OpenBGPD

Adding Diversity to Route Server Landscape

Claudio Jeker claudio@openbsd.org

Route server mono culture

From Job Snijders presentation at EuroIX 32 (April 2018):

> Virtually all IXPs who deploy ‘secure route servers’, are running BIRD
>
> A monoculture of all IXPs deploying only one BGP software is dangerous
>
> What if some corrupt BGP message ripples through the Internet and destroys all route servers?
>
> What if the organisation funding the one BGP software can’t support development any more?

“Let’s invest together
in OpenBGPD”

OpenBGPD

Very mature code base

Has all the features a route server needs

Fundamentally different from BIRD

Integration with auxiliary tools: bgpq3 and arouteserver

Ships with a basic looking glass

Speed problems

At that time (April 2018) too slow for running secure route server configs

- arouteserver config expanded easily to over 350’000 filter rules
- All filter rules were evaluated in order for each prefix
- Config reload stopped processing of UPDATES for 45 min to 1 hour

Projects to tackle

Work full time on OpenBGPD

- Implement fast lookups for origin-as and prefixes
- Add RPKI based BGP Origin Validation support
- Process UPDATES during soft-reconfiguration
- Make it portable (mainly support Linux but also other BSDs)

Funding the work

Community Projects Fund

and

Where do we stand one year later?

OpenBGPD in OpenBSD 6.4

- RFC8212 compliance (default deny policy)
  Be careful when updating to 6.4.
- RPKI ROA support (static table, no RTR support)
- Sets for AS numbers, prefixes and origins (prefix + source-as)
  Replace large amount of filter rules with a single fast lookup
- Background soft-reconfiguration on config reload
  On reload new withdraws and updates are now processed
- 154 commits since 6.3 (close to 8% of all commits)

Full report here:
https://labs.ripe.net/Members/claudio_jeker/openbgpd-adding-diversity-to-route-server-landscape

RPKI ROA validation

RFC 6811 BGP Origin Validation through the `roa-set` directive and `ovs` filter.

```
roa-set {
        203.119.88.0/23 maxlen 24 source-as 187
        2001:dd8:7::/48 maxlen 48 source-as 187
        2401:680::/32 maxlen 32 source-as 715
}
deny quick from ebgp ovs invalid
```

`ovs` can be `valid`, `invalid` or `not-found`

Use `bgpctl show rib ovs valid` to show only valid routes.

On my system reloading the roa-set takes a couple of seconds.

As-sets

Quick lookup table for large amount of AS numbers

```
as-set asns_AS6939 {
        2 3 4 5 6 8 10 13
        16 17 22 24 25 26 27 31
        32 37 38 42 43 44 45 47
}

match from AS 6939 source-as as-set asns_AS6939 set {
     localpref +50 community local-as:101 }
```

Works for `AS`, `transit-as` and `source-as`

Prefix-sets

`prefix-set` got a lot better, can be used also with `network` statements

```
prefix-set p4_AS6939 {
        1.0.0.0/24 prefixlen 24 - 32
        1.0.4.0/22 prefixlen 22 - 32
        1.0.16.0/22 prefixlen 22 - 32
}

match from AS 6939 prefix-set p4_AS6939 set { localpref +25 community local-as:102 }
```

Other `prefixlen` options are `or-longer`, `maxlen X`, or nothing.
Prefix-sets, with only prefixes, can use or-longer in the filter

```
deny quick from ebgp prefix-set mynetworks or-longer
```

Origin-sets

Similar to `roa-set` can be used when only prefixes matter which are `valid`

```
origin-set "ARINDB" {
        108.160.208.0/20 prefixlen 20 - 32 source-as 10242
        162.219.228.0/22 prefixlen 22 - 32 source-as 10242
        162.222.52.0/22 prefixlen 22 - 32 source-as 10242
}
# community local-as:101 is set when source-as is in IRR as-set
match from group clients community local-as:101 origin-set ARINDB set {
     localpref +25 community local-as:102 }
```

Allows to use ARIN Whois or RPKI ROA as an alternative to IRR DB to validate prefixes based on origin-as.

https://medium.com/@jobsnijders/a-new-source-for-authoritative-routing-data-arin-whois-5ea6e1f774ed
https://mailman.nanog.org/pipermail/nanog/2018-July/096359.html

Results

YYCIX (Calgary Internet Exchange) using arouteserver to generate config

- 6.3: generated config consists of 370’000 filter rules
- 6.4: with as-set, prefix-set and origin-set ruleset is now below 6000 rules

It takes less than 2 minutes for initial convergence

A config reload takes less than 30 seconds to finish soft-reconfiguration

OpenBGPD is now an alternative for mid-sized IXP

Portable OpenBGPD

Initial porting infrastructure created by Brent Cook (bcook@)

- It finally compiles on Linux systems

- Doesn’t really run just yet
  - Fighting with TCP-MD5 right now

Check out openbgpd-portable on github

First release will be together with OpenBSD 6.5 in May

What is next?

Future plans

Make the portable version as solid as the OpenBSD one

Keep up with the requirements of IXP route servers

Make it faster by going multithreading

Add more features: ADD_PATH, BMP, RTR, Outbound Max Prefix limits, JSON Looking Glass API

... but current funding lasts only until end of May 2019

Founding for another year

2019 Funding

Fundraising goal is 90’000 Euro again - have to feed my Swiss family!

- This allows me to work full time on OpenBGPD, no other distractions

Ideally supported by many IXP

- Ideally 20 - 30 contributors cover the cost

Getting commitments help me plan

- Email claudio@openbsd.org in case we can’t speak in person

Questions?