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

<table>
<thead>
<tr>
<th>System Resources</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>CPU utilization requirements are platform dependent.</td>
</tr>
<tr>
<td>Memory</td>
<td>The packet buffer is stored in DRAM. The size of the packet buffer is user specified.</td>
</tr>
<tr>
<td>Diskspace</td>
<td>Packets can be exported to external devices. No intermediate storage on flash disk is required.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | enable | Enables privileged EXEC mode.  
- Enter your password if prompted. |
| | Example: Router> enable | |
| **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 nth-packet | duration seconds | packet-count total-packets | packets-per-sec packets} | [max-size element-size] [size buffer-size] [circular| linear] | 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 element size of 100 bytes is defined. |
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>`nth-packet [duration seconds]</td>
<td><code>packet-count total-packets</code></td>
</tr>
<tr>
<td>`[packets-per-sec packets] [max-size element-size] [size buffer-size] [circular</td>
<td>linear]`</td>
</tr>
</tbody>
</table>

#### Step 3

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>`monitor capture point {ip</td>
<td>ipv6} {cef capture-point-name interface-name interface-type {both</td>
</tr>
<tr>
<td><code>Router# monitor capture point ip cef ipceffa0/1 fastEthernet 0/1 both</code></td>
<td>• 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.</td>
</tr>
<tr>
<td><code>Router# monitor capture point associate ipceffa0/1 pktrace1</code></td>
<td>• In this example, the capture point ipceffa0/1 is associated with the capture buffer pktrace1.</td>
</tr>
</tbody>
</table>

#### Step 4

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>`monitor capture point start {capture-point-name</td>
<td>all}`</td>
</tr>
<tr>
<td><code>Router# monitor capture point start ipceffa0/1</code></td>
<td>• In this example, the capture point ipceffa0/1 is enabled.</td>
</tr>
</tbody>
</table>

### 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

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

---

### 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**

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

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

**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:

```bash
Router> enable
```
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: AABBCD01 2D00ABB BCC013000 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
029E2920: ABCDABCD ABCDABCD ABCDABCD ABCDABCD +M+M+M+M+M+M+M
029E2930: ABCDABCD ABCDABCD ABCDABCD ABCDABCD +M+M+M+M+M+M+M
029E2940: ABCDABCD ABCDABCD ABCDABCD ABCDABCD +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
The following example shows how to enable the packet capture infra debugs:

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

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>--</td>
<td>No new or modified RFCs are supported, and support for existing RFCs has not been modified.</td>
</tr>
</tbody>
</table>

### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>

---

**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](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.
Table 2: Feature Information for Embedded Packet Capture

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded Packet Capture</td>
<td>12.2(33)SRE</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>12.4(20)T</td>
<td>Note: In Cisco IOS Release 12.2(33)SRE, EPC is supported only on 7200 platform. The following commands were introduced or modified: <code>debug packet-capture</code>, <code>monitor capture buffer</code>, <code>monitor capture point</code>, <code>monitor capture point associate</code>, <code>monitor capture point disassociate</code>, <code>monitor capture point start</code>, <code>monitor capture point stop</code>, <code>show monitor capture</code>.</td>
</tr>
</tbody>
</table>