Nftables strikes back

pablo@netfilter.org

Netfilter Workshop 2013
Copenhague, Denmark

nftables: Intro

- New kernel packet filtering framework to replace iptables.
- No changes in the core infrastructure:
  - Netfilter hooks
  - Connection Tracking System
  - NAT
- Designed from lessons learnt from iptables.
- Provides backward compatibility infrastructure.
- Nftables released in March 2009 by Patrick McHardy.
- Currently under active development.

Nftables: Architecture

- Pseudo-state machine in kernel-space (similar to BPF).
- Registers: 4 general purpose (128 bits long each) + 1 verdict.
- Provides instruction set (can be extended):
  - reg = pkt.payload[offset, len]
  - reg = immediate(value, len)
  - reg = cmp(reg1, reg2, EQ)
  - reg = byteorder(reg1, NTOH)
  - reg = pkt.meta(mark)
  - reg = (reg1 & mask) ^ xor)
  - reg = lookup(set, reg1)
  - reg = ct(reg1, state)
- New extensions are implemented using this instruction set.
- Netlink interface: kernel ↔ userspace (http://1984.lsi.us.es/~pablo/docs/spae.pdf)

Nftables: kernel code

- net/netfilter/nf_table_api.c (netlink interface)
- net/ipv4/netfilter/nft_chain_route_ipv4.c
- net/netfilter/nf_table_core.c (packet matching loop)
- net/netfilter/nft_payload.c (extensions)
- net/netfilter/nft_compat.c

Nftables: Commit operation

- Generation mask: 2 bits per rule
  - 00 active now, active in the next generation
  - 01 inactive now, active in the next generation
  - 10 active now, inactive in the next generation (will be deleted)
- Global generation counter can be 0 or 1.
- Transaction begin: open socket and send rule with commit flag (own by process), then add to chain list and the dirty list.
- Transaction end: send commit command, bump generation counter, iterate over the list
- In the nft_do_chain path:
  - Store current generation counter before entering rule matching loop.
  - If rule is ainactive (unlikely) skip.

Nftables: User-space code

- Libnftables
  - src/table.c
  - src/chain.c
  - src/rule.c
  - src/expression/payload.c
- Iptables-nftables:
  - iptables/nft.c
  - iptables/nft-ipv4.c

Nftables: Features

- **Backward** compatible:
  - Utility derivated from iptables/ip6tables with same syntax.
  - You can use existing and add new xtables modules.
  - No need to learn new utilities if you don't want to. No need for new documentation. No need to update your scripts.
- **But also, new features without breaking backward compatibility:**
  - xtables-event : Reporting changes in tables/chains/rules
  - Better incremental rule update support: Matches internal state is not lost
  - Enable/disable the chains per table that you want
  - … more improvements for xtables yet to come?

Nftables: examples

- Show iptables-like utility in action.

Jesper's has down to earth rule-sets..

- Around 100000 rules.
- ... in a fan-out tree. 4 - 8 rules per chain, eg.

  192.168.0.0/24 -> chain1
  192.168.1.0/24 -> chain2
  192.168.2.0/24 -> chain3
  192.168.3.0/24 -> chain4
  …

- Worst case: With iptables ~40 rule comparison until final action.
- With nftables, Jesper can arrange his rule-set using fast lookup data structures.

Pending tasks

- Bridge and ARP support.
- Object-oriented high level library for xtables (over nftables) developers.
- Add native interface nf_tables to xt_hashlimit.c (100% netlink).
- Documentation.

nftables summary

- One single kernel framework for packet filtering allowing long term evolution.
- Two userspace tools:
  - Backward compatible utility:
    - Same syntax + same features + new features
  - New utility:
    - New syntax + more cool new features
- Still work in progress.

Nftables summary (2)

- Grab the code
  - Backward compatible utility:
    - Kernel: git://git.netfilter.org/nftables
    - Library: git://git.netfilter.org/libnftables (requires libmnl)
    - User-space: git://git.netfilter.org/iptables-nftables
  - New utility:
    - Library: git://git.netfilter.org/libnl-nft
    - User-space: git://git.netfilter.org/nftables

Nftables strikes back

pablo@netfilter.org

Netfilter Workshop 2013
Copenhague, Denmark