

# 在Catalyst 9000交换机上配置FED CPU数据包捕获

## 目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[配置FED CPU数据包捕获](#)

[基本配置示例](#)

[修改数据包捕获](#)

[线性数据包捕获](#)

[循环数据包捕获](#)

[显示和捕获过滤](#)

[显示过滤](#)

[捕获过滤](#)

[按最大流量生成者\(17.6.X\)排序](#)

[相关信息](#)

## 简介

本文档介绍如何使用FED（转发引擎驱动程序）CPU捕获工具。

## 先决条件

### 要求

本文档没有任何特定的要求。

### 使用的组件

本文档仅限于运行Cisco IOS 16.X及更高版本的Catalyst交换平台。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

## 背景信息

FED CPU数据包捕获工具帮助识别流经控制平面的数据，并提供有关**传送**（从ASIC到CPU的数据包）或**注入**（从CPU到ASIC的数据包）的信息。

- 例如，此工具有助于识别触发CoPP（控制平面监视器）启动并导致有效流量被丢弃以保护

CPU的流量。

## 术语

- **转发引擎驱动程序(FED)**：负责从Cisco IOS-XE获取命令并对硬件ASIC进行编程。充当Catalyst交换机软件和硬件组件之间的桥梁。
- **控制平面(CP)**:涉及Catalyst交换机CPU的功能和流量的集合。这可能包括发往交换机或从交换机发送的流量，例如生成树协议(STP)、热备份路由器协议(HSRP)和路由协议。
- **数据平面(DP)**:包括ASIC和不是软件交换而是硬件转发的流量。
- **传送**：从数据平面发送到CPU的数据包的操作。
- **注入**：从CPU向CPU发送的数据包的操作。

## 配置FED CPU数据包捕获

使用此表作为配置选项

### 定义

用于传送或插入的数据包捕获的默认设置

显示捕获的数据包

定义您的缓冲区大小和捕获类型

为显示的数据包定义捕获过滤

显示捕获状态

### 配置

```
debug platform software fed switch active <punt | inject>数据包捕获  
<start | stop>
```

```
show platform software fed switch active <punt | inject> packet-cap  
<brief | detail>
```

```
debug platform software fed switch active <punt | inject> packet-ca  
buffer [circular] limit <#packets>
```

```
show platform software fed switch active <punt | inject> packet-cap  
display-filter <filter>
```

- 过滤器可以与逻辑&、||，和方括号。例如：“cdp ||(ipv.src== 10.1.1.11 && tcp.port == 179) || stp”

- 除了基于标准网络报头的过滤外，还添加了某些平台特定的过滤器。它们也可以与标准值混合使用。例如，从物理接口id 0x44收到的ARP数据包。

- 这不是Wireshark，因此它不支持所有Wireshark过滤器。可使用display-filter-help命令检查支持的过滤器。

```
show platform software fed switch active <punt | inject> packet-cap  
status
```

## 基本配置示例

此工具会创建一个缓冲区，用于捕获最多达4096个（默认设置）传送或注入的数据包，因为它已启用。

```
Cat9k#debug platform software fed switch active punt packet-capture start  
Punt packet capturing started.
```

```
Cat9k#debug platform software fed switch active punt packet-capture stop  
Punt packet capturing stopped. Captured 263 packet(s)
```

```
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)

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 suppressRewriteSideEffects = 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
00000000000000030 00200000000000B00 380000532F000100 0000002F00000000

```

要验证捕获的当前状态，可使用下一命令。

```

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

```

## 修改数据包捕获

传送/注入FED分组捕获工具得到增强，以允许分组缓冲器大小和类型配置调整以创建线性或循环分组捕获。

```

Cat9k#debug platform software fed switch active punt packet-capture buffer ?
circular      Circular capture
limit         Number of packets to capture

```

### 线性数据包捕获

第一个缓冲区配置选项是限制发送到缓冲区的数据包数量（默认大小为4096个数据包）。一旦达到缓冲区大小限制，就不会收集更多数据包（无缓冲区包装）。

```

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

```

### 循环数据包捕获

第二个缓冲区配置选项是为数据包设置循环缓冲区（默认缓冲区大小为4096个数据包）。一旦达到循环缓冲区大小限制，旧数据就会被缓冲区中的新数据取代（缓冲区包装）。

```

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

```

然后，可以使用相同的参数再次运行数据包捕获。

```

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--
```

## 显示和捕获过滤

Punt/Inject FED数据包捕获工具已增强以允许数据包显示和过滤器选项。

### 显示过滤

完成不带过滤器的捕获后，可以对其进行查看，以便仅显示您感兴趣的信息。

```
Cat9k#show platform software fed switch active punt packet-capture display-filter "ip.src==  
10.11.0.0/24" brief
```

```
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
```

由于这不是Wireshark，因此并非所有Wireshark过滤器都受支持。使用display-filter-help命令查看过滤的不同可用选项。

```
Cat9k#show platform software fed switch active punt packet-capture display-filter-help
```

```
FED Punct 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

## 捕获过滤

在开始捕获数据包之前，可以定义一个过滤器，以仅帮助捕获特定流量。

```
C9300#debug platform software fed switch active punt packet-capture set-filter "ip.src==
10.1.1.0/24 && tcp.port == 179"
```

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

## 按最大流量生成者(17.6.X)排序

从17.6.1开始，您可以根据指定字段对最大流量生成者捕获的数据包进行排序。

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

cause-code	occurences of cause-code
dst_ipv4	occurences on dst_ipv4
dst_ipv6	occurences on dst_ipv6
dst_l4	occurences of L4 destination
dst_mac	Occurences of dst_mac
eth_type	Occurences 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	occurences on src_ipv4
src_ipv6	occurences on src_ipv6
src_l4	occurences of L4 source
src_mac	Occurences of src_mac
summary	occurences of all in summary
ttl	occurences on ttl
vlan	Occurences 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

## 相关信息

有关Cat9K平台中CPU故障排除的更多详细信息：

[对运行Cisco IOS-XE 16.x的Catalyst交换机平台的CPU使用率过高进行故障排除](#)

### 附加阅读

- [Cisco IOS-XE 16 — 概览](#)
- [排除 Catalyst 3850 系列交换机高 CPU 使用率问题](#)
- [Cisco IOS和Cisco IOS-XE的嵌入式数据包捕获配置示例](#)
- [技术支持和文档 - Cisco Systems](#)



## 关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。