Measuring Relying Parties

John Kristoff, Randy Bush, Chris Kanich, George Michaelson, Amreesh Phokeer, Thomas C. Schmidt, Matthias Wählisch

Contact: jtk@depaul.edu
Overview: RPKI Ecosystem & Relying Parties (RPs)
Overview: RPKI Ecosystem & Relying Parties (RPs)

We are interested in monitoring and understanding this part.
Research and Operational Questions

Completeness
Do relying parties fetch data from all publication points?

Timeliness
How quickly do relying parties fetch data in the wild?

Spec compliance
Do relying parties implement the specs as expected?
Measurement Framework Synopsis

We run
(1) a delegated CA + compare access logs w/ logs from other publication points.
(2) Controlled RPs
(3) ROA Beacons
RPs do not fetch from all PPs equally.

Child PPs that support RRDP see rsync only from RPs that do not support RRDP.

December 2019, research CA/PP “fixes” RRDP.

March 2020 AFRINIC enables RRDP.
Overlap of RP IP Addresses

Surprising number of RPs do not fetch from all PPs, resulting in an incomplete view

Possible reasons for divergence:

IPv4 versus IPv6
Firewalls
Fetching data via an overlay
Inability to follow delegations
## Default Synchronization Interval Time

<table>
<thead>
<tr>
<th>RP Software</th>
<th>RRDP</th>
<th>rsync</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORT Validator</td>
<td>1 hour</td>
<td>1 hour</td>
</tr>
<tr>
<td>OctoRPKI</td>
<td>20 minutes</td>
<td>20 minutes</td>
</tr>
<tr>
<td>Routinator</td>
<td>10 minutes</td>
<td>10 minutes</td>
</tr>
<tr>
<td>Dragon Labs</td>
<td>1 hour</td>
<td>1 hour</td>
</tr>
<tr>
<td>OpenBSD rpki-client</td>
<td>not implemented</td>
<td>1 hour</td>
</tr>
<tr>
<td>RIPE NCC Validator 2</td>
<td>1 minute</td>
<td>10 minutes</td>
</tr>
<tr>
<td>RIPE NCC Validator 3</td>
<td>2 minutes</td>
<td>10 minutes</td>
</tr>
</tbody>
</table>
Observed RP Synchronization Intervals @ Research PP

Details depend on PP, but RRDP usually more predictable
Filtering RRDP vs rsync Experiment

**Objective:** Verify mandatory support of rsync

**Expectation:** When availability of RRDP stops, increase of rsync access because of rsync fall back

**Method:** Filter RRDP dynamically
Filtering RRDP vs rsync Experiment
Filtering RRDP vs rsync Experiment
Filtering RRDP vs rsync Experiment

At least Routinator and OctoRPKI did not fall back to rsync.
RP Software Popularity, March 2020

We leverage the RRDP HTTP User-Agent string

Most RPs use both RRDP and rsync

We see RRDP fingerprints for ~95% of all RPs that connect w/ rsync this way
Conclusions

Significant portion of RPs do not obtain a complete or timely view

Migration between RRDP and rsync does not work as defined

Surprisingly hard to map access on multiple PPs to a single RP
Future Work

Deploy delegated CA/PP infrastructure in each RIR region

Evaluate RP software object fetch and validation consistency

Evaluate data on RP cache servers

Evaluate timeliness between BGP updates and ROA updates
Backup
Our Contributions

Reproducible measurement framework to evaluate RPKI data dissemination completeness

Survey RP software, access protocols, and synchronization patterns

Characterize shortcomings in RP synchronization behavior

Identify mismatch in RP software behavior and protocol design expectations