draft-huston-sidr-repos-struct-01.txt

George Michaelson

Geoff Huston

APNIC

Where were we?

- Back at IETF66… -00.txt
  - Highly structured repository architecture
    - Proscriptive naming
    - Nested inheritance of name strictly enforced
    - Followed g(ski) name model
  - Nobody liked it
    - g(ski) names too much like stuff PKIX rejected
      - Didn't cope with key rollover well
  - Think I'll go and eat worms…

Where are we?

- No proscriptive language about terminal object names
- Talks about system behaviours
  - Persistent URLs (rsync/http…)
  - Some language about nesting
  - Example validation algorithm
- Clearer differentiation of entities in repository
  - EE vs CA vs signed-objects vs CRLs

Repository Structure

- Distributed Repository Framework
- CA publishes Certificates, CRL and Manifest
  - Name scheme used may allow the most recent published object to overwrite the older versions of the same logical object in the publication repository
- EE publishes objects signed by the EE's key pair
  - All subordinate EE certificates from a single CA may share a common repository publication point

Repository Good Housekeeping Guide

- Use a highly available platform
- RSYNC access should be supported
- Each repository to contain the products of a single CA

Relying Parties

- Relying Parties may elect to aggregate the repository collection through the maintenance of a local RPKI repository cache
- Draft suggests a regular “top-down” walk across the distributed repository set as a possible approach to maintenance of a local cache of all valid objects that have been published within the RPKI framework

Where are we going ?

- Talks about validation
  - Worst-case algorithm to fetch/validate a local repository
  - You can do *much* better than this if you are smart!
    - Index on g(ski), {issuer,subject} etc
    - Rsynic is smarter than this..
- Keep it simple
  - the naming, hierarchy will be local choice
  - Avoid cross contaminating CA/EE publication points in a consistent namespace
  - Persistence of the name across key rollover is useful
    - Can’t always do this (eg EE Cert outcomes, Manifests)

Are we there yet?

- Unlikely to converge on proscriptive naming
  - Check!
- Useful to understand behaviors of the system as a whole
  - Check!
- Specify minimum requirements and move on
  - Check?