DRAFT-SPAGHETTI-GROW-NRTM-V4

SASHA ROMIJN (RELIABLY CODED)
SASHA@RELIABLYCODED.NL

JOB SNIJDERS (FASTLY)
ED SHRYANE (RIPE NCC)
STAVROS KONSTANTARAS (AMS-IX)

IRR MIRRORING

- The Internet Routing Registry (IRR) is actually around a dozen authoritative registries
- Different operators, purposes, scope, policies, sizes
- Mirroring exists for:
  - Offering a more complete view to operators rather than just “your” IRR Database
  - Local performance (non-authoritative)
- A few common implementations:
  - Various versions of RIPE db software (RIPE, RIPE-NONAUTH, AFRINIC, APNIC)
  - Legacy IRRD v2/3 (RADB, LEVEL3, ALTDB, ...)
  - IRRD v4 (NTTCOM, ARIN, ARIN-NONAUTH, LACNIC, TC)

CURRENT: NRTM V3 + FTP

- Poorly specified
- Plain text over raw TCP sockets
- No integrity or authenticity check of any kind
- Poor scaling
- Poor status signalling (broken vs no new data)
- NRTM v3 stream poorly linked to FTP full downloads
- Many fascinating ways it can break quietly and weirdly

DRAFT-SPAGHETTI-GROW-NRTM-V4

- Published on HTTPS endpoint
- Small Update Notification File as an index
  - pointing to a snapshot and (usually) deltas with changes
- Snapshots are periodic with all records in the IRR Database
- Deltas only contain changes
  - batched into one minute timeframes, if any changes
- All files are JSON and support UTF-8
- Snapshots and Deltas immutable and cacheable
- Update Notification File is signed, and then contains hashes of all referred files
- Random session ID that must match, or otherwise reinitialise from snapshot
- Inspiration from RRDP

MIRROR SERVER USE: INITIALISATION

- Generate random new session ID
- Generate snapshot for version 1 (may be empty)
- Generate and sign new Update Notification File

- Note: always per IRR Database (“source”)

MIRROR SERVER USE: UPDATES

- Generate Delta File every minute, if changes happened
  - Catching up with one Delta File for a larger time frame is permitted
  - Sequential version number
- Generate and sign new Update Notification File

- Generate Snapshot File periodically, if there were changes
- Version number equal to latest Delta File version

- Update Notification File regenerated at least every 24 hours, even without changes

- No per-client variation in data

MIRROR CLIENT USE

- Retrieve Update Notification File and signature, verify

- If there is existing data, and session ID still matches:
  - Server has same version?
    - Do nothing
  - Server has newer version and deltas available from existing local version to latest?
    - Retrieve those deltas, verify hashes, process changes

- Otherwise, (re)initialise from snapshot:
  - Retrieve, verify hash, process snapshot

UPDATE NOTIFICATION FILE


```
{
  "nrtm_version": 4,
  "timestamp": "2022-01-00T15:00:00Z",
  "type": "notification",
  "source": "EXAMPLE",
  "session_id": "ca128382-78d9-41d1-8927-1ecef15275be",
  "version": 3,
  "snapshot": {
    "version": 2,
    "url": “https://example.com/ca..be/nrtm-snapshot.2.047595d0fae972fbed0c51b4a41c7a349e0c47bb.json",
    "hash": "9a..86"
  },
  "deltas": [
    {
      "version": 1,
      "url": “https://example.com/ca..be/nrtm-delta.1.784a2a65aba22e001fd25a1b9e8544e058fbc703.json",
      "hash": "62..a2"
    },
    {
      "version": 2,
      "url": “https://example.com/ca..be/nrtm-delta.2.0f681f07cfab5611f3681bf030ec9f6fa3442fb0.json",
      "hash": "25..9a"
    },
    {
      "version": 3,
      "url": “https://example.com/ca..be/nrtm-delta.3.d9c194acbb2cb0d4088c9d8a25d5871cdd802c79.json",
      "hash": "b4..13"
    }
  ]
}
```

SNAPSHOT FILE


```
{
  "nrtm_version": 4,
  "type": "snapshot",
  "source": "EXAMPLE",
  "session_id": "ca128382-78d9-41d1-8927-1ecef15275be",
  "version": 2,
  "objects": [
    "route: 192.0.2.0/24\norigin: AS65530\nsource: EXAMPLE",
    "route: 2001:db8::/32\norigin: AS65530\nsource: EXAMPLE"
  ]
}
```

DELTA FILE


```
{
  "nrtm_version": 4,
  "type": "delta",
  "source": "EXAMPLE",
  "session_id": "ca128382-78d9-41d1-8927-1ecef15275be",
  "version": 3,
  "changes": [
    {
      "action": "delete",
      "existing_object": "route: 192.0.2.0/24\norigin: AS65530\nsource: EXAMPLE"
    },
    {
      "action": "add_modify",
      "new_object": "route: 2001:db8::/32\norigin: AS65530\nsource: EXAMPLE"
    }
  ]
}
```

WIDE OPEN TO DISCUSSION: DATABASE CONFIGURATION FILE

- Update Notification File is signed and contains hashes of snapshots/deltas
  - For end-to-end integrity, in addition to HTTPS

- Problem: how does a mirror client know which public key to trust?

PROPOSED SOLUTION: DATABASE CONFIGURATION FILE

- Database Configuration File contains IRR Database name (source), public signing key, Update Notification File URL
- Published on well-known URI on the “generally known” authoritative domain, low traffic
- Enables automatic configuration and key rotation

```
# e.g. https://www.ripe.net/.well-known/irr-nrtm.json
[
  {
    "source": "EXAMPLE-A",
    "update_notification": "https://example.net/db/nrtmv4/example-a/notification.json",
    "key": "96..ae"
  },
  {
    "source": "EXAMPLE-B",
    "update_notification": "https://cdn.example.com/db/nrtmv4/example-b/notification.json",
    "key": "b6..3d"
  }
]
```

WIDE OPEN TO DISCUSSION: DELTA EXPIRATION

- Many clients will closely follow deltas, only retrieving 0-5 at a time
  - we want these to have low latency, hence delta every minute
- Some clients will lag behind and need to catch up with days, sometimes weeks
- Up to 1440 deltas generated per day if every minute has a change
  - though for many databases much less (~100s)

WIDE OPEN TO DISCUSSION: DELTA EXPIRATION

- Initial idea: mirror server must trim deltas so that their cumulative size is less than snapshot
  - Intention: a client downloading 1GB of deltas is better than 2GB of snapshot
  - Reality: rate of change in IRR is too low, and too spread out, too many tiny deltas
  - Estimate from real data: a client running behind 3 years for RIPE will still prefer deltas over snapshot and try to download (order of magnitude) 250.000 delta files

WIDE OPEN TO DISCUSSION: DELTA EXPIRATION

- New plan in draft: fast delta expiry after 24 hours
  - Simple to implement
  - Still may have up to 1440 Delta File downloads
  - Reasonable balance between allowing catch up and preventing excessive downloads
  - Remember re-download of snapshot is all automatic

- Alternate idea: delta aggregation
  - Aggregate older delta files into larger blocks in a single download
  - Longer recovery is possible without re-downloading snapshot
  - More optimal for some clients, at cost of more complexity

STATUS AND NEXT STEPS

- Draft authors already involved with IRRD v4 and RIPE db
- RIPE Database working group: broad strokes discussed in BoF and generally supported

- Posted in February on GROW mailing list, awaiting feedback
- Working group adoption?