Out with IRR, in with RPKI

Job Snijders - job@bsd.nl

Photo source: https://www.nationalparks.nsw.gov.au/things-to-do/events/new-years-eve-shark-island

Agenda

- Recap, what is it route filtering / routing security is trying to solve?
- Analysis of previous approach
- Plans for next approach
- Recommendations
- Q & A

What this talk is not about

This talk is ***not*** about advocating to only accept RPKI-valid routes.

The major risks to mitigate

- Misorigination of route announcements (aka *BGP Hijacking*)
- Unauthorized more-specifics (e.g. a government blackholing 1 IP)
- BGP Route leaks (see RFC 7908 for taxonomy)
- Spoofing of BGP information (see RFC 4272)

What folks have done over the years

Bolt one mechanism on top of the other:

1. Set max prefix limits
2. Add IRR based filters
3. Add RPKI based filters

In the future…

4. Add ASPA based filters
5. Add SPL based filters
6. Add XYZ …

Seems the industry has mostly just been “adding $new_thing on top of $existing”

The old approach (in Fastly)

- Daily generation of per-peer specific IRR-based filters (using bgpq4)
  - This IRR-based filter is used as an “allow list” what prefixes may be accepted from a peer
  - AS-SET used as input comes from PeeringDB or is manually set
- Set maximum prefix limit
  - When the threshold is reached the session is terminated under the assumption that whatever caused the sudden influx is bad for business

Analysis of old approach

A. IRR-based filtering is *very expensive*
   a. *IPv4: 173 MB (14,647,235 lines) - IPv6: 51 MB (3,901,510 lines)*
      i. *Cost is in CPU loading such filters and in memory holding the filters*

B. IRR-based filtering is *inefficient*
   a. *The larger an allowlist is, the less it blocks, the less efficient it is*
      i. *Both AS1273:AS-CWW and AS9498:AS-BHARTI-IN expand to 140,157 lines*

C. Maximum prefix limits **are** *effective*
   a. *Common case is a peer announcing a full routing table, max prefix blocks that*

D. IRR-based filtering doesn’t help against spoofing
   a. Spoofing is not stopped via IRR-filtering

Zooming in specifically on IRR data

- IRR is a plain-text database system from the 90s
- Seems to mostly just grow, *and grow, and grow* over time
- Objects tend to only get removed in two cases:
  - IRR route object became RPKI-invalid (RIPE-731, IRRdv4)
  - IRR maintainer terminates membership / service contract with RIR or RADB

A suggestion for a path forward

A. Continue to use maximum prefix limits
B. Stop using IRR-based filtering entirely
C. Use RPKI to reject RPKI-invalid routes on EBGP sessions
D. Use “peerlock-lite” on peering sessions
E. Use BGP-OPEN (RFC 9234) to help mitigate route leaks
F. Use ASPA (RFC-in-progress) to help mitigate route leaks

Analysis of new approach

- RPKI coverage *greatly improved* in recent years
  - More than 50% of routing table entries covered by ROAs
  - More than 75% of IP traffic is heading towards RPKI-valid destinations
- RFC 9234 is incrementally deployable
  - Supported in BIRD, OpenBGPD, FRR
  - Already stopped an IX-to-IX routeleak (source)
- ASPA is incrementally deployable
  - We can learn from ROA/ROV deployment lifecycle: community has to try to prevent proliferation of misconfigured ROAs before networks deploy validation.
- Spoofing still an issue
  - Remains an open issue, but we are no worse off today than yesterday

Analysis of new approach (continued)

- RPKI data is delivered to routers in a binary format
  - 3 copies of VRPs for every DFZ entry fits in a handful of megabytes
- RPKI data can be delivered using industry standards (RPKI-To-Router)
  - IRR-based data requires bespoke transformation via template uploading via SSH
- RFC 9234 is cheap on both CPU and memory
  - No cryptography involved
  - An extra 4 byte BGP Path Attribute per route
  - In Fastly we’ve already seen a few incidents avoided thanks to limited RFC 9234 deployment
- ASPA offers a granularity for anti-leak filters which simply is unattainable via IRR-driven configuration

Zooming in specifically on RPKI data

- RPKI is a cryptographically-protected distributed database
- Authority to issue ROAs is tied to the current INR holder
- Growth is good: more ROAs means more coverage (source)
- No proxy registrations that linger forever and ever
- ROA semantics differ from IRR route object semantics
    - ROAs tell us something about what should *not* exist, IRR objects only say something about the IRR object itself

Far better return-on-investment for RPKI-derived data

Takeaways for other operations

- IX route servers configurations are enormous, but are those filters *really* doing what the RS operator & participants hope they achieve?
- How relevant are AS-SETs (to guestimate customer cones) *really* in a world with RFC9234 & ASPA?
- If you use BIRD, OpenBGPD, FRR - already today you can deploy RFC 9234

Now might be time to re-evaluate existing routing security measures!

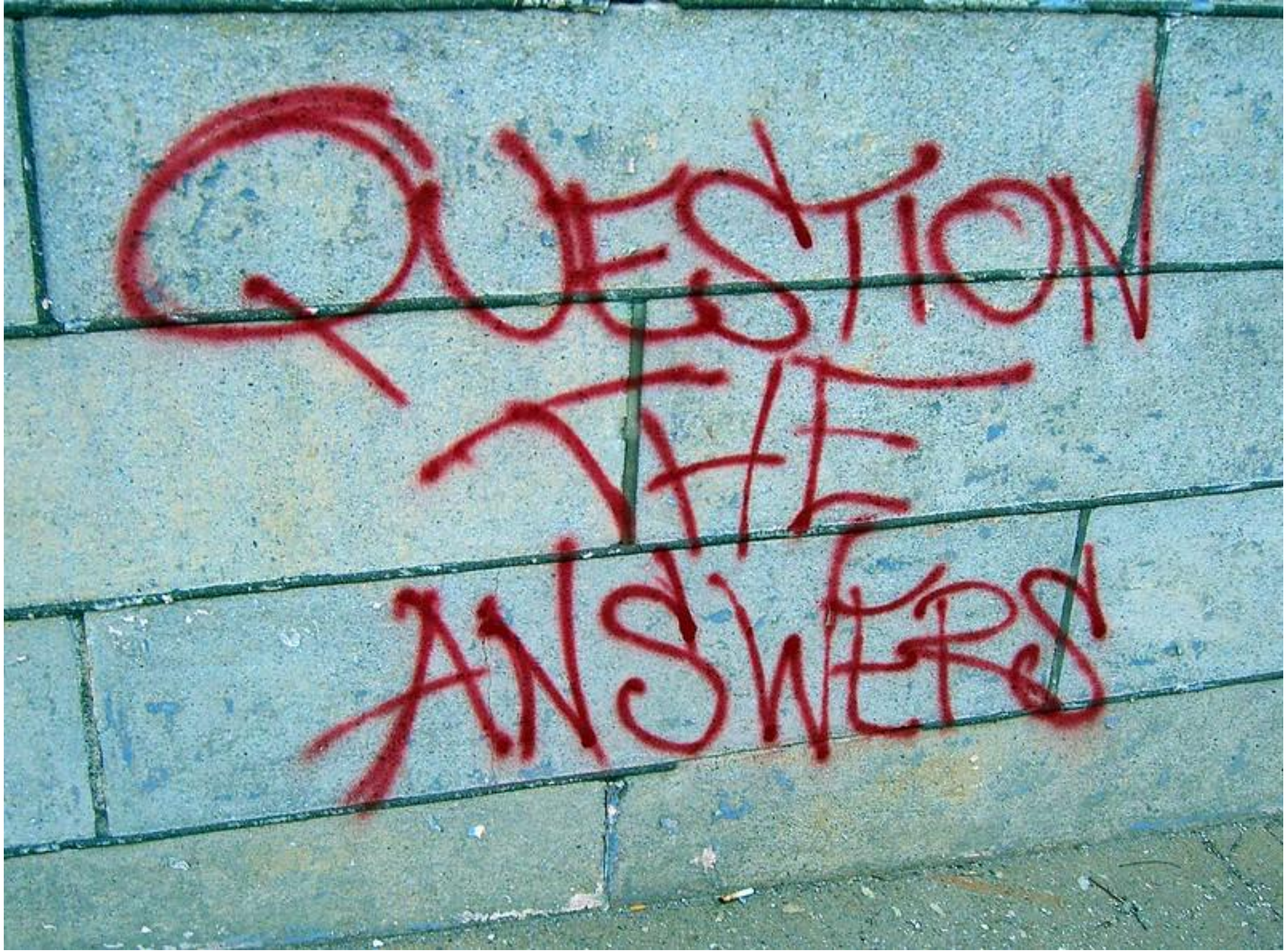

Photo source: https://www.flickr.com/photos/walkn/3526522573