操作Catalyst 9000交換機上的DHCP監聽並排除故障

目錄

簡介

本檔案介紹如何在Catalyst 9000系列交換器上執行DHCP窺探和疑難排解

必要條件

需求

思科建議您瞭解以下主題：

- Catalyst 9000系列交換器架構
- Cisco IOS® XE軟體架構

採用元件

本文中的資訊係根據以下軟體和硬體版本：

- C9200
- C9300
- C9400
- C9500
- C9600

Cisco IOS® XE 16.12.X

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

注意：有關在其他思科平台上啟用這些功能的命令，請參閱相應的配置指南。

背景資訊

DHCP窺探

動態主機配置協定(DHCP)監聽是一種安全功能，用於檢查DHCP流量以阻止任何惡意DHCP資料包。它充當網路上不受信任的使用者埠和DHCP伺服器埠之間的防火牆，以防止網路中的惡意DHCP伺服器，因為這會導致拒絕服務。

DHCP窺探操作

DHCP監聽使用可信和不可信介面的概念。通過DHCP流量的路徑，交換機驗證介面上接收到的DHCP資料包，並通過可信介面跟蹤預期的DHCP伺服器資料包（OFFER和ACK）。換句話說，不受信任的介面會阻止DHCP伺服器資料包。

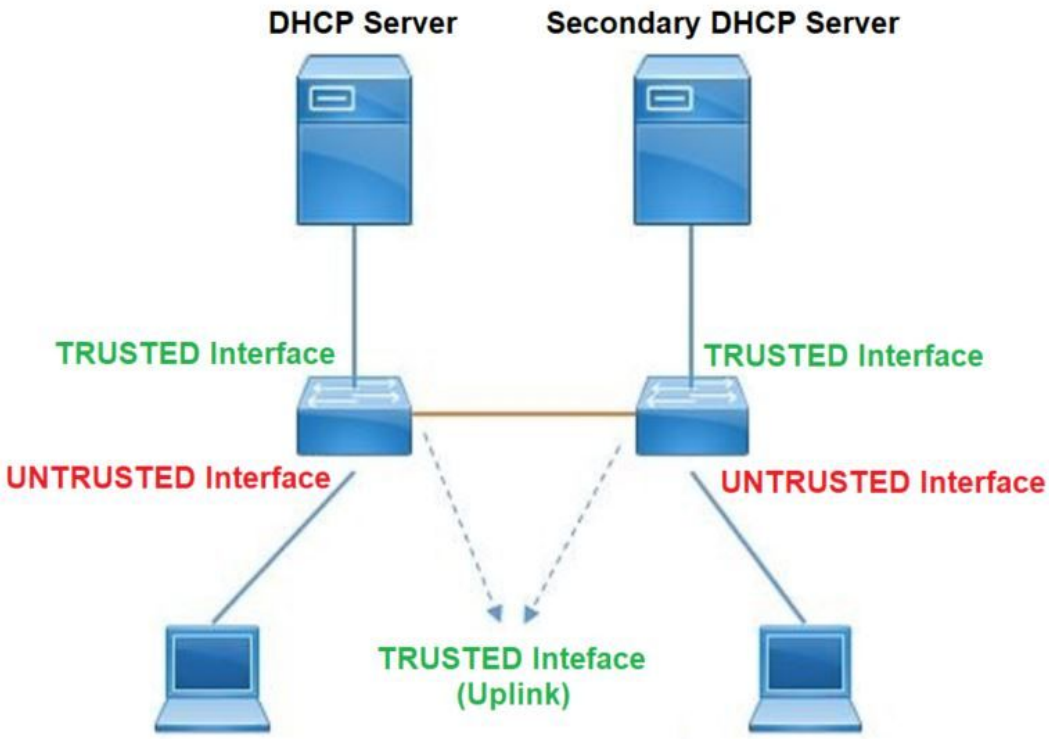
DHCP資料包在不受信任的介面上被阻止。

- 從DHCP伺服器（例如DHCPOFFER、DHCPACK、DHCPNAK或DHCPLEASEQUERY資料包）收到的資料包來自網路或防火牆外部。這可防止非法DHCP伺服器在不受信任的埠上攻擊網路。
- 不可信介面上收到的資料包與源MAC地址和DHCP客戶端硬體地址不匹配。這樣可以防止欺詐客戶端偽裝DHCP資料包，從而在DHCP伺服器上造成拒絕服務攻擊。
- 在DHCP監聽繫結資料庫中具有MAC地址的DHCPRELEASE或DHCPDECLINE廣播消息，但繫結資料庫中的介面資訊與接收消息的介面不匹配。這可以防止對客戶端的拒絕服務攻擊。
- 由DHCP中繼代理轉發的DHCP資料包包含非0.0.0.0的中繼代理IP地址，或者中繼代理將包含選項82資訊的資料包轉發到不可信埠。這樣可以防止網路中的中繼代理資訊被欺騙。

配置DHCP監聽的交換機構建DHCP監聽表或DHCP繫結資料庫。此表用於跟蹤從合法DHCP伺服器分配的IP地址。 繫結資料庫也用於其他IOS安全功能，如動態ARP檢測和IP源保護。

注意：要允許DHCP監聽正常工作，請確保您信任所有上行鏈路埠以到達DHCP伺服器，並取消信任終端使用者埠。

拓撲



設定

全域性配置

<#root>

1. Enable DHCP snooping globally on the switch
    switch(config)#

ip dhcp snooping

2. Designate ports that forward traffic toward the DHCP server as trusted
    switch(config-if)#

ip dhcp snooping trust

(Additional verification)

- List uplink ports according to the topology, ensure all the uplink ports toward the DHCP server a

trusted

- List the port where the Legitimate DHCP Server is connected (include any Secondary DHCP Server)
- Ensure that no other port is configured as trusted

3. Configure DHCP rate limiting on each untrusted port (Optional)
    switch(config-if)#

ip dhcp snooping limit rate 10 << ----- 10 packets per second (pps)

4. Enable DHCP snooping in specific VLAN
    switch(config)#

ip dhcp snooping vlan 10

<< ----- Allow the switch to snoop the traffic for that specific VLAN

5. Enable the insertion and removal of option-82 information DHCP packets
    switch(config)#

ip dhcp snooping information option

<-- Enable insertion of option 82

    switch(config)#

no ip dhcp snooping information option

<-- Disable insertion of option 82

Example

Legitimate DHCP Server Interface and Secondary DHCP Server, if available

Server Interface

interface FortyGigabitEthernet1/0/5
switchport mode access
switchport mode access vlan 11

ip dhcp snooping trust

end

Uplink interface

interface FortyGigabitEthernet1/0/10
switchport mode trunk

ip dhcp snooping trust

```
end

User Interface


 << ----- All interfaces are UNTRUSTED by default


interface FortyGigabitEthernet1/0/2
 switchport access vlan 10
 switchport mode access

ip dhcp snooping limit rate 10


<< ----- Optional

end
```

注意：要允許option-82資料包，必須啟用ip dhcp snooping information option allow-untrusted。

驗證

確認是否在所需的VLAN上啟用了DHCP監聽，並確保已列出受信任和不受信任的介面。如果配置了速率，請確保也列出了該速率。

```
<#root>

switch#show ip dhcp snooping


Switch DHCP snooping is

enabled


Switch DHCP gleaning is disabled
DHCP snooping is configured on following VLANs:

10-11


DHCP

snooping is operational on following VLANs

:

<<---- Configured and operational on Vlan 10 & 11


10-11
```

```
DHCP snooping is configured on the following L3 Interfaces:

Insertion of option 82 is disabled


<<---- Option 82 can not be added to DHCP packet


   circuit-id default format: vlan-mod-port
   remote-id: 00a3.d144.1a80 (MAC)
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Verification of giaddr field is enabled
DHCP snooping trust/rate is configured on the following Interfaces:

Interface

  Trusted

     Allow option    Rate limit (pps)
----------------------        -------     ------------    ----------------
FortyGigabitEthernet1/0/2

no

          no              10

<<--- Trust is NOT set on this interface


Custom circuit-ids:
FortyGigabitEthernet1/0/10

 yes

         yes             unlimited

<<--- Trust is set on this interface


Custom circuit-ids:
```

使用者通過DHCP收到IP後，會在此輸出中列出。

- DHCP監聽在IP位址租用到期或交換器從主機收到DHCPRELEASE訊息時移除資料庫中的專案。
- 確保所列的終端使用者MAC地址資訊正確。

<#root>

```
c9500#show ip dhcp snooping binding


MacAddress         IpAddress       Lease(sec) Type          VLAN Interface
------------------ --------------- ---------- ------------- ---- --------------------
00:A3:D1:44:20:46  10.0.0.3

85556
```

```
 dhcp-snooping 10   FortyGigabitEthernet1/0/2
Total number of bindings: 1
```

此表列出了可用於監視DHCP監聽資訊的各種命令。

指令	目的
show ip dhcp snooping binding show ip dhcp snooping binding [IP-address] [MAC-address] [interface ethernet slot/port] [vlan-id]	僅顯示DHCP監聽繫結資料庫（也稱為繫結表）中動態配置的繫結。 — 繫結條目IP地址 — 繫結條目Mac地址 — 繫結條目輸入介面 — 繫結條目VLAN
show ip dhcp snooping database	顯示DHCP監聽繫結資料庫狀態和統計資訊。
show ip dhcp snooping statistics	以摘要或詳細資訊形式顯示DHCP監聽統計資訊。
show ip source binding	顯示動態和靜態配置的繫結。
show interface vlan xyz show buffer input-interface Vlan xyz dump	DHCP資料包通過客戶端VLAN SVI傳送到客戶端VLAN中配置的中繼代理。如果輸入隊列顯示丟棄或達到最大限制，則可能是來自客戶端的DHCP資料包被丟棄，無法到達配置的中繼代理。 注意：確保輸入隊列中不會出現丟棄。 switch#show int vlan 670 5秒負載：13%/0%;1分鐘：10%;5分鐘：10% 時間來源為NTP，18:39:52.476 UTC Thu Sep 2020 Vlan670為up，線路協定為up，自動狀態已啟用 硬體為乙太網SVI，地址為00fd.227a.5920(bia 00fd.227a.5920) 說明：ion_media_client Internet地址為10.27.49.254/23 MTU 1500位元組，BW 1000000 Kbit/sec，DLY 10 usec， 可靠性255/255、txload 1/255、rxload 1/255

	封裝ARPA，未設定環回 不支援Keepalive ARP型別：ARPA，ARP超時04:00:00 上次輸入03:01:29，輸出00:00:02，輸出永不掛起 上次清除「show interface」計數器的時間從不 輸入佇列：375/375/4020251/0(size/max/drops/flushes)；總輸出捨棄次數：0 < — 輸入佇列中的375個封包/4020251已捨棄

疑難排解

軟體疑難排解

確認交換器收到什麼。這些資料包在CPU控制平面處理，因此請確保您看到所有資料包的注入和點入方向，並確認資訊是否正確。

⚠ 注意：請謹慎使用debug命令。請注意，許多debug命令都會影響實際網路，只有重現問題時才建議在實驗室環境中使用。

條件調試功能允許您根據您定義的一組條件為特定功能選擇性地啟用調試和日誌。這對於僅包含特定主機或流量的調試資訊非常有用。

條件是指功能或身份，其中身份可以是介面、IP地址或MAC地址等。

如何為資料包和事件調試啟用條件調試，以排除DHCP監聽故障。

指令	目的
debug condition mac <mac-address> 範例： switch#debug condition mac bc16.6509.3314	為指定的MAC地址配置條件調試。
debug condition vlan <VLAN Id> 範例： switch#debug condition vlan 10	為指定的VLAN配置條件調試。
debug condition interface <interface> 範例：	為指定的介面配置條件調試。

switch#debug condition interface twentyFiveGigE 1/0/8	

要調試DHCP監聽，請使用表中顯示的命令。

指令	目的
debug dhcp [detail | oper | 冗餘]	detail DHCP packet content oper DHCP內部OPER 冗餘DHCP客戶端冗餘支援
debug ip dhcp server packet detail	詳細解碼消息接收和傳輸
debug ip dhcp server events	報告地址分配、租約到期等。
debug ip dhcp snooping agent	Debug dhcp snooping database read and write
debug ip dhcp snooping event	每個元件之間的調試事件
debug ip dhcp snooping packet	在dhcp監聽模組中調試DHCP資料包

以下是debug ip dhcp snooping 指令的部分輸出範例。

<#root>

Apr 14 16:16:46.835: DHCP_SNOOPING: process new DHCP packet,

message type: DHCPDISCOVER, input interface: Fo1/0/2

, MAC da: ffff.ffff.ffff, MAC

sa: 00a3.d144.2046,

IP da: 255.255.255.255, IP sa: 0.0.0.0, DHCP ciaddr: 0.0.0.0, DHCP yiaddr: 0.0.0.0, DHCP siaddr: 0.0.0
Apr 14 16:16:46.835: DHCP_SNOOPING: bridge packet get invalid mat entry: FFFF.FFFF.FFFF, packet is floo

Apr 14 16:16:48.837: DHCP_SNOOPING:

received new DHCP packet from input interface (FortyGigabitEthernet1/0/10)

Apr 14 16:16:48.837: DHCP_SNOOPING:

process new DHCP packet, message type: DHCPOFFER, input interface: Fo1/0/10,

```
MAC da: ffff.ffff.ffff, MAC

sa: 701f.539a.fe46,

 IP da: 255.255.255.255, IP sa: 10.0.0.1, DHCP ciaddr: 0.0.0.0, DHCP yiaddr: 10.0.0.5, DHCP siaddr: 0.0
Apr 14 16:16:48.837: platform lookup dest vlan for input_if: FortyGigabitEthernet1/0/10, is NOT tunnel,
Apr 14 16:16:48.837: DHCP_SNOOPING: direct forward dhcp replyto output port: FortyGigabitEthernet1/0/2.
Apr 14 16:16:48.838: DHCP_SNOOPING: received new DHCP packet from input interface (FortyGigabitEthernet
Apr 14 16:16:48.838: Performing rate limit check

Apr 14 16:16:48.838: DHCP_SNOOPING: process new DHCP packet,

message type: DHCPREQUEST, input interface: Fo1/0/2,

 MAC da: ffff.ffff.ffff, MAC

sa: 00a3.d144.2046,

 IP da: 255.255.255.255, IP sa: 0.0.0.0, DHCP ciaddr: 0.0.0.0, DHCP yiaddr: 0.0.0.0, DHCP siaddr: 0.0.0
Apr 14 16:16:48.838: DHCP_SNOOPING: bridge packet get invalid mat entry: FFFF.FFFF.FFFF, packet is floo
Apr 14 16:16:48.839: DHCP_SNOOPING: received new DHCP packet from input interface (FortyGigabitEthernet

Apr 14 16:16:48.840: DHCP_SNOOPING: process new DHCP packet,

message type: DHCPACK, input interface: Fo1/0/10,

 MAC da: ffff.ffff.ffff, MAC

sa: 701f.539a.fe46,

 IP da: 255.255.255.255, IP

sa: 10.0.0.1,

 DHCP ciaddr: 0.0.0.0, DHCP yiaddr: 10.0.0.5, DHCP siaddr: 0.0.0.0, DHCP giaddr: 0.0.0.0, DHCP chaddr: 0
Apr 14 16:16:48.840: DHCP_SNOOPING: add binding on port FortyGigabitEthernet1/0/2 ckt_id 0 FortyGigabit
Apr 14 16:16:48.840: DHCP_SNOOPING: added entry to table (index 331)

Apr 14 16:16:48.840:

DHCP_SNOOPING: dump binding entry: Mac=00:A3:D1:44:20:46 Ip=10.0.0.5

 Lease=86400 Type=dhcp-snooping

Vlan=10 If=FortyGigabitEthernet1/0/2


Apr 14 16:16:48.840: No entry found for mac(00a3.d144.2046) vlan(10) FortyGigabitEthernet1/0/2
Apr 14 16:16:48.840: host tracking not found for update add dynamic (10.0.0.5, 0.0.0.0, 00a3.d144.2046)
Apr 14 16:16:48.840: platform lookup dest vlan for input_if: FortyGigabitEthernet1/0/10, is NOT tunnel,
Apr 14 16:16:48.840: DHCP_SNOOPING: direct forward dhcp replyto output port: FortyGigabitEthernet1/0/2.
```

要調試DHCP監聽事件，請執行以下步驟：

⚠ 注意：請謹慎使用debug命令。請注意，許多debug指令都會影響實際網路，且只有在重現問題時才建議在實驗室環境中使用。

摘要步驟

1. 啟用

2. debug platform condition mac {mac-address }
3. debug platform condition start
4. show platform condition OR show debug
5. debug platform condition stop
6. show platform software trace message ios R0 reverse | 包括DHCP
7. clear platform condition all

詳細步驟

	命令或操作	目的
步驟 1	啟用 範例： switch#enable	啟用特權執行模式。 • 如果系統提示，請輸入您的密碼。
步驟 2	debug platform condition mac {mac-address} 範例： switch#debug platform condition mac 0001.6509.3314	為指定的MAC地址配置條件調試。
步驟 3	debug platform condition start 範例： switch#debug platform condition start	啟動條件調試（如果其中一個條件匹配，則可以啟動放射性跟蹤）。
步驟 4	show platform condition OR show debug 範例： switch#show platform condition switch#show debug	顯示當前條件集。
步驟 5	debug platform condition stop 範例： switch#debug platform condition stop	停止條件調試（這可以停止放射性跟蹤）。

	命令或操作	目的
步驟 6	show platform software trace message ios R0 reverse | 包括DHCP 範例： switch#show platform software trace message ios R0 reverse | 包括DHCP	顯示從最新跟蹤檔案合併的HP日誌。
步驟 7	clear platform condition all 範例： switch# clear platform condition all	清除所有條件。

以下是d的部分輸出示例ebug平台 dhcp-snoop all命令。

<#root>

debug platform dhcp-snoop all

DHCP Server UDP port

(67)

DHCP Client UDP port

(68)

RELEASE

```
Apr 14 16:44:18.629: pak->vlan_id = 10
Apr 14 16:44:18.629: dhcp packet src_ip(10.0.0.6) dest_ip(10.0.0.1) src_udp(68) dest_udp(67) src_mac(00
Apr 14 16:44:18.629: ngwc_dhcpsn_process_pak(305): Packet handedover to SISF on vlan 10
Apr 14 16:44:18.629: dhcp pkt processing routine is called for pak with SMAC = 00a3.d144.2046{mac} and
```

DISCOVER

```
Apr 14 16:44:24.637: dhcp packet src_ip(0.0.0.0) dest_ip(255.255.255.255) src_udp(68) dest_udp(67) src_
Apr 14 16:44:24.637: ngwc_dhcpsn_process_pak(305): Packet handedover to SISF on vlan 10
Apr 14 16:44:24.637: dhcp pkt processing routine is called for pak with SMAC = 00a3.d144.2046{mac} and
Apr 14 16:44:24.637: sending dhcp packet out after processing with SMAC = 00a3.d144.2046{mac} and SRC_A
Apr 14 16:44:24.638: pak->vlan_id = 10
```

OFFER

```
Apr 14 16:44:24.638: dhcp packet src_ip(10.0.0.1) dest_ip(255.255.255.255) src_udp(67) dest_udp(68) src
Apr 14 16:44:24.638: ngwc_dhcpsn_process_pak(305): Packet handedover to SISF on vlan 10
Apr 14 16:44:24.638: dhcp pkt processing routine is called for pak with SMAC = 701f.539a.fe46{mac}  and
```

REQUEST

```
Apr 14 16:44:24.638: ngwc_dhcpsn_process_pak(284): Packet handedover to SISF on vlan 10
c9500#dhcp pkt processing routine is called for pak with SMAC = 0a3.d144.2046{mac} and SRC_ADDR = 0.0.0
```

ACK

```
Apr 14 16:44:24.640:  dhcp paket src_ip(10.10.10.1) dest_ip(255.255.255.255) src_udp(67) dest_udp(68) s
Apr 14 16:44:24.640: ngwc_dhcpsn_process_pak(284): Packet handedover to SISF on vlan 10dhcp pkt process
```

下表列出了可用於調試平台中的DHCP監聽的各種命令。

⚠ 注意：請謹慎使用debug命令。請注意，許多debug命令都會影響實際網路，因此建議僅在重現問題時在實驗室環境中使用。

指令	目的
switch#debug platform dhcp-snoop [all | 資訊包 | pd-shim]	所有NGWC DHCP監聽 資料包NGWC DHCP監聽資料包調試資訊 pd-shim NGWC DHCP監聽IOS填充程式調試資訊
switch#debug platform software infrastructure punt dhcp-snoop	在FP上接收的投切到控制平面的資料包)
switch#debug platform software infrastructure injection	從控制平面注入FP的資料包

對點數/路徑流量(CPU)進行故障排除

從FED的角度驗證每個CPU隊列中接收了哪些流量（DHCP監聽是控制平面處理的流量型別）。

- 流量進入交換器時，會以PUNT方向傳送到CPU，並傳送到dhcp snoop佇列。
- 交換器處理流量後，流量會透過INJECT方向離開。 DHCP OFFER和ACK資料包屬於L2控制/傳統隊列。

<#root>

c9500#show platform software fed switch active punt cause summary

Statistics for all causes

```
Cause   Cause Info                  Rcvd         Dropped
-------------------------------------------------------------------------------
21      RP<->QFP keepalive          8533         0
79      dhcp snoop                  71           0      <<---- If drop counter increases, there can be a
96      Layer2 control protocols    45662        0
109     snoop packets               100          0
-------------------------------------------------------------------------------
```

c9500#show platform software fed sw active inject cause summary

Statistics for all causes

```
Cause Cause Info                  Rcvd           Dropped
-------------------------------------------------------------------------------
1     L2 control/legacy
         128354         0     <<---- dropped counter must NOT increase
2     QFP destination lookup      18             0
5     QFP <->RP keepalive         8585           0
12    ARP request or response     68             0
25    Layer2 frame to BD          81             0
-------------------------------------------------------------------------------
```

您可以使用此命令確認發往CPU的流量，並驗證DHCP監聽是否丟棄流量。

<#root>

c9500#

show platform software fed switch active punt cpuq rates

Punt Rate CPU Q Statistics

Packets per second averaged over 10 seconds, 1 min and 5 mins

```
================================================================================
Q  |              Queue              |  Rx  |  Rx   |  Rx   |  Drop  |  Drop  |  Drop
no |              Name               | 10s  |  1min |  5min |  10s   |  1min  |  5min
================================================================================
0 CPU_Q_DOT1X_AUTH                      0        0      0        0        0        0
1 CPU_Q_L2_CONTROL                      0        0      0        0        0        0
2 CPU_Q_FORUS_TRAFFIC                   0        0      0        0        0        0
3 CPU_Q_ICMP_GEN                        0        0      0        0        0        0
4 CPU_Q_ROUTING_CONTROL                 0        0      0        0        0        0
5 CPU_Q_FORUS_ADDR_RESOLUTION           0        0      0        0        0        0
```

```
6 CPU_Q_ICMP_REDIRECT                  0        0      0        0        0        0
7 CPU_Q_INTER_FED_TRAFFIC              0        0      0        0        0        0
8 CPU_Q_L2LVX_CONTROL_PKT              0        0      0        0        0        0
9 CPU_Q_EWLC_CONTROL                   0        0      0        0        0        0
10 CPU_Q_EWLC_DATA                     0        0      0        0        0        0
11 CPU_Q_L2LVX_DATA_PKT                0        0      0        0        0        0
12 CPU_Q_BROADCAST                     0        0      0        0        0        0
13 CPU_Q_LEARNING_CACHE_OVFL           0        0      0        0        0        0
14 CPU_Q_SW_FORWARDING                 0        0      0        0        0        0
15 CPU_Q_TOPOLOGY_CONTROL              2        2      2        0        0        0
16 CPU_Q_PROTO_SNOOPING                0        0      0        0        0        0
```

17 CPU_Q_DHCP_SNOOPING

```
0        0      0        0        0
```

0   <<---- drop counter must NOT increase

```
18 CPU_Q_TRANSIT_TRAFFIC               0        0      0        0        0        0
19 CPU_Q_RPF_FAILED                    0        0      0        0        0        0
20 CPU_Q_MCAST_END_STATION_SERVICE     0        0      0        0        0        0
21 CPU_Q_LOGGING                       0        0      0        0        0        0
22 CPU_Q_PUNT_WEBAUTH                  0        0      0        0        0        0
23 CPU_Q_HIGH_RATE_APP                 0        0      0        0        0        0
24 CPU_Q_EXCEPTION                     0        0      0        0        0        0
25 CPU_Q_SYSTEM_CRITICAL               8        8      8        0        0        0
26 CPU_Q_NFL_SAMPLED_DATA              0        0      0        0        0        0
27 CPU_Q_LOW_LATENCY                   0        0      0        0        0        0
28 CPU_Q_EGR_EXCEPTION                 0        0      0        0        0        0
29 CPU_Q_FSS                           0        0      0        0        0        0
30 CPU_Q_MCAST_DATA                    0        0      0        0        0        0
31 CPU_Q_GOLD_PKT                      0        0      0        0        0        0

------------------------------------------------------------------------------------
```

硬體故障排除

轉送引擎驅動程式(FED)

FED是程式設計ASIC的驅動程式。FED命令用於驗證硬體和軟體狀態是否匹配。

獲取DI_Handle值

- DI控制代碼引用特定埠的目標索引。

<#root>

c9500#show platform software fed switch active security-fed dhcp-snoop vlan vlan-id 10

Platform Security DHCP Snooping Vlan Information

Value of Snooping DI handle

is::

0x7F7FAC23E438   <<---- If DHCP Snooping is not enabled the hardware handle can not be present

```
                                                          Port              Trust Mode
-------------------------------------------------------------------------------------
                                              FortyGigabitEthernet1/0/10
```

trust <<---- Ensure TRUSTED ports are listed

檢查ifm對映以確定端口的Asic和Core。

- IFM是對映到特定埠/核心/asic的內部介面索引。

<#root>

c9500#show platform software fed switch active ifm mappings

```
Interface                   IF_ID  Inst Asic Core Port SubPort Mac Cntx LPN GPN Type Active
FortyGigabitEthernet1/0/10
```

0xa

```
    3
```

1   1

```
    1    0       4    4   2   2   NIF  Y
```

使用DI_Handle獲取硬體索引。

<#root>

c9500#show platform hardware fed switch active fwd-asic abstraction print-resource-handle 0x7F7FAC23E438

```
 0
Handle:0x7f7fac23e438 Res-Type:ASIC_RSC_DI Res-Switch-Num:255 Asic-Num:255 Feature-ID:AL_FID_DHCPSNOOPI
priv_ri/priv_si Handle: (nil)Hardware Indices/Handles:
```

index0:0x5f03

```
 mtu_index/l3u_ri_index0:0x0 index1:0x5f03 mtu_index/l3u_ri_index1:0x0 index2:0x5f03 mtu_index/l3u_ri_i
<SNIP>
```

<-- Index is 0x5f03

將索引值0x5f03從十六進位制轉換為十進位制。

0x5f03 = 24323

使用此索引值十進位制，以及此命令中的ASIC和核心值，檢視為埠設定了哪些標誌。

<#root>

c9500#show platform hardware fed switch 1 fwd-asic regi read register-name SifDestinationIndexTable-2432

asic

1

core

1

For asic 1 core 1

Module 0 - SifDestinationIndexTable[0][

24323

]

<-- the decimal hardware index matches 0x5f03 = 24323

copySegment0 :

0x1 <<---- If you find this as 0x0, means that the traffic is not forwarded out of this port. (refer to

CSCvi39202)copySegment1 : 0x1
dpuSegment0 : 0x0
dpuSegment1 : 0x0
ecUnicast : 0x0
etherChannel0 : 0x0
etherChannel1 : 0x0
hashPtr1 : 0x0
stripSegment : 0x0

確保為特定VLAN啟用了DHCP監聽。

<#root>

c9500#show platform software fed switch 1 vlan 10

VLAN Fed Information

Vlan Id IF Id LE Handle STP Handle L3 IF Handle SVI IF
---

10 0x0000000000420011

0x00007f7fac235fa8

0x00007f7fac236798 0x0000000000000000 0x0000000000000000 15

c9500#

show platform hardware fed switch active fwd-asic abstraction print-resource-handle

0x00007f7fac235fa8 1 <<---- Last number might be 1 or 0, 1 means detailed, 0 means brief output

Handle:0x7f7fac235fa8 Res-Type:ASIC_RSC_VLAN_LE Res-Switch-Num:255 Asic-Num:255 Feature-ID:AL_FID_L2 Lk
priv_ri/priv_si Handle: (nil)Hardware Indices/Handles: index0:0xf mtu_index/l3u_ri_index0:0x0 sm handle
Cookie length: 56
00 00 00 00 00 00 00 00 0a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0

Detailed Resource Information (ASIC_INSTANCE# 0)

LEAD_VLAN_IGMP_MLD_SNOOPING_ENABLED_IPV4 value 1 Pass   <<---- Verify the highlighted values, if any are

LEAD_VLAN_IGMP_MLD_SNOOPING_ENABLED_IPV6 value 0 Pass

LEAD_VLAN_ARP_OR_ND_SNOOPING_ENABLED_IPV4 value 1 Pass

LEAD_VLAN_ARP_OR_ND_SNOOPING_ENABLED_IPV6 value 1 Pass
LEAD_VLAN_BLOCK_L2_LEARN value 0 Pass
LEAD_VLAN_CONTENT_MATCHING_ENABLED value 0 Pass
LEAD_VLAN_DEST_MOD_INDEX_TVLAN_LE value 0 Pass

LEAD_VLAN_DHCP_SNOOPING_ENABLED_IPV4 value 1 Pass

LEAD_VLAN_DHCP_SNOOPING_ENABLED_IPV6 value 1 Pass
LEAD_VLAN_ENABLE_SECURE_VLAN_LEARNING_IPV4 value 0 Pass
LEAD_VLAN_ENABLE_SECURE_VLAN_LEARNING_IPV6 value 0 Pass
LEAD_VLAN_EPOCH value 0 Pass
LEAD_VLAN_L2_PROCESSING_STP_TCN value 0 Pass
LEAD_VLAN_L2FORWARD_IPV4_MULTICAST_PKT value 0 Pass
LEAD_VLAN_L2FORWARD_IPV6_MULTICAST_PKT value 0 Pass
LEAD_VLAN_L3_IF_LE_INDEX_PRIO value 0 Pass
LEAD_VLAN_L3IF_LE_INDEX value 0 Pass
LEAD_VLAN_LOOKUP_VLAN value 15 Pass
LEAD_VLAN_MCAST_LOOKUP_VLAN value 15 Pass
LEAD_VLAN_RIET_OFFSET value 4095 Pass
LEAD_VLAN_SNOOPING_FLOODING_ENABLED_IGMP_OR_MLD_IPV4 value 1 Pass
LEAD_VLAN_SNOOPING_FLOODING_ENABLED_IGMP_OR_MLD_IPV6 value 1 Pass
LEAD_VLAN_SNOOPING_PROCESSING_STP_TCN_IGMP_OR_MLD_IPV4 value 0 Pass
LEAD_VLAN_SNOOPING_PROCESSING_STP_TCN_IGMP_OR_MLD_IPV6 value 0 Pass
LEAD_VLAN_VLAN_CLIENT_LABEL value 0 Pass
LEAD_VLAN_VLAN_CONFIG value 0 Pass
LEAD_VLAN_VLAN_FLOOD_ENABLED value 0 Pass
LEAD_VLAN_VLAN_ID_VALID value 1 Pass
LEAD_VLAN_VLAN_LOAD_BALANCE_GROUP value 15 Pass
LEAD_VLAN_VLAN_ROLE value 2 Pass
LEAD_VLAN_VLAN_FLOOD_MODE_BITS value 3 Pass
LEAD_VLAN_LVX_VLAN value 0 Pass
LEAD_VLAN_EGRESS_DEJAVU_CANON value 0 Pass
LEAD_VLAN_EGRESS_INGRESS_VLAN_MODE value 0 Pass
LEAD_VLAN_EGRESS_LOOKUP_VLAN value 0 Pass
LEAD_VLAN_EGRESS_LVX_VLAN value 0 Pass
LEAD_VLAN_EGRESS_SGACL_DISABLED value 3 Pass
LEAD_VLAN_EGRESS_VLAN_CLIENT_LABEL value 0 Pass
LEAD_VLAN_EGRESS_VLAN_ID_VALID value 1 Pass

```
LEAD_VLAN_EGRESS_VLAN_LOAD_BALANCE_GROUP value 15 Pass
LEAD_VLAN_EGRESS_INTRA_POD_BCAST value 0 Pass
```

LEAD_VLAN_EGRESS_DHCP_SNOOPING_ENABLED_IPV4 value 1 Pass

```
LEAD_VLAN_EGRESS_DHCP_SNOOPING_ENABLED_IPV6 value 1 Pass
LEAD_VLAN_EGRESS_VXLAN_FLOOD_MODE value 0 Pass
LEAD_VLAN_MAX value 0 Pass
<SNIP>
```

此表列出了可用於跟蹤實際網路上DHCP資料包路徑的各種常見Punject show/debug命令。

常用點選/注入show和debug命令
debug platt soft fed swit acti inject add-filter cause 255 sub_cause 0 src_mac 0 0 dst_mac 0 0 src_ipv4 192.168.12.1 dst_ipv4 0.0.0.0 if_id 0xf
set platform software trace fed [switch<num|active|standby>] inject verbose — >使用顯示的過濾器命令將跟蹤範圍限定到此特定主機
set platform software trace fed [switch<num|active|standby>] inject debug boot — > for reload
set platform software trace fed [switch<num|active|standby>] punt noise
show platform software fed [switch<num|active|standby>] inject cause summary
show platform software fed [switch<num|active|standby>] punt cause summary
show platform software fed [switch<num|active|standby>] inject cpuq 0
show platform software fed [switch<num|active|standby>] punt cpuq 17(dhcp queue)
show platform software fed [switch<num|active|standby>] active inject packet-capture det
show platform software infrastructure injection
show platform software infrastructure punt
show platform software infrastructure lsmpi driver
debug platform software infra punt dhcp
debug platform software infra injection

這些命令對於檢查是否收到特定客戶端的任何DHCP資料包非常有用。

- 此功能可讓您擷取與CPU透過IOS-DHCP軟體處理的指定使用者端MAC位址相關聯的所有DHCP窺探通訊。

- IPv4和IPv6流量均支援此功能。
- 此功能將自動啟用。

重要:Cisco IOS XE直布羅陀版16.12.X提供這些命令。

---

switch#show platform dhcpsnooping client stats {mac-address}
switch#show platform dhcpv6snooping ipv6 client stats {mac-address}

<#root>

```
C9300#

show platform dhcpsnooping client stats 0000.1AC2.C148

DHCPSN: DHCP snooping server
DHCPD:  DHCP protocol daemen
L2FWD:  Transmit Packet to driver in L2 format
FWD:    Transmit Packet to driver
Packet Trace for client MAC 0000.1AC2.C148:
Timestamp            Destination MAC  Destination Ip  VLAN  Message      Handler:Action
-------------------- ---------------- --------------- ----  ------------ --------------
06-27-2019 20:48:28  FFFF.FFFF.FFFF   255.255.255.255 88   DHCPDISCOVER PUNT:RECEIVED
06-27-2019 20:48:28  FFFF.FFFF.FFFF   255.255.255.255 88   DHCPDISCOVER PUNT:TO_DHCPSN
06-27-2019 20:48:28  FFFF.FFFF.FFFF   255.255.255.255 88   DHCPDISCOVER BRIDGE:RECEIVED
06-27-2019 20:48:28  FFFF.FFFF.FFFF   255.255.255.255 88   DHCPDISCOVER BRIDGE:TO_DHCPD
06-27-2019 20:48:28  FFFF.FFFF.FFFF   255.255.255.255 88   DHCPDISCOVER BRIDGE:TO_INJECT
06-27-2019 20:48:28  FFFF.FFFF.FFFF   255.255.255.255 88   DHCPDISCOVER L2INJECT:TO_FWD
06-27-2019 20:48:28  0000.0000.0000   192.168.1.1     0    DHCPDISCOVER INJECT:RECEIVED
06-27-2019 20:48:28  0000.0000.0000   192.168.1.1     0    DHCPDISCOVER INJECT:TO_L2FWD
06-27-2019 20:48:30  0000.0000.0000   10.1.1.3        0    DHCPOFFER    INJECT:RECEIVED
06-27-2019 20:48:30  0000.1AC2.C148   10.1.1.3        0    DHCPOFFER    INTERCEPT:RECEIVED
06-27-2019 20:48:30  0000.1AC2.C148   10.1.1.3        88   DHCPOFFER    INTERCEPT:TO_DHCPSN
06-27-2019 20:48:30  0000.1AC2.C148   10.1.1.3        88   DHCPOFFER    INJECT:CONSUMED
06-27-2019 20:48:30  FFFF.FFFF.FFFF   255.255.255.255 88   DHCPREQUEST  PUNT:RECEIVED
06-27-2019 20:48:30  FFFF.FFFF.FFFF   255.255.255.255 88   DHCPREQUEST  PUNT:TO_DHCPSN
06-27-2019 20:48:30  FFFF.FFFF.FFFF   255.255.255.255 88   DHCPREQUEST  BRIDGE:RECEIVED
06-27-2019 20:48:30  FFFF.FFFF.FFFF   255.255.255.255 88   DHCPREQUEST  BRIDGE:TO_DHCPD
06-27-2019 20:48:30  FFFF.FFFF.FFFF   255.255.255.255 88   DHCPREQUEST  BRIDGE:TO_INJECT
06-27-2019 20:48:30  FFFF.FFFF.FFFF   255.255.255.255 88   DHCPREQUEST  L2INJECT:TO_FWD
06-27-2019 20:48:30  0000.0000.0000   192.168.1.1     0    DHCPREQUEST  INJECT:RECEIVED
06-27-2019 20:48:30  0000.0000.0000   192.168.1.1     0    DHCPREQUEST  INJECT:TO_L2FWD
06-27-2019 20:48:30  0000.0000.0000   10.1.1.3        0    DHCPACK      INJECT:RECEIVED
06-27-2019 20:48:30  0000.1AC2.C148   10.1.1.3        0    DHCPACK      INTERCEPT:RECEIVED
06-27-2019 20:48:30  0000.1AC2.C148   10.1.1.3        88   DHCPACK      INTERCEPT:TO_DHCPSN
```

使用這些命令清除跟蹤。

switch#clear platform dhcpsnooping pkt-trace ipv4

```
switch#clear platform dhcpsnooping pkt-trace ipv6
```

CPU路徑資料包捕獲

確認DHCP監聽資料包是否到達並正確離開控制平面。

注意：有關如何使用轉發引擎驅動程式CPU捕獲工具的其他參考，請參閱進一步讀取部分。

<#root>

debug platform software fed

[switch<num|active|standby>]

punt/inject

packet-capture start

debug platform software fed

[switch<num|active|standby>]

punt/inject

packet-capture stop

show platform software fed

[switch<num|active|standby>]

punt/inject

packet-capture brief

PUNT

DISCOVER

------ Punt Packet Number: 16, Timestamp: 2021/04/14 19:10:09.924 ------
interface :

physical: FortyGigabitEthernet1/0/2

[if-id: 0x0000000a], pal: FortyGigabitEthernet1/0/2 [if-id: 0x0000000a]
metadata : cause: 79

[dhcp snoop],

sub-cause: 11, q-no: 17, linktype: MCP_LINK_TYPE_IP [1]
ether hdr : dest mac: ffff.ffff.ffff,

src mac: 00a3.d144.2046

ether hdr : ethertype: 0x0800 (IPv4)

ipv4 hdr : dest ip: 255.255.255.255, src ip: 0.0.0.0
ipv4 hdr : packet len: 347, ttl: 255, protocol: 17 (UDP)
udp hdr : dest port:

67

, src port:

68

OFFER

------ Punt Packet Number: 23, Timestamp: 2021/04/14 19:10:11.926 ------
interface :

physical: FortyGigabitEthernet1/0/10

[if-id: 0x00000012], pal: FortyGigabitEthernet1/0/10 [if-id: 0x00000012]
metadata : cause: 79

[dhcp snoop]

, sub-cause: 11, q-no: 17, linktype: MCP_LINK_TYPE_IP [1]
ether hdr : dest mac: ffff.ffff.ffff,

src mac: 701f.539a.fe46

ether hdr : vlan: 10, ethertype: 0x8100
ipv4 hdr : dest ip: 255.255.255.255,

src ip: 10.0.0.1

ipv4 hdr : packet len: 330, ttl: 255, protocol: 17 (UDP)
udp hdr : dest port:

68

, src port:

67

REQUEST

------ Punt Packet Number: 24, Timestamp: 2021/04/14 19:10:11.927 ------
interface :

physical: FortyGigabitEthernet1/0/2

[if-id: 0x0000000a], pal: FortyGigabitEthernet1/0/2 [if-id: 0x0000000a]
metadata : cause: 79

[dhcp snoop]

, sub-cause: 11, q-no: 17, linktype: MCP_LINK_TYPE_IP [1]
ether hdr : dest mac: ffff.ffff.ffff,

src mac: 00a3.d144.2046

ether hdr : ethertype: 0x0800 (IPv4)
ipv4 hdr : dest ip: 255.255.255.255, src ip: 0.0.0.0
ipv4 hdr : packet len: 365, ttl: 255, protocol: 17 (UDP)
udp hdr : dest port:

67

, src port:

68

ACK

------ Punt Packet Number: 25, Timestamp: 2021/04/14 19:10:11.929 ------
interface :

physical: FortyGigabitEthernet1/0/10

[if-id: 0x00000012], pal: FortyGigabitEthernet1/0/10 [if-id: 0x00000012]
metadata : cause: 79

[dhcp snoop]

, sub-cause: 11, q-no: 17, linktype: MCP_LINK_TYPE_IP [1]
ether hdr : dest mac: ffff.ffff.ffff,

src mac: 701f.539a.fe46

ether hdr : vlan: 10, ethertype: 0x8100
ipv4 hdr : dest ip: 255.255.255.255,

src ip: 10.0.0.1

ipv4 hdr : packet len: 330, ttl: 255, protocol: 17 (UDP)
udp hdr : dest port:

68

, src port:

67

INJECT

DISCOVER

------ Inject Packet Number: 33, Timestamp: 2021/04/14 19:53:01.273 ------
interface : pal:

FortyGigabitEthernet1/0/2

[if-id: 0x0000000a]

metadata : cause: 25 [Layer2 frame to BD], sub-cause: 1, q-no: 0, linktype: MCP_LINK_TYPE_IP [1]
ether hdr : dest mac: ffff.ffff.ffff,

src mac: 00a3.d144.2046

ether hdr : ethertype: 0x0800 (IPv4)
ipv4 hdr : dest ip: 255.255.255.255, src ip: 0.0.0.0
ipv4 hdr : packet len: 347, ttl: 255, protocol: 17 (UDP)
udp hdr : dest port:

67

, src port:

68

OFFER

------ Inject Packet Number: 51, Timestamp: 2021/04/14 19:53:03.275 ------
interface : pal:

FortyGigabitEthernet1/0/2

[if-id: 0x0000000a]
metadata : cause: 1 [L2 control/legacy], sub-cause: 0, q-no: 0, linktype: MCP_LINK_TYPE_LAYER2 [10]
ether hdr : dest mac: ffff.ffff.ffff,

src mac: 701f.539a.fe46

ether hdr : ethertype: 0x0800 (IPv4)
ipv4 hdr : dest ip: 255.255.255.255,

src ip: 10.0.0.1

ipv4 hdr : packet len: 330, ttl: 255, protocol: 17 (UDP)
udp hdr : dest port:

68,

src port:

67

REQUEST

------ Inject Packet Number: 52, Timestamp: 2021/04/14 19:53:03.276 ------
interface : pal:

FortyGigabitEthernet1/0/2

[if-id: 0x0000000a]
metadata : cause: 25 [Layer2 frame to BD], sub-cause: 1, q-no: 0, linktype: MCP_LINK_TYPE_IP [1]
ether hdr : dest mac: ffff.ffff.ffff,

src mac: 00a3.d144.2046

```
ether hdr : ethertype: 0x0800 (IPv4)
ipv4 hdr : dest ip: 255.255.255.255, src ip: 0.0.0.0
ipv4 hdr : packet len: 365, ttl: 255, protocol: 17 (UDP)
udp hdr : dest port:
```

67

```
, src port:
```

68

ACK

```
------ Inject Packet Number: 53, Timestamp: 2021/04/14 19:53:03.278 ------
interface : pal:
```

FortyGigabitEthernet1/0/2

```
 [if-id: 0x0000000a]
metadata : cause: 1 [L2 control/legacy], sub-cause: 0, q-no: 0, linktype: MCP_LINK_TYPE_LAYER2 [10]
ether hdr : dest mac: ffff.ffff.ffff,
```

src mac: 701f.539a.fe46

```
ether hdr : ethertype: 0x0800 (IPv4)
ipv4 hdr : dest ip: 255.255.255.255,
```

src ip: 10.0.0.1

```
ipv4 hdr : packet len: 330, ttl: 255, protocol: 17 (UDP)
udp hdr : dest port:
```

68

```
, src port:
```

67

有用跟蹤

這些是二進位制跟蹤，用於顯示每個進程或元件的事件。在本示例中，跟蹤顯示有關dhcpsn元件的資訊。

- 可以手動旋轉跟蹤，這意味著可以在開始進行故障排除之前建立新檔案，以便其中包含更乾淨的資訊。

<#root>

9500#

request platform software trace rotate all

9500#

set platform software trace fed [switch

] dhcpsn verbose

c9500#show logging proc fed internal | inc dhcp

<<---- DI_Handle must match with the output which retrieves the DI handle

2021/04/14 19:24:19.159536 {fed_F0-0}{1}: [dhcpsn] [17035]: (info):

VLAN event on vlan 10, enabled 1

2021/04/14 19:24:19.159975 {fed_F0-0}{1}: [dhcpsn] [17035]: (debug): Program trust ports for this vlan
2021/04/14 19:24:19.159978 {fed_F0-0}{1}: [dhcpsn] [17035]: (debug):

GPN (10) if_id (0x0000000000000012) <<---- if_id must match with the TRUSTED port

2021/04/14 19:24:19.160029 {fed_F0-0}{1}: [dhcpsn] [17035]: (debug): trusted_if_q size=1 for vlan=10
2021/04/14 19:24:19.160041 {fed_F0-0}{1}: [dhcpsn] [17035]: (ERR): update ri has failed vlanid[10]
2021/04/14 19:24:19.160042 {fed_F0-0}{1}: [dhcpsn] [17035]: (debug): vlan mode changed to enable
2021/04/14 19:24:27.507358 {fed_F0-0}{1}: [dhcpsn] [23451]: (debug): get di for vlan_id 10
2021/04/14 19:24:27.507365 {fed_F0-0}{1}: [dhcpsn] [23451]: (debug): Allocated rep_ri for vlan_id 10
2021/04/14 19:24:27.507366 {fed_F0-0}{1}: [inject] [23451]: (verbose): Changing di_handle from 0x7f7fac

0x7f7fac23e438

by dhcp snooping
2021/04/14 19:24:27.507394 {fed_F0-0}{1}: [inject] [23451]: (debug): TX: getting REP RI from dhcpsn fai
2021/04/14 19:24:29.511774 {fed_F0-0}{1}: [dhcpsn] [23451]: (debug): get di for vlan_id 10
2021/04/14 19:24:29.511780 {fed_F0-0}{1}: [dhcpsn] [23451]: (debug): Allocated rep_ri for vlan_id 10
2021/04/14 19:24:29.511780 {fed_F0-0}{1}: [inject] [23451]: (verbose): Changing di_handle from 0x7f7fac

0x7f7fac23e438

by dhcp snooping
2021/04/14 19:24:29.511802 {fed_F0-0}{1}: [inject] [23451]: (debug): TX: getting REP RI from dhcpsn fai

c9500#set platform software trace fed [switch

] asic_app verbose

c9500#show logging proc fed internal | inc dhcp

2021/04/14 20:13:56.742637 {fed_F0-0}{1}: [dhcpsn] [17035]: (info):

VLAN event on vlan 10

, enabled 0
2021/04/14 20:13:56.742783 {fed_F0-0}{1}: [dhcpsn] [17035]: (debug): vlan mode changed to disable
2021/04/14 20:14:13.948214 {fed_F0-0}{1}: [dhcpsn] [17035]: (info): VLAN event on vlan 10, enabled 1
2021/04/14 20:14:13.948686 {fed_F0-0}{1}: [dhcpsn] [17035]: (debug):

Program trust ports for this vlan

2021/04/14 20:14:13.948688 {fed_F0-0}{1}: [dhcpsn] [17035]: (debug):

GPN (10) if_id (0x0000000000000012) <<---- if_id must match with the TRUSTED port

2021/04/14 20:14:13.948740 {fed_F0-0}{1}: [dhcpsn] [17035]: (debug): trusted_if_q size=1 for vlan=10
2021/04/14 20:14:13.948753 {fed_F0-0}{1}: [dhcpsn] [17035]: (ERR): update ri has failed vlanid[10]
2021/04/14 20:14:13.948754 {fed_F0-0}{1}: [dhcpsn] [17035]: (debug): vlan mode changed to enable

Suggested Traces

set platform software trace fed [switch<num|active|standby>] pm_tdl verbose
set platform software trace fed [switch<num|active|standby>] pm_vec verbose
set platform software trace fed [switch<num|active|standby>] pm_vlan verbose

INJECT

set platform software trace fed [switch<num|active|standby>] dhcpsn verbose
set platform software trace fed [switch<num|active|standby>] asic_app verbose
set platform software trace fed [switch<num|active|standby>] inject verbose

PUNT

set platform software trace fed [switch<num|active|standby>] dhcpsn verbose
set platform software trace fed [switch<num|active|standby>] asic_app verbse
set platform software trace fed [switch<num|active|standby>] punt ver

系統日誌和說明

違反DHCP速率限制。

說明：DHCP監聽在指定介面上檢測到DHCP資料包速率限制衝突。

```
%DHCP_SNOOPING-4-DHCP_SNOOPING_ERRDISABLE_WARNING: DHCP Snooping received 300 DHCP packets on interface
%DHCP_SNOOPING-4-DHCP_SNOOPING_RATE_LIMIT_EXCEEDED: The interface Fa0/2 is receiving more than the thres
```

DHCP伺服器在不受信任的埠上進行欺騙。

解釋：DHCP監聽功能發現不可信介面上不允許的某些型別的DHCP消息，這表示某些主機嘗試充當DHCP伺服器。

```
%DHCP_SNOOPING-5-DHCP_SNOOPING_UNTRUSTED_PORT: DHCP_SNOOPING drop message on untrusted port, message typ
```

第2層MAC地址與DHCP請求中的MAC地址不匹配。

說明：DHCP監聽功能嘗試了MAC地址驗證，檢查失敗。乙太網報頭中的源MAC地址與DHCP請求消息的chaddr欄位中的地址不匹配。 可能存在試圖對DHCP伺服器進行拒絕服務攻擊的惡意主機。

```
%DHCP_SNOOPING-5-DHCP_SNOOPING_MATCH_MAC_FAIL: DHCP_SNOOPING drop message because the chaddr doesn't mat
```

選項82插入問題。

解釋：DHCP監聽功能發現一個具有不可信埠上不允許的選項值的DHCP資料包，這表示某些主機嘗試充當DHCP中繼或伺服器。

```
%DHCP_SNOOPING-5-DHCP_SNOOPING_NONZERO_GIADDR: DHCP_SNOOPING drop message with non-zero giaddr or option
```

錯誤埠上接收到第2層MAC地址。

說明：DHCP監聽功能檢測到主機試圖對網路中的另一台主機進行拒絕服務攻擊。

```
%DHCP_SNOOPING-5-DHCP_SNOOPING_FAKE_INTERFACE: DHCP_SNNOPING drop message with mismatched source interfa
```

在不可信介面上收到的DHCP消息。

解釋：DHCP監聽功能發現不可信介面上不允許的某些型別的DHCP消息，這表示某些主機嘗試充當DHCP伺服器。

```
%DHCP_SNOOPING-5-DHCP_SNOOPING_UNTRUSTED_PORT: DHCP_SNOOPING drop message on untrusted port: GigabitEth
```

DHCP監聽傳輸失敗。無法訪問URL。

說明：DHCP監聽繫結傳輸失敗。

```
%DHCP_SNOOPING-4-AGENT_OPERATION_FAILED: DHCP snooping binding transfer failed. Unable to access URL
```

DHCP窺探警告

思科錯誤ID編號	說明
CSCvi39202	在上行etherchannel上啟用DHCP監聽信任時，DHCP失敗。
CSCvp49518	重新載入後不刷新DHCP監聽資料庫。
CSCvk16813	使用DHCP窺探和埠通道或跨堆疊上行鏈路丟棄的DHCP客戶端流量。
CSCvd51480	解除繫結ip dhcp監聽和裝置跟蹤。
CSCvm55401	DHCP監聽可以丟棄dhcp選項82 packets with ip dhcp snooping information option allow-untrusted。
CSCvx25841	當REP網段發生更改時，DHCP監聽信任狀態中斷。
CSCvs15759	DHCP伺服器在DHCP續訂過程中發出NAK資料包。
CSCvk34927	重新載入時，DHCP監聽表不會從DHCP監聽DB檔案更新。

SDA邊界DHCP窺探

DHCP窺探統計資訊CLI。

一個新的CLI，可用於SDA，用於檢驗DHCP監聽統計資訊。

註：有關Cisco SD接入交換矩陣邊緣DHCP流程/資料包流和解碼的其他參考，請參閱「相關資訊」部分中的指南。

---

switch#show platform fabric border dhcp snooping ipv4統計資訊
switch#show platform fabric border dhcp snooping ipv6統計資訊

<#root>

```
SDA-9300-BORDER#

show platform fabric border dhcp snooping ipv4 statistics

Timestamp           Source IP    Destination IP  Source Remote Locator  Lisp Instance ID  VLAN  PROCESS
------------------- ----------   --------------  ---------------------  ----------------  ----- -------
08-05-2019 00:24:16 10.30.30.1   10.40.40.1      192.168.0.1             8189              88    10
08-05-2019 00:24:16 10.30.30.1   10.40.40.1      192.168.0.1             8189              88    11

SDA-9300-BORDER#

show platform fabric border dhcp snooping ipv6 statistics

Timestamp            Source IP              Destination IP          Source Remote Locator  Lisp Instanc
-------------------  ---------------------- ----------------------  ---------------------  ------------
08-05-2019 00:41:46  11:11:11:11:11:11:11:1 22:22:22:22:22:22:22:1  192.168.0.3             8089
08-05-2019 00:41:47  11:11:11:11:11:11:11:1 22:22:22:22:22:22:22:1  192.168.0.3             8089
```

相關資訊

IP編址服務配置指南，Cisco IOS XE Amsterdam 17.3.x（Catalyst 9200交換機）

IP編址服務配置指南，Cisco IOS XE Amsterdam 17.3.x（Catalyst 9300交換機）

IP編址服務配置指南，Cisco IOS XE Amsterdam 17.3.x（Catalyst 9400交換機）

IP編址服務配置指南，Cisco IOS XE Amsterdam 17.3.x（Catalyst 9500交換機）

IP編址服務配置指南，Cisco IOS XE Amsterdam 17.3.x（Catalyst 9600交換機）

Cisco SD存取光纖邊緣DHCP程式/封包流與解碼

在Catalyst 9000交換機上配置FED CPU資料包捕獲

技術支援與文件 - Cisco Systems

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。