运行Catalyst 9000交换机上的DHCP监听并排除故障

目录

简介

本文档介绍如何操作Catalyst 9000系列交换机上的DHCP监听并对其进行故障排除

先决条件

要求

Cisco 建议您了解以下主题：

- Catalyst 9000系列交换机架构
- Cisco IOS® XE软件架构

使用的组件

本文档中的信息基于以下软件和硬件版本：

- C9200
- C9300
- C9400
- C9500
- C9600

思科IOS® XE 16.12.X

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。

注意：有关在其他思科平台上启用这些功能的命令，请参阅相应的配置指南。

背景信息

DHCP 监听

动态主机配置协议(DHCP)监听是一项安全功能，用于检查DHCP流量以阻止任何恶意DHCP数据包。它充当网络上不受信任的用户端口和DHCP服务器端口之间的防火墙，以防止网络中的恶意DHCP服务器，因为这可能导致拒绝服务。

DHCP监听操作

DHCP监听使用可信和不可信接口的概念。通过DHCP流量的路径，交换机验证接口上收到的DHCP数据包，并在受信任接口上跟踪预期的DHCP服务器数据包（OFFER和ACK）。换句话说，不受信任的接口会阻止DHCP服务器数据包。

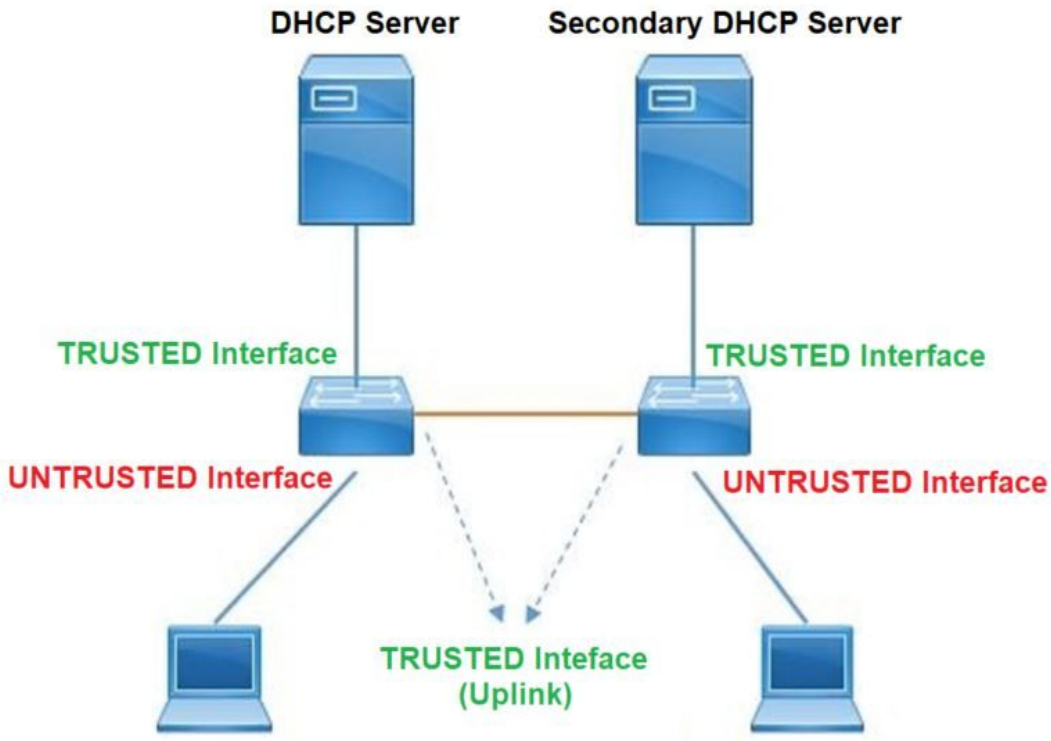
DHCP数据包在不受信任的接口上被阻止。

- 从网络或防火墙外部收到来自 DHCP 服务器的数据包，如 DHCPOFFER、DHCPACK、DHCPNAK 或 DHCPLEASEQUERY 数据包。这可防止非法DHCP服务器在不可信端口上攻击网络。
- 不可信接口上收到的数据包与源MAC地址和DHCP客户端硬件地址不匹配。这可防止欺诈客户端欺骗DHCP数据包，从而在DHCP服务器上造成拒绝服务攻击。
- DHCPRELEASE或DHCPDECLINE广播消息在DHCP监听绑定数据库中具有MAC地址，但绑定数据库中的接口信息与接收消息的接口不匹配。这可以防止对客户端的拒绝服务攻击。
- 由DHCP中继代理转发的DHCP数据包，包括非0.0.0.0的中继代理IP地址，或中继代理将包含选项82信息的数据包转发到不受信任的端口。这样可以防止网络上的中继代理信息欺骗。

配置DHCP监听的交换机建立DHCP监听表或DHCP绑定数据库。此表用于跟踪从合法DHCP服务器分配的IP地址。 绑定数据库也用于其他IOS安全功能，如动态ARP检测和IP源保护。

注意：要允许DHCP监听正常工作，请确保您信任所有上行链路端口以到达DHCP服务器，并取消信任最终用户端口。

拓扑



配置

全局配置

<#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：要允许option-82数据包，必须启用ip dhcp snooping information option allow-untrusted。

验证

确认是否在所需的VLAN上启用了DHCP监听，并确保已列出受信任和不受信任的接口。如果配置了速率，请确保也列出了该速率。

<#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:
```

用户通过DHCP接收IP后，会在此输出中列出。

- DHCP监听在IP地址租用到期或交换机从主机收到DHCPRELEASE消息时删除数据库中的条目。
- 确保为最终用户MAC地址列出的信息正确。

<#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
```

下表列出了可用于监控DHCP监听信息的各种命令。

命令	目的
show ip dhcp snooping binding show ip dhcp snooping binding [IP-address] [MAC-address] [interface ethernet slot/port] [vlan-id]	仅显示DHCP监听绑定数据库（也称为绑定表）中动态配置的绑定。 — 绑定条目IP地址 — 绑定条目Mac地址 — 绑定条目输入接口 — 绑定条目VLAN
show ip dhcp snooping database	显示DHCP监听绑定数据库状态和统计信息。
show ip dhcp snooping statistics	以摘要或详细信息形式显示DHCP监听统计信息。
show ip source binding	显示动态和静态配置的绑定。
show interface vlan xyz show buffer input-interface Vlan xyz dump	DHCP数据包通过客户端VLAN SVI发送到客户端VLAN中配置的中继代理。如果输入队列显示丢弃或达到最大限制，则可能是来自客户端的DHCP数据包被丢弃，无法到达配置的中继代理。 注意：确保输入队列中看不到丢弃。 switch#show int vlan 670 5秒的负载：13%/0%;1分钟：10%;5分钟：10% 时间来源为NTP，18:39:52.476 UTC 2020年9月10日星期四 Vlan670处于启用状态，线路协议处于启用状态，自动状态已启用 硬件为以太网SVI，地址为00fd.227a.5920(bia 00fd.227a.5920) 说明：ion_media_client Internet address is 10.27.49.254/23 MTU 1500字节，BW 1000000 Kbit/sec，DLY 10 usec，

	reliability 255/255, txload 1/255, rxload 1/255 Encapsulation ARPA, loopback not set 不支持Keepalive ARP类型：ARPA，ARP超时04:00:00 上次输入03:01:29，输出00:00:02，输出永不挂起 Last clearing of "show interface" counters never 输入队列：375/375/4020251/0(大小/最大/丢弃/刷新)；总输出丢弃：0 < — 输入队列/4020251中的375个数据包已丢弃

故障排除

软件故障排除

检验交换机收到什么。这些数据包在CPU控制平面处理，因此请确保您看到所有数据包的注入和传送方向，并确认信息是否正确。

⚠ 注意：请谨慎使用debug命令。请注意，许多debug命令会影响实时网络，因此建议仅在重现问题时在实验环境中使用。

Conditional Debug（条件调试）功能允许您根据您定义的一组条件选择性地启用特定功能的调试和日志。这对于仅包含特定主机或流量的调试信息非常有用。

条件是指功能或身份，其中身份可以是接口、IP地址或MAC地址等。

如何为数据包和事件调试启用条件调试，以排除DHCP监听故障。

命令	目的
debug condition mac <mac-address> 示例： switch#debug condition mac bc16.6509.3314	为指定的MAC地址配置条件调试。
debug condition vlan <VLAN Id> 示例： switch#debug condition vlan 10	为指定的VLAN配置条件调试。
debug condition interface <interface> 示例：	为指定的接口配置条件调试。

switch#debug condition interface twentyFiveGigE 1/0/8	

要调试DHCP监听，请使用表中显示的命令。

命令	目的
debug dhcp [detail | oper | 冗余]	detail DHCP packet content oper DHCP内部OPER 冗余DHCP客户端冗余支持
debug ip dhcp server packet detail	详细解码消息接收和传输
debug ip dhcp server events	报告地址分配、租赁到期等。
debug ip dhcp snooping agent	Debug dhcp snooping database read and write
debug ip dhcp snooping event	每个组件之间的调试事件
debug ip dhcp snooping packet	在DHCP监听模块中调试DHCP数据包

这是debug ip dhcp snooping命令的部分输出示例。

<#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.
```

要调试DHCP监听事件，请执行以下步骤：

⚠ 注意：请谨慎使用debug命令。请注意，许多debug命令会对实时网络产生影响，因此建议仅在重现问题的实验环境中使用。

总结步骤

1. enable

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

详细步骤

	命令或操作	目的
第 1 步	enable 示例： switch#enable	启用特权执行模式。 • 根据提示输入密码。
步骤 2	debug platform condition mac {mac-address} 示例： switch#debug platform condition mac 0001.6509.3314	为指定的MAC地址配置条件调试。
步骤 3	debug platform condition start 示例： switch#debug platform condition start	启动条件调试（如果其中一个条件匹配，则启动放射性跟踪）。
步骤 4	show platform condition OR show debug 示例： switch#show platform condition switch#show debug	显示当前条件集。
步骤 5	debug platform condition stop 示例： switch#debug platform condition stop	停止条件调试（这可以停止放射性跟踪）。

	命令或操作	目的
步骤 6	show platform software trace message ios R0 reverse | 包括DHCP 示例： switch#show platform software trace message ios R0 reverse | 包括DHCP	显示从最新跟踪文件合并的HP日志。
步骤 7	clear platform condition all 示例： switch# clear platform condition all	清除所有条件。

这是d的部分输出示例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
```

下表列出了可用于调试平台中的DHCP监听的各种命令。

⚠ 注意：请谨慎使用debug命令。请注意，许多debug命令会影响实际网络，因此建议仅在重现问题时在实验环境中使用。

命令	目的
switch#debug platform dhcp-snoop [all | 数据包 | pd-shim]	所有NGWC DHCP监听 数据包NGWC DHCP监听数据包调试信息 pd-shim NGWC DHCP监听IOS填充程序调试信息
switch#debug platform software infrastructure punt dhcp-snoop	在FP上接收并传送到控制平面的数据包)
switch#debug platform software infrastructure inject	从控制平面注入FP的数据包

排除Punt/Path流量(CPU)故障

从FED的角度验证每个CPU队列中接收了什么流量（DHCP监听是控制平面处理的流量类型）。

- 当流量进入交换机时，会以PUNT方向发送到CPU，并发送到dhcp snoop队列。
- 一旦流量被交换机处理，流量将通过INJECT方向离开。 DHCP OFFER和ACK数据包归入L2控制/传统队列。

<#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

您可以使用此命令确认传送到CPU的流量，并验证DHCP监听是否丢弃流量。

<#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 no	Queue Name	Rx 10s	Rx 1min	Rx 5min	Drop 10s	Drop 1min	Drop 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

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

硬件故障排除

转发引擎驱动程序(FED)

FED是对ASIC进行编程的驱动程序。FED命令用于验证硬件和软件状态是否匹配。

获取DI_Handle值

- DI句柄引用特定端口的目标索引。

<#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

检查ifm映射以确定端口的Asic和核心。

- IFM是映射到特定端口/核心/asic的内部接口索引。

<#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获取硬件索引。

<#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

将索引值0x5f03从十六进制转换为十进制。

0x5f03 = 24323

使用此十进制索引值以及此命令中的ASIC和核心值查看为端口设置了哪些标志。

<#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

确保为特定VLAN启用了DHCP监听。

<#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>
```

下表列出了可用于跟踪实际网络上DHCP数据包路径的各种常见Punject show/debug命令。

常用提示/注入show和debug命令
debug plat 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 — >使用显示的过滤器命令将跟踪范围限定到此特定主机
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 inject
show platform software infrastructure punt
show platform software infrastructure lsmpi driver
debug platform software infra punt dhcp
debug platform software infra inject

这些命令对于检查是否收到特定客户端的任何DHCP数据包非常有用。

- 此功能允许您捕获与CPU通过IOS-DHCP软件处理的给定客户端MAC地址关联的所有DHCP监听通信。

- IPv4和IPv6流量均支持此功能。
- 此功能将自动启用。

重要信息：这些命令可从Cisco IOS XE Gibraltar 16.12.X获得。

---

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
```

使用这些命令清除跟踪。

switch#clear platform dhcpsnooping pkt-trace ipv4

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

CPU路径数据包捕获

确认DHCP监听数据包是否到达并正确离开控制平面。

注意：有关如何使用转发引擎驱动程序CPU捕获工具的其他参考，请参阅进一步阅读部分。

<#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

有用跟踪

这些是显示每个进程或组件事件的二进制跟踪。在本示例中，跟踪显示有关dhcpsn组件的信息。

- 可以手动旋转轨迹，这意味着可以在开始进行故障排除之前创建新文件，使其包含更干净的信息。

<#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

系统日志和说明

违反DHCP速率限制。

解释：DHCP监听检测到指定接口上存在DHCP数据包速率限制冲突。

```
%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服务器在不可信端口上欺骗。

解释：DHCP监听功能发现不可信接口上不允许的特定类型的DHCP消息，这表示某些主机尝试充当DHCP服务器。

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

第2层MAC地址与DHCP请求中的MAC地址不匹配。

说明：DHCP监听功能尝试了MAC地址验证，检查失败。以太网报头中的源MAC地址与DHCP请求消息的chaddr字段中的地址不匹配。 可能存在试图对DHCP服务器实施拒绝服务攻击的恶意主机。

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

第82项插入问题。

解释：DHCP监听功能发现具有不可信端口上不允许的选项值的DHCP数据包，这表示某些主机尝试充当DHCP中继或服务器。

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

错误端口上收到第2层MAC地址。

解释：DHCP监听功能检测到主机试图对网络中的另一台主机进行拒绝服务攻击。

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

在不可信接口上收到的DHCP消息。

解释：DHCP监听功能发现不可信接口上不允许的特定类型的DHCP消息，这表示某些主机尝试充当DHCP服务器。

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

DHCP监听传输失败。无法访问URL。

解释：DHCP监听绑定传输失败。

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

DHCP监听警告

Cisco Bug ID号	描述
CSCvi39202	在上行链路etherchannel上启用DHCP监听信任时，DHCP失败。
CSCvp49518	DHCP监听数据库在重新加载后不刷新。
CSCvk16813	通过DHCP监听和端口通道或跨堆叠上行链路丢弃的DHCP客户端流量。
CSCvd51480	解除ip dhcp监听和设备跟踪的绑定。
CSCvm55401	DHCP监听可以丢弃dhcp选项82数据包，带ip dhcp snooping information option allow-untrusted。
CSCvx25841	当REP网段发生更改时，DHCP监听信任状态中断。
CSCvs15759	DHCP服务器在DHCP续订过程中发出NAK数据包。
CSCvk34927	DHCP监听表在重新加载时不会从DHCP监听DB文件更新。

SDA边界DHCP监听

DHCP监听统计信息CLI。

一个新的CLI，可用于SDA，以验证DHCP监听统计信息。

> 注意：有关思科SD接入交换矩阵边缘DHCP流程/数据包流和解码的其他参考，请参阅“相关信息”部分中的指南。

switch#show platform fabric border dhcp snooping ipv4统计信息
switch#show platform fabric border dhcp snooping ipv6统计信息

<#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
```

相关信息

IP编址服务配置指南，Cisco IOS XE Amsterdam 17.3.x（Catalyst 9200交换机）

IP编址服务配置指南，Cisco IOS XE Amsterdam 17.3.x（Catalyst 9300交换机）

IP编址服务配置指南，Cisco IOS XE Amsterdam 17.3.x（Catalyst 9400交换机）

IP编址服务配置指南，Cisco IOS XE Amsterdam 17.3.x（Catalyst 9500交换机）

IP编址服务配置指南，Cisco IOS XE Amsterdam 17.3.x（Catalyst 9600交换机）

思科SD访问交换矩阵边缘DHCP流程/数据包流和解码

在Catalyst 9000交换机上配置FED CPU数据包捕获

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）。