Summary
SIDR Interim 27 Jul 2012

Agenda

1) Deployment Considerations in RPKI – Measurements and Data
   a) Tim Bruinzeels
   b) Randy Bush
2) Deployment Considerations in RPKI – Alternative Communication Designs
   a) Tim Bruinzeels
   b) Rob Austein
3) Deployment and BGPSEC Protocol

Measurements and Data

- Tim Bruiinzeels
- Difficulties with current deployment – repository reliability, no proxying/cacheing for performance, object organization choices at some places (flat rather than hierarchic) causes problems for RPs, etc.
- Rsync issues – no transactions, heavy load on server, no libraries, no real spec, error msgs bad, etc

Measurements and Data

- Recounted experiment with volunteer test objects
  - Simulated hierarchic by prefetching and modifying URIs
  - 15K reports from 37 different instances – great differences between clients and between runs
    - See slides for graphs
  - V6 problems
- Ran experiment in small lab machines to test what **eventual** load might be
  - **This load is some time in the future**
  - 12K CAs, 70K objects
  - Rsync thoughput vs # concurrent clients had cliff as number of forks exhausted memory
  - Can disallow recursive fetching (but then loose advantage of hierarchic orgnanization)

Measurements and Data

- Randy Bush
- Early report on RPKI Propagation Emulation Measurement
  - Propagation: time from CA publish to Relying Party
  - See slides for nifty keeno experimental setup
- Measurements Desired
  - Propagation characteristics (sensitivity to cache RTT, timers)
  - Split between propagation and validation
- Distinguishing gatherers (who synchronize with global repository system) and caches (who feed off gatherers or other caches)
- Results (**EARLY**) say propagation time is 500-2000 sec depending on to/from RIPE/RIR/gatherer/cache/router
  - … and flat organization moves curve to right

Measurements and Data

- Randy Bush
- Measurements of current deployments
  - See slides for graphs
- Some repositories have poor reliability records
  - RIRs aren’t 24x7 operators (outages on weekends, etc)
  - Lack of response to reports of problems
- But RIPE number of objects is up and to the right – which is an EXCELLENT thing

Measurements and Data

Consensus

- Problems are NOT barrier to deployment
- rsync useful first implementation
  - no re-inventing of syncing protocol or incremental fetching
  - works, mostly
  - good enough to build up experience
  - but see next topic about beginning to look at alternatives
- flat organization causes problems for RPs
  - this can be (should be) fixed in those repositories that do it
  - some discussion of overt way to communicate this to community
- relying party software needs to expect problems
- need more monitoring (more eyes, more tools)
- Some discussion of need for doc describing repository ops

Alternative Communication Designs

- Tim Bruiinzeels
- Rsync
  - pros: can retrieve incrementally for RPs,
  - cons: incremental is hard for servers
- Http
  - Proven protocol, implementations, etc
    - Implementation ease: native libraries, error msgs, etc
  - Difficult to retrieve increments, so hard for RPs
    - And latency has huge impact on performance unless parallelize fetching

Alternative Communication Designs

- Rsync deltas are good for RPs, bad for servers
- Discussion of alternate with update notifications and http (RPs compute the deltas)

Alternative Communication Designs

- Rob Austein
- Rsync issues; Flat hierarchy means more connections
- Discussed a few ideas to play with
  - Dns-like zones, ATOM+Bittorrent, etc
- Important to consider data freshness
  - How close can RP come to CA published data
    - Need more measurements

Alternative Communication Designs

CONSENSUS

- We need to deploy rsync basis **NOW**
- We will need an alternative **eventually**
- We need to begin work on the alternative **NOW**
- Just what do we need in an alternative, i.e., requirements? (and do we need a doc on this?)
  - Incremental fetching
  - Fetch object structure should reflect logical object structure (ie cert hierarchy)
- There are only so many ways to slice this bread
  - Choose impact: server, client, or (and/or) network

BGPSEC Protocol

- New version has section on confed handling
- Discussion of need to include the target AS in the sig attribute – to handle AS aliasing cases
  - Using pcount=0 to solve this
  - Discussion of separate doc to show how this works
- Discussion of (in)stable signatures
  - ECDSA produces different signatures over the exact same data
  - So duplicate updates won't look like duplicate updates if you are just doing strict compare
  - Could be just "advice to implementers" – Matt will do