

Fehlerbehebung beim beschädigten Ethernet-Paket auf dem Cisco Nexus 9000

Inhalt

[Einführung](#)

[Hintergrundinformationen](#)

[Wie wird ein Paket von einem Switch verarbeitet?](#)

[Mit getaggten VLANs geändertes Padding bei der Übertragung von Datenverkehr über N9K](#)

[Lösung](#)

Einführung

In diesem Dokument wird beschrieben, wie Sie das beschädigte Ethernet-Paket auf dem Cisco Nexus 9000 beheben, wenn eine Padding-Information beschädigt oder falsch gebildet wird.

Hintergrundinformationen

Die minimale Größe eines Ethernet-Frames beträgt 64 Byte, unabhängig davon, welches VLAN-Tag ist dort vorhanden sein oder nicht.

Die minimale Ethernet-Nutzlastgröße ist:

- 46 Byte, wenn das VLAN-Tag nicht vorhanden ist.
- 42 Byte, wenn das VLAN-Tag vorhanden ist.

Sie können Folgendes überprüfen:

- Auf Wikipedia, Abschnitt **Payload**: https://en.wikipedia.org/wiki/Ethernet_frame
- Beim IEEE 802.3-Standard (http://people.ee.duke.edu/~mbrooke/EE164.02/Spring_2004/group_2/index_files/8023.pdf) Hier wird das MAC-Frame-Format (ohne VLAN) in Abschnitt 3.1.1, Seite definiert. 39 und die Elemente eines markierten MAC-Frames sind in Abschnitt 3.5 auf Seite 43 definiert.

Die minimale Größe eines Ethernet-Paketes beträgt 64 Byte, unabhängig davon, ob ein VLAN-Header vorhanden ist oder nicht. Der Server darf ein 64-Byte-Paket mit einem VLAN senden, das Sie akzeptieren und korrekt verarbeiten sollten.

Hinweis: Dieses Verhalten wird von einem Catalyst 4500x richtig gehandhabt, nicht von Nexus 9000.

Wie wird ein Paket von einem Switch verarbeitet?

Schritt 1: Empfangen eines **GÜLTIGEN** 64-Byte-Ethernet-Frames.

Schritt 2: Entfernen Sie die Frame Check Sequence (FCS), sodass das Paket 60 Byte lang wird.

Schritt 3: Entfernen Sie das VLAN-Tag, sodass das Paket 56 Byte lang wird.

Schritt 4: Fügen Sie Padding hinzu, um das Paket 60 Byte lang zu machen.

Schritt 5: Durch das Hinzufügen des FCS wird das Paket 64 Byte lang.

Beim Durchstellen eines Pakets über den Cut-Through-Switch darf das Padding nicht geändert werden.

Mit getaggten VLANs geändertes Padding bei der Übertragung von Datenverkehr über N9K

Anstatt mit Nullen zu paddeln, wird das Paket mit Garbage-Zeichen gepaart. In den meisten Fällen hat es keine Auswirkungen, da Prüfsummen nicht geändert werden und daher keine dieser Daten verwendet wird. Wenn Kunden jedoch eine spezielle Verwendung haben und die Prüfsummen neu berechnen müssen, führen diese überflüssigen Daten letztendlich zu Beschädigungen der Prüfsummen (das Problem kann auch bei anderen Appliances wie NAT/Load Balancern auftreten)

Gerät ist ein N9K 93120TX (wurde ursprünglich auf einem 9372TX erkannt), Version ist neueste NXOS 7.0(3)I2(2a)

Hier Linux-Hosts mit direkt mit der N9K verbundener Hardware (keine Virtualisierung irgendeiner Art) verwenden (1000Base-T-Verbindungen)

Verwenden Sie diese Konfiguration:

```
interface Ethernet1/59
    switchport mode trunk
!
interface Ethernet1/60
    switchport mode trunk

linux configurations:
inet 10.2.1.1/24 brd 10.2.1.255 scope global eth1 <= native vlan
inet 10.1.1.1/24 brd 10.1.1.255 scope global eth1.100 <= tagged vlan 100
```

oder

Schließen Sie einfach den Windows-Host an und senden Sie die getaggten Frames, sollten sie das Problem auslösen. Stellen Sie außerdem sicher, dass die Netzwerkkarte (NIC) das Paket kennzeichnen kann.

Switch fügt den durchlaufenden Frames das Padding von 0 (0) hinzu.

Beispiel: Host — [Trunk] N9K [Trunk] — Host

Sie können netcat zum Senden und Empfangen von Paketen verwenden.

Wie im Bild gezeigt, sendet er Side (VLAN 100 getaggt), Port e1/59 am Switch

```
6: eth1.100@eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc noqueue state UP group default
link/ether 44:a8:42:2c:5f:c4 brd ff:ff:ff:ff:ff:ff
inet 10.1.1.1/24 brd 10.1.1.255 scope global eth1.100
    valid_lft forever preferred_lft forever
inet6 fe80::46a8:42ff:fe2c:5fc4/64 scope link
    valid_lft forever preferred_lft forever
```

```
root@s35-c2-0:~# nc 10.1.1.2 3002 -u
a
^C
root@s35-c2-0:~#
```

Es empfängt Side (VLAN 100 markiert), Port e1/60 am Switch, wie im Bild gezeigt:

```
7: eth1.100@eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc noqueue state UP group default
link/ether 44:a8:42:2c:63:d1 brd ff:ff:ff:ff:ff:ff
inet 10.1.1.2/24 brd 10.1.1.255 scope global eth1.100
    valid_lft forever preferred_lft forever
inet6 fe80::46a8:42ff:fe2c:63d1/64 scope link
    valid_lft forever preferred_lft forever
```

```
root@s35-c2:~# nc -l -u -p 3002
a
^C
root@s35-c2:~#
```

Wie im Bild gezeigt, wird das Paket übertragen.

```
root@s35-c2-0:~# tcpdump -i eth1.100 -nvex
tcpdump: listening on eth1.100, link-type EN10MB (Ethernet), capture size 65535 bytes
10:42:20.953994 44:a8:42:2c:5f:c4 > 44:a8:42:2c:63:d1, ethertype IPv4 (0x0800), length 44: (tos 0x0, ttl 64, id 64283, offset 0, flags [DF], proto UDP (17), length 30)
    10.1.1.1.41675 > 10.1.1.2.3002: UDP, length 2
    0x0000: 4500 001e fb1b 4000 4011 29af 0a01 0101
    0x0010: 0a01 0102 a2cb 0bba 000a dca5 610a 0000
^C
1 packet captured
1 packet received by filter
0 packets dropped by kernel
root@s35-c2-0:~#
```

Das Paket wird empfangen, wie im Bild gezeigt:

```
10:43:12.665897 44:a8:42:2c:5f:c4 > 44:a8:42:2c:63:d1, ethertype IPv4 (0x0800), length 60: (tos 0x0, ttl 64, id 64283, offset 0, flags [DF], proto UDP (17), length 30)
    10.1.1.1.41675 > 10.1.1.2.3002: UDP, length 2
    0x0000: 4500 001e fb1b 4000 4011 29af 0a01 0101
    0x0010: 0a01 0102 a2cb 0bba 000a dca5 610a 0000
    0x0020: 0000 0000 0000 0000 0000 7562 710e
^C
7 packets captured
7 packets received by filter
0 packets dropped by kernel
root@s35-c2:~#
```

Wie im Bild gezeigt, wird das falsche Padding hervorgehoben.

The screenshot shows a packet capture in Wireshark. The top section is a table with columns: No., Time, Source, Destination, Protocol, Length, and Info. The first entry is No. 1, Time 0.000000, Source 10.1.1.1, Destination 10.1.1.2, Protocol UDP, Length 60, and Info Source port: 40849 Destination port: 3002.

The packet details pane shows the following structure:

- Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
- Ethernet II, Src: Dell_2c:5f:c4 (44:a8:42:2c:5f:c4), Dst: Dell_2c:63:d1 (44:a8:42:2c:63:d1)
 - Destination: Dell_2c:63:d1 (44:a8:42:2c:63:d1)
 - Source: Dell_2c:5f:c4 (44:a8:42:2c:5f:c4)
 - Type: IP (0x0800)
- Padding: 000000000000000000000000f1b7bc5c
- Internet Protocol Version 4, Src: 10.1.1.1 (10.1.1.1), Dst: 10.1.1.2 (10.1.1.2)
 - 0100 = Version: 4
 - 0101 = Header Length: 20 bytes
 - Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
 - Total Length: 30
 - Identification: 0xfbd (64285)
 - Flags: 0x02 (Don't Fragment)
 - Fragment offset: 0
 - Time to live: 64
 - Protocol: UDP (17)
 - Header checksum: 0x29ad [validation disabled]
 - Source: 10.1.1.1 (10.1.1.1)
 - Destination: 10.1.1.2 (10.1.1.2)
 - [Source GeoIP: Unknown]
 - [Destination GeoIP: Unknown]
- User Datagram Protocol, Src Port: 40849 (40849), Dst Port: 3002 (3002)
 - Source Port: 40849 (40849)
 - Destination Port: 3002 (3002)
 - Length: 10
 - Checksum: 0xdd7f [validation disabled]
 - [Good Checksum: False]
 - [Bad Checksum: False]
 - [Stream index: 0]
- Data (2 bytes)
 - Data: 610a
 - [Length: 2]

The bottom section shows the raw packet bytes in hexadecimal and ASCII. The padding bytes are highlighted in blue:

```
0000 44 a8 42 2c 63 d1 44 a8 42 2c 5f c4 08 00 45 00  D.,c.D. B,....E.
0010 00 1e fb 1d 40 00 40 11 29 ad 0a 01 01 01 0a 01  ....@. ).....
0020 01 02 9f 91 0b ba 00 0a dd 7f 61 0a 00 00 00 00  .....a.....
0030 00 00 00 00 00 00 00 00 f1 b7 bc 5c          ..... \
```

Dies wird auch mit einem Paketanalyser angezeigt (in einem anderen Paket unterscheiden sich die Daten von den vorherigen Screenshots, Test und Fehler sind jedoch identisch).

Lösung

Die Umgehungsarbeit besteht darin, [Puffer-Boost](#) auf der Schnittstelle deaktivieren, an der dieser Server angeschlossen ist.

```
C9396PX-1(config)# int et 1/7
```

```
C9396PX-1(config-if)# kein Buffer-Boost
```

Ähnlicher Fehler:

[CSCva46849](#) 60-Byte-Frame mit dot1q-Header L2-Switching auf N9k