False immunity: long prefixes that bypass ROV

Bryton Herdes
Principal Network Engineer

CHI-NOG 13

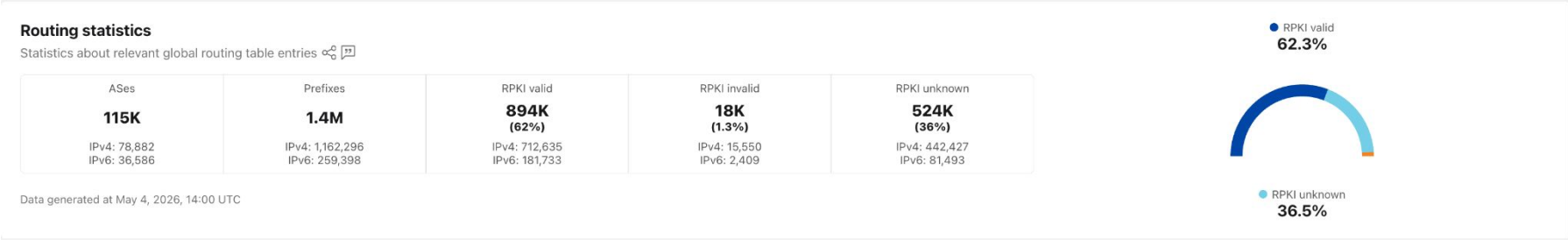
Agenda

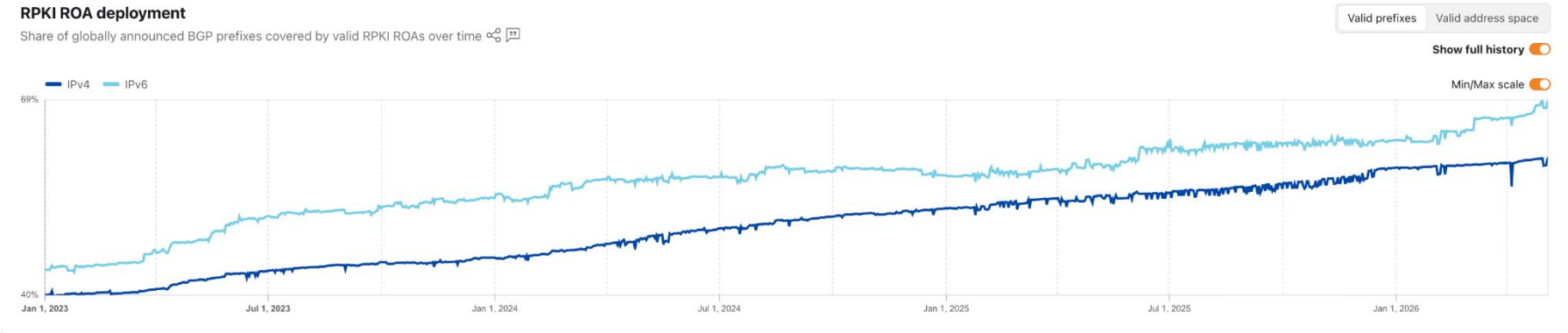
1. A surprising route hijack
2. Edge case: Traffic Engineering (TE)
3. Edge case: Remote-Triggered Blackhole (RTBH)
5. Questions?

A surprising route hijack

RPKI-ROV milestone

- ROAs exist for around 60% of BGP prefixes





Validating routes

- Every major network should be running ROV
- Enforce BGP route matches ROA
- Feeling pretty safe against origin hijacks

NAME	TYPE	DETAILS	STATUS	ASN
Lumen	transit	signed + filtering	safe	3356
Arelion (formerly Telia)	transit	signed + filtering	safe	1299
Cogent	transit	signed + filtering	safe	174
NTT	transit	signed + filtering	safe	2914
Sparkle	transit	signed + filtering	safe	6762
Hurricane Electric	transit	signed + filtering	safe	6939
GTT	transit	signed + filtering	safe	3257
TATA	transit	signed + filtering	safe	6453
Zayo	transit	signed + filtering	safe	6461
PCCW	transit	signed + filtering	safe	3491
Vodafone	transit	signed + filtering	safe	1273
RETN	transit	partially signed + filtering	safe	9002
Orange	transit	signed + filtering	safe	5511
Telstra International	transit	signed + filtering	safe	4637
Telefonica/Telxius	transit	signed + filtering	safe	12956
Comcast	ISP	signed + filtering	safe	7922
AT&T	ISP	signed + filtering	safe	7018
Verizon	ISP	signed + filtering	safe	701
Liberty Global	transit	signed + filtering	safe	6830

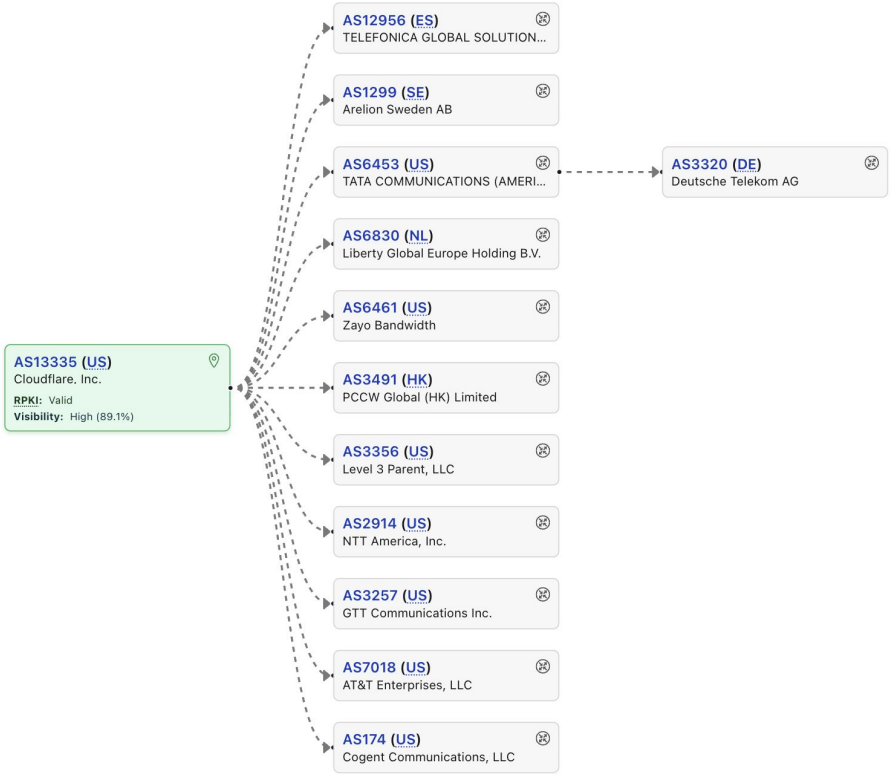
https://isbgpsafeyet.com/

General filtering assumptions

- Longer than /24 + /48 are often filtered out (independent of ROV)
- ROV coverage spans **all** prefix lengths
- Use of ROA **maxLength** makes ROV more effective
  - RFC9319: BGP routes should match *minimal* ROA
  - Provides some protection against forged-origin hijacks

Normal route to 1.1.1.1

- AS13335 advertises 1.1.1.0/24
- No more-specifics expected anywhere



https://radar.cloudflare.com/routing/prefix/1.1.1.0/24

1.1.1.0/24 ROA covering 1.1.1.1

Found *1* ROAs and *11* certificates

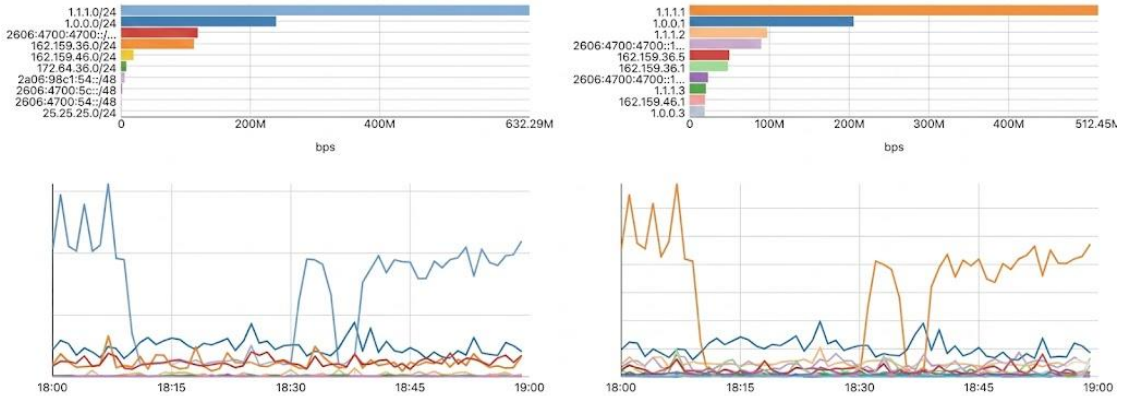
ROAs

ASN	Prefix	Max Length
AS13335	1.1.1.0/24	/24

1–1 of **1** items

https://rpki.cloudflare.com/?view=explorer&prefix=1.1.1.0%2F24

1.1.1.1/32 hijack



1.1.1.1/32 (1 entry, 1 announced)

***BGP Preference: 170/-101**

Next hop type: Router, Next hop index: 2490

Source: 212.73.201.90

Next hop type: Router, Next hop index: 3381

Next hop: 212.73.201.90 via et-7/0/17:1.0, selected

Indirect next hop: 0xf8268d88 2097429 INH Session ID: 23852

Local AS: 3356 Peer AS: 3356

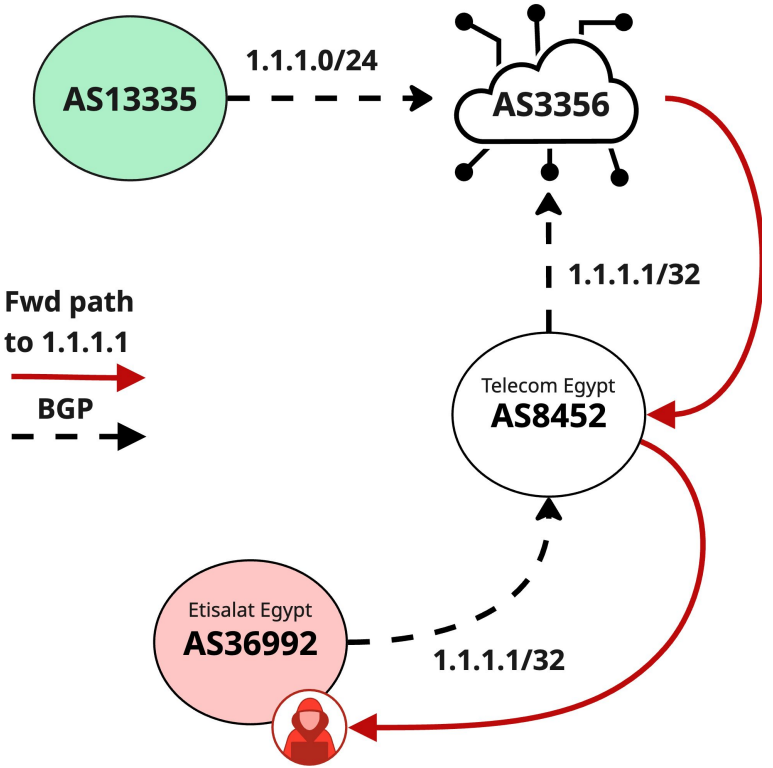
Validation State: unverified

AS path: 8452 36992 I

```
Tracing the route to one.one.one.one (1.1.1.1)
VRF info: (vrf in name/id, vrf out name/id)
  1 [redacted]                                                    1 msec 48 msec 49 msec
  2 ae1.3505.edge4.mrs1.neo.colt.net (171.75.8.243) [AS 3356] 145 msec 144 msec 193 msec
  3 TELECOM-EGY.edge4.Marseille1.Level3.net (212.73.253.58) [AS 3356] 194 msec 188 msec *
  4 81.10.87.99 [AS 8452] 173 msec *  *
  5  *  *  *
  6  *  *  *
  7  *
```

Unpacking what happened

- AS13335: valid *originator*
- AS36992: *hijacker*
- AS8452: hijack acceptor, *enabler*
- AS3356: hijack *acceptor*



ROV Bypass

- For >/24 prefixes, it appeared AS3356 was running **IRR-only filtering**, allowing invalids
- AS3356: motivation to support customer “TE”
- AS13335 is found (indirectly) within Telecom Egypt’s expanded AS-SET

```
➜  ~
➜  ~ whois -h rr.ntt.net '!iAS-TEDATA,1' | grep -qw AS13335 && echo "FOUND" || echo "NOT FOUND"
FOUND
➜  ~
➜  ~ bgpq4 -J4 AS-TEDATA | grep -F " 1.1.1.0/24"
    1.1.1.0/24;
➜  ~
```

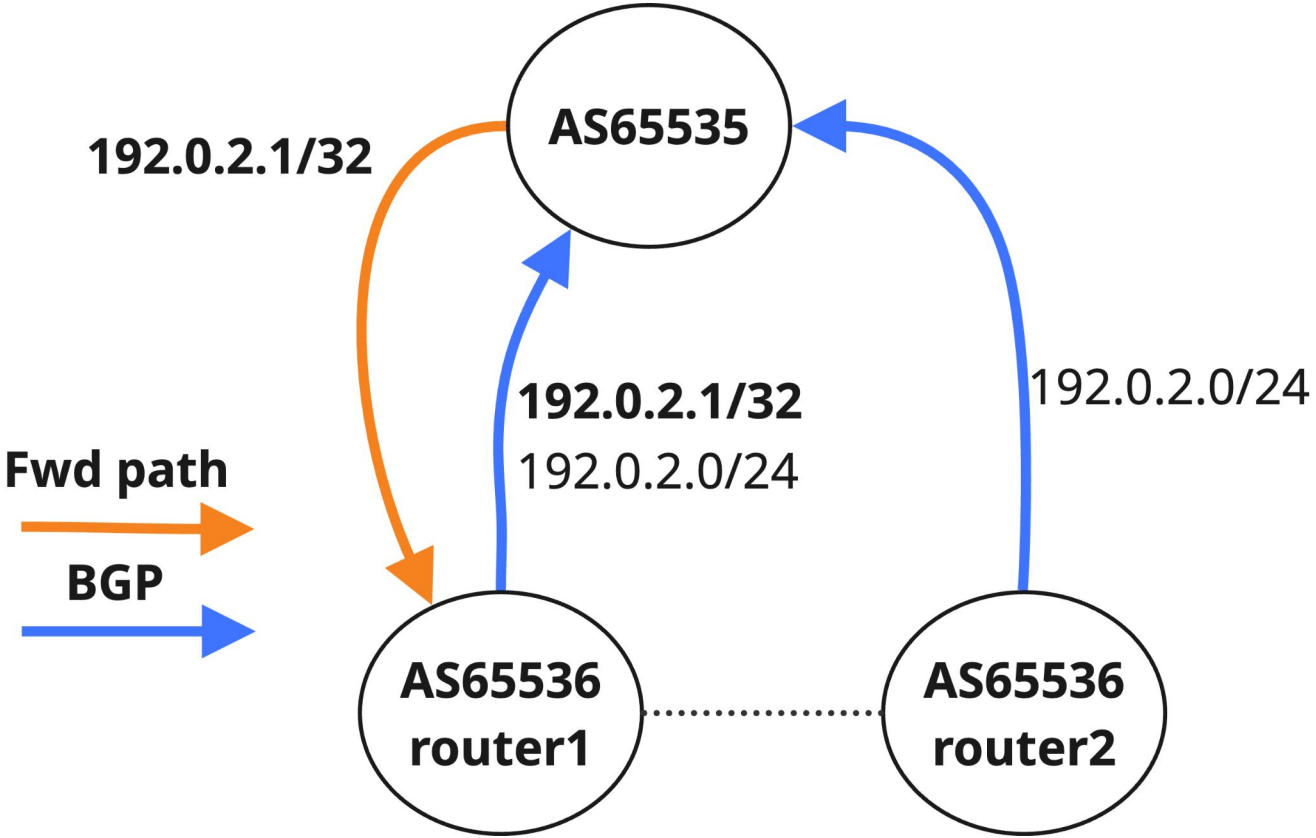
False immunity

Edge case: Traffic Engineering (TE)

TE with long prefixes - why?

- Some customers want to advertise >/24 and >/48 *within* provider networks
  - Not propagated further to peers outside provider's network
- Motivated by IPv4 address starvation?
- Poor management of load per-prefix?

TE with long prefixes - how?



Potential provider config: ROV bypass

```
bryton@edge01.stl01# show policy-options policy-statement 4-CUSTOMER-FOO-BAR-IN
/* Only check DFZ-eligible prefixes for RPKI-ROV invalid status  */
term ROV-DFZ-INVALID {
    from {
        validation-database invalid;
        route-filter 0.0.0.0/0 upto /24;
    }
    then {
        validation-state invalid;
        reject;
    }
}
term ROV-VALID {
    from validation-database valid;
    then {
        validation-state valid;
        next term;
    }
}
/* Here IRR filtering takes place, and long-prefixes may only hit this term */
term ACCEPT-CUSTOMER-IRR {
    from {
        prefix-list-filter 4-foo-bar-customer orlonger;
    }
    then {
        local-preference 140;
        community add CUSTOMER;
        accept;
    }
}
then reject;
```

Test: who allows TE to bypass ROV?

- Advertise /32 + /128 prefixes to peers
  - BGP origin AS != ROA origin AS
  - Find who filters via IRR-only and allows ROV bypass

- Tests limited by:
  - Type of BGP relationship AS13335 has (customer-provider, vs lateral Peer)

Purposely invalid announcements

Validating route **162.159.225.1/32**

from origin **AS13335**

✕ **Invalid**

1 covering ROA found

Covering ROAs:

Trust Anchor	Prefix	Max Length	ASN	Expiration	Match
ARIN	162.159.225.0/24	24	209242	in 2 months	✕

Validating route **2606:4700:4f::1/128**

from origin **AS13335**

✕ **Invalid**

1 covering ROA found

Covering ROAs:

Trust Anchor	Prefix	Max Length	ASN	Expiration	Match
ARIN	2606:4700:4f::/48	48	209242	in 2 months	✕

```
~ bgpq4 -J4 AS13335:AS-CLOUDFLARE | grep 162.159.225.0
  162.159.225.0/24;
~ bgpq4 -J6 AS13335:AS-CLOUDFLARE | grep 2606:4700:4f
  2606:4700:4f::/48;
```

Testing method

- Advertise long-invalid prefixes to peers one-by-one
- Ping reachability from looking-glasses, or vantage points
- Record results

NO ROV examples

```
bryton@edge02.dub01> show bgp summary | grep " 6461 "
94.31.43.25            6461    1447406      88975       0       0 3w6d 19:31:15 Establ
2a00:16f8:11::22d        6461    3349233      84049       0       0 3w6d 19:30:59 Establ

bryton@edge02.dub01> traceroute 162.159.225.1 next-hop 94.31.43.25
traceroute to 162.159.225.1 (162.159.225.1) from 94.31.43.26, 30 hops max, 52 byte packets
 1  ae52.mpr1.dub2.ie.zip.zayo.com.zip.zayo.com (94.31.43.25)  1.637 ms  1.684 ms  1.640 ms
 2  162.159.225.1 (162.159.225.1)  1.298 ms  1.229 ms  1.274 ms

bryton@edge02.dub01> traceroute 2606:4700:4f::1 next-hop 2a00:16f8:11::22d
traceroute6 to 2606:4700:4f::1 (2606:4700:4f::1), 64 hops max, 12 byte packets
 1  2a00:16f8:11::22d (2a00:16f8:11::22d)  3.791 ms  1.284 ms  1.444 ms
 2  2606:4700:4f::1 (2606:4700:4f::1)  1.290 ms  1.395 ms  1.374 ms
```

162.159.225.1/32 (2 entries, 1 announced)

***BGP Preference: 170/-101**

Next hop type: Indirect, Next hop index: 0

Address: 0x1ba49dadc

Next-hop reference count: 459848

Source: 171.75.50.22

AS path: 13335 I (Originator)

ROV examples

```
Router: gva-b1 / Geneva (Equinix GV2)
Command: show bgp ipv6 unicast 2606:4700:4f::1

BGP routing table entry for 2606:4700:4f::1/128
Last Modified: May  1 03:13:31.269 for 00:05:46
Paths: (1 available, no best path)

  Path #1: Received by speaker 0
  13335, (received-only)
    2001:2035:0:2854::2 from 2001:2035:0:2854::2 (172.69.108.1)
      Origin IGP, localpref 100, valid, external
```

```
Router: gva-b1 / Geneva (Equinix GV2)
Command: ping ipv6 2606:4700:4f::1 timeout 1 source Loopback0

Sending 5, 100-byte ICMP Echos to 2606:4700:4f::1, timeout is 1 seconds:
..UUU
Success rate is 0 percent (0/5)
```

Results: who allows TE to bypass ROV?

ASN	Name	Result
174	Cogent	**ROV**
701	Verizon	**ROV**
1299	Arelion	**ROV**
2914	NTT	**ROV**
3257	GTT	**ROV**
3356	Lumen/Colt/Cirion	**NO ROV** (*IPv4 only*)
3491	PCCW	**ROV**
5511	Orange	**ROV**

Results: who allows TE to bypass ROV?

ASN	Name	Result
6453	Tata	ROV
6461	Zayo	NO ROV
6762	Sparkle	ROV
6830	Liberty Global	ROV
6939	HE	ROV
7018	AT&T	ROV
12956	Telefonica	ROV

Testing takeaway

- At least two tier-1 ISPs appear to accept RPKI-ROV long-invalid prefixes, ignoring ROV
- Acceptance of IPv4 invalid prefixes may be more likely
- Downstreams of these ISPs will be impacted

Solutions for long prefix TE

- Option 1: don’t offer this “TE” method
  - How valid is this customer use-case?

- Option 2: require long maxLength ROAs
  - Advertising upto /32 requires maxLength=32 ROA
  - Customers accept going against RFC9319
    - Forged-origin hijacks may be more effective

Edge case: Remote-Triggered Blackhole (RTBH)

Destination-based RTBH

- Customer is getting DDoS attacked toward IP(s)
- Links are congesting
- Customer network signals via BGP for provider to discard traffic to IP(s)
- Congestion resolved since traffic is dropped on provider routers

Blackhole route filtering

- Specialized routing announcement
  - kept internal-only (BLACKHOLE, NO_EXPORT)
- Generally no RPKI-based filtering at all
- Based on IRR objects (AS-SET, route + route6)
- Announcements are expected to be long
  - Customers don’t want to blackhole more IPs than they absolutely have to

RTBH hijack

- AS13335: *valid* prefix owner
- AS55720: RTBH *hijacker*
- Accepted by Tier-1 ISP *based on IRR data*

Query Results:
Router: Ashburn, VA - US
Command: show route table inet.0 protocol bgp 162.159.140.238 terse

```
inet.0: 980113 destinations, 8470679 routes (978951 active, 2448 holddown, 12
+ = Active Route, - = Last Active, * = Both

A V Destination        P Prf   Metric 1   Metric 2  Next hop        AS path
* ? 162.159.140.238/32 B 170        120             to Discard      55720 I
  ?                    B 170        120             to Discard      55720 I
  ?                    B 170        120             to Discard      55720 I
```

Validating route **162.159.140.238/32**
from origin **AS55720**
Invalid
2 covering ROAs found

Covering ROAs:

Trust Anchor	Prefix	Max Length	ASN	Expiration	Match
ARIN	162.159.128.0/19	19	13335	in 2 months	✗
ARIN	162.159.128.0/19	19	13335	in 2 years	✗

RTBH solution: more RPKI

- Dedicated RPKI object for RTBH
- Similar to ROA, but “Discard” Origin Authorization
- https://datatracker.ietf.org/doc/draft-spaghetti-sidrops-rpki-doa/
- Problem: re-do of ROA signing effort for DOA

RTBH solution: loose ROV

- Use ROA but extend to /32 prefix-length for **only** RTBH routes - AKA origin-only ROV

- ROV with “loose” method **must** be restricted to BLACKHOLE (RFC7999) routes

- Problem: *slippery* slope
  - Can ruin maxLength ROV (RFC9319) if implemented incorrectly

RTBH loose ROV in the wild (BIRD)

`ignore max length` *switch*

Ignore received max length in ROA records and use max value (32 or 128) instead. This may be useful for implementing loose RPKI check for blackholes. Default: disabled.

- Is the route a blackhole (RFC 7999)?
  - **If no, the route undergoes strict RPKI validation filtering** (both `origin` and maximum prefix length (`maxLength`)):
    - if the result is `RPKI Valid` , the route is accepted (a missing route object will have no implication in this case)
    - if the result is `RPKI Invalid` , the route is rejected
    - if the result is `RPKI NotFound` , we check if the route is resolvable for its origin ASN (this will be the case if a proper route object exists) and it might get accepted or rejected depending on the result**
  - **If yes, the route undergoes loose RPKI validation filtering** (`origin` only):
    - if the result is `RPKI Valid` , the route is accepted
    - if the result is `RPKI Invalid` , the route is rejected
    - if the result is `RPKI NotFound` , we check if the route is resolvable for its origin ASN (this will be the case if a proper route object exists) and it might get accepted or rejected depending on the result**

Configuration from BIRD
https://bird.network.cz/?get_doc&v=20&f=bird-6.html

- Requires multiple ROA tables to maintain separation of ROV (regular routes) and loose-ROV (RTBH)

Example in the wild: DE-CIX route servers
https://docs.de-cix.net/article/o80or2w5dl-de-cix-chicago-globepeer-route-server-guide

RTBH solution: covering route lookup

- Leverage the covering route that *already* passed ROV checks
- Check if RTBH route is received from same neighbor AS (or NEXT_HOP) as covering route
  - If yes: ACCEPT
  - If no: REJECT
- https://iepg.org/2019-03-24-ietf104/blackholing_reconsidered_ietf104_snijders.pdf

Questions?

Bryton Herdes
E: bryton@cloudflare.com
X: https://x.com/next_hopself
M: https://mastodon.social/@next_hopself