

Dépannage du paquet Ethernet endommagé sur Cisco Nexus 9000

Contenu

[Introduction](#)

[Informations générales](#)

[Comment un paquet est-il traité par un commutateur ?](#)

[Remplissage modifié avec des VLAN étiquetés lorsque le trafic traverse N9K](#)

[Solution](#)

Introduction

Ce document décrit comment dépanner le paquet Ethernet endommagé sur Cisco Nexus 9000 lorsqu'une information de remplissage est endommagée ou malformée.

Informations générales

La taille minimale d'une trame Ethernet est de 64 octets, quelle que soit la balise VLAN qui y est présente ou non.

La taille minimale de la charge utile Ethernet est la suivante :

- 46 octets si la balise VLAN est absente.
- 42 octets si la balise VLAN est présente.

Vous pouvez vérifier ce fait :

- Sur Wikipedia, section **Payload** : https://en.wikipedia.org/wiki/Ethernet_frame
- Sur la norme IEEE 802.3 (http://people.ee.duke.edu/~mbrooke/EE164.02/Spring_2004/group_2/index_files/8023.pdf), où le format de trame MAC (sans VLAN) est défini à la section 3.1.1, page 39, et les éléments d'une trame MAC étiquetée sont définis à la page 43, section 3.5.

La taille minimale d'un paquet Ethernet est de 64 octets, quel que soit l'en-tête du VLAN présent ou non. Le serveur est autorisé à envoyer un paquet de 64 octets qui contient un VLAN, que vous devez accepter et traiter correctement.

Note: Ce comportement est correctement géré par un Catalyst 4500x et non par Nexus 9k.

Comment un paquet est-il traité par un commutateur ?

Étape 1. Recevez une trame Ethernet **VALIDE** de 64 octets.

Étape 2. Supprimez la séquence de contrôle de trame (FCS, Frame Check Sequence), de sorte que le paquet devient de 60 octets.

Étape 3. Supprimez la balise VLAN, de sorte que le paquet soit long de 56 octets.

Étape 4. Ajoutez un remplissage pour faire 60 octets de paquet.

Étape 5. Il ajoute le FCS, ce qui fait que le paquet a 64 octets de long.

Le remplissage ne doit pas être modifié lorsqu'un paquet passe par un commutateur cut-through.

Remplissage modifié avec des VLAN étiquetés lorsque le trafic traverse N9K

Au lieu de remplir le paquet avec des zéros, le paquet est rempli de caractères d'ordures, dans la plupart des cas, il n'a aucun impact car les sommes de contrôle ne sont pas modifiées et donc personne n'utilise ces données. Cependant, si les clients ont une utilisation particulière et ont besoin de recalculer les sommes de contrôle, ces données de déchets conduisent à la corruption des sommes de contrôle au bout du compte (d'autres appliances, comme NAT/load-balancers, peuvent également voir le problème).

Le périphérique est un N9K 93120TX (détecté initialement sur un 9372TX, cependant), la version est la dernière NXOS 7.0(3)I2(2a).

Utilisez ici des hôtes Linux avec du matériel directement connecté au N9K (aucune virtualisation de quelque sorte que ce soit) (liaisons 1000base-T).

Utilisez cette configuration :

```
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
```

ou

Il suffit de connecter l'hôte Windows et d'envoyer les trames marquées, elles doivent déclencher le problème. En outre, assurez-vous que la carte réseau (NIC) est capable d'étiqueter le paquet.

Le commutateur ajoute le remplissage non nul aux trames qui passent.

Exemple : Hôte — [Trunk] N9K [Trunk] — Hôte

Vous pouvez utiliser netcat pour envoyer et recevoir les paquets.

Comme l'illustre l'image, il envoie Side (VLAN 100 balisé), le port e1/59 sur le commutateur.

```
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:~#
```

Il reçoit côté (VLAN 100 balisé), port e1/60 sur le commutateur, comme illustré sur l'image :

```
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:~#
```

Comme l'illustre l'image, le paquet est transmis.

```
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 1620 610a
^C
1 packet captured
1 packet received by filter
0 packets dropped by kernel
root@s35-c2-0:~#
```

Le paquet est reçu, comme le montre l'image :

```
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 da45 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:~#
```

Comme le montre l'image, le mauvais remplissage est mis en surbrillance.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.1.1.1	10.1.1.2	UDP	60	Source port: 40849 Destination port: 3002

▶ 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: 0xfb1d (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]

0000	44 a8 42 2c 63 d1 44 a8 42 2c 5f c4 08 00 45 00	D.B,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 00a.....
0030	00 00 00 00 00 00 00 00 f1 b7 bc 5c\

Ceci est également affiché avec un analyseur de paquets (dans un autre paquet, les données sont différentes des captures d'écran précédentes mais le test et le bogue sont identiques),

Solution

La solution de contournement est de désactiver [buffer-boost](#) sur l'interface où ce serveur est connecté.

```
C9396PX-1(config)# int et 1/7
C9396PX-1(config-if)# no buffer-boost
```

Défaillance associée :

[CSCva46849](#) Trame de 60 octets avec commutation L2 en-tête dot1q sur N9k