Using BGP Flow-Spec for distributed micro-segmentation

Davide Pucci / 12019364
Attila de Groot / Cumulus Networks
Data Center micro-segmentation

Layer 2 segmentation

VLANs to isolate multiple flows over the same link.

Layer 3 segmentation

VRFs to separate routing tables.

Micro-segmentation

Apply custom security filtering within the same VLAN.
Border Gateway Protocol (BGP)

BGP is the *de-facto* Internet routing protocol.

Pulls intra-Autonomous System prefixes, relying on iBGP.

Exchanges these internal prefixes with neighbouring Autonomous Systems to enable proper routing, relying on eBGP.
BGP Flow Specification

Extension of BGP, born with the only aim of DDoS attacks mitigation.

The Flow-Spec controller spreads filtering policies to its neighbours, the clients.

Regulate actions against given prefixes with extended communities, relying on BGP for the diffusion.
BGP in Data Centers

Third-wave applications moved most of the traffic to an east-west direction.

This change introduced the need for more elastic Data Centers.

All the switches represent a (private) Autonomous System.
Is the **BGP Flow Specification** applicable for Data Center **micro-segmentation**?
Distributed micro-segmentation with Flow-Spec

route flow4 {
    src 2.0.0.1/32;
    dst 1.0.0.1/32;
} {
    bgp_ext_community.add(
        (generic, 0x80060000, 0x0)
    );
};

Flow Specification controller

Flow Specification clients
Open source implementations

**Bird** for controller capabilities  
**FRR** for client capabilities

as none of them implements routes injection over the underlying system

**Custom utility** for rules injection
Open source implementations

**Bird**

Starting from version 2.0, it correctly implements the whole Flow-Spec specification.

**FRR**

Used to be unable to relay Flow-Spec announcements, later patched by working together with Cumulus Networks developers.
fs-controller:~# iptables -L FORWARD

Chain FORWARD (policy DROP)

num target prot opt source   destination
1   ACCEPT all  --  2.0.0.1   1.0.0.1
Rules fetcher ~ Flow-Spec routes on Bird

# default policy
route flow4 {
    src 0.0.0.0/0;
    dst 0.0.0.0/0;
} {
    # traffic drop
    bgp_ext_community.add(
        (generic, 0x80060000, 0x0)
    );
    
};

# rule 1
route flow4 {
    src 2.0.0.1/32;
    dst 1.0.0.1/32;
} {
    # traffic-mark as rule number
    bgp_ext_community.add(
        (generic, 0x80090000, 0x1)
    );
    
};
Rules transit

BIRD controller

FRR clients

BGP UPDATE
Rules injector ~ Flow-Spec routes on FRR

fs-client# show bgp ipv4 flowspec detail json

```
{
  "to": "1.0.0.1/32",
  "from": "2.0.0.1/32"
},
{
  "ecomlist": "FS:marking 1"
},
{
  "time": "00:00:09"
}
```

```
{
  "to": "0.0.0.0/0",
  "from": "0.0.0.0/0"
},
{
  "ecomlist": "FS:rate 0.000000"
},
{
  "time": "00:00:09"
}
```
Rules injector ~ iptables on the controller

```
fs-client:~# iptables -L FORWARD
Chain FORWARD (policy ACCEPT)
num target    prot opt    source     destination
1  FLOWSPEC  all  --  anywhere  anywhere
 [...]

fs-client:~# iptables -L FLOWSPEC
Chain FLOWSPEC (1 references)
num target    prot opt    source     destination
1  ACCEPT    all  --  2.0.0.1  1.0.0.1
2  DROP      all  --  anywhere  anywhere
```
Flow Specification is suitable for such a purpose and

A. **Rules numbering** must be carried along with routes, preferably with own extended community sub-type

B. A proper implementation of **routes injection** in the underlying system is still missing

C. Rules application can be filtered at a BGP level, using the **Route Target** extended community to achieve higher scalability
Thank you.

Davide Pucci
https://davidepucci.it

Cumulus Networks
https://cumulusnetworks.com

Security and Network Engineering
https://os3.nl

University of Amsterdam
https://uva.nl