this example, data from the capture buffer pktrace1 is exported using the TFTP protocol.

Monitoring and Maintaining Captured Data

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

SUMMARY STEPS

1. **enable**
2. **show monitor capture** {**buffer** {*capture-buffer-name* [**parameters**] | **all parameters** | **merged** *capture-buffer-name1 capture-buffer-name2*}[**dump**] [**filter** *filter-parameters*]} | **point** {**all** | *capture-point-name*}}
3. **debug packet-capture**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show monitor capture** {**buffer** {*capture-buffer-name* [**parameters**]	**all parameters**	**merged** *capture-buffer-name1 capture-buffer-name2*}[**dump**] [**filter** *filter-parameters*]}	**point** {**all**	*capture-point-name*}} **Example:** `Router# show monitor capture buffer pktrace1 dump`	Displays the data captured. • In this example, data from the capture buffer pktrace1 is displayed.
Step 3	**debug packet-capture** **Example:** `Router# debug packet-capture`	Enables packet capture infra debugs.

Configuration Examples for Embedded Packet Capture

Starting Packet Data Capture Example

The following example shows how to capture packets to and from Fast Ethernet 0/1 interface:

```
Router> enable
```

```
Router# monitor capture buffer pktrace1
Router# monitor capture point ip cef ipceffa0/1 fastEthernet 0/1 both
Router# monitor capture point associate ipceffa0/1 pktrace1
Router# monitor capture point start ipceffa0/1
Mar 21 11:13:34.023: %BUFCAP-6-ENABLE: Capture Point ipceffa0/1 enabled.
Router# show monitor capture point all
Status Information for Capture Point ipceffa0/1
IPv4 CEF
Switch Path: IPv4 CEF            , Capture Buffer: pktrace1
Status : Inactive
Configuration:
monitor capture point ip cef ipceffa0/1 FastEthernet0/1 both
Router# show monitor capture buffer all
Capture buffer pktrace1 (circular buffer)
Buffer Size : 262144 bytes, Max Element Size : 256 bytes, Packets : 31
Allow-nth-pak : 0, Duration : 0 (seconds), Max packets : 0, pps : 0
Associated Capture Points:
Name : ipceffa0/1, Status : Active
Configuration:
monitor capture buffer pktrace1 size 256 max-size 256 circular
monitor capture point associate ipceffa0/1 pktrace1
```

Stopping Packet Data Capture Example

The following example shows how to stop capturing packet data:

```
Router> enable
Router# monitor capture point stop ipceffa0/1
Mar 21 11:14:20.152: %BUFCAP-6-DISABLE: Capture Point ipceffa0/1 disabled.
```

Exporting Packet Data Example

The following example shows how to export data for analysis through an external tool:

```
Router> enable
Router# monitor capture buffer pktrace1 export tftp://10.1.88.9/pktrace1
```

Monitoring and Maintaining Captured Data Example

The EPC feature provides the ability to dump packets in ASCII. The following example shows an IPv4 ICMP echo reply packet from one host to another:

```
<timestamp>: IPv4 packet received on Ethernet0/0 in the IPv4 CEF LES switch path
029E28E0: AABBCC01 2D00AABB CC013000 08004500  *;L.-.*;L.0...E.
029E28F0: 00640001 0000FE01 A8950A00 00020A00  .d....~.(.......
029E2900: 00010000 D5C80001 00000000 00000000  ....UH..........
029E2910: B080ABCD ABCDABCD ABCDABCD ABCDABCD  0.+M+M+M+M+M+M+M
029E2920: ABCDABCD ABCDABCD ABCDABCD ABCDABCD  +M+M+M+M+M+M+M+M
029E2930: ABCDABCD ABCDABCD ABCDABCD ABCDABCD  +M+M+M+M+M+M+M+M
029E2940: ABCDABCD ABCDABCD ABCDABCD ABCDABCD  +M+M+M+M+M+M+M+M
029E2950: ABCD
```

The following example shows how to view the contents of the capture buffer pktrace1. This output is displayed using the **show monitor capture buffer** *capture-buffer-name* **dump** command. This command supports two modes: the default mode and the dump mode. In the dump mode, the hexadecimal dump of the captured packet is also shown.

```
Router> enable
Router# show monitor capture buffer pktrace1 dump
```

```
11:13:00.593 EDT Mar 21 2007 : IPv4 Turbo      : Fa2/1 Fa0/1
65B6F500: 080020A2 44D90009 E94F8406 08004500  .. "DY..iO....E.
65B6F510: 00400F00 0000FE01 92AF5801 13025801  .@....~../X...X.
65B6F520: 58090800 4D1A1169 00000000 0005326C  X...M..i......2l
65B6F530: 01CCABCD ABCDABCD ABCDABCD ABCDABCD  .L+M+M+M+M+M+M+M
65B6F540: ABCDABCD ABCDABCD ABCDABCD ABCD00    +M+M+M+M+M+M+M.
11:13:20.593 EDT Mar 21 2007 : IPv4 Turbo      : Fa2/1 Fa0/1

65B6F500: 080020A2 44D90009 E94F8406 08004500  .. "DY..iO....E.
65B6F510: 00400F02 0000FE01 92AD5801 13025801  .@....~..-X...X.
65B6F520: 58090800 FEF91169 00000000 0005326C  X...~y.i......2l
65B6F530: 4FECABCD ABCDABCD ABCDABCD ABCDABCD  Ol+M+M+M+M+M+M+M
65B6F540: ABCDABCD ABCDABCD ABCDABCD ABCDFF    +M+M+M+M+M+M+M
```

The following example shows how to enable the packet capture infra debugs:

```
Router> enable
Router# debug packet-capture
Buffer Capture Infrastructure debugging is on
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Network Management commands (including EEM commands): complete command syntax, defaults, command mode, command history, usage guidelines, and examples.	*Cisco IOS Network Management Command Reference*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

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	12.2(33)SRE 12.4(20)T	Cisco IOS Embedded Packet Capture (EPC) is an onboard packet capture facility that allows network administrators to capture packets flowing to, through or from the device and to analyze them locally or save and export them for offline analysis using a tool like 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 on packet format. It also facilitates application analysis and security. This feature was introduced in Cisco IOS Release 12.4(20)T and integrated into Cisco IOS Release 12.2(33)SRE. **Note** In Cisco IOS Release 12.2(33)SRE, EPC is supported only on 7200 platform. The following commands were introduced or modified: **debug packet-capture** , **monitor capture buffer**, **monitor capture point**, **monitor capture point associate**, **monitor capture point disassociate**, **monitor capture point start**, **monitor capture point stop**, **show monitor capture.**