Configure FED CPU Packet Capture on Catalyst 9000 Switches

Contents

Introduction

This document describes how to use the FED (Forwarding Engine Driver) CPU capture tool.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is restricted to Catalyst switching platforms that run Cisco IOS 16.X and above.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background information

The FED CPU packet capture tool helps identify data that traverses the control-plane and provides information on traffic **punted** (packets from ASIC to CPU) or **injected** (packets from CPU to ASIC).

- For example, this tool is helpful to identify traffic that triggered CoPP (control-plane policer) to kick in, causing valid traffic to be dropped in an effort to protect the CPU.

Terminology

- **Forwarding Engine Driver (FED)**: Responsible for taking commands from Cisco IOS-XE and programming hardware ASICs. Serves as a bridge between software and hardware components of a Catalyst switch.
- **Control Plane (CP):**Collection of functions and traffic that involve the CPU of the Catalyst Switch. This can include traffic such asSpanning Tree Protocol (STP),Hot Standby Router Protocol (HSRP), and routing protocols that are destined to the switch or sent from the switch.
- **Data Plane (DP):** Encompasses the ASIC(s) and traffic that is not software-switched, but hardware forwarded.
- **Punt:** Action of a packet sent up to the CPU from the data plane.
- **Inject:**Action of a packet sent down from the CPU towards the CPU.

Configure FED CPU Packet Capture

Use this table for configuration options

Definition	**Configuration**
Default setting of packet capture for punt or inject	debug platform software fed switch active <punt | inject> packet-capture <start | stop>
Display the captured packets	show platform software fed switch active <punt | inject> packet-capture <brief | detail>
Define your buffer size and type of capture	debug platform software fed switch active <punt | inject> packet-capture buffer [circular] limit <#packets>
Define capture filtering for displayed packets	show platform software fed switch active <punt | inject> packet-capture display-filter <filter> • Filters can be combined with logical && , || , and brackets. For example: “cdp || (ipv.src== 10.1.1.11 && tcp.port == 179) || stp” • In addition to standard network header based filtering, some platform specific filters have been added. They also can be mixed along with standard ones. For example, ARP packets received from physical interface id 0x44. • This is not Wireshark therefore it does not support all Wireshark filters. A display-filter-help command is available to check supported filters.
Display capture status	show platform software fed switch active <punt | inject> packet-capture status

Basic Configuration Example

This tool creates a buffer for the capture of up to 4096 (default setting) punted or injected packets since it has been enabled.

<#root>

```
Cat9k#

debug platform software fed switch active punt packet-capture start


Punt packet capturing started.
```

<#root>

```
Cat9k#

debug platform software fed switch active punt packet-capture stop

Punt packet capturing stopped. Captured 263 packet(s)
```

<#root>

```
Cat9k#

show platform software fed switch active punt packet-capture brief

Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 263 packets. Capture capacity : 4096 packets


------ Punt Packet Number: 1, Timestamp: 2020/04/10 18:15:53.499 ------

 interface : physical: GigabitEthernet1/0/1[if-id: 0x00000008], pal: Vlan20 [if-id: 0x00000076]
 metadata  : cause: 29 [RP handled ICMP], sub-cause: 0, q-no: 6, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: 084f.a940.fa56, src mac: 380e.4d77.4f66
 ether hdr : vlan: 20, ethertype: 0x8100
 ipv4  hdr : dest ip: 10.11.0.3, src ip: 10.11.0.3
 ipv4  hdr : packet len: 40, ttl: 255, protocol: 17 (UDP)
 udp   hdr : dest port: 3785, src port: 49152


------ Punt Packet Number: 2, Timestamp: 2020/04/10 18:15:53.574 ------

 interface : physical: GigabitEthernet1/0/1[if-id: 0x00000008], pal: Vlan20 [if-id: 0x00000076]
 metadata  : cause: 45 [BFD control], sub-cause: 0, q-no: 27, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: 084f.a940.fa56, src mac: 380e.4d77.4f66
 ether hdr : vlan: 20, ethertype: 0x8100
 ipv4  hdr : dest ip: 10.11.0.1, src ip: 10.11.0.1
 ipv4  hdr : packet len: 40, ttl: 254, protocol: 17 (UDP)
```

<#root>

```
Cat9k#

show platform software fed switch active punt packet-capture detailed

F340.04.11-9300-1#$e fed switch active punt packet-capture detailed
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 263 packets. Capture capacity : 4096 packets

------ Punt Packet Number: 1, Timestamp: 2020/04/10 18:15:53.499 ------
 interface : physical: GigabitEthernet1/0/1[if-id: 0x00000008], pal: Vlan20 [if-id: 0x00000076]
 metadata  : cause: 29 [RP handled ICMP], sub-cause: 0, q-no: 6, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: 084f.a940.fa56, src mac: 380e.4d77.4f66
 ether hdr : vlan: 20, ethertype: 0x8100
 ipv4  hdr : dest ip: 10.11.0.3, src ip: 10.11.0.3
 ipv4  hdr : packet len: 40, ttl: 255, protocol: 17 (UDP)
 udp   hdr : dest port: 3785, src port: 49152

 Packet Data Hex-Dump (length: 68 bytes) :
   084FA940FA56380E  4D774F668100C014  080045C00028CC8E  0000FF11DA5A0A0B
   00030A0B0003C000  0EC90014B6BE0000  0000000000010009  6618000000000000
   D54ADEEB

 Doppler Frame Descriptor :
   fdFormat                  = 0x4            systemTtl                 = 0xc
   loadBalHash1              = 0x10           loadBalHash2              = 0x2
   spanSessionMap            = 0              forwardingMode            = 0
   destModIndex              = 0x1            skipIdIndex               = 0x38
   srcGpn                    = 0x1            qosLabel                  = 0
   srcCos                    = 0x4            ingressTranslatedVlan     = 0x5
   bpdu                      = 0              spanHistory               = 0
   sgt                       = 0              fpeFirstHeaderType        = 0
   srcVlan                   = 0x14           rcpServiceId              = 0x3
   wccpSkip                  = 0              srcPortLeIndex            = 0
   cryptoProtocol            = 0              debugTagId                = 0
   vrfId                     = 0              saIndex                   = 0
   pendingAfdLabel           = 0              destClient                = 0xb
   appId                     = 0              finalStationIndex         = 0
   decryptSuccess            = 0              encryptSuccess            = 0
   rcpMiscResults            = 0              stackedFdPresent          = 0
   spanDirection             = 0              egressRedirect            = 0x1
   redirectIndex             = 0              exceptionLabel            = 0x20
   destGpn                   = 0x1            inlineFd                  = 0x1
   suppressRefPtrUpdate      = 0              suppressRewriteSideEfects = 0
   cmi2                      = 0x320          currentRi                 = 0x1
   currentDi                 = 0              dropIpUnreachable         = 0
   srcZoneId                 = 0              srcAsicId                 = 0
   originalDi                = 0x5338         originalRi                = 0
   srcL3IfIndex              = 0x2f           dstL3IfIndex              = 0x2f
   dstVlan                   = 0              frameLength               = 0x44
   fdCrc                     = 0x4c           tunnelSpokeId             = 0
   isPtp                     = 0              ieee1588TimeStampValid    = 0
   ieee1588TimeStamp55_48    = 0              lvxSourceRlocIpAddress    = 0
   sgtCachingNeeded          = 0

 Doppler Frame Descriptor Hex-Dump :
   0000010044004C02  8004424C00000100  0000000040000100  0000230514000000
   0000000000000030  0020000000000B00  380000532F000100  0000002F00000000
```

To validate the current status for the capture, you can use the next command.

<#root>

```
Cat9k#
```

```
show platform software fed switch active punt packet-capture status
```

```
Punt packet capturing: enabled. Buffer wrapping: enabled (wrapped 0 times)
Total captured so far: 110 packets. Capture capacity : 6000 packets
```

Modify the Packet Capture

The punt/inject FED packet capture tool is enhanced to allow packet buffer size & type configuration adjustment to create linear or circular packet captures.

<#root>

```
Cat9k#

debug platform software fed switch active punt packet-capture buffer ?

  circular  Circular capture
  limit     Number of packets to capture
```

Linear Packet Capture

The first buffer configuration option is to limit the number of packets (the default size is 4096 packets) that are sent the buffer. Once the buffer size limit is reached, no further packets are collected (no buffer wrapping).

<#root>

```
Cat9k#

debug platform software fed switch active punt packet-capture buffer limit ?

  <256-16384>  Number of packets to capture

Cat9k#

debug platform software fed switch active punt packet-capture buffer limit 5000

Punt PCAP buffer configure: one-time with buffer size 5000...done
```

Circular Packet Capture

The second buffer configuration option is to set a circular buffer for packets (the default buffer size is 4096 packets). Once the circular buffer size limit is reached, old data is replaced by new data in the buffer (buffer wrapping).

<#root>

```
Cat9k#

debug platform software fed switch active punt packet-capture buffer circular ?

  limit  Number of packets to capture

Cat9k#

debug platform software fed switch active punt packet-capture buffer circular limit ?

  <256-16384>  Number of packets to capture
```

```
Cat9k#

debug platform software fed switch active punt packet-capture buffer circular limit 6000

Punt PCAP buffer configure: circular with buffer size 6000...done
```

The packet capture can then be run again with the same parameters.

<#root>

```
Cat9k#

debug platform software fed switch active punt packet-capture start

Punt packet capturing started.

Cat9k#

show platform software fed switch active punt packet-capture status

Punt packet capturing: enabled. Buffer wrapping: enabled (wrapped 0 times)
Total captured so far: 110 packets. Capture capacity : 6000 packets

Cat9k#

debug platform software fed switch active punt packet-capture stop

Punt packet capturing stopped. Captured 426 packet(s)

Cat9k#

show platform software fed switch active punt packet-capture brief

Punt packet capturing: disabled. Buffer wrapping: enabled (wrapped 0 times)
Total captured so far: 426 packets. Capture capacity : 6000 packets

------ Punt Packet Number: 1, Timestamp: 2020/04/10 23:37:14.884 ------
 interface : physical: GigabitEthernet1/0/1[if-id: 0x00000008], pal: Vlan20 [if-id: 0x00000076]
 metadata  : cause: 29 [RP handled ICMP], sub-cause: 0, q-no: 6, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: 084f.a940.fa56, src mac: 380e.4d77.4f66
 ether hdr : vlan: 20, ethertype: 0x8100
 ipv4  hdr : dest ip: 10.11.0.3, src ip: 10.11.0.3
 ipv4  hdr : packet len: 40, ttl: 255, protocol: 17 (UDP)
 udp   hdr : dest port: 3785, src port: 49152

------ Punt Packet Number: 2, Timestamp: 2020/04/10 23:37:14.899 ------
 interface : physical: GigabitEthernet1/0/1[if-id: 0x00000008], pal: Vlan20 [if-id: 0x00000076]
 metadata  : cause: 45 [BFD control], sub-cause: 0, q-no: 27, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: 084f.a940.fa56, src mac: 380e.4d77.4f66
 ether hdr : vlan: 20, ethertype: 0x8100
 ipv4  hdr : dest ip: 10.11.0.1, src ip: 10.11.0.1
 ipv4  hdr : packet len: 40, ttl: 254, protocol: 17 (UDP)
 udp   hdr : dest port: 3785, src port: 49152
--snip--
```

Display and Capture Filtering

The Punt/Inject FED packet capture tool has been enhanced to allow packet display and filter options.

Display Filtering

Once a capture without a filter has been completed, it can be reviewed to display only the information in which you are interested in.

<#root>

```
Cat9k#

show platform software fed switch active punt packet-capture display-filter "ip.src== 10.11.0.0/24" brie

Punt packet capturing: disabled. Buffer wrapping: enabled (wrapped 0 times)
Total captured so far: 426 packets. Capture capacity : 6000 packets

------ Punt Packet Number: 2, Timestamp: 2020/04/10 23:37:14.899 ------
 interface : physical: GigabitEthernet1/0/1[if-id: 0x00000008], pal: Vlan20 [if-id: 0x00000076]
 metadata  : cause: 45 [BFD control], sub-cause: 0, q-no: 27, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: 084f.a940.fa56, src mac: 380e.4d77.4f66
 ether hdr : vlan: 20, ethertype: 0x8100
 ipv4  hdr : dest ip: 10.11.0.1, src ip: 10.11.0.1
 ipv4  hdr : packet len: 40, ttl: 254, protocol: 17 (UDP)
 udp   hdr : dest port: 3785, src port: 49152

------ Punt Packet Number: 4, Timestamp: 2020/04/10 23:37:15.023 ------
 interface : physical: GigabitEthernet1/0/1[if-id: 0x00000008], pal: Vlan20 [if-id: 0x00000076]
 metadata  : cause: 29 [RP handled ICMP], sub-cause: 0, q-no: 6, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: 084f.a940.fa56, src mac: 380e.4d77.4f66
 ether hdr : vlan: 20, ethertype: 0x8100
 ipv4  hdr : dest ip: 10.11.0.3, src ip: 10.11.0.3
 ipv4  hdr : packet len: 40, ttl: 255, protocol: 17 (UDP)
 udp   hdr : dest port: 3785, src port: 49152
```

Since this is not Wireshark, not all Wireshark filters are supported. Use the display-filter-help command to see the different available options for filtering.

<#root>

```
Cat9k#

show platform software fed switch active punt packet-capture display-filter-help

FED Punject specific filters :
   1. fed.cause              FED punt or inject cause
   2. fed.linktype           FED linktype
   3. fed.pal_if_id          FED platform interface ID
   4. fed.phy_if_id          FED physical interface ID
   5. fed.queue              FED Doppler hardware queue
   6. fed.subcause           FED punt or inject sub cause
Generic filters supported :
   7. arp                    Is this an ARP packet
   8. bootp                  DHCP packets [Macro]
   9. cdp                    Is this a CDP packet
  10. eth                    Does the packet have an Ethernet header
  11. eth.addr               Ethernet source or destination MAC address
  12. eth.dst                Ethernet destination MAC address
  13. eth.ig                 IG bit of ethernet destination address (broadcast/multicast)
  14. eth.src                Ethernet source MAC address
  15. eth.type               Ethernet type
  16. gre                    Is this a GRE packet
  17. icmp                   Is this a ICMP packet
  18. icmp.code              ICMP code
  19. icmp.type              ICMP type
  20. icmpv6                 Is this a ICMPv6 packet
  21. icmpv6.code            ICMPv6 code
  22. icmpv6.type            ICMPv6 type
  23. ip                     Does the packet have an IPv4 header
  24. ip.addr                IPv4 source or destination IP address
  25. ip.dst                 IPv4 destination IP address
  26. ip.flags.df            IPv4 dont fragment flag
  27. ip.flags.mf            IPv4 more fragments flag
  28. ip.frag_offset         IPv4 fragment offset
  29. ip.proto               Protocol used in datagram
  30. ip.src                 IPv4 source IP address
  31. ip.ttl                 IPv4 time to live
  32. ipv6                   Does the packet have an IPv4 header
  33. ipv6.addr              IPv6 source or destination IP address
  34. ipv6.dst               IPv6 destination IP address
  35. ipv6.hlim              IPv6 hot limit
  36. ipv6.nxt               IPv6 next header
  37. ipv6.plen              IPv6 payload length
  38. ipv6.src               IPv6 source IP address
  39. stp                    Is this a STP packet
  40. tcp                    Does the packet have a TCP header
  41. tcp.dstport            TCP destination port
  42. tcp.port               TCP source OR destination port
  43. tcp.srcport            TCP source port
  44. udp                    Does the packet have a UDP header
  45. udp.dstport            UDP destination port
  46. udp.port               UDP source OR destination port
  47. udp.srcport            UDP source port
  48. vlan.id                Vlan ID (dot1q or qinq only)
  49. vxlan                  Is this a VXLAN packet
```

Capture Filtering

Before the start of the packet capture, you can define a filter to help capture specific traffic only.

<#root>

```
C9300#
```

```
debug platform software fed switch active punt packet-capture set-filter "ip.src== 10.1.1.0/24 && tcp.po
```

```
Filter setup successful. Captured packets will be cleared
```

```
C9300#
```

```
show platform software fed switch active punt packet-capture status
```

```
Punt packet capturing: disabled. Buffer wrapping: enabled (wrapped 0 times)
Total captured so far: 0 packets. Capture capacity : 6000 packets
Capture filter : "ip.src== 10.1.1.0/24 && tcp.port == 179"
```

```
C9300#
```

```
debug platform software fed switch active punt packet-capture clear-filter
```

```
Filter cleared. Captured packets will be cleared
```

```
C9300#
```

show platform software fed switch active punt packet-capture status

```
Punt packet capturing: disabled. Buffer wrapping: enabled (wrapped 0 times)
Total captured so far: 0 packets. Capture capacity : 6000 packets
```

Sort by Top Talker (17.6.X)

From 17.6.1 onward, you can sort the packets captured by top talkers based on a specified field.

<#root>

Switch#

show platform software fed switch active punt packet-capture cpu-top-talker ?

```
  cause-code          occurences of cause-code
  dst_ipv4            occurrences on dst_ipv4
  dst_ipv6            occurrences on dst_ipv4
  dst_l4              occurences of L4 destination
  dst_mac             Occurrences of dst_mac
  eth_type            Occurrences of eth_type
  incoming-interface  occurences of incoming-interface
  ipv6_hoplt          occurences of hoplt
  protocol            occurences of layer4 protocol
  src_dst_port        occurences of layer4 src_dst_port
  src_ipv4            occurrences on src_ipv4
  src_ipv6            occurrences on src_ipv6
  src_l4              occurences of L4 source
  src_mac             Occurrences of src_mac
  summary             occurences of all in summary
  ttl                 occurrences on ttl
  vlan                Occurrences of vlan
```

```
Switch#
```

show platform software fed switch active punt packet-capture cpu-top-talker dst_mac

```
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 224 packets. Capture capacity : 4096 packets
Sr.no.  Value/Key       Occurrence
1       01:80:c2:00:00:00       203
2       01:00:0c:cc:cc:cc       21
```

```
Switch#
```

show platform software fed switch active punt packet-capture cpu-top-talker summary

```
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 224 packets. Capture capacity : 4096 packets

L2 Top Talkers:
224     Source mac      00:27:90:be:20:84
203     Dest mac        01:80:c2:00:00:00

L3 Top Talkers:
```

```
L4 Top Talkers:

Internal Top Talkers:
224      Interface        FortyGigabitEthernet2/1/2
224      CPU Queue        Layer2 control protocols
```

Related Information

For further details about CPU Troubleshooting in Cat9K platforms:

Troubleshoot High CPU Usage in Catalyst Switch Platforms Running Cisco IOS-XE 16.x

Additional Reading

- Cisco IOS-XE 16 - At a Glance
- Catalyst 3850 Series Switch High CPU Usage Troubleshoot
- Embedded Packet Capture for Cisco IOS and Cisco IOS-XE Configuration Example
- Technical Support & Documentation - Cisco Systems