ASPA

RPKI for path validation!

Anurag Bhatia, 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

How filtering is 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 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

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"
}
```

How to create ASPA object?

- ASPA is part of RPKI and ASPA object can be created in same way as ROA
- 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

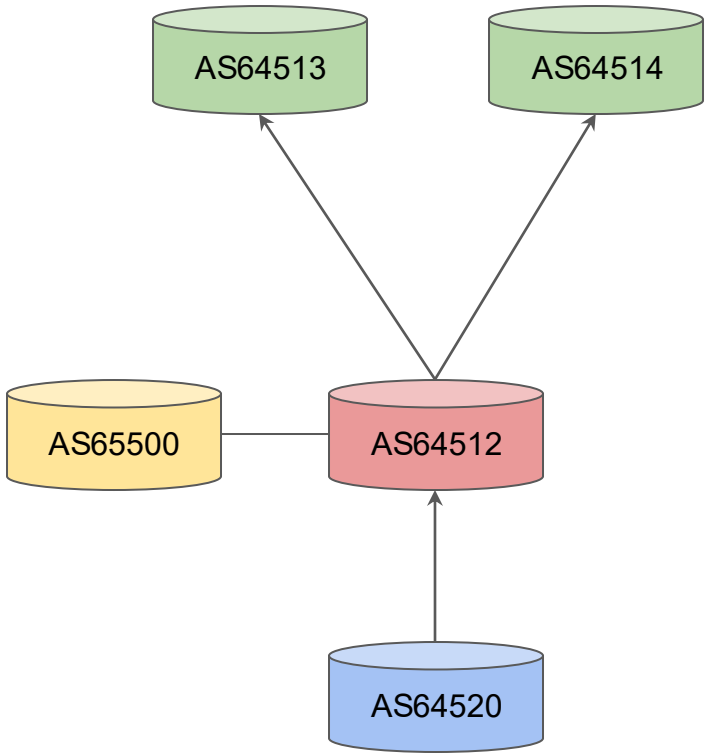
ASPA validation logic

- 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. No ASPA is not dropped (for now)
- 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

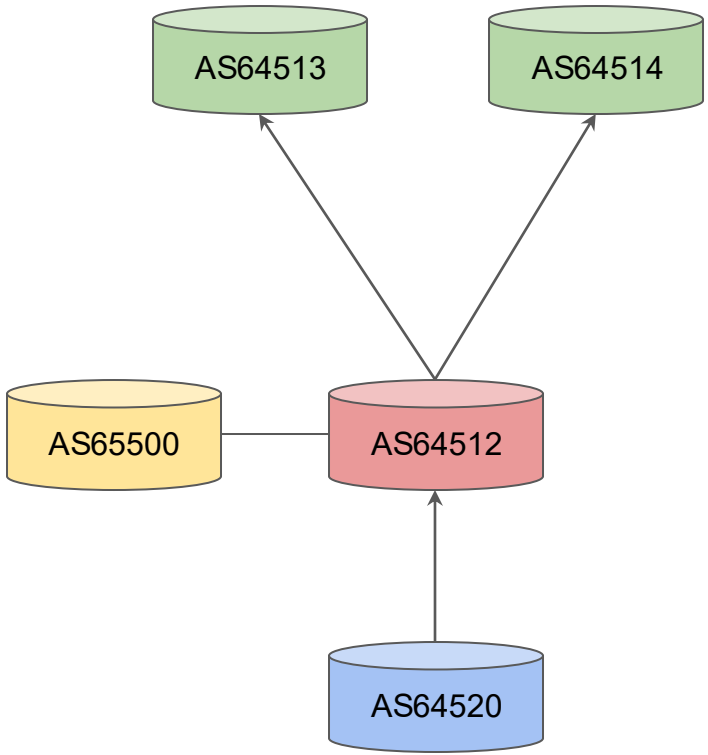
- AS64512 has upstream AS64513 and AS64514
- AS64512 has peering with AS65500
- AS64512 has a customer AS64520
- ASPA object:
  customer: AS64512
  providers: “AS64513”, “AS64514”

  Customer: AS64520
  providers: AS64512



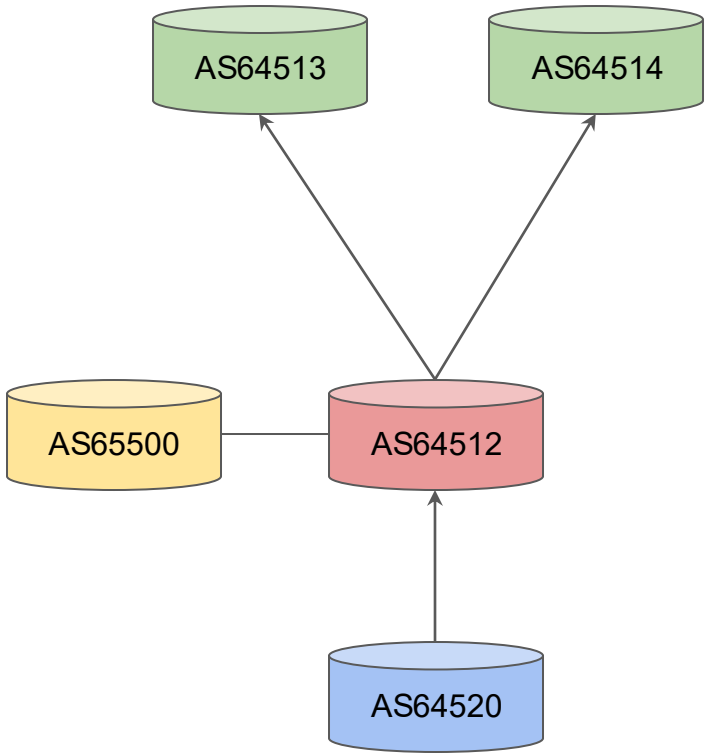
ASPA validation logic

- For AS64513 - they would see AS64520 as: 64513 64512 64520
- They can validate from ASPA object of AS64520 and will mark it as valid route



Case of route leak

- In this case AS64512 can leak AS65500 because of missing ASPA object by AS65500
- If AS65500 has a object, leak would be detected & dropped by network doing the validation

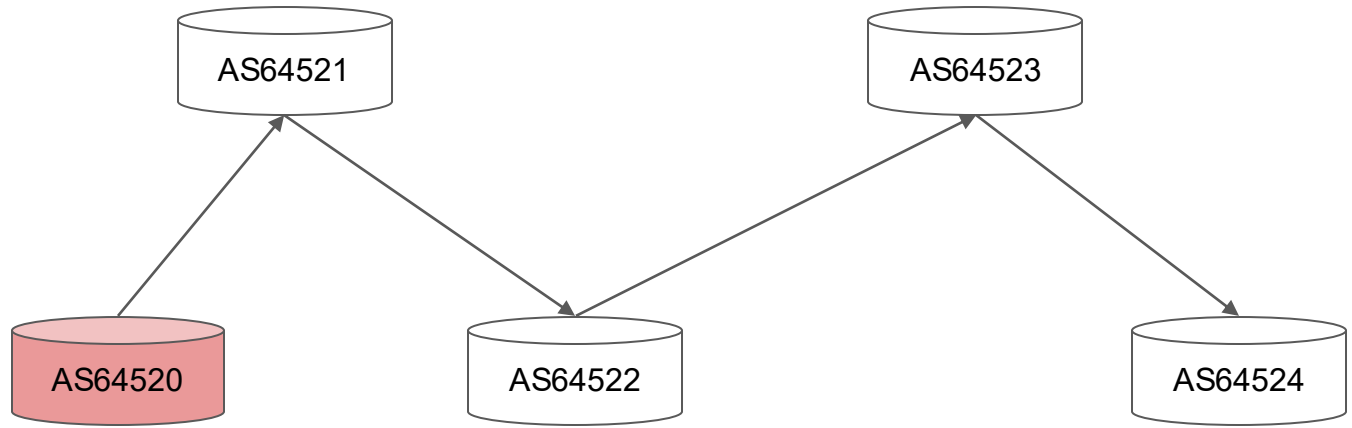


Concept of valley free routing...

Example of valley in routing

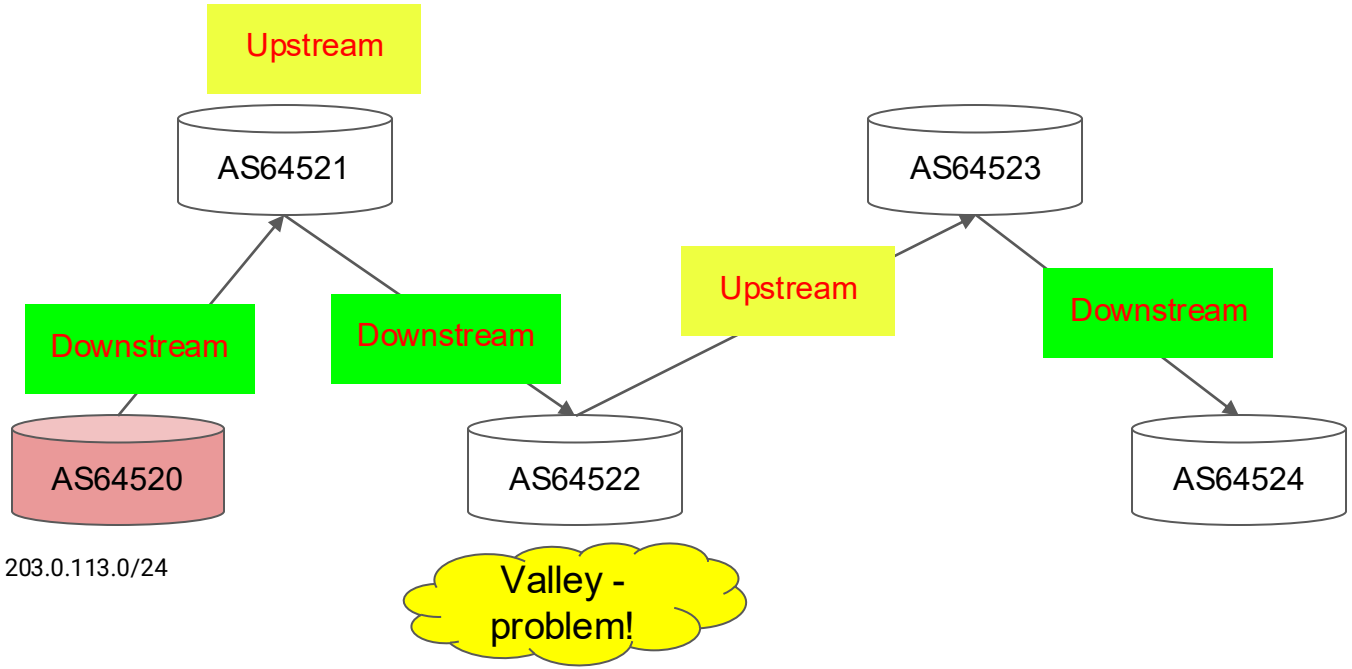
AS64524 table:

203.0.113.0/24
64523 64522 64521 64520



203.0.113.0/24

Example of valley in routing

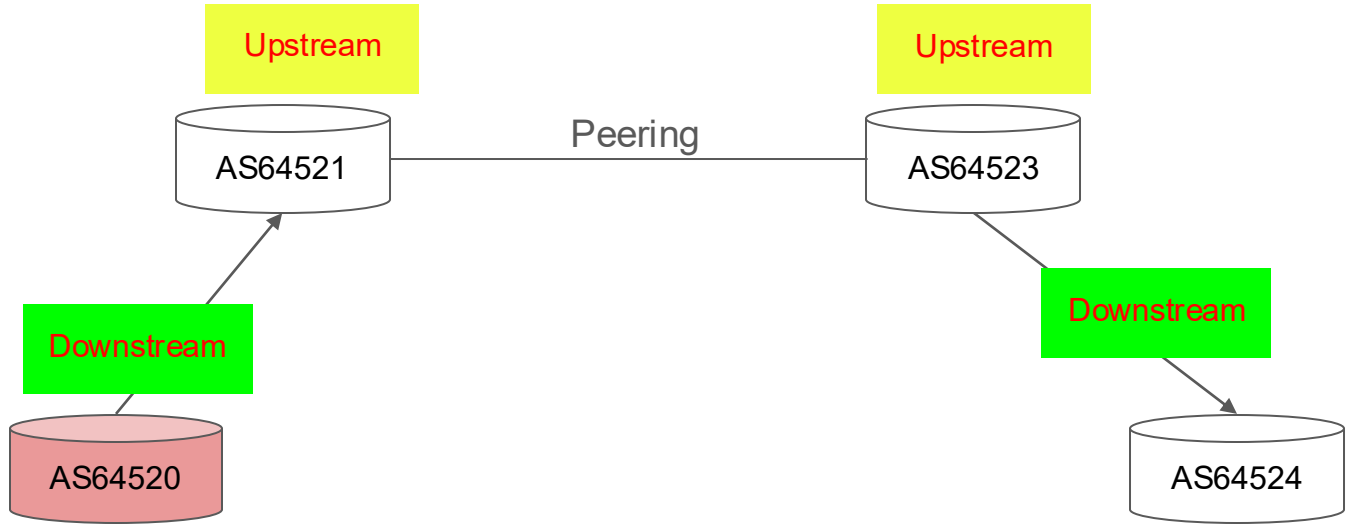


AS64524 table:

203.0.113.0/24
64523 64522 64521 64520

AS64521 does not have AS64522 as upstream (hence AS64522 is downstream) but if AS64522 marked AS64523 as upstream then it's a leak

Example of valley in routing



AS64524 table:

203.0.113.0/24
64523 64522 64521 64520

AS64520 -> AS64521 - Valid!

AS64521 -> AS64523 - No mention in ASPA object (peers)

AS64523 -> AS64524 - Valid!

No Valley - all good!

How ASPA is configured for path validation?

- Routers supporting ASPA 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 ~>
```

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
- 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

ASPA adoption coverage on bgp.he.net

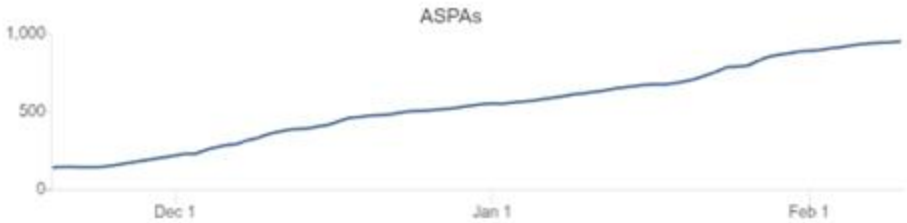
ASPA Adoption report - bgp.he.net/report/rpki_and_aspa

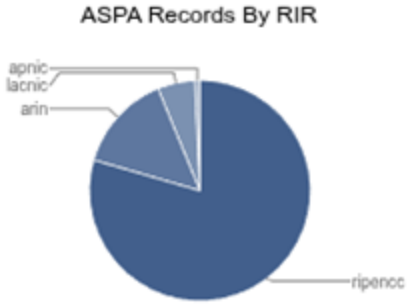
ASPA

ASNs In Global Routing Table: 86,733

Routed ASNs Covered by ASPA: 870

Total ASPA Records: 955





RIR	Count
ripencc	759
arin	137
lacnic	52
apnic	7

ASNs By RIR

RIR	With ASPA	Total	Coverage (%)
arin	130	20,301	0.64
ripencc	683	31,178	2.19
apnic	5	20,026	0.02
afrinic	0	2,085	0.00
lacnic	52	12,440	0.42

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