ASPAs and IETF Updates

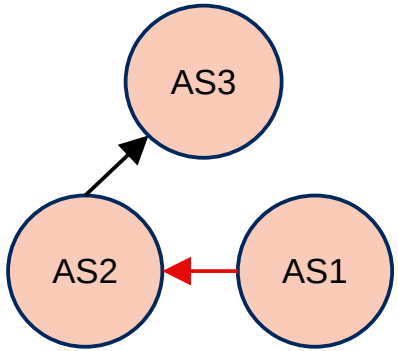
Routing Security SIG

RPKI ASPAs (1)

- **A**utonomous **S**ystem **P**rovider **A**uthorization
- Provides protection against inadvertent route leaks, as well as some types of malicious activity
  - See https://www.kentik.com/blog/beyond-their-intended-scope-bgp-goof-upx/ for a recent example of a problem that ASPAs would have addressed
- Drafts close to finished
  - Will implement based on drafts, but wait for RFCs before releasing
- Planned for completion in Q4 2025

AS	Providers
AS1	AS3

ASPAs



Route path: AS3 AS2 AS1

Route is ASPA-invalid: AS1 does not list AS2 as provider – therefore route leak

RPKI ASPAs (2)

- What are the remaining issues?
  - Size limits on ASN provider set in ASPA objects and rpki-rtr PDUs
  - Total ordering of PDUs in rpki-rtr
  - Various miscellaneous rpki-rtr handling issues
- Importantly, none of these relate to ASPA fundamentals, like the verification algorithm
- Finalising these issues should not take too long

RFC 9697: RRDP Session Desynchronisation

```
<notification session_id="1" serial="1">
  <snapshot uri=".../1/snapshot.xml" hash="s1">
</notification>
```

- T0
  - Client fetches notification file, retrieves snapshot

```
<notification session_id="1" serial="2">
  <snapshot uri=".../2/snapshot.xml" hash="s2">
  <delta    uri=".../2/delta.xml"    hash="d2"
            serial="2">
</notification>
```

- T1
  - Client fetches notification file, retrieves delta to stay up to date

```
<notification session_id="1" serial="3">
  <snapshot uri=".../3/snapshot.xml" hash="s3">
  <delta    uri=".../3/delta.xml"    hash="d3"
            serial="3">
  <delta    uri=".../2/delta.xml"    hash="d9"
            serial="2">
</notification>
```

- T2
  - Client fetches notification file, but hash for the second delta has changed (at T1, the hash was “d2”, but it’s now “d9”)
  - This indicates that something has gone wrong on the server side
  - This RFC requires clients to reinitialise by using the snapshot when this happens

RFC 9674: Same-Origin Policy

```
$ curl -s https://rrdp.example.net/notification.xml
<notification session_id="1" serial="1">
  <snapshot uri="https://rrdp.example.net/1/snapshot.xml"
            hash="s1">
</notification>
$
```

- Notification file origin (in green) matches snapshot origin: file is fine

```
$ curl -s https://rrdp.example.net/notification.xml
<notification session_id="1" serial="1">
  <snapshot uri="https://rrdp.example.org/1/snapshot.xml"
            hash="s1">
</notification>
$
```

- Notification file origin (in green) does not match snapshot origin: file is rejected

- For notification files, snapshots, deltas, and relevant links in RPKI objects
- Also requires checking that RRDP redirects do not send clients to a different origin
- Helps to avoid unnecessary resource usage (one server pointing to another)

RFC 9691: Trust Anchor Keys

- It is difficult for a TA to change its TA key in RPKI:
  - Publish new TAL
  - Alert users to existence of new TAL
  - Keep old and new TALs functional for extended transition period
- This RFC defines a signed object for publishing a new TA key, so that clients can detect the key and transition in-band
- Over the longer term, this should make key transition much easier

2025 APRICOT APNIC 59

PETALING JAYA, MALAYSIA

19 – 27 February 2025

#apricot2025