IXP Manager & Route Servers

Route Servers Video Tutorial Series - Part 5

Barry O'Donovan - @barryo79, June 2021

Route Server IRRDB Filtering with IXP Manager

Demonstration

— [x] Build the route server
— [x] Show clients connected and routes
— [x] IPv6 instance
— [x] Looking glass
— [x] Community filtering
— [ ] IRRDB filtering
— [ ] RPKI filtering

IRRDB Filtering

IRRDB Filtering

IRRDB filtering (and RPKI) used to ensure that a route server participant can only advertise routes that they should be able to advertise.

- LIRs register routes with routing registries.
  - APNIC, RIPE, etc. but also commercial such as RADB
- Quality of records vary greatly
- IRRDB based filtering has been - and is - the standard

IRRDB Filtering - Example Records

```
route:     192.0.2.0/24
descr:     Packet Loss Ltd
origin:    AS65501
mnt-by:    JOE-MNT
source:    APNIC

route:     2001:db8::/32
descr:     Packet Loss Ltd
origin:    AS65501
mnt-by:    JOE-MNT
source:    RIPE
```

IRRDB Filtering - Generating Prefix Lists

```
$ bgpq3 -j as58372
{ "NN": [
    { "prefix": "103.29.204.0\/22", "exact": true },
    { "prefix": "103.29.204.0\/24", "exact": true },
    { "prefix": "103.29.205.0\/24", "exact": true },
    { "prefix": "103.29.206.0\/24", "exact": true },
    { "prefix": "103.29.207.0\/24", "exact": true }
] }

$ bgpq3 -6j as58374
{ "NN": [
    { "prefix": "2402:9100::\/32", "exact": true }
] }
```

IRRDB Filtering - AS Sets

— Important for members with downstream networks
— Currently a gap in RPKI functionality (AS Cones?)
— BGPQ3 and IXP Manager will recursively unwrap AS sets

```
$ whois AS-HEANET
as-set:          AS-HEANET
descr:           Autonomous Systems routed by HEAnet
members:         AS1213, AS2128, AS112, AS42310, AS2850, AS-IEDR
...
```

IRRDB Filtering - IXP Manager

— Database updated every 6 hours via the scheduler
— Route server config updated via the scripts
— Transaction safe - won't trip over each other
— Manually via the UI
— Manually via Artisan (command line tool)

```
$ ./artisan irrdb:update-asn-db -vvv
Aptus: [IPv4: 1 total; 0 stale; 0 new; DB updated] [IPv6: 1 total; 0 stale; 0 new; DB updated]
    Time for net/database/processing: 0.921408/0.010303/0.000834 (secs)
$ ./artisan irrdb:update-prefix-db -vvv
Aptus: [IPv4: 7 total; 0 stale; 0 new; DB updated] [IPv6: 1 total; 0 stale; 0 new; DB updated]
    Time for net/database/processing: 1.100500/0.014494/0.000373 (secs)
```

IRRDB Filtering - IXP Manager

Previous route server filter config when IRRDB was disabled:

```
# Skipping RPKI check -> RPKI not enabled / configured correctly.
bgp_large_community.add( IXP_LC_INFO_RPKI_NOT_CHECKED );

# This ASN was configured not to use IRRDB filtering
bgp_large_community.add( IXP_LC_INFO_IRRDB_NOT_CHECKED );
```

What does it look like now?

IRRDB Filtering - IXP Manager

```
allas = [ 49567 ];

# Ensure origin ASN is in the neighbors AS-SET
if !(bgp_path.last_nonaggregated ~ allas) then {
    bgp_large_community.add( IXP_LC_FILTERED_IRRDB_ORIGIN_AS_FILTERED );
    accept;
}

# Skipping RPKI check -> RPKI not enabled / configured correctly.
bgp_large_community.add( IXP_LC_INFO_RPKI_NOT_CHECKED );

allnet = [ 31.217.240.0/21, 45.154.100.0/22, ... ];

if ! (net ~ allnet) then {
    bgp_large_community.add( IXP_LC_FILTERED_IRRDB_PREFIX_FILTERED );
    bgp_large_community.add( IXP_LC_INFO_IRRDB_FILTERED_STRICT );
    accept;
} else {
    bgp_large_community.add( IXP_LC_INFO_IRRDB_VALID );
}
```

Demonstration

— [x] Build the route server
— [x] Show clients connected and routes
— [x] IPv6 instance
— [x] Looking glass
— [x] Community filtering
— [x] IRRDB filtering
— [ ] RPKI filtering

Coming in Part 6:

Securing Route Servers with RPKI and IXP Manager

Thanks for watching!

- https://www.ixpmanager.org/
- https://docs.ixpmanager.org/
- https://www.barryodonovan.com/
- @barryo79 on Twitter
- barry.odonovan@inex.ie