

# RomandIX

## Year one.

# Agenda

- Presenter Bio
- What is RomandIX
- Why another IXP
- What we did
- Lesson learned / Next steps
- Questions

# Bio

## -Nicolas Desir

Founder of Saitis (AS6893) an ISP in Lausanne, working with community networks for 20+ years, Datacenter builder, hardware tinkerer [...]

## -Will van Gulik

Running AS2613, Network Engineer at AS6893, Ripe Connect-WG co-chair, Ripe Atlas Ambassador, MANRS advocate, Dj, Writer [...]

# What is RomandIX

RomandIX is a Lausanne IXP founded as a non-profit organisation the 13<sup>th</sup> March 2017  
by :

- AlpineDC – AS198385
- AS2613
- DFI Services – AS12333
- EDSI-Tech – AS202194
- Evok – AS203882
- Infraly – AS50963
- Openbusiness – AS49457
- Saitis Networks – AS6893
- Telcopack – AS206214

Currently 9 Active networks, 3 more connecting

# Why another IXP

- Keep your traffic local
- Simpler interconnections (no full mesh)
- Additional services
- We were bored and didn't have enough stuff to do
- The Fun ;)

# What we did #1

- Build an association
- Get IPv4/6 space and an ASN from Ripe NCC
- Buy a switch and rack it (Arista 7050)
- Get two servers for our services in two different locations (AlpineDC room in Brain hosted by EDSI and in Provence hosted by Saitis)
- Create the PeeringDB entries / IRR Records
- Use virtualization, VMs, Install IXP Manager and ...

## What we did #2

... install the Route Servers, activate filtering day one, daily updates

```
./gen_prefix_filters_verbose.sh  
[...]  
AS2613: [IPv4: 1 total; 0 stale; 0 new; DB updated] [IPv6: 3 total; 0 stale; 0 new; DB updated]  
PCH3856: [IPv4: 1 total; 0 stale; 0 new; DB updated] [IPv6: 1 total; 0 stale; 0 new; DB updated]  
PCH42: [IPv4: 272 total; 0 stale; 0 new; DB updated] [IPv6: 237 total; 0 stale; 0 new; DB updated]  
Saitis: [IPv4: 72 total; 0 stale; 0 new; DB updated] [IPv6: 69 total; 0 stale; 0 new; DB updated]  
Telcopack: [IPv4: 35 total; 0 stale; 0 new; DB updated] [IPv6: 6 total; 0 stale; 0 new; DB updated]  
ROIX: [IPv4: 0 total; 0 stale; 0 new; DB updated] [IPv6: 0 total; 0 stale; 0 new; DB updated]  
Customer not a RS client or IRRDB filtered. Prefixes, if any, wiped from database.  
[...]
```

- Configure the switch with a Reseller VLAN mechanism, one VLAN per port
- But what's the point, everyone can do that. Nico wanted to do more.



WOULD YOU LIKE TO KNOW MORE?

# Let's hack a switch!

Broadcast/mac learning, ARP/NDP inspection, sponge: Is there another paradigm?

- We need something not too expensive/energy consuming
- NDP inspection can't be found today on available second hand hardware like ours
- Is there an alternative way?
- Can we also solve other problems:
  - broadcast when mac is forgotten?
  - ARP broadcast load?



# L2 Part

- Separation first
- Each participant in a separate VLAN
- Let's give a try to the directflow feature of Arista

```
flow to_peer_001_ipv4
  Table Trident ifp
  match destination mac 42:07:f8:b5:00:00 mask ff:ff:ff:ff:f0
  match ethertype ip
  match vlan 512 mask 3584
  action output interface Et47
  action set vlan 512
!
flow to_peer_001_ipv6
  Table Trident ifp
  match destination mac 42:07:f8:b5:00:00 mask ff:ff:ff:ff:f0
  match ethertype 34525
  match vlan 512 mask 3584
  action output interface Et47
  action set vlan 512
!
```

# L3 Part

It won't work, ARP and NDP need [b/m]cast !

- No inspecting/filtering: let's generate ARP and NDP response
- so easy access to the shell in EOS
- executable build on debian7/amd64 run on the switch
- each VLAN can be seen as a standard interface in the switch

# ARP: the simple case

- arp-dictator : a home made simple executable
- static config file which IP/MAC mapping
- one per VLAN with rate-limiting

```
bash-4.3# pwd
/mnt/flash/bin
bash-4.3# cat v4tomac
185.1.93.1 42:07:F8:B5:00:01
185.1.93.2 42:07:F8:B5:00:02
185.1.93.3 42:07:F8:B5:00:11
185.1.93.4 42:07:F8:B5:00:04
185.1.93.5 42:07:F8:B5:00:05
[...]
```

# NDP: the tricky case

- ndp-dictator
- must answer for the real and link-local addresses
- multicast
- no specific Ethernet protocol
  - directflow will forward unicast response
  - can be faked
  - we must filter
- let's deny inbound all ND except Neighbor Solicitation

```
ipv6 access-list ndonlyns
  20 deny icmpv6 any any 133
  30 deny icmpv6 any any 134
  40 deny icmpv6 any any 136
  50 deny icmpv6 any any 137
  60 permit ipv6 any any

interface Ethernet11
  switchport access vlan 11
  ipv6 access-group ndonlyns in
```

# NDP: same idea as ARP

```
bash-4.3# cat v6tomac  
# [syntax ipv6 address] [mac]  
2001:7f8:b5::10:1 00:1b:21:99:a4:21  
fe80::21b:21ff:fe99:a421 00:1b:21:99:a4:21  
2001:7f8:b5::11:1 e4:8D:8C:A4:B7:4E  
fe80::e68d:8cff:fea4:b74e e4:8D:8C:A4:B7:4E  
[...]
```

- local IPv6 addresses are made with MAC addresses
- will be automatic in the future ;)

# Caveats

- Broken ARP/NDP reachability information
  - to be addressed/implemented by dynamic checks and communication between daemons
  - BGP's responsibility
- AFAIK not so big and bad problem:
  - ARP can be also very slow

# Mac Addressing

- Can be configured on more and more devices
- Some still cause trouble
- Exception are accepted
- Better to give the MAC address to the participant
  - insure the participant is able to configure it
  - participant will be able to do it again when choosing/changing equipment

<https://kb.juniper.net/InfoCenter/index?page=content&id=KB28918> :

"As confirmed by the example above, the MAC address configured under the edit interfaces IRB unit <xyz> hierarchy is not effective for the inet6 family.  
This has always been the case. There are no plans to include this support in the future."

## MAC Addressing #2

-Got an OUI-36 for RomandIX:

-<http://standards-oui.ieee.org/oui36/oui36.txt>

```
[...]
70-B3-D5          (hex)      Association Romandix
AEB000-AEBFFF    (base 16)
      Association Romandix
                  rue de Sebeillon 9b
                  Lausanne Vaud 1004
                  CH
[...]
```

-Arista works with mac mask

-one directflow entry can accommodate several peers

-Participant get a /44 on their port

-So no excuse of possible conflicts...



## Also an other political paradigm?

- Common IXP way
  - collaborative (democratic/free speech)
  - inspection/blocking (police)
  - quarantine (jail)
- This proposed way
  - separation (Divide),
  - and dictate (Rule)
  - suppress unwanted packets (kill)

Todo for L2/L3 part :

- code cleaning and optimization
- dynamic peer accessibility between daemon (e.g. via IPC)
- ring for maximum performance: PACKET\_RX\_RING/PACKET\_TX\_RING
- external code/methodology reviewing (YOU?)

# Lessons learned / What's next

## Conclusions

- Our switches run Linux, we used it, so could you
- Doing things non-profit takes longer
- We suck at web mastering

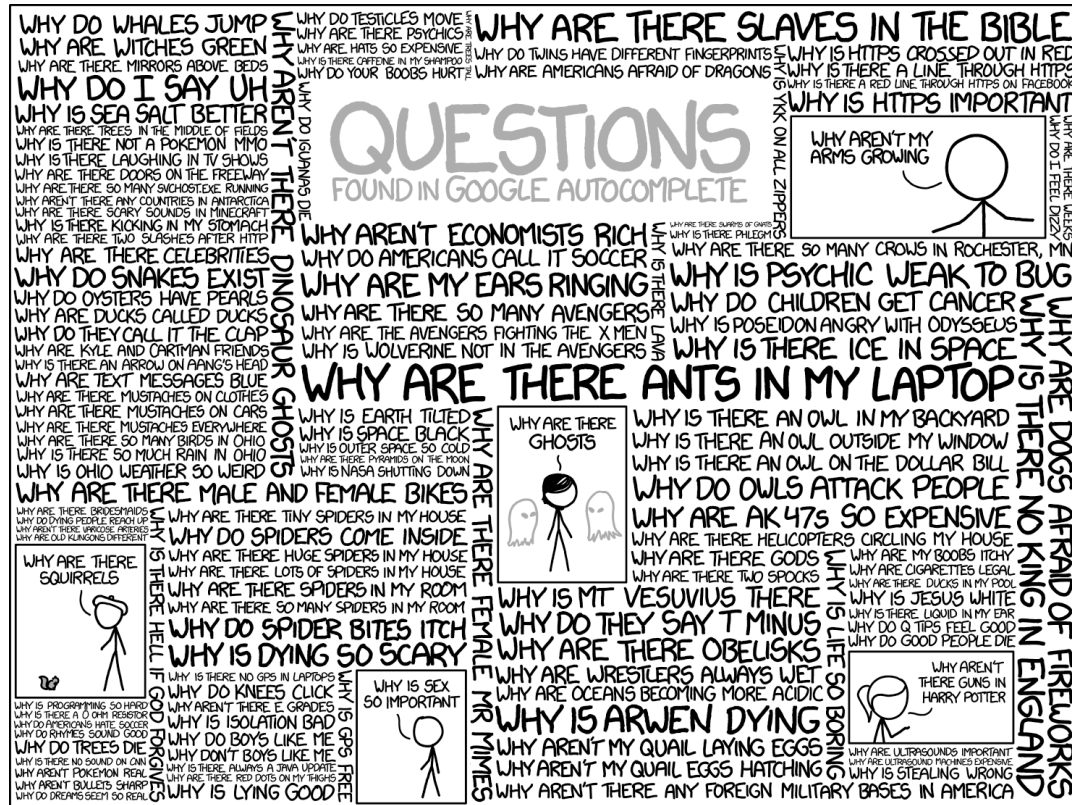
But :

- First Swiss member of MANRS for IXPs
- We enforce IRR based filtering on our RS

Todo :

- Reach ZRH (Colozüeri)
- Reach GVA (Cern - IXEu^H^H^HEquinix's GV1/2)
- More members (you ?), more traffic, rpki, röstigraben packet translation, ...
- Our tools are available on  
<https://www.romandix.ch/code/>
- Contact us : [info@romandix.ch](mailto:info@romandix.ch)

# Questions ?



- Thanks to xkcd.com for the image
- Thanks to Sébastien Keller, Florian Hibler and Romain Aviolat
- <https://Romandix.ch> (com/net/org)
- <https://peeringdb.com/ix/1984>
- <https://www.manrs.org/ixps/>