Troubleshoot Unknown Protocol Drops in Catalyst 9000 Switches

Contents

Introduction

This document describes common causes for unknown protocol drops in Catalyst 9000 series switches.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Dynamic Trunking Protocol (DTP)
- Link Layer Discovery Protocol (LLDP)
- Cisco Discovery Protocol (CDP)
- Encapsulation 802.1Q

Components Used

The information in this document is based on these software and hardware versions:

- Catalyst 9000 series switches
- Cisco IOS® XE

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

Background information

Unknown protocol drops occur when a frame's ethertype is not recognized, which means the encapsulated protocol is unsupported or not configured at the switch interface. Also, the frame's destination MAC address must be a multicast control plane address, which are listed in this command.

<#root>

```
Switch#

show mac address-table | include CPU

 All    0100.0ccc.cccc    STATIC      CPU
 All    0100.0ccc.cccd    STATIC      CPU
 All    0180.c200.0000    STATIC      CPU
 All    0180.c200.0001    STATIC      CPU
 All    0180.c200.0002    STATIC      CPU
 All    0180.c200.0003    STATIC      CPU
 All    0180.c200.0004    STATIC      CPU
 All    0180.c200.0005    STATIC      CPU
 All    0180.c200.0006    STATIC      CPU
 All    0180.c200.0007    STATIC      CPU
 All    0180.c200.0008    STATIC      CPU
 All    0180.c200.0009    STATIC      CPU
 All    0180.c200.000a    STATIC      CPU
 All    0180.c200.000b    STATIC      CPU
 All    0180.c200.000c    STATIC      CPU
 All    0180.c200.000d    STATIC      CPU
 All    0180.c200.000e    STATIC      CPU
 All    0180.c200.000f    STATIC      CPU
 All    0180.c200.0010    STATIC      CPU
 All    0180.c200.0021    STATIC      CPU
 All    ffff.ffff.ffff    STATIC      CPU
```

Note: Unknown protocol drops do not increment when destination MAC address is broadcast.

Troubleshoot

Step 1. Ensure that unknown protocol drops are incrementing.

```
<#root>

Switch#

show interface ten1/0/5 | include protocol

TenGigabitEthernet1/0/5 is up, line protocol is up (connected)

85 unknown protocol drops

Switch#

show interface ten1/0/5 | include protocol
```

```
TenGigabitEthernet1/0/5 is up, line protocol is up (connected)

90 unknown protocol drops
```

Step 2. Configure a packet capture in the affected interface and match destination MAC addresses starting with 01.

<#root>

```
Switch#

monitor capture port5 interface ten1/0/5 in


Switch#

monitor capture port5 match mac any 0100.0000.0000 00ff.ffff.ffff

Switch#

monitor capture port5 buffer size 100
```

Step 3. Start the packet capture and check the unknown-protocol-drops counter.

<#root>

```
Switch#

monitor capture port5 start

Started capture point : port5

Switch#

show interface ten1/0/5 | include protocol

TenGigabitEthernet1/0/5 is up, line protocol is up (connected)
     541 unknown protocol drops
```

Step 4. Stop the packet capture after a few unknown protocol drops.

<#root>

```
Switch#

show interface ten1/0/5 | include protocol

TenGigabitEthernet1/0/5 is up, line protocol is up (connected)
     544 unknown protocol drops

Switch#

monitor capture port5 stop
```

```
Capture statistics collected at software:
        Capture duration - 68 seconds
        Packets received - 38
        Packets dropped - 0
        Packets oversized - 0

Bytes dropped in asic - 0

Capture buffer will exists till exported or cleared

Stopped capture point : port5
```

Step 5. Export packet capture contents.

<#root>

```
Switch#

monitor capture port5 export location flash:drops.pcap

Export Started Successfully

Switch#
Export completed for capture point port5
```

Step 6. Transfer the packet capture to your computer.

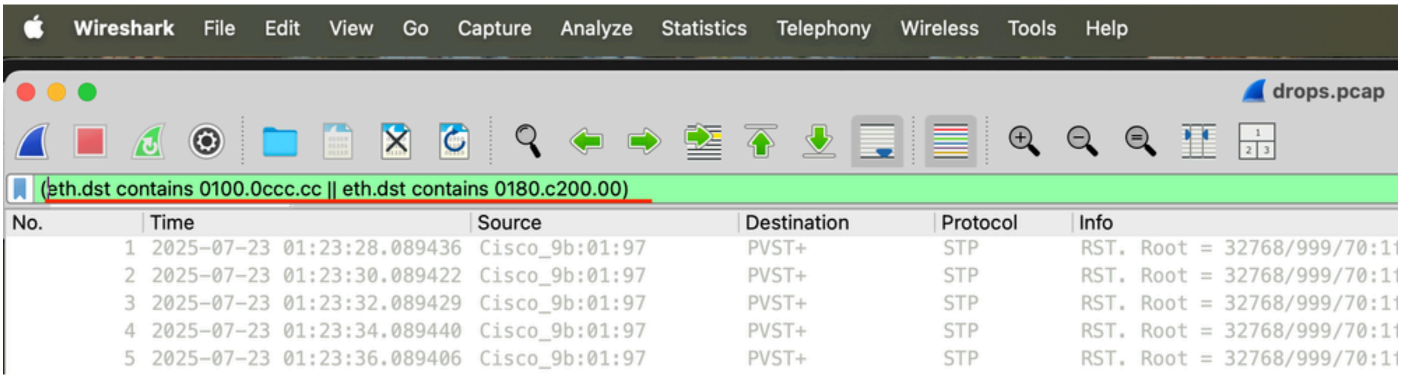
<#root>

```
Switch#

copy flash: ftp: vrf Mgmt-vrf

Source filename [drops.pcap]?
Address or name of remote host []? 10.10.10.254
Destination filename [drops.pcap]?
Writing drops.pcap !
4024 bytes copied in 0.026 secs (154769 bytes/sec)
```

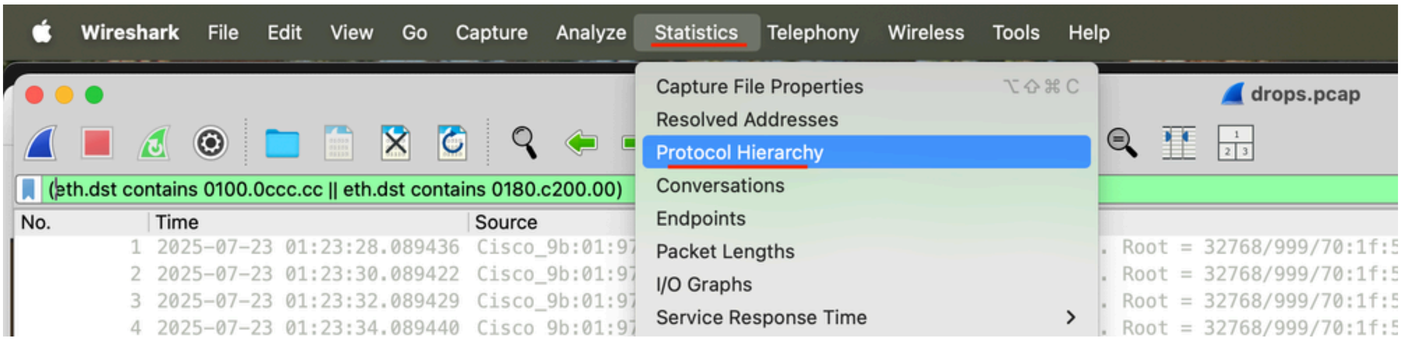
Step 7. Open the packet capture in Wireshark and use this filter **(eth.dst contains 0100.0ccc.cc || eth.dst contains 0180.c200.00)** to focus on CPU multicast addresses.



(eth.dst contains 0100.0ccc.cc || eth.dst contains 0180.c200.00)

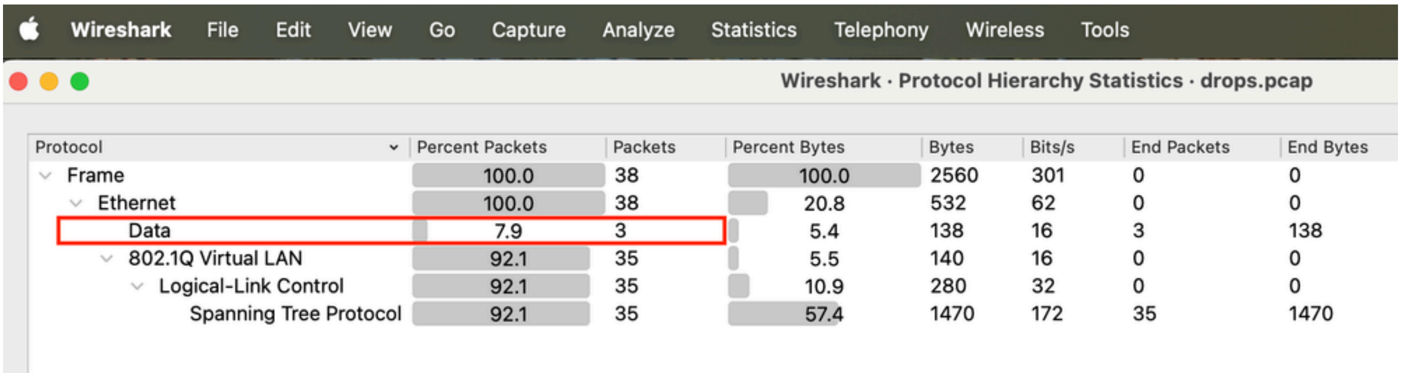
No.	Time	Source	Destination	Protocol	Info
1	2025-07-23 01:23:28.089436	Cisco_9b:01:97	PVST+	STP	RST. Root = 32768/999/70:1
2	2025-07-23 01:23:30.089422	Cisco_9b:01:97	PVST+	STP	RST. Root = 32768/999/70:1
3	2025-07-23 01:23:32.089429	Cisco_9b:01:97	PVST+	STP	RST. Root = 32768/999/70:1
4	2025-07-23 01:23:34.089440	Cisco_9b:01:97	PVST+	STP	RST. Root = 32768/999/70:1
5	2025-07-23 01:23:36.089406	Cisco_9b:01:97	PVST+	STP	RST. Root = 32768/999/70:1

Step 8. Go to **Statistics** and then click **Protocol Hierarchy**.



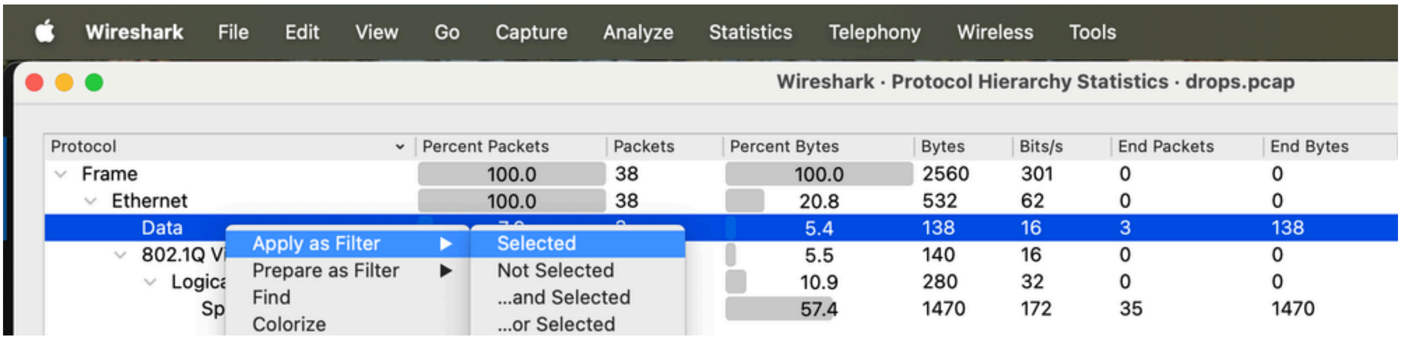
```
(eth.dst contains 0100.0ccc.cc || eth.dst contains 0180.c200.00)
No.   Time                          Source
   1  2025-07-23 01:23:28.089436  Cisco_9b:01:97   . Root = 32768/999/70:1f:5
   2  2025-07-23 01:23:30.089422  Cisco_9b:01:97   . Root = 32768/999/70:1f:5
   3  2025-07-23 01:23:32.089429  Cisco_9b:01:97   . Root = 32768/999/70:1f:5
   4  2025-07-23 01:23:34.089440  Cisco_9b:01:97   . Root = 32768/999/70:1f:5
```

Step 9. Expand the protocol tree and verify that the switch interface is configured for these protocols. Anything labeled as **Data** causes unknown protocol drops because the ethertype is unknown.



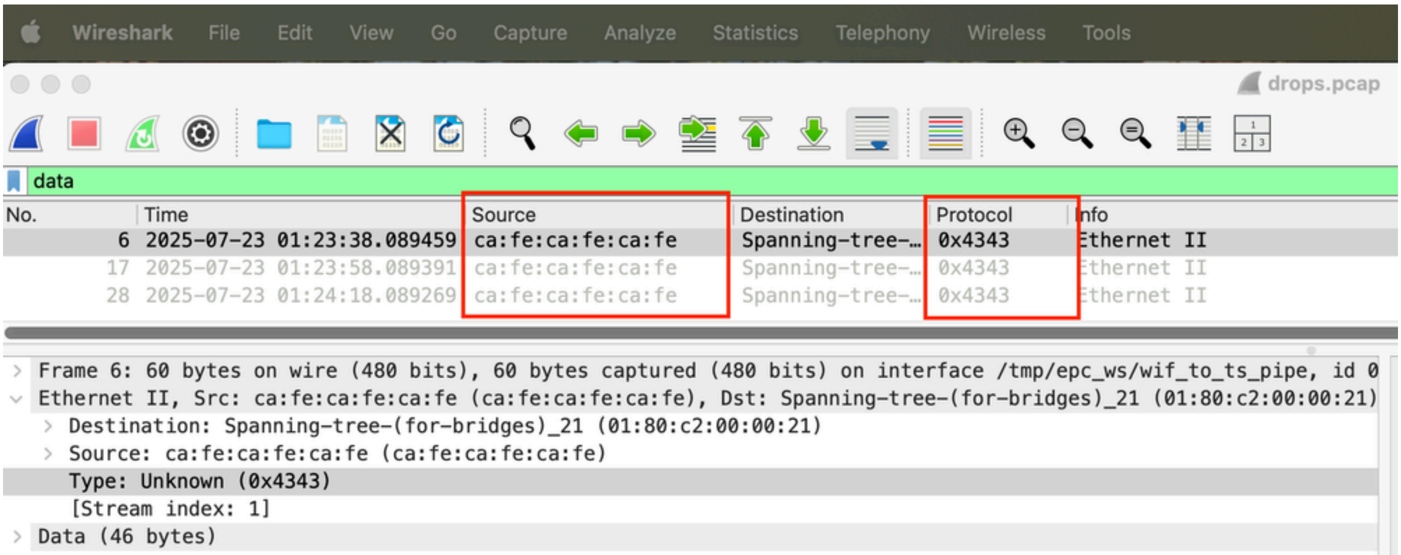
Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes
Frame	100.0	38	100.0	2560	301	0	0
Ethernet	100.0	38	20.8	532	62	0	0
Data	7.9	3	5.4	138	16	3	138
802.1Q Virtual LAN	92.1	35	5.5	140	16	0	0
Logical-Link Control	92.1	35	10.9	280	32	0	0
Spanning Tree Protocol	92.1	35	57.4	1470	172	35	1470

Step 10. Right-click **Data**, navigate to **Apply as Filter** and click **Selected** to filter unknown protocol frames.



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes
Frame	100.0	38	100.0	2560	301	0	0
Ethernet	100.0	38	20.8	532	62	0	0
Data			5.4	138	16	3	138
802.1Q Vi			5.5	140	16	0	0
Logica			10.9	280	32	0	0
Sp			57.4	1470	172	35	1470

Step 11. Go back to Wireshark's main window to determine source MAC address and ethertype for unknown protocols.



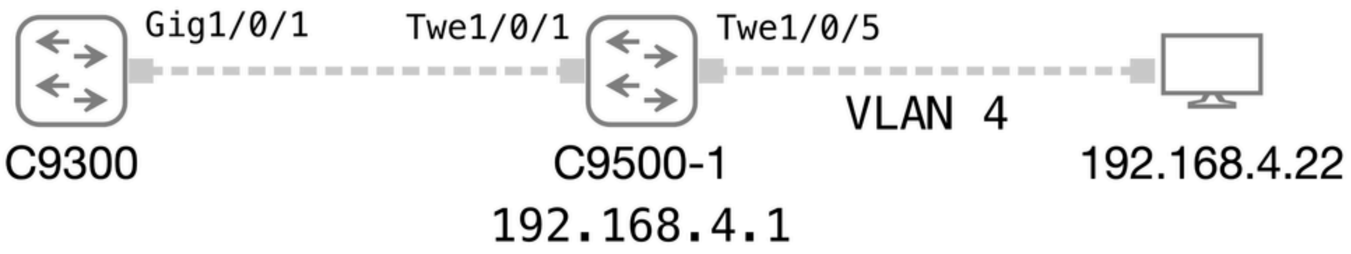
No.	Time	Source	Destination	Protocol	Info
6	2025-07-23 01:23:38.089459	ca:fe:ca:fe:ca:fe	Spanning-tree-…	0x4343	Ethernet II
17	2025-07-23 01:23:58.089391	ca:fe:ca:fe:ca:fe	Spanning-tree-…	0x4343	Ethernet II
28	2025-07-23 01:24:18.089269	ca:fe:ca:fe:ca:fe	Spanning-tree-…	0x4343	Ethernet II

```
> Frame 6: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface /tmp/epc_ws/wif_to_ts_pipe, id 0
v Ethernet II, Src: ca:fe:ca:fe:ca:fe (ca:fe:ca:fe:ca:fe), Dst: Spanning-tree-(for-bridges)_21 (01:80:c2:00:00:21)
  > Destination: Spanning-tree-(for-bridges)_21 (01:80:c2:00:00:21)
  > Source: ca:fe:ca:fe:ca:fe (ca:fe:ca:fe:ca:fe)
    Type: Unknown (0x4343)
    [Stream index: 1]
> Data (46 bytes)
```

In this case, source MAC address CAFE.CAFE.CAFE is causing unknown protocol drops because ethertype 0x4343 is unsupported.

Common problems

The examples in this section are based on this network topology diagram.



Dynamic Trunking Protocol (DTP)

DTP messages could potetially cause unknown protocol drops if they are received on a port where DTP is disabled. You can enable DTP by using the command **no switchport nonegotiate** in interface configuration mode.

<#root>

```
C9500-1#

show running-config interface Twe1/0/1


interface TwentyFiveGigE1/0/1
 description C9300
 switchport mode trunk
end

C9300#

show running-config interface Gi1/0/1
```

```
interface GigabitEthernet1/0/1
 description C9500-1
 switchport mode trunk
 switchport nonegotiate
end

C9300#

show interface gi1/0/1 | include unknown

     350 unknown protocol drops
```

Link Layer Discovery Protocol (LLDP)

LLDP messages can also cause unknown protocol drops if they are received on port where LLDP is disabled. You can enable LLDP by using the command **lldp run** in global configuration mode.

<#root>

```
C9500-1#

show lldp

Global LLDP Information:
    Status: ACTIVE
    LLDP advertisements are sent every 30 seconds
    LLDP hold time advertised is 120 seconds
    LLDP interface reinitialisation delay is 2 seconds

C9300#

show lldp

% LLDP is not enabled

C9300#

show interface gi1/0/1 | include unknown

     423 unknown protocol drops
```

Cisco Discovery Protocol (CDP)

Similarly, unknown protocol drops can increment if CDP messages are received on a port where CDP is disabled. You can enable CDP by using the command **cdp run** in global configuration mode.

<#root>

```
C9500-1#

show cdp


Global CDP information:
        Sending CDP packets every 60 seconds
        Sending a holdtime value of 180 seconds
        Sending CDPv2 advertisements is  enabled
```

```
C9300#

show cdp

% CDP is not enabled

C9300#

show interface gi1/0/1 | include unknown

     434 unknown protocol drops
```

All-zeros VLAN identifier in 802.1Q header

Catalyst 9000 series switches also drop 802.1Q frames with VLAN ID 0 when they received on access ports. However, these packets do not increment the unknown protocol drops counter. In this example, let us investigate why the Catalyst 9500 switch cannot get an ARP entry for host 192.168.4.22.

<#root>

```
C9500-1#

ping 192.168.4.22

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.22, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

C9500-1#

show ip arp vlan 4

Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  192.168.4.1              -   ecc0.18a4.b1bf  ARPA   Vlan4
C9500-1#

C9500-1#

show running-config interface Twe1/0/5

interface TwentyFiveGigE1/0/5
 switchport access vlan 4
 switchport mode access
 load-interval 30
end
```

Step 1. Start a packet capture in the interface connecting to the end device.

<#root>

```
C9500-1#

show monitor capture TAC parameter

   monitor capture TAC interface TwentyFiveGigE1/0/5 both
   monitor capture TAC match any
   monitor capture TAC buffer size 100 circular
   monitor capture TAC limit pps 1000

C9500-1#

monitor capture TAC start

Started capture point : TAC
```

Step 2. Try to ping the end device to generate some ARP traffic.

<#root>

```
C9500-1#

ping 192.168.4.22

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.22, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Step 3. Stop the packet capture.

<#root>

```
C9500-1#

monitor capture TAC stop

Capture statistics collected at software:
        Capture duration - 35 seconds
        Packets received - 28
        Packets dropped - 0
        Packets oversized - 0

Bytes dropped in asic - 0

Capture buffer will exists till exported or cleared

Stopped capture point : TAC
```

Step 4. Notice that the end device is sending an ARP reply, which in this case is frame 17.

<#root>

```
C9500-1#

show monitor capture TAC buff brief | include ARP

   15  19.402191 ec:c0:18:a4:b1:bf b^F^R ff:ff:ff:ff:ff:ff ARP 60 Who has 192.168.4.22? Tell 192.168.4.
   17  21.347022 fe:af:ea:fe:af:ea b^F^R ec:c0:18:a4:b1:bf ARP 60 192.168.4.22 is at fe:af:ea:fe:af:ea
```

Step 5. Notice that the ARP reply is encapsulated in an 802.1Q header using VLAN ID 0.

<#root>

```
C9500-1#

show monitor capture TAC buff detailed | begin Frame 17

Frame 17: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
<output omitted>
Ethernet II, Src: fe:af:ea:fe:af:ea (fe:af:ea:fe:af:ea), Dst: ec:c0:18:a4:b1:bf (ec:c0:18:a4:b1:bf)
    Destination: ec:c0:18:a4:b1:bf (ec:c0:18:a4:b1:bf)
        Address: ec:c0:18:a4:b1:bf (ec:c0:18:a4:b1:bf)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: fe:af:ea:fe:af:ea (fe:af:ea:fe:af:ea)
        Address: fe:af:ea:fe:af:ea (fe:af:ea:fe:af:ea)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: 802.1Q Virtual LAN (0x8100)

802.1Q Virtual LAN

, PRI: 0, DEI: 0, ID: 0
    000. .... .... .... = Priority: Best Effort (default) (0)
    ...0 .... .... .... = DEI: Ineligible
    ....

0000 0000 0000 = ID: 0

    Type: ARP (0x0806)
    Padding: 0000000000000000000000000000
Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: fe:af:ea:fe:af:ea (fe:af:ea:fe:af:ea)
    Sender IP address: 192.168.4.22
    Target MAC address: ec:c0:18:a4:b1:bf (ec:c0:18:a4:b1:bf)
    Target IP address: 192.168.4.1
```

Step 6. Export packet capture contents.

<#root>

```
C9500-1#

monitor capture TAC export location flash:ARP.pcap

Export Started Successfully
```

Step 7. Determine what the switch does with packet 17 by using the packet tracer tool.

<#root>

```
C9500-1#

show platform hardware fed active forward interface Twe1/0/5 pcap flash:ARP.pcap number 17 data

Show forward is running in the background. After completion, syslog will be generated.

C9500-1#
*Sep 29 17:45:29.091: %SHFWD-6-PACKET_TRACE_DONE: R0/0: fed: Packet Trace Complete:  Execute (show plat
*Sep 29 17:45:29.091: %SHFWD-6-PACKET_TRACE_FLOW_ID: R0/0: fed: Packet Trace Flow id is 6881284
```

Step 8. Display packet tracer results.

<#root>

```
C9500-1#

show platform hardware fed active forward last summary


Input Packet Details:
###[ Ethernet ]###
  dst       = ec:c0:18:a4:b1:bf
  src=fe:af:ea:fe:af:ea
  type      = 0x8100
###[ 802.1Q ]###
     prio      = 0
     id        = 0
     vlan      = 0
     type      = 0x806
###[ ARP ]###
        hwtype    = 0x1
        ptype     = 0x800
        hwlen     = 6
        plen      = 4
        op        = is-at
        hwsrc=fe:af:ea:fe:af:ea
        psrc=192.168.4.22
        hwdst     = ec:c0:18:a4:b1:bf
        pdst      = 192.168.4.1
###[ Padding ]###
           load      = '00 00 00 00 00 00 00 00 00 00 00 00 00 00'
<output omitted>

Packet DROPPED

 Catch-all for phf.finalFdPresent==1.
```

Note: Packet is dropped because it includes VLAN ID 0.

There are two options to prevent this type of drops.

Option 1: use the command **switchport voice vlan dot1p**. This way frames received with vlan 0 are assigned to the access vlan.

```
interface TwentyFiveGigE1/0/5
 switchport access vlan 4
 switchport mode access
 switchport voice vlan dot1p
 load-interval 30
```

Option 2: configure the interface as a trunk port. This way, frames received with vlan 0 are assigned to the native vlan.

```
interface TwentyFiveGigE1/0/5
 switchport trunk native vlan 4
 switchport mode trunk
 load-interval 30
end
```

Note: This has been commonly seen with Profinet devices.

Related Defects

- See Cisco bug ID CSCwe88812 for more information.

Related Fnformation

- VLAN 0 Priority Tagging Support