About Packet Filtering

Aleksi Suhonen
3rd Euro-IX Virtual Forum
December 2021

What's This All about?

- Packet filters are an important tool to handle DDoS and abuse
  - Sometimes a misconfiguration is indistinguishable from abuse
- I've learnt a lot from running services that get DDoS
- Not just L3, but other layers too
- Few vendors have good tools for what I want to do
- I alone don't have enough purchasing power to encourage other vendors to create better tools, but maybe IXP wish list can

Juniper Love Affair

- When I had to transition from Cisco to Juniper, I noticed that Juniper packet filters are very expressive and have very powerful tools, like named counters and rate-limiters
- You can build interface filters from multiple smaller segments
- Now I feel like I can no longer live without these features
- Other vendors have things like policy-maps, but they feel awkward and inefficient

IRC Example

- IRC used to get a lot of DDoS back then
- Packet types that weren’t used by the IRC server were easy to discard
- But handling packets to production ports was harder
- Using a stateful firewall was right out of the question

```
term irc-clients
    from
        protocol tcp
        port 6660-6670
    then accept
term dns
    from
        port 53
    then accept
term finally
    then discard
```

Simple Rate Limiter

- Limit traffic to levels that the server can handle
  - But this can make it easier for the attackers to achieve their goals
    - Making the network split
- Differentiate between server links and client connections

```
term irc-servers
    from
        source-prefix-list irc-servers
    then
        policer 10Mbps
        accept

term irc-clients
    from
        protocol tcp
        port 6660-6670
    then
        policer 1Mbps
        accept
```

More Elaborate Rate Limiter

- TCP connections are divided into stages
- The connection setup stage is often attacked with a SYN flood
- A separate policer for SYNs will protect existing connections from this type of attack

```
term irc-syns
    from
        protocol tcp
        port 6660-6670
        tcp-initial
    then
        policer 100kbps
        accept
term ssh-syns
    from
        protocol tcp
        port 22
        tcp-initial
    then
        policer 100kbps
        accept
```

Off The Shelf Attack Tools

- Most attackers use off the shelf attack tools
- They often target just one or a few ports or mechanisms
- Having separate rate limiters for everything means that such attacks will just take out some functionality
  - e.g. new connections aren’t possible, but existing ones are OK

IXP L2 Example

- Same principles can be applied to Layer 2:
- Only allow IPv4, IPv6 and ARP traffic, discard the rest
- Rate limit ARP traffic as a fail safe against DoS and misconfigurations

```
term ipv4
    from protocol ipv4
    then accept
term arp
    from protocol arp
    then
        policer 10Mbps
        accept
term ipv6
    from protocol ipv6
    then accept
```

Refined L2 Example

- Same principles can be applied to Layer 2:
- Block specific IPv4 traffic
  - OSPF
  - VRRP
  - BGP
    - TTL Security
    - RFC 8327
- Rate limit IPv6 link local traffic similar to ARP
- Block blackhole MAC addr

```
term rfc8327-ipv4
    from
        ether-type ipv4
        protocol tcp
        destination-port bgp
        address 195.140.192.0/24
    then discard
term router-adv
    from
        ether-type ipv6
        protocol icmp6
        icmp6-type router-advertisement
        destination-address ff02::1
    then deny
```

Other Vendors

- Many other vendors use Cisco style configuration structure, where adding new filter and rate-limit features can be very challenging
- Cisco IOS specifically has accumulated a dozen different ACL formats and syntax over the decades
- A lot of new *NOSes* run on Linux, and a few of them even make use of existing Linux features instead of writing their own

Idea: nftables

- Linux seems to be switching from iptables to nftables
- Base idea is to combine iptables, ip6tables, ebtables and whatever else into a single framework, where duplication of code, work and effort is minimized
- It’s even possible to combine IPv4 and IPv6 filter rules
- Rule language is incredibly powerful, and I think it could be integrated into Cisco style configuration structure
- Implementing an nftables to merchant silicon compiler would leapfrog a vendor past Juniper in my eyes

Nftables Example

- This example handles both IPv4 and IPv6 traffic
- First rule is completely protocol agnostic, as it only matches on incoming interface

```
iifname "lo" counter accept
ip saddr 195.140.192.0/22 counter accept
ip6 saddr 2001:7f8:1d::/48 accept

udp dport 53 jump dport53
udp sport 53 counter accept
tcp dport 53 counter accept

udp sport 123 accept
```

Baby Steps

- Implementing every feature nftables already has into merchant silicon would take a lot of time
- Some features are probably seldom used
- Start with some basic core functionality
  - e.g. implement static prefix lists before dynamic address lists
- Work your way up according to
  - what is easy to implement
  - what there is customer demand for

Ingress vs Egress Filtering

- Some switch platforms only support filter rules before lookup
  - This doesn't matter much for general switch operations
- This can make it difficult to protect the control plane
  - You don't know whether the packet is going to the control plane before lookup
  - Workaround: protect control plane in every ingress filter

simplified ASIC workflow:

Ingress Filter

Lookup

Egress Filter

Call to Action

- Do we – as a community – want more powerful L2 filtering?
- When do we want it?!?
- Should some ideas from today's presentations be added to the **IXP Switch Wish List**?

Thank you!

Time for questions