Route Object processes in WHOIS

George Michaelson, APNIC R&D

Route Object Template

route:	[mandatory]	[single]	[primary/look-up key]
descr:	[mandatory]	[multiple]	[ ]
origin:	[mandatory]	[single]	[primary/inverse key]
holes:	[optional]	[multiple]	[ ]
country:	[optional]	[single]	[ ]
member-of:	[optional]	[multiple]	[ ]
inject:	[optional]	[multiple]	[ ]
aggr-mtd:	[optional]	[single]	[ ]
aggr-bndry:	[optional]	[single]	[ ]
export-comps:	[optional]	[single]	[ ]
components:	[optional]	[single]	[ ]
remarks:	[optional]	[multiple]	[ ]
notify:	[optional]	[multiple]	[inverse key]
mnt-lower:	[optional]	[multiple]	[inverse key]
mnt-routes:	[optional]	[multiple]	[inverse key]
mnt-by:	[mandatory]	[multiple]	[inverse key]
changed:	[mandatory]	[multiple]	[ ]
source:	[mandatory]	[single]	[ ]

Route Object Template

```
route:          [mandatory]   [single]     [primary/look-up key]
descr:          [mandatory]   [multiple]   [ ]
origin:         [mandatory]   [single]     [primary/inverse key]
holes:          [optional]    [multiple]   [ ]
country:        [optional]    [single]     [ ]
member-of:      [optional]    [multiple]   [ ]
inject:         [optional]    [multiple]   [ ]
aggr-mtd:       [optional]    [single]     [ ]
aggr-bndry:     [optional]    [single]     [ ]
export-comps:   [optional]    [single]     [ ]
components:     [optional]    [single]     [ ]
remarks:        [optional]    [multiple]   [ ]
notify:         [optional]    [multiple]   [inverse key]
mnt-lower:      [optional]    [multiple]   [inverse key]
mnt-routes:     [optional]    [multiple]   [inverse key]
mnt-by:         [mandatory]   [multiple]   [inverse key]
changed:        [mandatory]   [multiple]   [ ]
source:         [mandatory]   [single]     [ ]
```

Asserts an origin-AS in origin: over the prefixes in route:

ROA Object ASN.1 (RFC6482)

```
RouteOriginAttestation ::= SEQUENCE {
         version [0] INTEGER DEFAULT 0,
         asID  ASID,
         ipAddrBlocks SEQUENCE (SIZE(1..MAX)) OF ROAIPAddressFamily }

      ASID ::= INTEGER

      ROAIPAddressFamily ::= SEQUENCE {
         addressFamily OCTET STRING (SIZE (2..3)),
         addresses SEQUENCE (SIZE (1..MAX)) OF ROAIPAddress }

      ROAIPAddress ::= SEQUENCE {
         address IPAddress,
         maxLength INTEGER OPTIONAL }

      IPAddress ::= BIT STRING
```

ROA Object ASN.1 (RFC6482)

```
RouteOriginAttestation ::= SEQUENCE {
        version [0] INTEGER DEFAULT 0,
        asID  ASID,
        ipAddrBlocks SEQUENCE (SIZE(1..MAX)) OF ROAIPAddressFamily }

    ASID ::= INTEGER

    ROAIPAddressFamily ::= SEQUENCE {
       addressFamily OCTET STRING (SIZE (2..3)),
       addresses SEQUENCE (SIZE (1..MAX)) OF ROAIPAddress }

    ROAIPAddress ::= SEQUENCE {
       address IPAddress,
       maxLength INTEGER OPTIONAL }

    IPAddress ::= BIT STRING
```

Asserts an origin-AS in asID over the prefixes in ipAddrBlocks with optional maxLength

Observations

- Route objects
  - Simple use, specify origin-as/prefix relationship
  - *Complex use, to add route qualifiers/aggregation*
- Simple use invokes behavior similar to a ROA
- ROA creation vests with prefix holder only
  - No AS holder input
  - Strong trust in the system to make ROA
- ROUTE: object creation vests in AS and prefix holder
  - Both must countersign a create or update request, if not the same maintainer.

Observations

- Process delays at APNIC
  - Post- final /8 policy, more people present with prefix only, to be routed by ISP/provider
  - But process to get route object countersigned is clumsy, different provisioning paths in ISP
  - APNIC increasingly asked to intervene, adds delay
- Goal: simplify processes and speed up announcement of new prefixes

Route objects in RIPE IRR

- 211,640 **`route`** objects
- 6,924 **`route6`** objects
- How may use ‘complex’ substructure?
  - ‘boring’ objects consist only of:
    - 'route’, 'descr’, 'origin’, 'mnt-by’, 'notify’, 'changed’, 'source', 'remarks’, 'org’, 'holes’, 'mnt-lower’, 'mnt-routes’
  - ‘boring’ because these are unrelated to AS, or IRR support not related to either inetnum/inet6num or aut-num

- 3,077 objects reference any of pingable, ping-hdl (arguably boring) member-of, inject, export-comps, components, aggr-bndry, aggr-mtd

Non-boring Route/Route6 objects

4 aggr-bndry

7 export-comps

13 components

50 ping-hdl

106 aggr-mtd

111 pingable

145 inject

2995 member-of

1.45% of all route objects

Route objects in APNIC IRR

- 83,312 **`route`** objects
- 52,919 **`route6`** objects

- 129 ‘not boring’
  - 2 components
  - 2 export-comps
  - 127 member-of

0.15% of all route objects

Route objects in APNIC IRR

- 83,312 **`route`** objects
- 52,919 **`route6`** objects

Wait. What?

Route objects in APNIC IRR

- 83,312 **`route`** objects
- 52,919 **`route6`** objects

Wait. What? 52,000 route6????

Route objects in APNIC IRR

- 83,312 **`route`** objects
- 52,919 **`route6`** objects

```
$ zmore apnic.db.route6.gz | \
    grep route6 | awk '{print $2}' | cut -d: -f1,2| \
      sort –u |wc
     318      318     3148
$
```

Indian /32 holder de-aggregating to /48

Ideas?

- Searching for feedback and suggestions
  - Modest suggestions of our own.
  - Or, what you'd like automated
- Bring RPKI and IRR into alignment
  - "say the same things"
  - Getting a lot to manage
  - Avoid adding to workload

- How about an 'Automatic ROA' creation?
  - Match ROA with route objects. Consistent state in both systems
  - If we go down the 'create a ROA' path
    - Automatic route object
    - Sensible idea? Worth exploring?

Change WHOIS permissions model for route/route6 objects?

- Change WHOIS permission model to permit simple route: object creation solely on permission of prefix holder.
  - Risk: route objects used for filters, AND used for announcement
    - Problem vests with filters: inetnum holder can't be seen without a route object, as holder not actively using IRR
    - For IRR user configuring a router route: object can cause injection of routes AS holder doesn't want to announce.
  - Considering HM authorized override with communication to AS maintainer OOB
    - Eg 72h notice of change unless countermanded

Whois Tags?

- Root cause: route: object serves dual purpose
  - Configure filters to accept origin-as
  - Configure routers to define announcement
- If we tag inetnum-authorized only route: objects then by ‘definition’ they are filter-suitable only.
- AS operator can choose to add auth and then invoke local IRRtoolset to make announcement
- Route object will gatekeep filters for origin-AS that do not use IRR