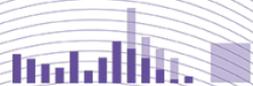


CSNOG 2022
2022-06-21

QRATOR
LABS

Automatic updating of prefix lists

Alexander Zubkov
green@qrator.net



- control prefixes announced by a peer
 - fool proof
 - strict compliance
- peer declares expected announces
 - explicit list of prefixes
 - ASN or as-set (IRR)

- This presentation
 - <https://gitlab.com/qratorlabs/example-automatic-filters>
 - <https://gitlab.com/qratorlabs/plag>
- Other useful links
 - <https://github.com/rfc1036/rpsltool/> (suite of tools)
 - <https://juni.pr/3zJshTZ> (Juniper documentation)

- where
 - BIRD routing daemon on Linux system
 - 10x nodes
 - configuration is generated with Ansible
- what
 - configuration parts
 - identify as-sets to fetch
 - fetch prefixes & generate prefix lists
 - reload bgp daemon

- ansible
 - install scripts to nodes
 - bgp & filters configuration
- plag-http
 - docker image
 - HTTP API for prefix lists
- updatefilter
 - scripts that do the job on nodes

```
clients:  
  name3:  
    asn: 64498  
    prefix:  
      - 2001:db8:1::/48  
      - AS64498  
      - AS64496:AS-TEST  
    ip6: 2001:db8::2  
    peer6: 2001:db8::3
```

```
define pfx6_client_name3 = [ 2001:db8:1::/48 ];
function match6_client_name3() {
    if net ~ pfx6_client_name3 then return true;
    if net = fdfd::fdfd/128 then return false;
    if net ~ 'pfx6_dyn_AS64498' then return true;
    if net ~ 'pfx6_dyn_AS64496:AS-TEST' then return true;
    return false;
}
filter from6_client_name3 {
    if ! match6_client_name3() then reject;
    if ! common6_client() then reject;
    accept;
}
protocol bgp bgp6_client_name3 from bgp6_client {
    local 2001:db8::2;
    neighbor 2001:db8::3 as 64498;
    ipv6 { import filter from6_client_name3; };
}
```

prefix:

- 233.252.0.0/24
- 2001:db8:1::/48
- AS-TEST
- 192.0.2.0/24
- AS64496

```

define pfx4_name = [
    233.252.0.0/24,
    192.0.2.0/24
];
function match4_name() {
    if net ~ pfx4_name then return true;
    if net = 127.127.127.127/32 then return false;
    if net ~ 'pfx4_dyn_AS-TEST' then return true;
    if net ~ 'pfx4_dyn_AS64496' then return true;
    return false;
}
...
if ! match4_name() then reject;

```

- filter.list

```
4 AS-CLIENT 'pfx4_dyn_AS-CLIENT'  
4 AS64496 'pfx4_dyn_AS64496'  
6 AS64496:AS-TEST 'pfx6_dyn_AS64496:AS-TEST'
```

- filter.conf

```
define 'pfx4_dyn_AS-CLIENT' = [  
    ...  
];
```

- ' pfx4_dyn_AS-CLIENT '
 - quoted name symbol in bird
 - pretty unique substring
- identify with regex
 - / ' pfx[46]_dyn_AS[-_ :0-9a-zA-Z]+ ' /
 - no need to process ansible config
- grep config
 - on node
 - in ansible template

```

{% macro gen_list() %}
{%   set list = caller().split('\n')
      | select('regex', "'pfx[46]_dyn_AS([-_:0-9a-zA-Z]+)'" )
      | map('regex_replace',
            "'.*' (pfx([46])_dyn_(AS([-_:0-9a-zA-Z]+)))' .*",
            "\\2\t\\3\t'\\1'")
    %}
{%   for v in list | sort | unique %}
{{ v }}
{%   endfor %}
{% endmacro %}
{% call gen_list() %}
{%   include 'bird-bgp.conf.j2' %}
{% endcall %}

```

- bgpq4
 - `bgpq4 -F '%n/%l\n' -4 AS-CLIENT`
 - default IRR: rr.ntt.net
- many nodes, need cache
- private server with HTTP API
 - nginx + lua + scripts
 - cache by nginx at port 80
 - at port 81 fetch prefixes with scripts over bgpq4
- wrapped with docker-compose
 - `http://127.0.0.1:8000/ipv4/plain/AS-QRATOR.txt`

```
proxy_cache_lock on;
proxy_cache_lock_age 50s;
proxy_cache_lock_timeout 50s;

server {
    listen 80 default_server;
    location ~ ^/ipv[46]/(plain|max)/(AS[-_:0-9a-zA-Z]+)\.txt$ {
        proxy_cache prefix;
        proxy_hide_header Cache-Control;
        proxy_pass http://127.0.0.1:81;
    }
}
```

```
server {
    listen 81;
    location ~ ^/ipv(?<ver>[46])/plain/(?<q>AS[-_:0-9a-zA-Z]+)\.txt$ {
        content_by_lua_block {
            local out = ... "/app/bin/wrap_bgpq" ...
            ngx.header["Cache-Control"] = "max-age=600"
            ngx.print(out)
        }
    }
}
```

- strict prefix list
 - compact by folding groups: 192.0.2.0/24{26,26}
- loose prefix list
 - drop subprefixes & fold: 192.0.2.0/24+
- plag tool
 - <https://gitlab.com/qratorlabs/plag>
 - fast & low memory usage
 - strict & loose variants
- available in HTTP API
 - <http://127.0.0.1:8000/ipv4/max/AS-QRATOR.txt>

- filter.list → filter.conf
 - for each entry fetch list & print “define” statement
 - empty list workaround
 - 127.127.127.127/32
 - fdfd::fdfd/128
- error checking
 - http status
- temporary file, then move

- when
 - cron
 - trigger update on filter.list changes
- “configure soft”
 - protocol states are updated
 - <https://bird.network.cz/pipermail/bird-users/2022-January/015896.html>
 - <https://static.qrator.net/bird/bird-keep-state.patch>
- if you need to refilter
 - “reload in”
- be careful when editing configuration

- Alexander Zubkov
 - green@qrator.net