RPKI validated cache Update in SLURM over HTTPs (RUSH)
draft-madi-sidrops-rush-01

Di Ma
ZDNS and RPSTIR
madi@rpstir.net

IETF 108, Online

RPKI Cache Deployment Scenarios, in RFC 8210

- Not all networks have to set up its own relying party system to do RPKI data synchronization and verification.

- RPKI validated cache is not necessarily transported to routers, but between two intermediate nodes arranged in a hierarchical scheme.

What do we expect for RPKI cache transfer?

- Format / in SLURM
  - Facilitate the data parse for both visualization and applications other than ROV
  - Take advantage of the SLURM (RFC 8416) to do local control

- Transport / over HTTPS
  - Widely supported
  - Security enhancement to offer integrity protection
  - Protocol-Data-Unit independent

RUSH Operations

- Use of SLURM-Incremental Update
  - SLURM-Filters (RFC 8416-Section 3.3) to delete
  - SLURM-Assertions (RFC 8416-Section 3.4) to insert

- Use of HTTPS as Transport
  - The RUSH-aware HTTP server/client MUST be prepared to process media type "application/json-slurm".

```
POST /rpki-cache HTTP/2
Host: rpki.example.com
Content-Type : application/json-slurm
Content-Length:XXX
<XXX bytes represented by the
following json string>
{
 "slurmVersion": 1,
 "validationOutputFilters": {
    "prefixFilters": [...],
    "bgpsecFilters": [...]
  },
  "locallyAddedAssertions": {
    "prefixAssertions": [...],
    "bgpsecAssertions": [...]
  }
}
```

RTR is over there, why bother RUSH to do this

- RTR is designed exclusively for provisioning RPKI cache for routers
  - PDU is bound to transport, which is difficult to add extensions to support VC synchronization management
  - Binary data unit can not be parsed by applications, which by the way do not support RTR generally
- HTTPs is widely deployed and data format independent
- RPKI VC Servers just need ONE API to do both synchronization and local control since JSON is employed by SLURM （RFC 8416） to override global RPKI data

Usecases

- Cache Distribution
  - RUSH can be used to distribute a RPKI validated cache within a single ASN or network, for example a confederation composed of a number of ASes.
  - A small site or enterprise network MAY also use RUSH by synchronizing with a third-party RPKI cache provider over external networks.
- Local Control over Networks
  - Network operators may want to inject SLURM Assertions/Filters via an API offered by RPKI validator/cache. RUSH is therefore able to carry out such local control signals inside an administrative bailiwick in a secure manner.
- AS0 SLURM File Delivery
  - The Regional Internet Registries (RIRs) need to publish assertions with origin AS0 ([RFC6491]) for all the unallocated and unassigned address space (IPv4 and IPv6) for which it is the current administrator. RUSH is able to deliver those assertions to RPKI relying parties if so called AS0 SLURM file would be generated by the RIR.
- Maybe more…

Standardization Considerations

- RUSH merely focuses on a standardized transport and data format of the RPKI cache data.
  - Registration of the "application/ json-slurm " Media Type

- RUSH has nothing to do with synchronization at the RUSH end system
  - Use of RUSH in different usecases may call for different signaling mechanisms.
  - Sophisticated functions could be left to private implementation.
    - PUSH or PULL
    - Automatic re-synchronization
    - Access Control
    - ...

Security Considerations

- Updating RPKI validated cache over HTTPs relies on the security of the underlying HTTP transport.

- An HTTPS connection provides transport security for the interaction between servers, but it does not provide data integrity detection. An adversary that can control the cache used by the subscriber can affect that subscriber's view of the RPKI.

- The RPKI cache server security and the trust model for the interaction between cache server and subscriber is out of the scope of this document.

Adopted?

Questions?