

Catalyst 9000スイッチでのFED CPUパケットキャプチャの設定

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[FED CPUパケットキャプチャの設定](#)

[基本設定の例](#)

[パケットキャプチャの変更](#)

[リニアパケットキャプチャ](#)

[循環パケットキャプチャ](#)

[表示およびキャプチャフィルタリング](#)

[表示のフィルタリング](#)

[キャプチャフィルタリング](#)

[トップトーカー\(17.6.X\)でソート](#)

[関連情報](#)

概要

このドキュメントでは、FED(フォワーディングエンジンドライバ)CPUキャプチャツールの使用方法について説明します。

前提条件

要件

このドキュメントに特有の要件はありません。

使用するコンポーネント

このドキュメントは、Cisco IOS 16.X以降が稼働するCatalystスイッチングプラットフォームに限定されています。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期(デフォルト)設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

FED CPUパケットキャプチャツールは、コントロールプレーンを通過するデータの識別に役立ち、パント (ASICからCPUへのパケット) または挿入 (CPUからASICへのパケット) のトラフィックに関する情報を提供します。

- たとえば、このツールは、CoPP (コントロールプレーンポリサー) をキックインさせるトリガーとなったトラフィックを特定し、CPUを保護するために有効なトラフィックをドロップするのに役立ちます。

用語

- **Forwarding Engine Driver(FED):**Cisco IOS-XEからコマンドを取得し、ハードウェアASICをプログラミングします。Catalystスイッチのソフトウェアコンポーネントとハードウェアコンポーネントの間のブリッジとして機能します。
- **コントロールプレーン(CP):**CatalystスイッチのCPUに関係する機能とトラフィックの集合。これには、スパニングツリープロトコル(STP)、ホットスタンバイルータプロトコル(HSRP)、およびスイッチ宛てまたはスイッチから送信されるルーティングプロトコルなどのトラフィックが含まれます。
- **データプレーン(DP):**ソフトウェアでスイッチングされず、ハードウェアで転送されるASICとトラフィックを含みます。
- **パント:**データプレーンからCPUに送信されるパケットの動作。
- **インジェクト:**CPUからCPUに向けて送信されるパケットのアクション。

FED CPUパケットキャプチャの設定

次の表を設定オプションに使用します

定義

パントまたはインジェクトのパケットキャプチャのデフォルト設定
キャプチャされたパケットを表示する
バッファサイズとキャプチャのタイプを定義します。

表示されたパケットのキャプチャフィルタリングを定義する

キャプチャステータスの表示

コンフィギュレーション

```
debug platform software fed switch active <punt | inject> packet-ca  
<start |停止>  
show platform software fed switch active <punt | inject> packet-cap  
<brief |詳細>  
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>
```

- フィルタは、論理&&、||、および角カッコで囲みます。例：
|| (ipv.src== 10.1.1.11 && tcp.port == 179) || stp"
- 標準のネットワークヘッダーベースのフィルタリングに加えて、一部のプラットフォーム固有のフィルタが追加されています。極端なものと一緒に混ぜることもできます。たとえば、物理インターフェイスID 0x44から受信したARPパケットです。

- これはWiresharkではないため、すべてのWiresharkフィルタをサポートしているわけではありません。サポートされているフィルタを確認するには、display-filter-helpコマンドを使用できます。

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

基本設定の例

このツールは、有効になってから最大4096 (デフォルト設定) のパントまたはインジェクトされたパケットをキャプチャするためのバッファを作成します。

```
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
```

キャプチャの現在のステータスを検証するには、次のコマンドを使用します。

```
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
```

パケットキャプチャの変更

パント/インジェクトFEDパケットキャプチャツールが拡張され、パケットバッファのサイズとタイプの設定を調整して、線形または円形のパケットキャプチャを作成できるようになりました。

```
Cat9k#debug platform software fed switch active punt packet-capture buffer ?
```

```
  circular  Circular capture
```

```
  limit     Number of packets to capture
```

リニアパケットキャプチャ

最初のバッファ設定オプションは、バッファに送信されるパケットの数を制限することです (デフォルトサイズは4096パケット)。バッファサイズの制限に達すると、それ以上のパケットは収集されません (バッファラッピングは行われません)。

```
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
```

循環パケットキャプチャ

2番目のバッファ設定オプションは、パケットの循環バッファを設定することです (デフォルトのバッファサイズは4096パケット)。循環バッファサイズの制限に達すると、古いデータがバッファ内の新しいデータに置き換えられます (バッファラッピング)。

```
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
```

パケットキャプチャは、同じパラメータを使用して再実行できます。

```
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--
```

表示およびキャプチャフィルタリング

パント/インジェクトFEDパケットキャプチャツールが拡張され、パケットの表示とフィルタオプションが可能になりました。

表示のフィルタリング

フィルタを使用しないキャプチャが完了すると、目的の情報のみが表示されるように確認できます。

```
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
```

これはWiresharkではないので、すべてのWiresharkフィルタがサポートされているわけではありません。display-filter-helpコマンドを使用して、フィルタリングに使用できるさまざまなオプションを表示します。

```
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 hop 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

キャプチャフィルタリング

パケットキャプチャを開始する前に、特定のトラフィックだけをキャプチャするフィルタを定義できます。

```
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
```

トップトーカー(17.6.X)でソート

17.6.1以降では、指定されたフィールドに基づいて、トップトーカーによってキャプチャされたパケットをソートできます。

```
Switch#show platform software fed switch active punt packet-capture cpu-top-talker ?
cause-code      occurrences of cause-code
dst_ipv4        occurrences on dst_ipv4
dst_ipv6        occurrences on dst_ipv6
dst_l4          occurrences of L4 destination
dst_mac         Occurrences of dst_mac
eth_type        Occurrences of eth_type
incoming-interface occurrences of incoming-interface
ipv6_hoplt     occurrences of hoplt
protocol        occurrences of layer4 protocol
src_dst_port    occurrences of layer4 src_dst_port
src_ipv4        occurrences on src_ipv4
src_ipv6        occurrences on src_ipv6
src_l4          occurrences of L4 source
src_mac         Occurrences of src_mac
summary        occurrences 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
```

関連情報

Cat9KプラットフォームにおけるCPUのトラブルシューティングの詳細については、次を参照してください。

[Cisco IOS-XE 16.xを実行するCatalystスイッチプラットフォームでCPU使用率が高い場合のトラブルシューティング](#)

追加の資料

- [Cisco IOS-XE 16 – 概要](#)
- [Catalyst 3850 シリーズ スイッチでの CPU 高使用率のトラブルシューティング](#)
- [Cisco IOSおよびCisco IOS-XEの組み込みパケットキャプチャの設定例](#)
- [テクニカル サポートとドキュメント – Cisco Systems](#)

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。