

RFC3849 and RFC5737

Documentation! Documentation! Documentation!



Documentation matters

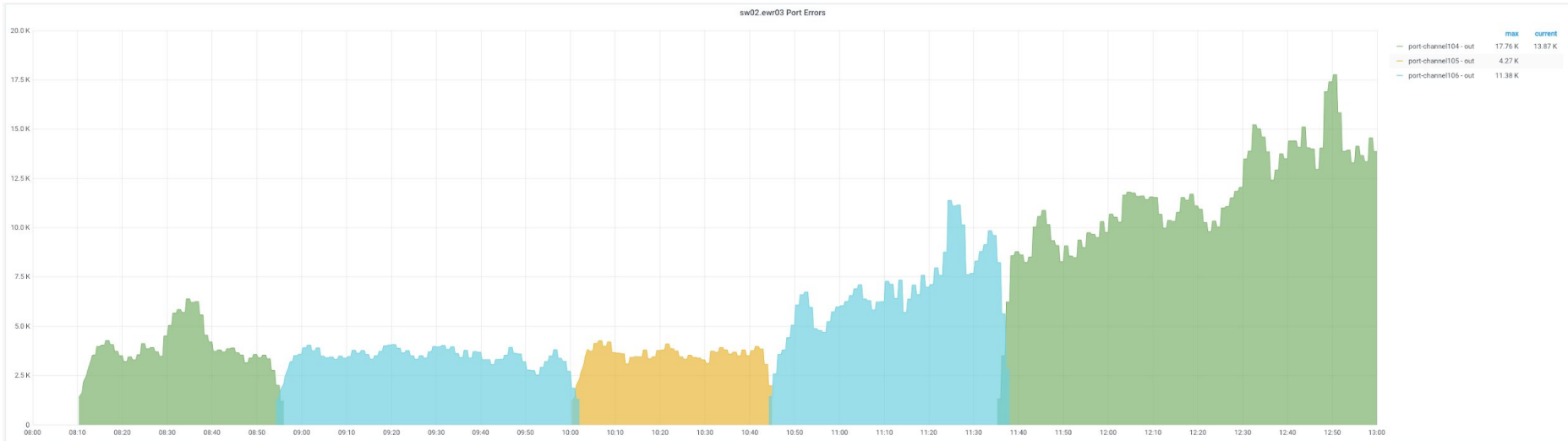
Why?



Network / NET-16171

sw02.ewr03 spitting errors

Flipperty Flopperty



Nothing is straightforward

added a comment - 18/Sep/19 10:15 AM - edited

so I think what's happening is that the Nexus receive a broken frame, because of cut-through the frames is forwarded to the dest () with a bad CRC on purpose (stomping on the Nexus)

```
~$ /sbin/ifconfig ext0 | grep err
RX errors 1746054969 dropped 195799 overruns 0 frame 1746054969
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
~$ /sbin/ifconfig ext0 | grep err
RX errors 1746064939 dropped 195799 overruns 0 frame 1746064939
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
~$ /sbin/ifconfig ext0 | grep err
RX errors 1746076243 dropped 195799 overruns 0 frame 1746076243
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

BUT for some reason the Nexus won't tell us where the broken frames are coming from (no CRC counter increasing)

so imho the issue here is a display issue more than a switch behavior

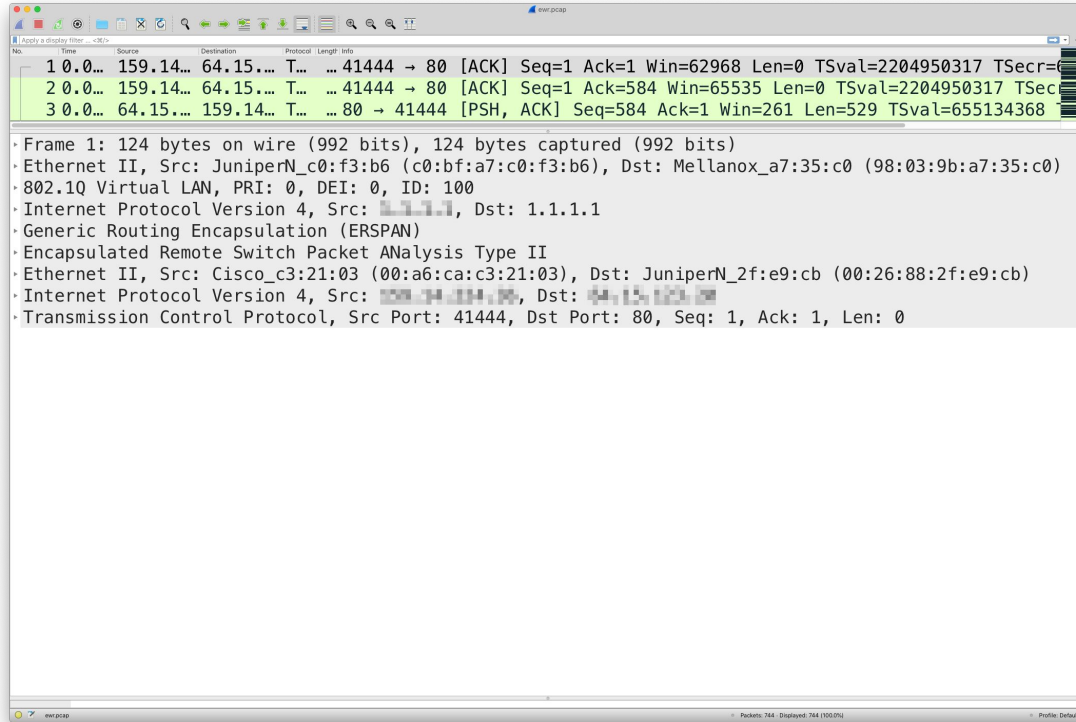
I'm pretty sure the problematic traffic comes from the same ToR since we would see CRC on the core01.ewr03 as well.

TCPDump!

 added a comment - 01/Nov/19 11:47 AM

I enabled 'rx-fcs' on ext0 and did a ethernet capture with tcpdump:

TCPDump! (contd.)



```
1 0.0... 159.14... 64.15... T... 41444 → 80 [ACK] Seq=1 Ack=1 Win=62968 Len=0 TSval=2204950317 TSecr=
2 0.0... 159.14... 64.15... T... 41444 → 80 [ACK] Seq=1 Ack=584 Win=65535 Len=0 TSval=2204950317 TSecr=
3 0.0... 64.15... 159.14... T... 80 → 41444 [PSH, ACK] Seq=584 Ack=1 Win=261 Len=529 TSval=655134368
```

- Frame 1: 124 bytes on wire (992 bits), 124 bytes captured (992 bits)
- Ethernet II, Src: JuniperN_c0:f3:b6 (c0:bf:a7:c0:f3:b6), Dst: Mellanox_a7:35:c0 (98:03:9b:a7:35:c0)
- 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 100
- Internet Protocol Version 4, Src: 1.1.1.1, Dst: 1.1.1.1
- Generic Routing Encapsulation (ERSPAN)
- Encapsulated Remote Switch Packet ANalysis Type II
- Ethernet II, Src: Cisco_c3:21:03 (00:a6:ca:c3:21:03), Dst: JuniperN_2f:e9:cb (00:26:88:2f:e9:cb)
- Internet Protocol Version 4, Src: 159.14.159.14, Dst: 64.15.159.14
- Transmission Control Protocol, Src Port: 41444, Dst Port: 80, Seq: 1, Ack: 1, Len: 0


Combined problems == hard problems



 admin

Tom Strickx

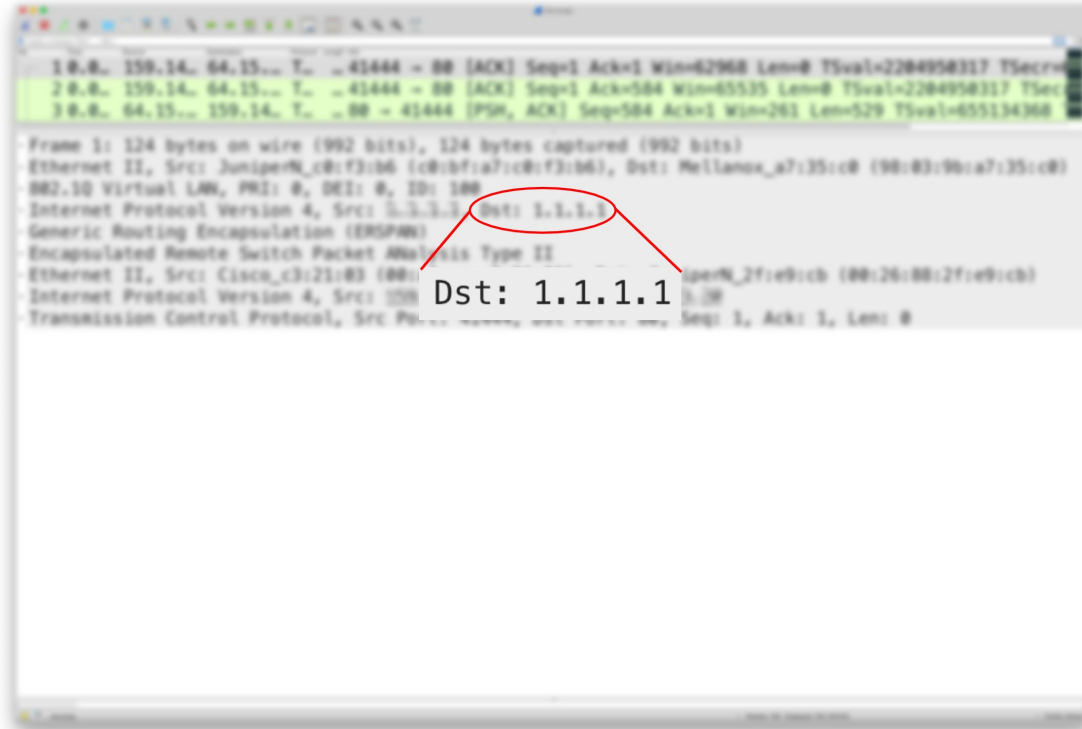
Yup, but odd that the QFXs didn't spot it

nexus has a long feature of reading deeper into packets than it should. had similar issues in 
with the N7K

Documentation?



And the penny drops



```
10.0. 159.14. 64.15... T. 41444 - 80 [ACK] Seq=1 Ack=1 Win=62968 Len=0 TSval=2284958317 TSecr=
20.0. 159.14. 64.15... T. 41444 - 80 [ACK] Seq=1 Ack=504 Win=65535 Len=0 TSval=2284958317 TSecr=
30.0. 64.15... 159.14. T. 80 - 41444 [PSH, ACK] Seq=504 Ack=1 Win=261 Len=529 TSval=655134368

-Frame 1: 124 bytes on wire (992 bits), 124 bytes captured (992 bits)
-Ethernet II, Src: JuniperN_c0:f3:b6 (c0:bfa7:c0:f3:b6), Dst: Mellanox_a7:35:c0 (90:83:90:a7:35:c0)
-IEEE 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 100
-Internet Protocol Version 4, Src: 1.1.1.1, Dst: 1.1.1.1
-Generic Routing Encapsulation (GRE)
-Encapsulated Remote Switch Packet 400...
-Ethernet II, Src: Cisco_c3:21:83 (00:0c:29:c3:21:83), Dst: JuniperN_2f:e9:cb (00:26:00:2f:e9:cb)
-Internet Protocol Version 4, Src: 100...
-Transmission Control Protocol, Src Port: 80, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
```

Documentation Matters!

<https://www.cisco.com/c/en/us/products/collateral/switches/nexus-5000-series-switches/white-paper-c11-733110.pdf>

You next need to define the source for the ERSPAN session: an interface, VLAN, or VSAN from which traffic will be mirrored. By default, traffic is monitored in both directions if the source is a port or PortChannel. To monitor traffic in only one direction, you must specify the transmit (tx) or receive (rx) parameter. To define a port and a PortChannel as the source of the ERSPAN session, use these commands:

```
switch(config-erspan-src)# source interface ethernet 1/1 [tx|rx|both]
switch(config-erspan-src)# source interface port-channel 101 [tx|rx|both]
```

To define a VLAN and a VSAN as the source of the ERSPAN session, use these commands:

```
switch(config-erspan-src)# source vlan 1
switch(config-erspan-src)# source vsan 1
```

The ERSPAN source session needs a defined destination IP address to which to forward traffic. The ERSPAN session can have only one destination IP address. Use this command to configure the destination IP address:

```
switch(config-erspan-src)# destination ip 1.1.1.1
```

Define the ERSPAN ID for the ERSPAN flow with this command:

```
switch(config-erspan-src)# erspan-id 1
```

To define a specific VRF to use instead of the global VRF, use either of these commands:

```
switch(config-erspan-src)# vrf default
```

or

```
switch(config-erspan-src)# vrf vrf-name
```

You can also define optional parameters: access control list (ACL), TTL, maximum transmission unit (MTU) for truncated ERSPAN, and differentiated services code point (DSCP) values.

In the source session, you can configure ACLs to filter packets in the ERSPAN session:

```
switch(config-erspan-src)# filter access-group erspan_acl_filter
```

To limit the number of jumps from the source to the destination and stop traffic from going to an unvented device, you can define the IP TTL. To configure the IP TTL for packets in ERSPAN traffic, use this command:

```
switch(config-erspan-src)# ip ttl 1
```

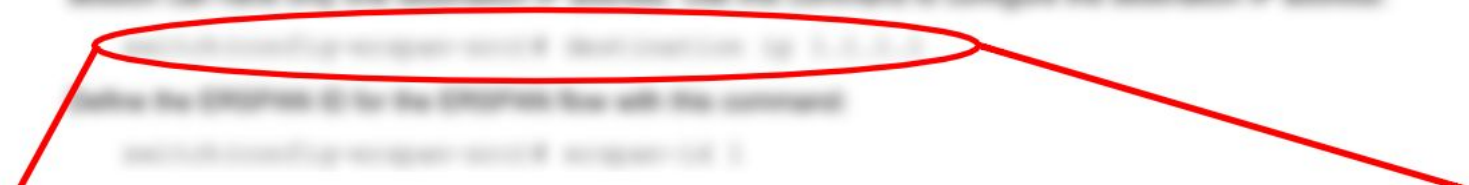
To prioritize ERSPAN traffic over some less important production traffic, you can change the DSCP value. By default, ERSPAN traffic is assigned a DSCP value of 0. To define a different DSCP value for packets in ERSPAN traffic, use this command:

```
switch(config-erspan-src)# ip dscp 1
```

If ERSPAN traffic is oversubscribing a network link, you can reduce the load by defining an MTU value. The MTU value for truncated ERSPAN packets can be between 64 and 1518 bytes. Use this command:

```
switch(config-erspan-src)# mtu 64
```

```
switch(config-erspan-src) # destination ip 1.1.1.1
```



Welp

<https://blog.cloudflare.com/fixing-reachability-to-1-1-1-globally/>

Announcing 1.1.1.1: the fastest, privacy-first consumer DNS service



01/04/2018



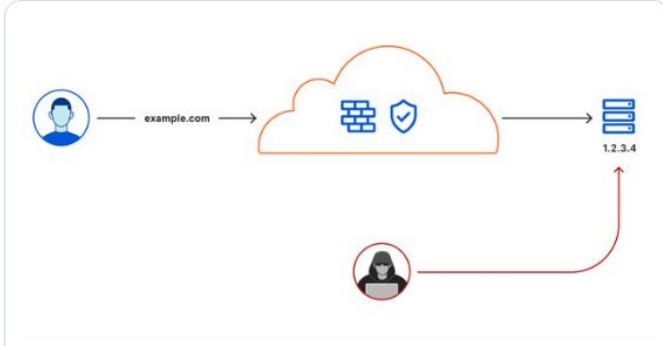
Matthew Prince

Awkward

<https://twitter.com/Cloudflare/status/1316062417416327169>

 **Cloudflare** 
@Cloudflare

Argo Tunnels that live forever. cfl.re/3dqjzN6
#ZeroTrustWeek



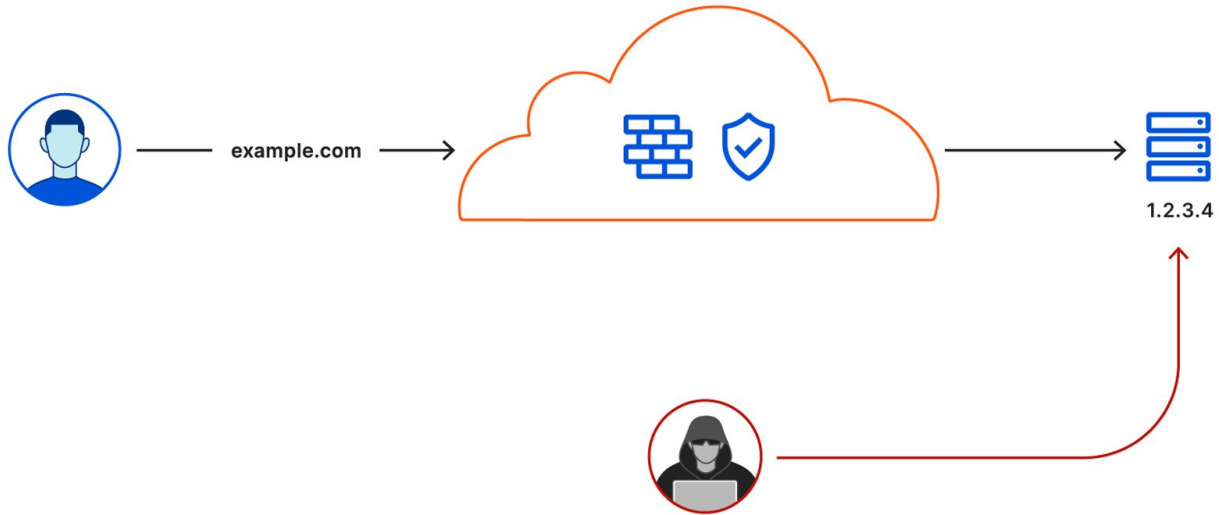
Argo Tunnels that live forever
Securely connecting your infrastructure to Cloudflare's network just became easier.
blog.cloudflare.com

6:05 pm · 13 Oct 2020 · Sprout Social

19 Retweets 3 Quote Tweets 19 Likes

Awkward

<https://twitter.com/Cloudflare/status/1316062417416327169>



Awkward

<https://twitter.com/Cloudflare/status/1316062417416327169>



RFC3849 and RFC5737

July 2004

IPv6 Address Prefix Reserved for Documentation

January 2010

IPv4 Address Blocks Reserved for Documentation

For those in the back

RFC3849

and

RFC5737

Summary - RFC3849

2001:DB8::/32

Summary - RFC5737

192.0.2.0/24
198.51.100.0/24
203.0.113.0/24

Summary



Thank you!

 @tstrickx

 tstrickx@cloudflare.com

