Some interesting stuff I found on IX LANs

Ben Cartwright-Cox
BGP.Tools / Port 179 LTD

NetUK 2 - June 2025

IX Lans are

- Ethernet MAC Switched Environment
  - (There used to be FDDI or other fun tech, these are all dead now, it's only ethernet)
- The goal is to be able to send frames directly to another router, using a IPv4/IPv6 LAN address
- Because it's a ethernet switched fabric, there is also the ability to send broadcast traffic in the same way your Home/SMB has

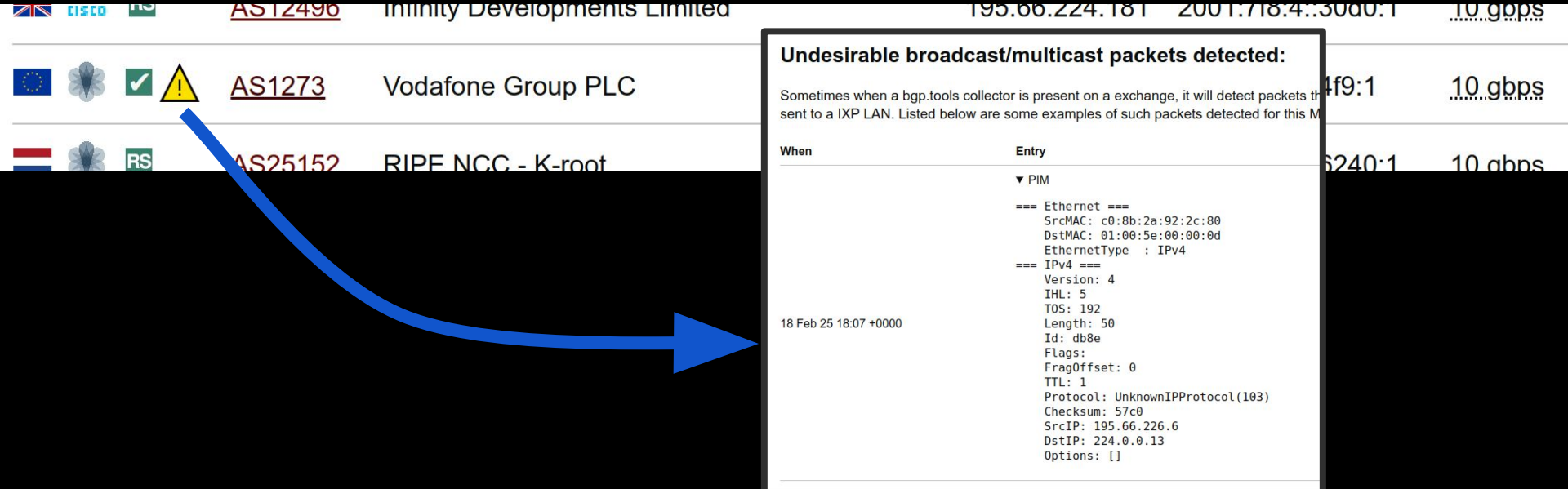
However

- A lot of network equipment has defaults or options that are more suited to internal LANs, and not network edge deployments
  - This includes things like:
  - Auto discovery (Hostname/OS Version/Port descriptions)
  - Auto configuration (DHCP/DNS/etc)
  - Interior Gateway Protocols that can autonomously setup with other network neighbors

- Problem is that these defaults on Internet eXchanges are at best "messy", at worst outright dangerous!
  - IXs are shared with various members, of varying trust
  - You do not want a random ASN on a IX being able to configure your router, or see that you are running a very old version of IOS-XE

Context to this adventure

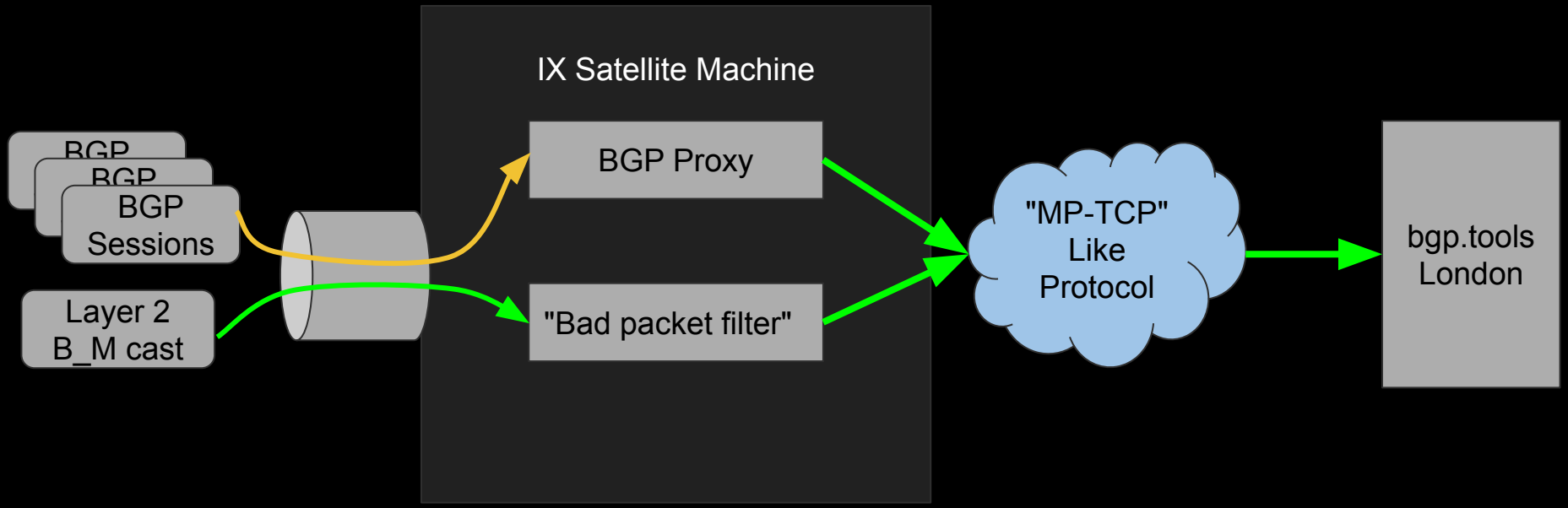
- bgp.tools has many (100~) IX ports for BGP route collection
- It also listens for stray Broadcast/Multicast traffic and tries to identify "bad packets", and marks the IX member (if it can figure it who)



Context to this adventure

- bgp.tools
- It also list entify "bad packets",

How does that work?



bgp.tools does not capture "Unknown Unicast" traffic!

Router/Switch Identification

- LLDP
- CDP

Router/Switch Identification

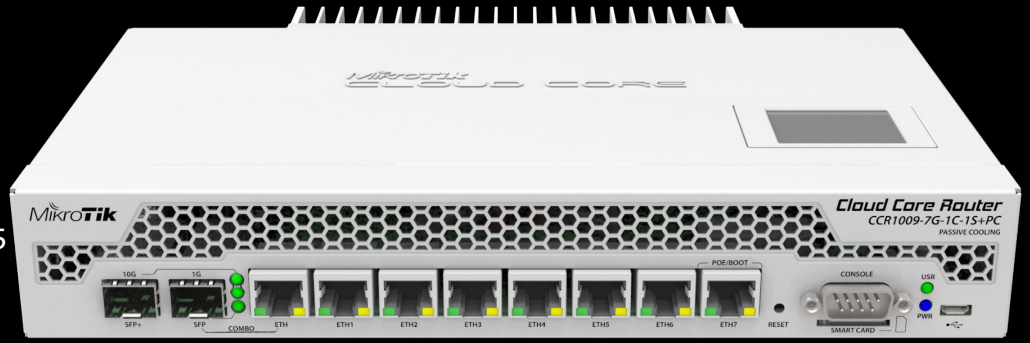
- LLDP
- CDP
- MNDP - Mikrotik Neighbor Discovery Protocol
  - A custom protocol, **enabled by default on all ports**

Router/Switch Identification

- LLDP
- CDP
- MNDP - Mikrotik Neighbor Discovery Protocol
  - A custom protocol, **enabled by default on all ports**
  - `=== MNDP ===`
  - `Seq: 134398`
  - `MAC Address: 08:55:31:1b:9c:aa`
  - `Identity: DC2.A23.CCR02`
  - `Version: 6.49.6 (stable)`
  - `Platform: MikroTik`
  - `Uptime: 2239h52m39s`
  - `SoftwareID: HBWH-7QHV`
  - `Board: CCR1009-7G-1C-1S+`
  - `IPv6Address: 2001:7f8:54::1:85`
  - `InterfaceName: FRANCE_IX`
  - `IPv4Address: 37.49.237.85`

Router/Switch Identification

- LLDP
- CDP
- MNDP - Mikrotik Neighbor Discovery Protocol
  - A custom protocol, **enabled by default on all ports**
  - `=== MNDP ===`
  - `Seq: 134398`
  - `MAC Address: 08:55:31:1b:9c:aa`
  - `Identity: DC2.A23.CCR02`
  - `Version: 6.49.6 (stable)`
  - `Platform: MikroTik`
  - `Uptime: 2239h52m39s`
  - `SoftwareID: HBWH-7QHV`
  - `Board: CCR1009-7G-1C-1S+`
  - `IPv6Address: 2001:7f8:54::1:85`
  - `InterfaceName: FRANCE_IX`
  - `IPv4Address: 37.49.237.85`



Router/Switch Identification

- LLDP
- CDP
- MNDP
- Some TPLink Switch someone added to the MAC Address Filter
  - Just blasts JSON payloads wrapped in UDP to 255.255.255.255
  - "Thanks"
  - Thanks to the JSON payload we know it was a
    - "JetStream 8-Port 10GE SFP+ L2+ Managed Switch"

Addressing?

- DHCP / DHCPv6
- IPv6 Router Advertisements

IPv6 Router Advertisements

- RA's say "Hey, you can send all of your stuff (aka ::/0) to me!"
- This is called "transit", it is something that people on IXs want to avoid
  - Giving away transit bandwidth for no cost is bad for business, most of the time
- Cisco by default will send RA's to anything with IPv6 addresses configured
  - Arista is so Cisco compatible, that they also have this default behaviour!
- Both can be disabled by

```
(Cisco) - ipv6 nd suppress-ra

(Arista) - ipv6 nd ra disabled all
```

IPv6 Router Advertisements

- RA's say "Hey, you can send all of your stuff (aka ::/0) to me!"
- This is called "transit", it is something that people on IXs want to avoid
  - Giving away transit bandwidth for no cost is bad for business, most of the time
- Cisco by default will send RA's to anything with IPv6 addresses configured
  - Arista is so Cisco compatible, that they also have this default behaviour!
- Both can be disabled by

```
(Cisco) - ipv6 nd suppress-ra

(Arista) - ipv6 nd ra disabled all
```

Unexpected Routing Protocol in IX Area

- OSPF / OSPFv3
- IS-IS / ES-IS
- RIP/RIPv2
- MPLS LDP

Unexpected Routing Protocol in IX Area

- OSPF / OSPFv3
- IS-IS / ES-IS
- RIP/RIPv2
- MPLS LDP

I have witnessed two different ASNs accidentally merge their IGPs because of these protocols left configured on IX ports

Testing

- IEEE802.1Q
  - ¯_(ツ)_/¯

Wireshark · Packet 270794 · unknown-ixp-packets.pcap-wip

```
▸ Frame 270794: 189 bytes on wire (1512 bits), 189 bytes captured (1512 bits) on interface intf0, id 0
▸ Ethernet II, Src: 02:00:c0:a8:62:62 (02:00:c0:a8:62:62), Dst: OAM-Multicast-DA-Class-1_07 (01:80:c2:00:00:37)
▾ CFM EOAM IEEE 802.1Q/ITU-T Y.1731 Protocol, Type Continuity Check Message (CCM)
    111. .... = CFM MD Level: 7
    ...0 0000 = CFM Version: 0
    CFM OpCode: Continuity Check Message (CCM) (1)
  ▾ CFM CCM PDU
    ▸ Flags: 0x00
      First TLV Offset: 74
      Sequence Number: 0
      ...0 0000 0000 0000 = Maintenance Association Endpoint Identifier: 0
    ▾ Maintenance Association Identifier
        MD Name Format: Reserved for IEEE 802.1 (0)
        MD Name Length: 0
        Short MA Name (MEG ID) Format: Reserved for IEEE 802.1 (0)
        Short MA Name (MEG ID) Length: 0
        Short MA Name: <MISSING>
        Zero-Padding
    ▸ Defined by ITU-T Y.1731
      Unknown data: 40000600
  ▾ CFM TLVs
    ▸ TLV: End TLV (t=0,l=0)

0000  01 80 c2 00 00 37 02 00  c0 a8 62 62 89 02 e0 01   ·····7·· ··bb····
0010  00 4a 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ·J····· ........
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 00 00 00 00 00 00  40 00 06 00 00 00 01 10   ........ @······ 
0060  00 01 00 42 40 01 30 32  2d 43 30 2d 41 38 2d 36   ···B@·02 -C0-A8-6
0070  32 2d 36 35 00 00 00 00  00 00 01 00 00 00 00 00   2-65···· ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

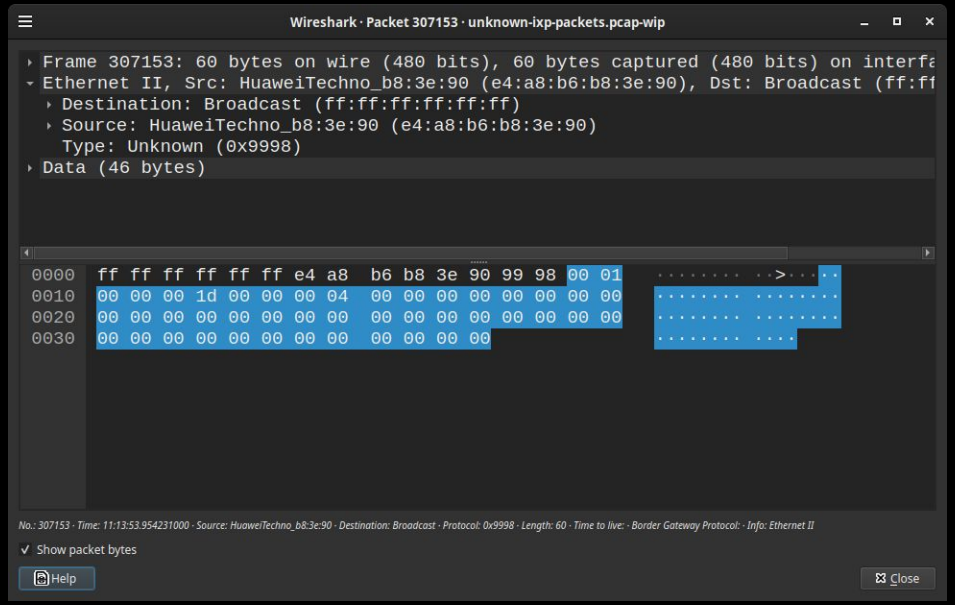
CFM EOAM IEEE 802.1Q/ITU-T Y.1731 Protocol (cfm), 79 bytes

✓ Show packet bytes

Help Close

Testing

- IEEE802.1Q
- Various vendors loop testing protocols



Testing

- IEEE802.1Q
- Various vendors loop testing protocols
- Spanning Tree (of various flavors)

SONiC's lazy workarounds

- SONiC is a free NOS, and has a script called arp_update [1]
- This script pings ff02::1 (all devices) so it can update the neighbor table…
  - This appears to be a "hack"/workaround so they don't need hardware counters for next-hops
  - At the expense of sending crap into broadcast domains
  - If more than a handful for IX members are running SONiC, this would be even more obnoxious

No.	Time	Source	Destination	Protocol	Length	Info
2728…	06:08:44.43490…	fe80::30b1:e8ff:feb6:2c6d	ff02::1	ICMPv6	118	Echo (ping) request id=0xa81f, seq=1, hop limit=1 (multicast)
2728…	06:11:21.10683…	fe80::b6a9:fcff:fe6f:8b23	ff02::1	ICMPv6	118	Echo (ping) request id=0x467a, seq=1, hop limit=1 (multicast)
2728…	06:11:21.10686…	fe80::ce37:abff:fecb:1c	ff02::1	ICMPv6	118	Echo (ping) request id=0x0577, seq=1, hop limit=1 (multicast)
2728…	06:11:21.10688…	fe80::b6db:91ff:fe8b:3d66	ff02::1	ICMPv6	118	Echo (ping) request id=0xa95c, seq=1, hop limit=1 (multicast)
2728…	06:11:21.10688…	fe80::aca7:76ff:fe57:a70a	ff02::1	ICMPv6	118	Echo (ping) request id=0x6590, seq=1, hop limit=1 (multicast)
2728…	06:14:09.04282…	fe80::2e0:ecff:fee4:dbcc	ff02::1	ICMPv6	118	Echo (ping) request id=0x0251, seq=1, hop limit=1 (multicast)
2728…	06:14:09.04285…	fe80::30b1:e8ff:feb6:2c6d	ff02::1	ICMPv6	118	Echo (ping) request id=0xa856, seq=1, hop limit=1 (multicast)
2728…	06:14:09.04287…	fe80::b6a9:fcff:fe6f:8b23	ff02::1	ICMPv6	118	Echo (ping) request id=0x8bc3, seq=1, hop limit=1 (multicast)
2728…	06:16:50.83516…	fe80::ce37:abff:fecb:1c	ff02::1	ICMPv6	118	Echo (ping) request id=0x0201, seq=1, hop limit=1 (multicast)
2728…	06:16:50.83543…	fe80::b6db:91ff:fe8b:3d66	ff02::1	ICMPv6	118	Echo (ping) request id=0xabb9, seq=1, hop limit=1 (multicast)
2728…	06:16:50.83545…	fe80::aca7:76ff:fe57:a70a	ff02::1	ICMPv6	118	Echo (ping) request id=0x0f87, seq=1, hop limit=1 (multicast)
2728…	06:16:50.83551…	fe80::2e0:ecff:fee4:dbcc	ff02::1	ICMPv6	118	Echo (ping) request id=0x0fc5, seq=1, hop limit=1 (multicast)
2728…	06:19:38.77086…	fe80::30b1:e8ff:feb6:2c6d	ff02::1	ICMPv6	118	Echo (ping) request id=0xa88d, seq=1, hop limit=1 (multicast)
2728…	06:19:38.77088…	fe80::b6a9:fcff:fe6f:8b23	ff02::1	ICMPv6	118	Echo (ping) request id=0x5f10, seq=1, hop limit=1 (multicast)
2728…	06:19:38.77089…	fe80::ce37:abff:fecb:1c	ff02::1	ICMPv6	118	Echo (ping) request id=0x1b01, seq=1, hop limit=1 (multicast)
2728…	06:22:26.70685…	fe80::b6db:91ff:fe8b:3d66	ff02::1	ICMPv6	118	Echo (ping) request id=0xdb22, seq=1, hop limit=1 (multicast)

[1] https://github.com/sonic-net/sonic-buildimage/blob/0fa211d531766b953ce280d21c56075f7b2a85b4/files/scripts/arp_update#L71

SONiC's lazy workarounds

- SONiC is a free NOS, and has a script called arp_update [1]
- This script pings ff02::1 (all devices) so it can update the neighbor table…
  - This appears to be a "hack"/workaround so they don't need hardware counters for next-hops
  - At the expense of sending crap into broadcast domains
  - If more than a handful for IX members are running SONiC, this would be even more obnoxious

No.	Time	Source	Destination	Protocol	Length	Info
2728…	06:08:44.43490…	fe80::30b1:e8ff:feb6:2c6d	ff02::1	ICMPv6	118	Echo (ping) request id=0xa81f, seq=1, hop limit=1 (multicast)
2728…	06:11:21.10683…	fe80::b6a9:fcff:fe6f:8b23	ff02::1	ICMPv6	118	Echo (ping) request id=0x467a, seq=1, hop limit=1 (multicast)
2728…	06:11:21.10686…	fe80::ce37:abff:fecb:1c	ff02::1	ICMPv6	118	Echo (ping) request id=0x0577, seq=1, hop limit=1 (multicast)
2728…	06:11:21.10688…	fe80::b6db:91ff:fe8b:3d66	ff02::1	ICMPv6	118	Echo (ping) request id=0xa95c, seq=1, hop limit=1 (multicast)
2728…	06:14:09.04287…	fe80::b6a9:fcff:fe6f:8b23	ff02::1	ICMPv6	118	Echo (ping) request id=0x8bc3, seq=1, hop limit=1 (multicast)
2728…	06:16:50.83516…	fe80::ce37:abff:fecb:1c	ff02::1	ICMPv6	118	Echo (ping) request id=0x0201, seq=1, hop limit=1 (multicast)
2728…	06:16:50.83543…	fe80::b6db:91ff:fe8b:3d66	ff02::1	ICMPv6	118	Echo (ping) request id=0xabb9, seq=1, hop limit=1 (multicast)
2728…	06:16:50.83545…	fe80::aca7:76ff:fe57:a70a	ff02::1	ICMPv6	118	Echo (ping) request id=0x0f87, seq=1, hop limit=1 (multicast)
2728…	06:16:50.83551…	fe80::2e0:ecff:fee4:dbcc	ff02::1	ICMPv6	118	Echo (ping) request id=0x0fc5, seq=1, hop limit=1 (multicast)
2728…	06:19:38.77086…	fe80::30b1:e8ff:feb6:2c6d	ff02::1	ICMPv6	118	Echo (ping) request id=0xa88d, seq=1, hop limit=1 (multicast)
2728…	06:19:38.77088…	fe80::b6a9:fcff:fe6f:8b23	ff02::1	ICMPv6	118	Echo (ping) request id=0x5f10, seq=1, hop limit=1 (multicast)
2728…	06:19:38.77089…	fe80::ce37:abff:fecb:1c	ff02::1	ICMPv6	118	Echo (ping) request id=0x1b01, seq=1, hop limit=1 (multicast)
2728…	06:22:26.70685…	fe80::b6db:91ff:fe8b:3d66	ff02::1	ICMPv6	118	Echo (ping) request id=0xdb22, seq=1, hop limit=1 (multicast)

```
ping6cmd="timeout 0.2 ping6 -I $intf -n -q -i 0 -c 1 -W 0 ff02::1 >/dev/null"
```

[1] https://github.com/sonic-net/sonic-buildimage/blob/0fa211d531766b953ce280d21c56075f7b2a85b4/files/scripts/arp_update#L71

Stuff that just should not be here

- Broadcast NTP
  - This is just rude, spamming NTP time packets into the LAN is not useful
  - All times where this has been detected, it's been a Mikrotik MAC address
  - The time being broadcast was also often wrong

Stuff that just should not be here

- Broadcast NTP
- MikroTIK RoMON/WinBox
  - MikroTik "Router Management Overlay Network" allows you to hop-by-hop connect and manage MikroTik devices. The data is not encrypted in most cases

Stuff that just should not be here

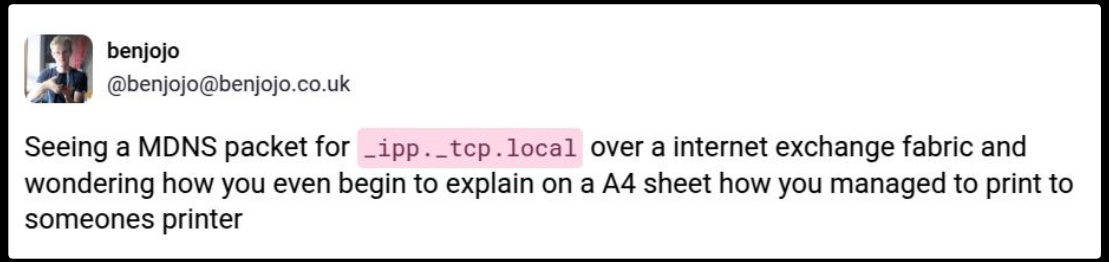
- Broadcast NTP
- MikroTIK RoMON/WinBox
- DEC-MOP
  - Default enabled on Cisco Enterprise IOS releases on all ethernet interfaces
  - DEC is for OpenVMS deployments, It feels extremely overdue being disabled by default

Stuff that just should not be here

- Broadcast NTP
- MikroTIK RoMON/WinBox
- DEC-MOP
- SSDP (UPNP)
  - Typically SOHO routers have this enabled, I have no idea why these kinds of routers are on IX LANs

Stuff that just should not be here

- Broadcast NTP
- MikroTIK RoMON/WinBox
- DEC-MOP
- SSDP (UPNP)
- LLMNR / MDNS
  - Link-Local Multicast Name Resolution packets are almost always generated by systemd-resolved, as it is enabled by default, and more and more people are moving to "Linux Software Routers"
  - Multicast DNS is common as well on Linux Software Routers, triggered normally by CUPS (the printer system) or Avahi



Stuff that just should not be here

- Broadcast NTP
- MikroTIK RoMON/WinBox
- DEC-MOP
- SSDP (UPNP)
- LLMNR / MDNS
- NETBIOS
  - Desktop Windows does this by default, Server windows does this if the network is set to trusted, I would typically not expect Windows computers to be in IXPs
  - Fun fact: There is a BGP Implementation in windows, it is not very good:
    - https://blog.benjojo.co.uk/post/flexing-windows-rras-bgp-stack-ipv6

Stuff that just should not be here

- Broadcast NTP
- MikroTIK RoMON/WinBox
- DEC-MOP
- SSDP (UPNP)
- LLMNR / MDNS
- NETBIOS
- VRRP
  - Or HSRP I guess, But it's the Router/Gateway High Availability protocol

Stuff that just should not be here

- Broadcast NTP
- MikroTIK RoMON/WinBox
- DEC-MOP
- SSDP (UPNP)
- LLMNR / MDNS
- NETBIOS
- VRRP
- DNS-Broadcast

DNS-Broadcast

```
$ tcpdump -nr dns.pcap |& pcregrep -o1 '\] A{1,4}\? ([^\)]+)' | awk '{print $1}' | sort |
uniq -c | sort -n | tail -n 15
     875 cslu-local.{CORP-F}
     998 cslu-local.{CORP-F}
    1162 tools.cisco.com.{Military-A}.
    1648 cslu-local.{CORP-G}
    1659 cslu-local.{CORP-F}
    1880 cslu-local.{CORP-E}
    2088 tools.cisco.com.{CORP-A}.
    2910 cslu-local.{CORP-D}
    4213 cslu-local.{Military-A}.
    5125 cslu-local.{CORP-C}.com.
    5515 cslu-local.{CORP-B}.fr.
    7367 cslu-local.{CORP-A}.com.
    7675 tools.cisco.com.
    9934 cslu-local.{Military-A}
   10838 cslu-local.
```

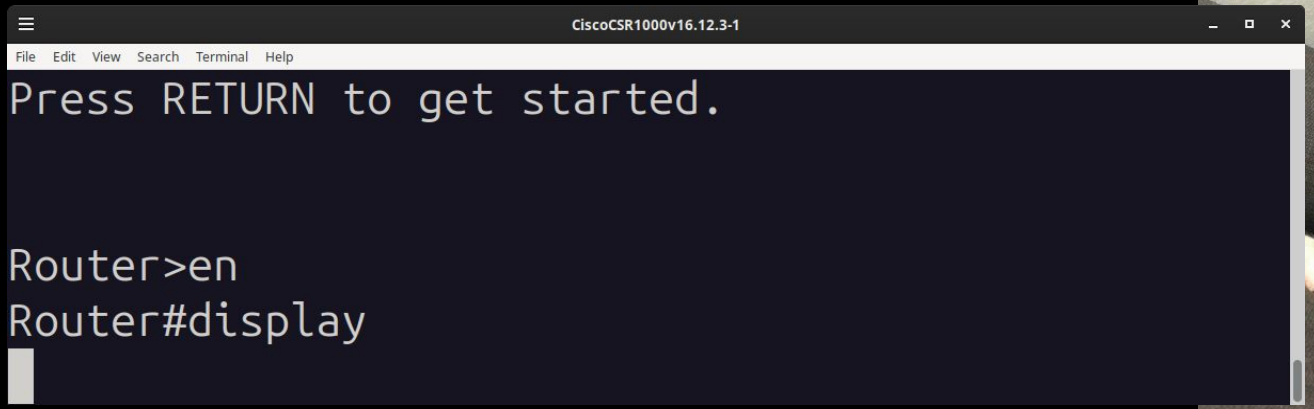
DNS-Broadcast

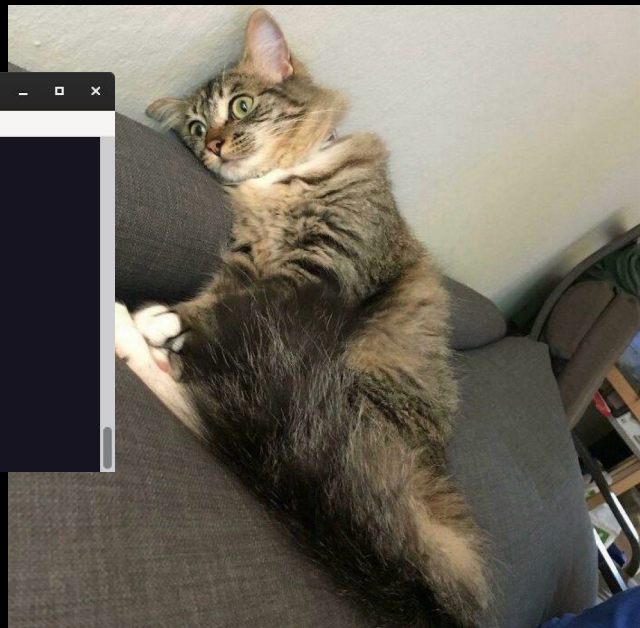
```
$ tcpdump -nr dns.pcap |& pcregrep -o1 '\] A{1,4}\? ([^\)]+)' | awk '{print $1}' | sort |
uniq -c | sort -n | egrep -v  'cslu|tools.cisco.com'

     1 cls.basetelco.com.
     1 configre.jato3.com.
     1 conft.asn28176.com.br.
     1 conft.powernet.net.br.
     1 cont.wanfiber.net.br.
     1 cpnf.cd.net.za.
     1 end.3cta.eb.mil.br.
     1 end.as37497.net.
     1 end.cd.net.za.
     1 end.spnet.com.br.
     1 exiexit.cd.net.za.
     1 exit-address-family.
     1 exitr.jfsc.local.
     1 expression.jato3.com.
```

DNS-Broadcast

Normally Cisco's do this, Half of the time it's because someone typo'd on the CLI, other half is the device looking for a working DNS resolver





DNS-Broadcast / Top Typo's on the CLI

```
$ tcpdump -nr dns.pcap |& pcregrep -o1 '\] A{1,4}\? ([^\)]+)' | awk '{print $1}' |& pcregrep -o1 '^([^.]+)' | sort | uniq -c | sort -n
...
      1 access-list
      1 configre
      1 cont
      1 cpnf
      1 exiexit
      1 exit-address-family
      1 exitr
      1 exti
      1 extit
      1 ifconfig
      1 int
      1 interface
      1 qconft
      1 qqq
      1 reboot
      1 Routes-Advertised-To-IX
      1 shut
      1 test
      1 uptime
      1 vf
      1 virma
      1 vr10
      1 vrf
      2 coinf
      2 cssm
      2 ext
      2 hidekeys
      2 ipv6
      2 load-interval
      2 ls
      2 next-hop-override
      2 Please
      2 Port-channel1
      2 Port-channel20
      2 shorun
      2 top
      2 vlan
      2 vrf-mgmt
      3 address
      3 cls
      4 history
      4 ip
      4 ping
      6 bgp
      6 save
      9 conft
      9 end
     10 pool
     87 y
     92 quit
    134 q
    289 summary
  58101 cslu-local
```

What can IXs do about this?

What does bgp.tools detect? And how is it to filter out?

Easy L2

- DEC-MOP
- RoMON
- STP
- CDP
- IS-IS
- ES-IS
- LLDP
- VRRP
- IEEE 802.1Q

Easy L3

- LLMNR
- OSPFv2 /OSPFv3
- NetBIOS
- PIM
- OSPF
- LDP
- MDNS
- DHCPv4 /DHCPv6
- SSDP
- DNS-Broadcast
- Broadcast NTP
- MikroTik Discovery

Hard L3

- RA
  - **You cannot block ICMPv6 otherwise the IX breaks**
  - So you need the ability to ACL on ICMPv6 type, and that is not possible on a lot of things

Filtering these things out (If you are a IX operator)

- Until *relatively* recently, most common IX switches didn't have a good ability to ACL these L3 things off
  - You can however most of the time target the multicast group MAC address they send to
  - However you should have or look to have L3 filtering capabilities by now so you can do BGP Session culling (BCP 214 / RFC 8327)
  - However *again* some switches are unable to filter L3 broadcast traffic correctly, check your manuals or test environments
- Another route is to "simply" not have a real broadcast domain
  - More IXs are using EVPN, and depending on how you set it up, multi/broadcast traffic is just ignored
  - Still depends on vendors, a "standard" Juniper/Arista setup will still leak B.U.M traffic between the same switch, Nokia setups typically don't (see DE-CIX's very quiet broadcast domains)

If you look carefully, nearly all of these are L2 ACL-able!

- Things like OSPF/IS-IS etc use specific MAC addresses
- Same with the CDP/LLDP/STP/VRRP's of the world
- DHCP is harder (since it's just broadcast mac, same as ARP)
- IPv6 RA's can be filtered out by (multicast) MAC address alone

- So there is actually no excuse here to be filtering most of this!
- Two kinds of providers:
  - "I hope my customers are competent"
  - "I protect against my customers being incompetent ***just in case***"

Enforcement (If you are a IX operator)

- Tools like ixp-watch exist to generate reports on your broadcast domain
  - https://github.com/euro-ix/IXP-Watch (LONAP and LINX have run this for many years!)
- If bgp.tools/AS212232 is on your exchange, you can keep an eye on bgp.tools!
  - If AS212232 is not on your exchange, why not? :)
  - I will probably start sending email alerts if there is interest in that (Please put your hand up in Q&A if you would like that)

- Have a policy for what you are going to do when someone is sending silly packets into the fabric
  - This policy will probably have to be different to say, a L2 loop, because it's not life threatening

Thanks!

Questions? Comments? Stories?
Shy? Email netuk@benjojo.co.uk
(or fedi/mastodon @benjojo@benjojo.co.uk)