

ASPA

RPKI for path validation!

Anurag Bhatia, Hurricane Electric

Presentation overview

1. **How filtering is done presently**
2. Introduction to ASPA
3. Creating ASPA record
4. Configuring ASPA for path validation
5. ASPA deployment at Hurricane Electric

How is filtering done presently?

How filtering is done so far...

- Varies from operator to operator
- Some filter based on route object, some based on custom logic
- Some filter only customers, some peers, very few filter upstream
- No easy/standardized end to end tooling (e.g bgpq3 exists but needs mechanism to “push” rules to the router)

No standardised way to filter, it’s just broken...

Presentation overview

1. How filtering is done presently
2. **Introduction to ASPA**
3. Creating ASPA record
4. Configuring ASPA for path validation
5. ASPA deployment at Hurricane Electric

Introduction to ASPA

- Stands for Autonomous System Provider Authorization (ASPA)
- Brings RPKI into path validation like ROA did for origin validation
- Brings cryptography as well as easy to integrate tooling for filtering route leaks
- More secure than AS-SETs

Example of ASPA record

- Customer AS number
- Provider AS number (can be multiple)
- RPKI TA (Trust Anchor)
- Note: Provider = upstream only & not a peer


```
{
  "customer": "AS11358",
  "providers": [
    "AS835",
    "AS6939",
    "AS34927"
  ],
  "ta": "arin"
}
{
  "customer": "AS11708",
  "providers": [
    "AS32097"
  ],
  "ta": "arin"
}
{
  "customer": "AS11967",
  "providers": [
    "AS835",
    "AS1299",
    "AS6939",
    "AS34872",
    "AS34927",
    "AS50917",
    "AS58057",
    "AS213753",
    "AS214809",
    "AS215828"
  ],
  "ta": "arin"
}
```

ASPA validation logic

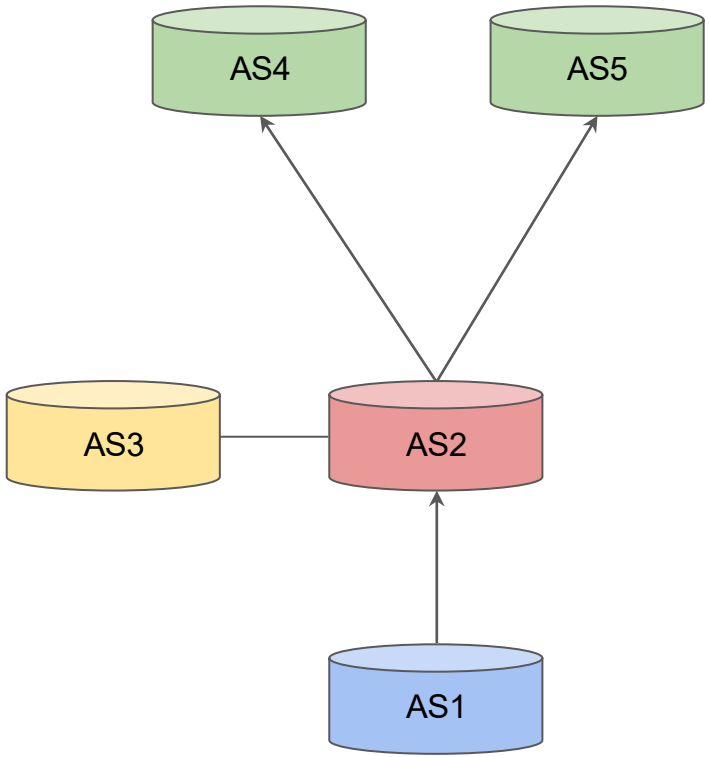
- Validation is done across the AS_PATH with possible states as: Provider, Not Provider or Unknown state
- Similar to RPKI ROA, routes are rejected if a “Not Provider” comes in the path. Lack of ASPA is not dropped
- ASPA doesn’t care if adjacency is upstream or downstream or peer but follows validation based on declared upstream by each ASN
- Logic for rejection is applied in policy and hence can be selectively applied for the rollout

ASPA validation logic

- AS1 has upstream AS2
- AS2 has peering with AS3 + transit with AS4 & AS5
- ASPA object:

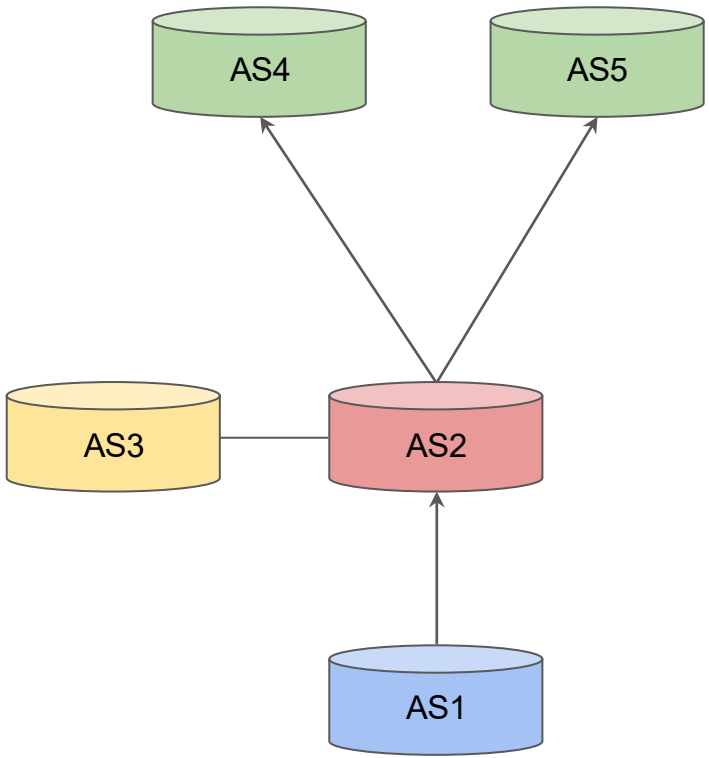
  Customer: AS1
  Providers: AS2

  Customer: AS2
  Providers: AS4, AS5



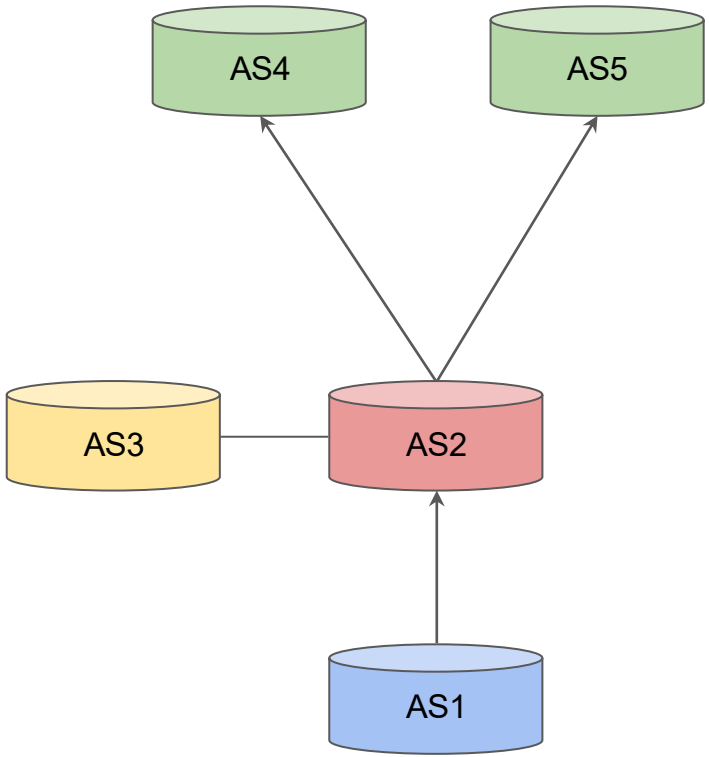
ASPA validation logic

- For AS4 - It would see AS1 as: 4 2 1
- They can validate from ASPA object of AS2 and AS1 and validate it
- For AS1, AS1 will see AS3, AS4 & AS5 all behind AS2



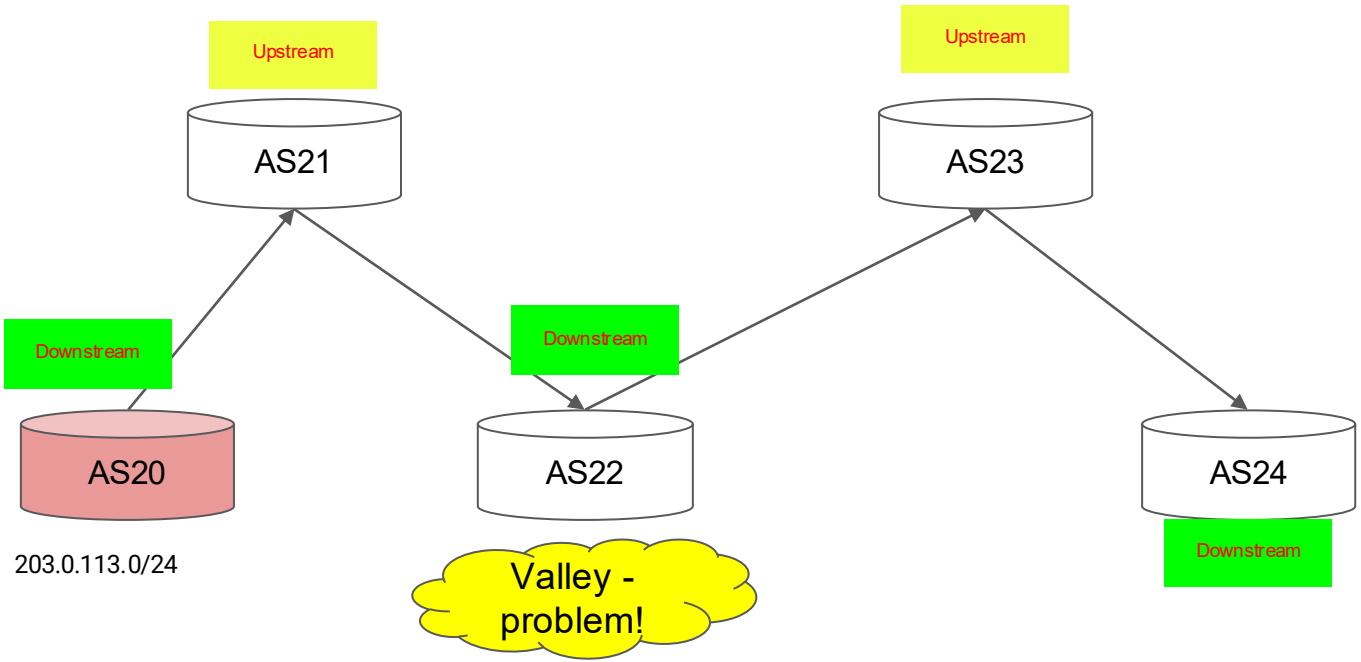
Case of route leak

- In this if AS2 leaks AS3, leak will go through because of missing ASPA record of AS3
- If AS3 has a ASP record, leak would be detected & dropped by network doing the validation *(AS3 will not list AS2 as provider)*



Concept of valley free routing...

Example of valley in routing

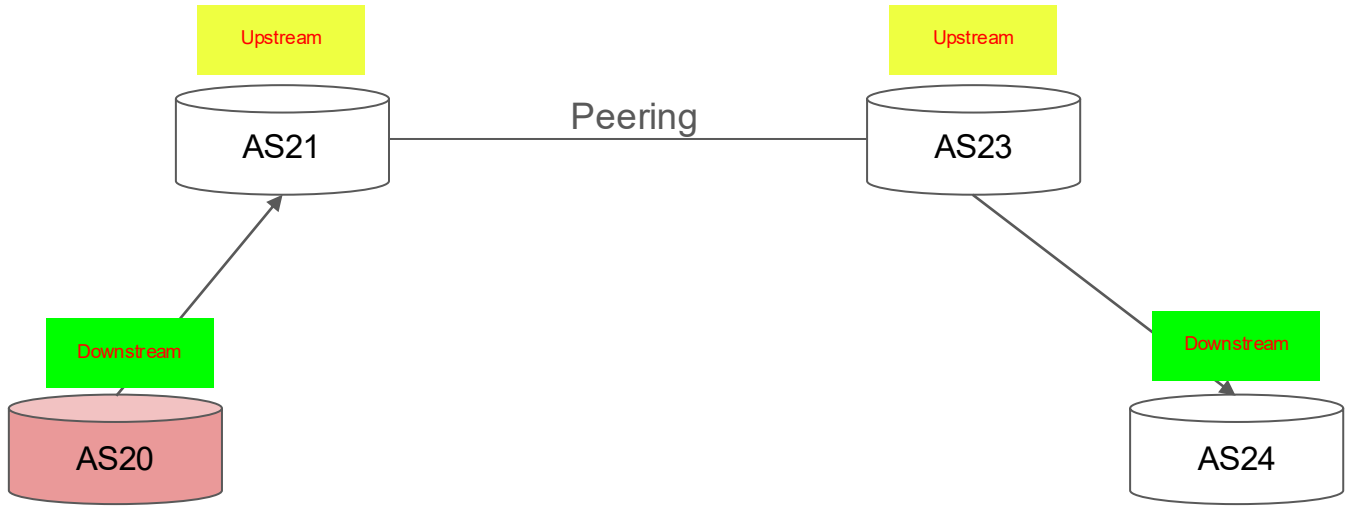


AS24 table:

203.0.113.0/24 23 22 21 20

AS21 does not have AS22 as upstream (hence AS22 is downstream) and if AS22 marked AS23 as upstream then it's a leak

Example of valley in routing



203.0.113.0/24

AS24 table:

203.0.113.0/24 23 22 21 20

AS20 -> AS21 - Valid!

AS21 -> AS23 - No mention in ASPA object (peers)

AS23 -> AS24 - Valid!

No Valley - all good!

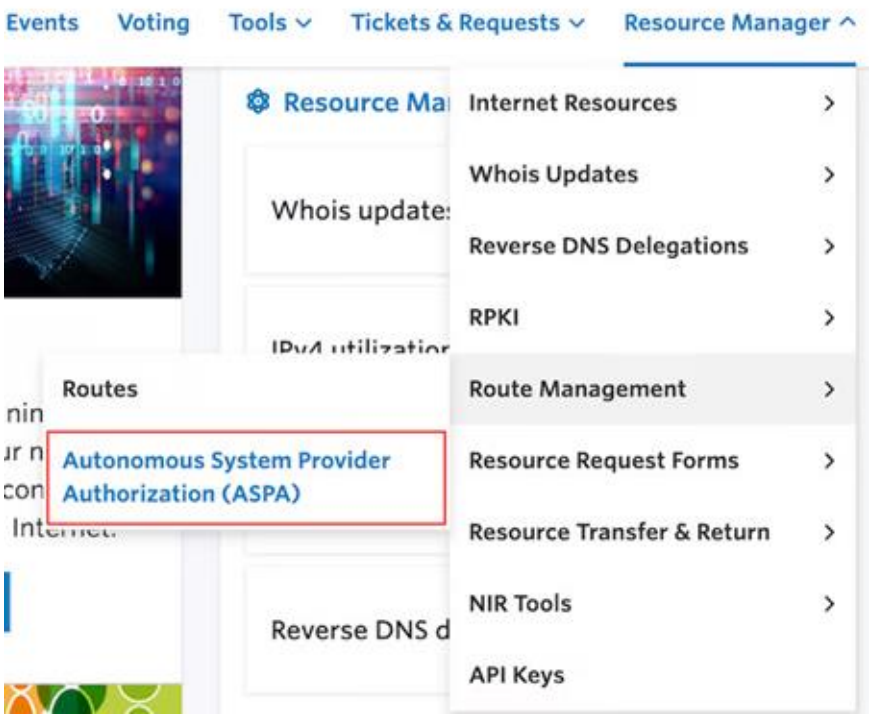
Presentation overview

1. How filtering is done presently
2. Introduction to ASPA
3. **Creating ASPA record**
4. Configuring ASPA for path validation
5. ASPA deployment at Hurricane Electric

How to create ASPA object?

- ASPA is part of RPKI and ASPA object can be created in same way as ROA on APNIC, ARIN & RIPE NCC
- APNIC launched ASPA support since 9th July 2026
- In RIR hosted model, ASPA is created at RIR via RIR interface and certificates are held in the RIR's infrastructure
- In delegated / self hosted model, ASPA is held in own infrastructure e.g on open source software like Krill

Creating ASPA record in APNIC



Source: https://help.apnic.net/s/article/ASPA

Presentation overview

1. How filtering is done presently
2. Introduction to ASPA
3. Creating ASPA record
4. **Configuring ASPA for path validation**
5. ASPA deployment at Hurricane Electric

Hardware support

- It is supported in BIRD 2.16
- IOS-XR - Cisco is running field trials since last year
- JunOS likely to support it by end of the year (~26.4)
- Nokia likely to support it by end of 2026

How ASPA is configured for path validation?

- Routers supporting ASPA will be able to connect with RPKI validator like routinator over RTR
- Enable the ASPA validation
- Apply policy to reject “ASPA Invalid state” on the import policy

Routinator support

- Supported in latest version 0.15.1
- Enabled by --enable-aspa when bringing up or enable-aspa=true as static option in the config
- Gives output via RTR as well as json dump (alongside ROAs)

```
anurag@fmt01 ~> curl -s https://routinator.fmt.anuragbhatia.com/json | jq keys
[
  "aspas",
  "metadata",
  "roas"
]
anurag@fmt01 ~>
```

Presentation overview

1. How filtering is done presently
2. Introduction to ASPA
3. Creating ASPA record
4. Configuring ASPA for path validation
5. **ASPA deployment at Hurricane Electric**

ASPA deployment at Hurricane Electric AS6939 backbone

ASPA validation at Hurricane Electric / AS6939 backbone

- HE / AS6939 is doing ASPA validation across it’s backbone for downstreams and peers via our reactive system
- HE rejects routes where ASPA exists and upstream ASN is not matching the AS_PATH from the routing table
- Setup is integrated within existing reactive route filtering system

Hurricane Electric Route Filtering Algorithm

This is the route filtering algorithm for customers and peers that have explicit filtering:

1. Attempt to find an as-set to use for this network.

1.1 In peeringdb, for this ASN, check for an IRR as-set name.
Validate the as-set name by retrieving it. If it exists, use it.

1.2 In IRR, query for an aut-num for this ASN. If it exists, inspect the aut-num for this ASN to see if we can extract from their IRR policy an as-set for what they will announce to Hurricane by finding export or mp-export to AS6939, ANY, or AS-ANY.
Precedence is as follows: The first match is used, "export" is checked before "mp-export", and "export: to AS6939" is checked before "export: to ANY" or "export: to AS-ANY".
Validate the as-set name by retrieving it. If it exists, use it.

1.3 Check various internal lists maintained by Hurricane Electric's NOC that map ASNs to as-set names where we discovered or were told of them.
Validate the as-set name by retrieving it. If it exists, use it.

1.4 If no as-set name is found by the previous steps use the ASN.

2. Collect the received routes for all BGP sessions with this ASN. This details both accepted and filtered routes.

3. For each route, perform the following rejection tests:

3.1 Reject default routes 0.0.0.0/0 and ::/0.

3.2 Reject AS paths that use BGP AS_SET notation (i.e. {1} or {1 2}, etc). See RFC 9774 "Deprecation of AS_SET and AS_CONFED_SET in BGP".

3.3 Reject prefix lengths less than minimum and greater than maximum. For IPv4 this is 8 and 24. For IPv6 this is 16 and 48.

3.4 Reject bogons (RFC 1918, documentation prefix, etc).

3.5 Reject exchange prefixes for all exchanges Hurricane Electric is connected to.

3.6 Reject AS paths that exceed 50 hops in length. Excessive BGP AS Path Prepending is a Self-Inflicted Vulnerability.

3.7 Reject AS paths that use unallocated 32-bit ASNs between 1000000 and 4199999999. https://www.iana.org/assignments/as-numbers/as-numbers.xhtml

3.8 Reject AS paths that use unassigned 32-bit ASNs between 4200000000 and 4294967294. Reserved for Private Use by RFC 6996.

3.9 Reject AS paths that use AS 23456. AS 23456 should not be encountered in the AS paths of BGP speakers that support 32-bit ASNs.

3.10 Reject AS paths that use AS 0. As per RFC 7606, "A BGP speaker MUST NOT originate or propagate a route with an AS number of zero".

3.11 Reject routes that fail an ASPA (Autonomous System Provider Authorization) check for any hop of the path where a customer ASPA record exists and the provider ASN is not listed as a provider.

3.12 Reject routes that were learned via a route server and that contain a hop that is an ASN that has specified "Never via route servers" (peeringdb flag).

3.13 Reject routes that have RPKI status INVALID_ASN or INVALID_LENGTH based on the origin AS and prefix.

4. For each route, perform the following acceptance tests:

4.1 If the origin is the neighbor AS, accept routes that have RPKI status VALID based on the origin AS and prefix.

4.2 If the prefix is an announced downstream route that is a subnet of an accepted originated prefix that was accepted due to either RPKI or an RIR handle match, accept the prefix.

4.3 If RIR handles match for the prefix and the peer AS, accept the prefix.

4.4 If this prefix exactly matches a prefix allowed by the IRR policy of this peer, accept the prefix.

4.5 If the first AS in the path matches the peer and path is two hops long and the origin AS is in the expanded as-set for the peer AS and either the RPKI status is VALID or there is an RIR handle match for the origin AS and the prefix, accept the prefix.

5. Reject all prefixes not explicitly accepted

Some live rejections as on 21 Oct 2026

95.213.235.0/24,rejected,AS path 3491 20764 49505 ASPA record exists for 20764 and 3491 is not listed as a provider.

95.213.236.0/24,rejected,AS path 3491 20764 49505 ASPA record exists for 20764 and 3491 is not listed as a provider.

Record: { "customer": "AS20764", "providers": ["AS3356"], "ta": "ripe" },

Some live rejections as on 21 Oct 2026

103.149.123.0/24,rejected,AS path 6453 9829 9885 55824 140110 ASPA record exists for 9885 and 9829 is not listed as a provider.

Record: { "customer": "AS9885", "providers": ["AS4755", "AS9498", "AS11537", "AS20965", "AS23855", "AS24029", "AS24490", "AS55836"], "ta": "apnic" }

103.19.16.0/24,rejected,AS path 3356 135607 7629 23930 ASPA record exists for 23930 and 7629 is not listed as a provider.

Record:
{ "customer": "AS23930", "providers": ["AS1299", "AS3491", "AS4635", "AS4637", "AS4779", "AS6648", "AS7707", "AS9299", "AS15169", "AS15412", "AS17639", "AS18187", "AS18392", "AS23942", "AS24095", "AS24115", "AS55821", "AS59360", "AS135607"], "ta": "apnic" }

Who else has Google AS15169 in their ASPA record? - 11 ASNs!

{ "customer": "AS1436", "providers": ["AS209", "AS3356", "AS6128", "AS6461", "AS15169", "AS46887", "AS54004", "AS62532"], "ta": "arin" },

{ "customer": "AS7629", "providers": ["AS1299", "AS3491", "AS4635", "AS4637", "AS4779", "AS6648", "AS7707", "AS9299", "AS15169", "AS15412", "AS17639", "AS18187", "AS18392", "AS23942", "AS24095", "AS24115", "AS55821", "AS59360", "AS135607"], "ta": "apnic" },

{ "customer": "AS17747", "providers": ["AS9498", "AS15169", "AS21357", "AS24029", "AS55836", "AS134375", "AS135188"], "ta": "apnic" },

{ "customer": "AS23930", "providers": ["AS1299", "AS3491", "AS4635", "AS4637", "AS4779", "AS6648", "AS7707", "AS9299", "AS15169", "AS15412", "AS17639", "AS18187", "AS18392", "AS23942", "AS24095", "AS24115", "AS55821", "AS59360", "AS135607"], "ta": "apnic" },

{ "customer": "AS24221", "providers": ["AS1299", "AS3491", "AS4635", "AS4637", "AS4779", "AS6648", "AS7707", "AS9299", "AS15169", "AS15412", "AS17639", "AS18187", "AS18392", "AS23942", "AS24095", "AS24115", "AS55821", "AS59360", "AS135607"], "ta": "apnic" },

{ "customer": "AS32277", "providers": ["AS812", "AS3257", "AS5645", "AS5690", "AS6461", "AS11342", "AS11666", "AS11814", "AS14116", "AS15169", "AS19752", "AS20940", "AS40838"], "ta": "arin" },

{ "customer": "AS35378", "providers": ["AS174", "AS3150", "AS6939", "AS12741", "AS15169", "AS20804", "AS24115", "AS36040", "AS50952", "AS212914"], "ta": "ripe" },

{ "customer": "AS38491", "providers": ["AS1299", "AS3491", "AS4635", "AS4637", "AS4779", "AS6648", "AS7707", "AS9299", "AS15169", "AS15412", "AS17639", "AS18187", "AS18392", "AS23942", "AS24095", "AS24115", "AS55821", "AS59360", "AS135607"], "ta": "apnic" },

{ "customer": "AS135244", "providers": ["AS42", "AS714", "AS2635", "AS2906", "AS3856", "AS6509", "AS8075", "AS9498", "AS13335", "AS15169", "AS20940", "AS21859", "AS23879", "AS26415", "AS32590", "AS32934", "AS54994", "AS55836", "AS59200", "AS133296"], "ta": "apnic" },

{ "customer": "AS200750", "providers": ["AS2906", "AS5617", "AS6507", "AS13335", "AS15169", "AS16509", "AS29535", "AS32934", "AS50607", "AS62047", "AS63293", "AS201054", "AS202520"], "ta": "ripe" },

{ "customer": "AS262393", "providers": ["AS6939", "AS11344", "AS15169", "AS28202", "AS53427", "AS60503", "AS61568", "AS61832", "AS262999", "AS263165", "AS264531", "AS265269", "AS266931", "AS267991", "AS269194"], "ta": "lacnic" },

ASPA adoption coverage on bgp.he.net

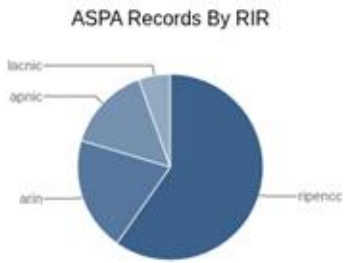
ASPA Adoption report - bgp.he.net/report/rpki_and_aspa



ASNs In Global Routing Table: 88,746

Routed ASNs Covered by ASPA: 3,285

Total ASPA Records: 3,630



RIR	Count
ripencc	2,169
arin	721
apnic	540
lacnic	200

ASNs By RIR

RIR	With ASPA	Total	Coverage (%)
arin	646	20,580	3.14
ripencc	1,931	32,154	6.01
apnic	508	20,546	2.47
afrinic	0	2,172	0.00
lacnic	200	12,705	1.57

Limitations

- Has network effect: Useful when large number of networks do it and large number of AS_PATHs can be validated completely with no gaps
- Does not differ between mixed relation like if a network is “peer” in one continent, while transit in other
- No IPv4/IPv6 granularity: ASN is either provider or not a provider. No option to mark as provider only in IPv4 or IPv6.
- Both ASPA objects and validation has to happen to actually be effective. Similar to RoV situation in RPKI ROA.

Questions?

anurag@he.net