nft fib expression

netdev 1.2 netfilter workshop

Florian Westphal

Oct 2016

motivation

nftables lacks feature equivalent of xtables

1. rpfilter match (we have in-stack rpfilter for ipv4 but not for ipv6)
2. addrtype match (true if saddr/daddr is local/local only on this interface, uni/broad/multicast etc)

partial overlap, addrtype also sometimes needs to do route/fib lookup

$\rightarrow$ FIB expression to implement 1) and later extend it for 2)

FIB expression

```
add rule ... input fib daddr . mark . oif oif
```

- query fib to obtain one fib key (here: oif) which is placed in a register
- based on inputs represented as tuple, here:
  - ip destination address
  - skb mark (nfmark/fwmark)
  - ask for route via (fixed) oif
  - only saddr or daddr is needed

```
add rule ... fib daddr oif != eth0
add rule ... fib saddr oif eth0
add rule ... fib daddr . mark . oif eth0
```

FIB expression (2)

how do do equivalent of
`iptables .. -m rpfilter --invert ... -j DROP`?
(i.e. drop packets whose reply would be sent via different interface than the one the packet arrived on)?

- Not so easy as nftables expressions are more generic
- `nft add rule ... fib saddr . oif oif` asks to do a reverse lookup (get index of interface that a packet sent to the source address of the current packet)
- ... but there is no `--invert`
- would need something like:

```
nft add rule ... fib saddr . oif \
  oif != "the_input_interface" drop
```

FIB expression (3)

- right side of = must be a constant
- use 0 as "invalid" result?
- `nft add rule ... fib saddr . oif oif = 0 drop`
- would need to extend nft userspace to accept "type compatible" rhs
- should also work with future "oifname" key:

```
nft add rule ... fib saddr . oif oifname = "" drop
nft add rule ... fib saddr . oif oifname != "" accept
nft add rule ... fib saddr . oif oifname eth0 accept
```

- What if interface named "0" exists...?
- add explicit casting that forces a type?

```
nft add rule .. oifname = (cast integer) 0
```

- or just use "`eq 0`" vs. "`eq "0"`"?

Current status

- kernel part done (oif only at the moment, ipv6 wip)
- userspace parts done, using 0 register approach for "not found"