RPKI Repository Fetch Protocols

Rob Austein <sra@hactrn.net>
Randy Bush <randy@psg.com>
Michael Elkins <Michael.Elkins@sparta.com>
… and a lot of help from our friends

IETF 84
Vancouver
July 2012

Fun With rsync

rsync was a reasonable first choice while developing the RPKI, but has some issues:

- Race conditions due to object level granularity
  - . . . and the black box interface between rsync and the underlying filesystem (on which Unix variant?) will likely defeat any attempt to code around this
- rsync is not well designed to be run by other programs
  - Quick, what does exit code #23 mean again?
- Repository tree stucture has heavy impact on efficiency
  - Connection setup is expensive, want to amortize cost

(Summarized from a report by Tim Bruijnzeels—Thanks!)

More On Connection Setup

- Connection setup is expensive for relying party: process slot, delay waiting for setup, comparision of local disk against server
- Connection setup is also expensive for repository: process slot, comparison of local disk against relying party.
- This happens every time, whether anything has changed or not
- For big repositories, this happens hundreds or thousands of times per relying party, per validation session
- And there's no analogue of DNS Time-To-Live (TTL), so there's no way for repository to signal how long relying party should wait before pestering repository again

Hierarchy & URIs

- The hierarchy that really matters for RPKI validation is the certificate hierarchy
- URI hierarchy (rsync or otherwise) is just a (sometimes) convenient way of fetching big chunks of the certificate hierarchy efficiently
- But we still want some kind of canonical URI for every object
- Since rsync is already mandatory-to-implement and every object already has an rsync URI, we may as well keep that as the canonical URI unless and until we have a better idea

Zones

- Term borrowed from DNS: A sub-tree of objects which can be transfered as a unit, with an “apex” at the top and “zone cuts” where leaves of the sub-tree point to further data outside the zone
- How do we identify zone cuts?
- Well, we still have SIA pointers, RPKI manifests, and canonical URIs
- And any bulk operation is going to have to include some kind of list of what’s in the bulk transfer
- So an SIA pointer to a directory that’s not in the zone would be a zone cut
- . . . or a corrupt zone, but this is (probably) no worse than the risk of an attack on rsync transfers—in either case, the real defenses are the RPKI structure, manifests, and signatures

Data Freshness

- How close does the relying party get to having an accurate picture of what the CA wants the relying party to see?
- We don’t have a good handle on this yet
- Experiments in progress hope to give us more data
- See other presentations, if they’re cooked in time

Possible approaches

It's probably a bit early to be diving into solution space, but there are at least two general approaches here:

1. Add optional SIA URI for some protocol which works like DNS AXFR/IXFR
2. Add optional SIA URI for something like an ATOM feed

These should be viewed as examples of possible approaches, not literal proposals (yet), so don't get hung up on questions like whether we really mean "ATOM" here

AXFR/IXFR-like approach

Optional SIA URI for a protocol which works like rpki-rtr or DNS AXFR/IXFR:

- Some kind of versioned zones, with serial numbers or timestamps
- Data within a zone consists of <URI, object> pairs
- Relying party keeps track of version it saw last, asks for changes from that version to current
- Simple for relying party
- Requires direct contact with authoritative servers

ATOM-like approach

Optional SIA URI points to an ATOM feed:

- Elements in feed point to real data source
- Fetch is two step process
- This approach works for:
  - HTTP-cachable blobs *e.g.*:

```
<zone serial="...">
  <object uri="...">Base64</object>
  <object uri="...">Base64</object>
</zone>
```

  - BitTorrent, perhaps using "magnet links"
  - (Your favorite protocol here)
- More work for relying party
- An awful lot of rope, would want to profile tightly
- Requires direct contact with authoritative server (feed), but extra level of indirection means that real data could be elsewhere

Thanks To. . .

- The other RPKI relying party software implementers, particularly Tim Bruijnzeels
- DHS:[1] Taking away your scissors and beating them into plowshares

[1]This work is sponsored in part by the US Department of Homeland Security under an interagency agreement with the Air Force Research Laboratory (AFRL)

Questions?