Netfilter updates since last NetDev conference

Pablo Neira Ayuso
<pablo@netfilter.org>

Netdev Conference (Vancouver, Canada)
Oct 30th 2023

Topics

- nftables releases since last NetDev
- NFWS’23 summary

nftables releases

- 4 releases for userspace, since last NetDev conference in 2022:
  - 1.0.6: 21 Dec 2022 (69 commits)
  - 1.0.7: 13 Mar 2023 (61 commits)
  - 1.0.8: 14 Jul 2023 (91 commits)
  - 1.0.9: 19 Oct 2023 (233 commits)

nftables 1.0.6: --optimize fixes

- Squash common selectors into concatenation:

  meta iifname eth1 **ip saddr** 1.1.1.1 **ip daddr** 2.2.2.3 accept
  meta iifname eth1 **ip saddr** 1.1.1.2 **ip daddr** 2.2.3.0/24 accept
  meta iifname eth1 **ip saddr** 1.1.1.2 **ip daddr** 2.2.4.0-2.2.4.10 accept

```
# nft -o -c -f ruleset.nft
   Merging:
   …
   into:
           meta iifname . ip saddr . ip daddr {
                           eth1 . 1.1.1.1 . 2.2.2.3, \
                           eth1 . 1.1.1.2 . 2.2.3.0/24, \
                           eth1 . 1.1.1.2 . 2.2.4.0-2.2.4.10 } accept
```

nftables 1.0.6: --optimize fixes

- Squash common selectors into concatenation *with sets*:

```
meta iifname "enp0s31f6" ip saddr { 209.115.181.102, 216.197.228.230 } \
            ip daddr 10.0.0.149 udp sport 123 udp dport 32768-65535 accept
meta iifname "enp0s31f6" ip saddr { 64.59.144.17, 64.59.150.133 } \
            ip daddr 10.0.0.149 udp sport 53 udp dport 32768-65535 accept

# nft -o -c -f ruleset.nft
  Merging:
  ...
  into:
       meta iifname . ip saddr . ip daddr . udp sport . udp dport {
             enp0s31f6 . 209.115.181.102 . 10.0.0.149 . 123 . 32768-65535, \
             enp0s31f6 . 216.197.228.230 . 10.0.0.149 . 123 . 32768-65535, \
             enp0s31f6 . 64.59.144.17 . 10.0.0.149 . 53 . 32768-65535, \
             enp0s31f6 . 64.59.150.133 . 10.0.0.149 . 53 . 32768-65535 } accept
```

nftables 1.0.6: more updates

- Already possible in previous releases: Matching destination port of both UDP and TCP
  - meta l4proto { tcp, udp } @th,16,16 { 53, 80 }
- Uncommon match with raw expression (unknown layer 4 protocol):
  - meta l4proto 91 @th,400,16 0x0 accept
  - … this reads as @base,offset,length
    - Base: ll,nh,th, ih
    - Offset and length in **bits**
- Planned 1.0.6.x -stable release: Backport of 41 fixes
  - … maybe more to be included soon, this is WIP

nftables 1.0.7: inner header match

- Match on inner headers for VxLAN, GENEVE, GRE, GRETAP.
  - No need for round-trip to decapsulate packet then filter packet
  - Stateless filtering for inner headers
  - Requires Linux kernel >= 6.2
- Match on any layer 2/3/4 header:

```
... udp dport 4789 vxlan ether daddr aa:bb:cc:dd:ee:01
... udp dport 4789 vxlan ip protocol udp
... udp dport 4789 vxlan ip saddr 1.2.3.0/24
... udp dport 4789 tcp dport 22
```

- Combine it with concatenations:

```
... udp dport 4789 vxlan ip saddr . vxlan ip daddr { 1.2.3.4 . 4.3.2.1 }
```

- … sets/maps and so on.

nftables 1.0.7: set auto-merge improvements

- Auto-merge with interval deletions
  - nft list ruleset
    ```
    table ip x {
            set y {
                typeof tcp dport
                flags interval
                auto-merge
                elements = { 24-30, 40-50 }
            }
        }
    ```
  - nft delete element ip x y { 25 }
  - => Results in { 24, 26-30, 40-50 }

nftables 1.0.7: scalable NAT

- DNAT based on concatenations:
  - dnat to ip daddr . tcp dport map { \
    10.1.1.136 . 80 : 1.1.2.69 . 1024, \
    10.1.1.10-10.1.1.20 . 8888-8889 : 1.1.2.69 . 2048-2049 }
    persistent
- IP daddr and TCP dport determines IP daddr and TCP dport to be used for DNAT.
- Ranges can be used.
- ‘persistent' flag => tells core to hash the IPv4 source and IPv4 destination to evenly distribute the load to backend servers.

nftables 1.0.7: lastuse

- Update set from datapath to maintain last seen matching packet:

```
table ip x {
 set y {
        typeof ip daddr . tcp dport
        size 65535
        flags dynamic,timeout
        last
        timeout 1h
    }
    chain z {
        type filter hook output priority filter; policy accept;
        update @y { ip daddr . tcp dport }
    }
}
```

nftables 1.0.7: lastuse (2)

- Combine *last use* with dynamic sets:

```
# nft list set ip x y
  table ip x {
      set y {
         typeof ip daddr . tcp dport
         size 65535
         flags dynamic,timeout
         last
         timeout 1h
        elements = {
                    192.168.100.2 . 443 last used 1s591ms expires 59m58s409ms,
                    192.168.100.10 . 443 last used 4s636ms expires 59m55s364ms,
                    192.168.201.20 . 443 last used 4s748ms expires 59m55s252ms,
                    192.168.168.24 . 443 last used 4s436ms expires 59m55s564ms }
         }
  }
```

nftables 1.0.7: scalable quota

- Set with quota per element.

```
table netdev x {
    set y {
        typeof ip daddr
        size 65535
        quota over 10000 mbytes
    }

    chain y {
        type filter hook egress device "eth0" priority filter; policy accept;
        ip daddr @y drop
    }
}

# nft add element netdev x y  { 8.8.8.8 }
```

nftables 1.0.7: scalable quota (2)

- # nft list set netdev x y

```
table netdev x {
    set y {
        type ipv4_addr
        size 65535
        quota over 10000 mbytes
        elements = { 8.8.8.8 quota over 10000 mbytes used 196 bytes }
    }
```

- Override default quota is possible:
  # nft add element inet x y { 1.2.3.5 quota 5000 mbytes }

nftables 1.0.7: destroy command

- destroy command to delete object, requires Linux kernel >= 6.3.
  - Unlike ‘delete’, it never fails if object does not exist

    ```
    destroy table ip x
    destroy chain ip x y
    destroy set ip x y
    destroy map ip x y
    destroy counter ip x y
    ```

nftables 1.0.8: updates

- Stateful statements in anonymous maps, such as counters (No kernel update required)

```
... meta mark { 0xa counter, 0xb counter }
```

- also with verdict maps:

```
… ct state vmap { established counter : accept, \
                  related counter : accept,
                  invalid counter : drop }

... ip saddr vmap { 127.0.0.1 counter : drop, * counter : accept }
```

- Set packet ct and mark based on IP dscp:

```
... meta mark set ip dscp and 0x0f
... meta mark set ip dscp << 8
... meta mark set (ip dscp and 0xf) << 8
```

nftables 1.0.8: -o/--optimize

- Compact masquerade statements:

```
Merging:
masq.nft:3:3-36:          ip saddr 10.141.11.0/24 masquerade
masq.nft:4:3-36:          ip saddr 10.141.13.0/24 masquerade
into:
        ip saddr { 10.141.11.0/24, 10.141.13.0/24 } masquerade
```

- ... and redirect statements too:

```
Merging:
redir.nft:3:3-32:          tcp dport 83 redirect to :8083
redir.nft:4:3-32:          tcp dport 84 redirect to :8084
into:
        redirect to :tcp dport map { 83 : 8083, 84 : 8084 }
```

nftables 1.0.8: error reporting

- Location based error reporting with misspell support already available for many releases

  Error: No such file or directory; did you mean table ‘**filter**’ in family ip?
  add chain **filtro** input
  ```
                ^^^^
  ```

- Improve error reporting with suggestions on datatype mistypes:

  test.nft:3:11-14: Error: Could not parse Differentiated Services Code Point expression; did you you mean `**cs0**`?
  ```
  ip dscp ccs0
          ^^^^
  ```

nftables 1.0.8: set updates

- Support for *constant values* in concatenation

```
table inet x {
        set s1 {
                typeof ip saddr . ip daddr . tcp dport
                size 65535
                timeout 1m
                flags dynamic
        }
}
```

- Then, from ruleset:

... update @s1 { ip saddr . **10.180.0.4** . **80** }

nftables 1.0.9: speed up listing

- Speed up listing chain listing (when table contains large sets)

```
# time nft list chain inet raw input
table inet raw {
    chain input {
        type filter hook input priority filter; policy accept;
        ip6 saddr @bogons6 counter drop
    }
}
```

```
before:                                   after:
real    0m2,913s                          real    0m0,056s
user    0m1,345s                          user    0m0,018s
sys     0m1,568s                          sys     0m0,039s
```

nftables 1.0.9: NAT with numgen

- Allow to combine dnat with numgen

```
... dnat to numgen inc mod 8 offset 0xc0a864c8
```

offset 0xc0a864c8 => 192.168.100.200 to fan out packets using stateful DNAT from 192.168.100.200 to 192.168.100.207.

Nftables 1.0.9: set updates

- Allow for using constants as key in dynamic sets.

```
table inet x {
    map dynmark {
       typeof ip saddr : meta mark
       flags timeout
    }

    chain y {
       type filter hook input priority 0; policy drop;
       udp dport 1024 add @dynmark { 10.2.3.4 timeout 3s : 0x00000002 }
    }
}
```

nftables 1.0.9: fixes

- Memleak with wildcard interface, eg. abcde*

```
… meta iifname { abcde*, xyz }
```

- Restore interval maps

```
table inet filter {
        counter TEST { }

        map testmap {
                type ipv4_addr : counter
                flags interval
                elements = { 192.168.0.0/24 : "TEST" }
        }
}
```

nftables 1.0.9: fixes

- Restore bitwise operations with verdict maps:

```
table ip x {
        map sctm_o0 {
            type mark : verdict
            elements = { 0x00000000 : jump sctm_o0_0, \
                         0x00000001 : jump sctm_o0_1 }
        }
        chain sctm_o0_0 {
             counter
        }
        chain sctm_o0_1 {
             counter
        }
        chain SET_ctmark_RPLYroute {
                meta mark >> 8 & 0xf vmap @sctm_o0
        }
}
```

Summary of NFWS’23

NFWS’23 summary

- 2 days meetings in Dresden
  - Discussion on existing bug reports (~1 day)
  - 1 day with assorted topics:
    - br_netfilter
    - Improve test infrastructure
    - Flowtables updates to speed up forwarding

NFWS’23 summary (2)

- Expand and improve test infrastructure at kernel API level (WIP)
  - Address recent bugs reported due to lack of sanitization and error unwinding path (syzbot)
- Existing tests mostly cover userspace interactions with the kernel, through nft.
- Add tests for kernel API.
  - Minimalistic
  - Which utilizes and resides in libnftnl
  - Provide common framework to report kernel API bugs

NFWS’23 summary (3)

```
                    // test API, add unbound chain
#include "test.h"

int main(void)
{
        struct test_batch batch;

        setup_batch(&batch);
        add_table(&batch, NFPROTO_IPV4, "test");
        add_chain(&batch, "__test0", "0x1", “NFT_CHAIN_BINDING”);

        if (batch_commit(&batch) < 0)
                return EXIT_FAILURE;

        return EXIT_SUCCESS;
}
```

NFWS’23 summary (4)

- br_netfilter issues
  - Allows user to use iptables from bridge with -m physdev
    - very popular because ebtables lacks many features
    - broken by design in many aspects
- Native replacement:
  - stateful filtering: nftables bridge + nf_conntrack_bridge
    - Use ‘ct state’ expression in nftables bridge rules.
  - NAT: Use ‘ether daddr set’ to specify br0 MAC address to pass up packets to IP stack
    - NAT is done from ‘inet’ family