IXP Manager & Route Servers

Route Servers Video Tutorial Series - Part 6

Barry O'Donovan - @barryo79, June 2021

Securing Route Servers with RPKI and IXP Manager

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

RPKI

RPKI Filtering

— An IRRDB entry was a prefix and an origin ASN
— RPKI is a cryptographically secure replacement
    — Adds maximum prefix length
— Yields route origin triplets that have been validated

```
( Origin AS, Prefix,          Max Length )
( AS65500,   2001:db8::/32,   /48        )
( AS65501,   192.0.2.0/24,    /24        )
```

RPKI Validators

An RPKI Validator (aka Relying Party software- rely: depend on with full trust) downloads and verifies the global RPKI data set and can be used to feed the resultant data to our route servers.

— NLnetLabs Routinator
— Cloudflare's OctoRPKI
— FORT Validator, OpenBSD's rpki-client, rpki-prover, RPSTIR2

RPKI and IXP Manager

1. You need **two** RPKI validators
   — See https://docs.ixpmanager.org/features/rpki/
2. Add simple config to IXP Manager's `.env`:

```
# IP address and port of the first RPKI local cache:
IXP_RPKI_RTR1_HOST=192.168.140.211
IXP_RPKI_RTR1_PORT=3323
```

3. Enable RPKI for the router(s) in IXP Manager

RPKI and Origin ASNs

- RPKI provides a prefix and an origin AS
- It **does not** provide any information about whether a particular peer should be able to advertise such a prefix and origin ASN
- E.g. if a peer accidentally advertised a Netflix prefix with Netflix's ASN as the origin, it would pass RPKI's test!
- **You cannot have RPKI without the IRRDB origin AS check**

RPKI Filtering and IXP Manager

We had:

```
# IRRDB origin ASN check
...

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

# IRRDB prefix check
```

RPKI Filtering and IXP Manager

We now have:

```
# IRRDB origin ASN check
...

# RPKI test - if it's INVALID or VALID, we are done
if filter_rpki() then accept;

# IRRDB prefix check
```

RPKI Filtering and IXP Manager

```
function filter_rpki()
{
    if( roa_check( t_roa, net, bgp_path.last_nonaggregated ) = ROA_INVALID ) then {
        bgp_large_community.add( IXP_LC_FILTERED_RPKI_INVALID );
        return true;
    }

    if( roa_check( t_roa, net, bgp_path.last_nonaggregated ) = ROA_VALID ) then {
        bgp_large_community.add( IXP_LC_INFO_RPKI_VALID );
        return true;
    }

    bgp_large_community.add( IXP_LC_INFO_RPKI_UNKNOWN );
    return false;
}
```

Demonstration

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

Recap on "Securing Route Servers"

1. Small prefixes (default is > /24 for ipv4 and /48 for ipv6)
2. Martians / bogons
3. Ensure at least 1 ASN and <= 64 ASNs in path
4. Ensure peer AS is the same as first AS in the prefix’s AS path
5. Prevent next-hop hijacking
6. Filter known transit networks
7. Ensure origin AS is in set of ASNs from member AS-SET
8. RPKI:
   - Valid -> accept
   - Invalid -> drop
   - Unknown -> revert to standard IRRDB prefix filtering

Thanks for watching!

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