OARC 47 VANCOUVER

Trust your paths

Routing security for DNS infrastructure operators

Ritesh Mukherjee

ANYCAST DNS RPKI / ROV ASPA BMP

You instrument everything above the routing layer

YOUR DASHBOARDS WATCH

- Query rate and QPS per instance
- Response latency percentiles
- RCODE mix — NOERROR, SERVFAIL, NXDOMAIN
- DNSSEC validation failures
- Zone serial propagation

THEY DON'T SEE

- Who is originating your prefix right now
- Whether a more specific one exists somewhere
- Which AS path traffic takes to reach you
- Why a query landed on the instance it did
- Whether the "instance" is yours at all

When routing breaks, DNS signals often **stay healthy** — right up until they don't.

Three incidents, three different failures

Each needed a different mechanism to catch it

2018

Amazon Route 53

More-specific hijack of an authoritative DNS service. Resolvers worldwide cached forged answers.

2023 – 2025

Root server prefixes

Repeated origin hijacks — mostly invisible because anycast localizes the damage.

2024

Cloudflare 1.1.1.1

An origin hijack and a separate route leak in the same hour. ROV catches one of them.

INCIDENT 1 — AMAZON ROUTE 53

April 24, 2018

11:05 UTC

eNet (AS10297), an ISP in Columbus, Ohio, starts announcing **five /24s carved out of Route 53's authoritative DNS space.**

The Cascade

AS10297 announces 205.251.192.0/24, .193, .195, .197, .199

▼

Peers prefer the more-specifics over Amazon's /23

▼

Resolvers querying Route 53 reach an impostor authoritative

▼

It answers for one name only: myetherwallet.com

▼

Users land on a phishing clone with a self-signed certificate

The Impact

~2 hrs	bogus routes present, 11:05 – 13:03 UTC
~1,300	Route 53 IP addresses covered by the hijacked /24s
~$150K	in Ether drained from the wallets of users who clicked through the certificate warning
0	AWS systems compromised — the attack lived entirely in routing

You didn't have to accept the route to be hit

Large public resolvers—including 1.1.1.1—sat behind routers that accepted the bogus /24s. They queried the impostor and cached its answers for everyone behind them.

Caches outlive the hijack

Withdrawing the route doesn't flush a resolver. A forged answer keeps being served until its TTL runs out.

ROV would have caught it

A ROA for the /23 with maxLength /23 makes every one of those /24s — from any origin — RPKI-Invalid.

DNSSEC limits the damage

A signed zone and a validating resolver turn a forged answer into SERVFAIL. Integrity holds; reachability doesn't.

INCIDENT 2 — ROOT SERVER PREFIXES

A recurring target

Jan 2023 — **F-root IPv6** originated by AS24028. Seen by a single RIS peer — the prefix had a ROA

Feb 2023 — **E-root and F-root IPv6** originated by AS17639 for 18+ hours. E-root had no ROA — its blast radius was **about 3× larger** than F-root's.

Apr 2023 — **L-root IPv4** originated by AS137661, prepended seven times. Best-path selection hid it from almost every collector.

Jun 2023 — **K-root IPv4** announced from a hosting partner's own AS instead of AS25152. Visible only in Italy.

Jun 2025 — **Eight root prefixes, including A-root,** originated with AS35168 — the first documented BGP hijack of root server prefixes.

Jun 2025 — **L-root IPv4** originated by AS3132. Only 19 BGP monitors saw it, yet ~10% of ThousandEyes agents sent L-root queries through it.

Sources: ThousandEyes / Code BGP root-prefix monitoring; RSO statement on the June 2025 hijack.

JUNE 20, 2025

Eight root identifiers, one bogus origin

AS35168
TNS-Plus, Kazakhstan

bogus origin

AS28910
Uzbektelekom — no ROV

Regional resolvers
query unauthorized "roots"

Further spread stopped by policy, topology, best-path —not by validation.

A-root 198.41.0.0/24	**B-root** 170.247.170.0/24	**C-root** 192.33.4.0/24	**F-root** 192.5.5.0/24
G-root 192.112.36.0/24	**H-root** 198.97.190.0/24	**J-root** 192.58.128.0/24	**M-root** 202.12.27.0/24

Three of these prefixes had covering ROAs. It didn't matter: the one network that accepted the routes **doesn't do Route Origin Validation.** Verisign's own tooling raised the alarm within two minutes.

19:40 routes appear · 19:42 alert · 21:10 withdrawn

Anycast is camouflage

The property that makes anycast resilient is the same one that hides a hijack from you.

The damage is local

Only the catchment behind the bad route is affected. Global health checks from your usual vantage points stay green.

Best-path hides it

A router only propagates its best path. A bogus route that loses locally never reaches the collectors — L-root in 2023.

Seeing it takes your own vantage

Verisign caught 2025 in two minutes with its own tooling. Your edge routers already receive what your peers send — if you look.

ROAs measurably shrink the blast radius. Same hijacker, same day: the unsigned E-root prefix spread about three times farther than the signed F-root prefix.

INCIDENT 3 — JUNE 27, 2024

1.1.1.1

An origin hijack and a route leak of the same resolver,
starting within one minute of each other.

Unreachable or degraded from 300+ networks in 70 countries

1.1.1.1 — what happened, minute by minute

18:51 **AS267613 (Eletronet)** starts originating **1.1.1.1/32** to peers, providers, and customers.

18:52 **AS262504 (Nova)** leaks **1.1.1.0/24** to its upstream AS1031. Path: 1031 262504 267613 13335.

18:52 **AS1031** redistributes the leak to IX peers and route servers. A Tier 1 takes the /32 as a **blackhole route**.

20:03 Cloudflare raises an internal incident — **72 minutes** after onset.

20:10 The leak reappears via 262504 53072 7738 13335. Queries now **succeed — with high latency.**

22:16 Blackholing appears to be resolved after Cloudflare shuts down two peering points with AS267613. The leak continues to pull traffic to São Paulo.

02:28 +1 AS262504 fully withdraws the leak — **7.5 hours** after it began.

All times UTC. Source: Cloudflare incident report, July 4, 2024.

Two failures, two different verdicts

Cloudflare's ROA: 1.1.1.0/24 → AS13335, maxLength /24

THE HIJACK

1.1.1.1/32

origin AS267613

Wrong origin *and* longer than maxLength. A /32 also breaks the usual "nothing longer than /24 in the DFZ" filter.

ROV: INVALID

THE LEAK

1.1.1.0/24

path 1031 262504 267613 13335

Right prefix, right origin. The problem is the path: a customer re-announced a peer's route to its own provider.

ROV: VALID

The leak was **RPKI-valid**. Every router doing ROV accepted it.

What users — and DNS monitoring — saw

SYMPTOM 1

Timeouts

Queries dropped inside AS267613 or at Tier 1, which blackholed the /32. Looks like packet loss.

SYMPTOM 2

Correct answers, slowly

Traffic from the US and Germany landed in Brazilian data centers. Responses were right; the round trip wasn't.

THE TELL

BMP saw it arrive

Cloudflare's own pre-policy BMP data showed the /32 coming in from route servers — and their import policy rejected it.

Neither symptom says "routing." A latency spike with NOERROR responses is the kind of alert that gets acknowledged and closed.

MECHANISM 1

ROV: Is this origin allowed to announce this prefix?

ROA in RPKI
1.1.1.0/24 · AS13335 · max /24

1.1.1.1/32 from AS267613
✗ **Invalid**

1.1.1.0/24 via leak
✓ **Valid**

right origin, wrong path — ROV can't tell

No covering ROA → NotFound → accepted by default

CATCHES

Origin hijacks and unauthorized more-specifics — Route 53 2018, every root-prefix event, the 1.1.1.1/32.

MISSES

Anything with the right origin and the wrong path: route leaks, forged-origin hijacks.

ONLY WORKS IF

You sign, and the networks between you and your users drop invalid. Uzbektelekom in 2025 and the Tier 1 in 2024 didn't.

MECHANISM 2

ASPA: Does this path match real provider relationships?

Each AS signs one statement: “These are my upstream providers." A path must climb from customer → provider, cross at most one peering, then descend.

WALKING THE 1.1.1.0/24 LEAK FROM THE ORIGIN

AS13335 → ✗ → AS267613 → ✗ → AS262504 → ✓ → AS1031 → their peers

13335 → 267613
A peer, not a provider. Cloudflare's ASPA wouldn't list it.

267613 → 262504
Nova is Eletronet's customer. Eletronet's ASPA wouldn't list it.

262504 → 1031
A real provider. This hop is fine — the damage was done earlier.

Either ASPA on its own makes the path **Invalid** for anyone who receives it from AS1031 and validates it. Signing is in production at RIPE NCC, ARIN, and APNIC; LACNIC is due by the end of 2026. About 3,600 ASes have signed.

Which mechanism catches which incident

INCIDENT	ROV	ASPA	DNSSEC
Route 53 more-specifics, 2018	✓	—	◐
Root prefix origin hijacks, 2023 – 2025	✓	—	◐
1.1.1.1/32 hijack, 2024	✓	—	✗
1.1.1.0/24 route leak, 2024	✗	✓	✗

✓ catches it, given signed objects and validating receivers · ◐ protects the answers, not the reachability · — not what it's for

DNSSEC protects your answers. ROV protects your origin. ASPA protects your path.

LIVE DEMO

One Scenario, live

Twenty minutes doesn't fit all three incidents, so we'll replay the one ROV can't see.

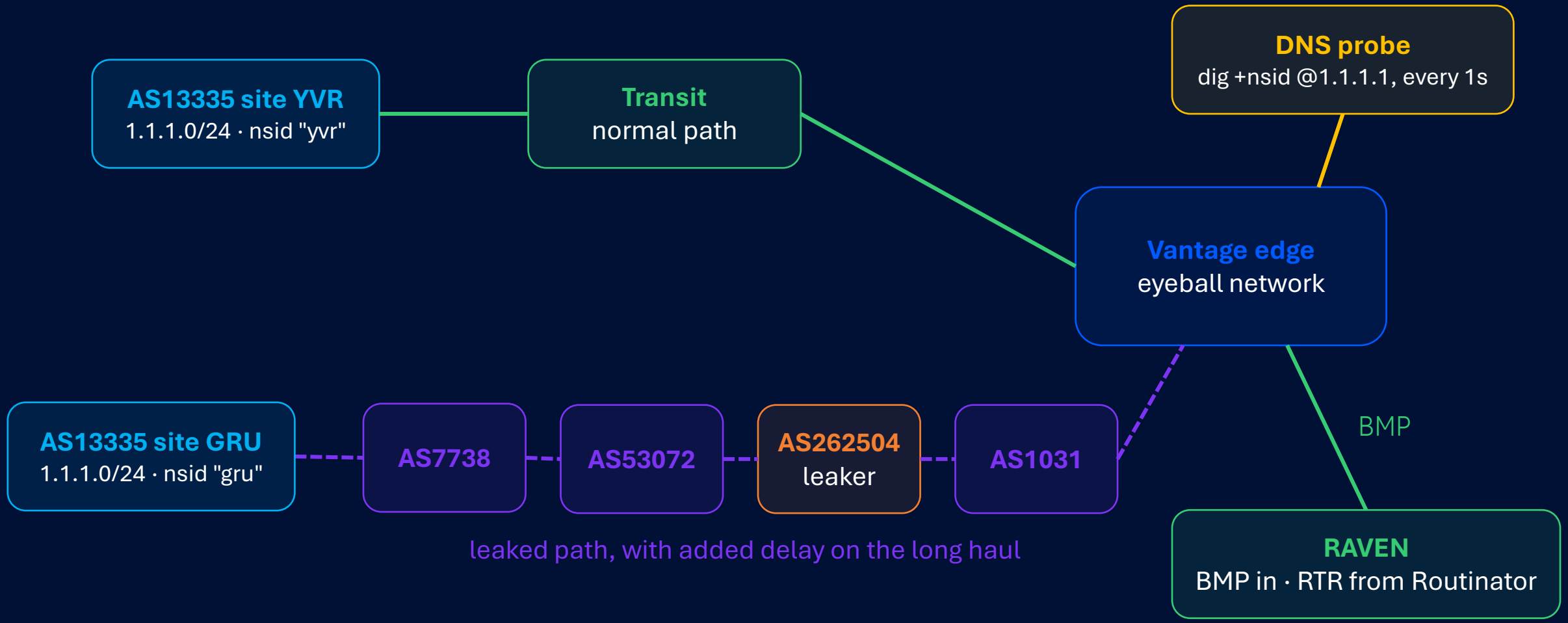
20:10 UTC, JUNE 27, 2024

262504 53072 7738 13335

The leaked path successfully delivered queries to 1.1.1.1—from São Paulo, with high latency.

The lab: an anycast resolver with two sites

Containerlab · FRR · Routinator validating against the live RPKI · RAVEN



Two panes, one event

Real output from the lab — Watch what each one says when the leak arrives

```
pane 1 — DNS probe · dig +nsid @1.1.1.1 demo.oarc A

11:36:03  rcode=NOERROR  nsid=yvr  latency=6ms
11:36:04  rcode=NOERROR  nsid=yvr  latency=6ms
11:36:05  rcode=NOERROR  nsid=yvr  latency=5ms
── AS262504 starts leaking ──
11:36:16  rcode=NOERROR  nsid=gru  latency=151ms
11:36:17  rcode=NOERROR  nsid=gru  latency=151ms
11:36:18  rcode=NOERROR  nsid=gru  latency=151ms
```

Every answer is correct. Nothing here says "routing" — it reads as a slow site.

```
pane 2 — raven watch

watching for route events... (Ctrl+C to stop)

[path-suspect] 1.1.1.0/24 via 10.47.2.2 origin:AS13335 AS_PATH:[1031 262504
53072 7738 13335] rov:Valid ASPA:Invalid hop(53072→262504) posture:path-suspect
```

Same moment, with the reason: the origin is legitimate, the path isn't.

What a DNS operator can do this quarter

Sign every anycast prefix

ROAs with maxLength equal to what you announce. If you use a unique origin AS per site (RFC 6382), every one of them needs a ROA.

Publish ASPA

One object per ASN listing its actual transit providers— including per-site ASNs. Keep it current when a transit provider changes.

Watch from your own edge

Stream pre-policy BMP from your anycast edge routers. Alert on new origins, new more-specifics, and paths that fail ASPA.

Ask your transits

Do they drop RPKI-Invalid routes? How do they filter blackhole requests? Their answer decides whether your ROAs matter.

And keep signing zones and validating. DNSSEC is the layer that still holds when routing doesn't.

OPEN SOURCE

The demo ran on RAVEN

A single Go binary that takes BMP from your routers and RPKI from your validator and tells you which routes have a valid origin and a valid path.

BMP	pre-policy Adj-RIB-In from any vendor's router
RTR	VRPs and ASPAs from Routinator or any RTR v2 cache
Out	CLI, live stream, Prometheus

quick start

```
raven serve --bmp-listen :11019 \
  --rtr rpki1.example.com:3323

raven watch --posture path-suspect
```

github.com/nokia-steward/bgp-routing-security-monitor

BSD-3-Clause · no external measurement infrastructure required

Your DNS can be perfectly healthy
and still be answering from the wrong place.

Questions?

Ritesh Mukherjee

OARC 47 VANCOUVER