nf_tables 2-phase commit protocol speedup

NFWS 2017 Faro, Portugal

Pablo Neira Ayuso <pablo@netfilter.org>

Loading nf_tables ruleset

- Netlink interface w/2-phase commit protocol
  - Preparation phase
  - Commit phase (never fails)
- More detailed explanation to educate developers here...

Loading nf_tables ruleset (2)

- Updates via nft -f are atomic
  - You can perform incremental updates.
  - Takes ~60 milliseconds in my laptop.
- time nft -f ruleset.nft
  real 0m0.058s
  user 0m0.000s
  sys 0m0.008s

Loading nf_tables ruleset (3)

- Running tests/py/ is slow
  - Lots of individual rule addition/deletions in a row
- Problem is two synchronize_rcu() calls in the nf_tables_commit() path.
  - First call makes sure no packets in the previous generation: First bump generation counter, then synchronize_rcu().
  - Second call makes sure no packets walk over the ruleset data structure.

Loading nf_tables ruleset (4)

- Removing second synchronize_rcu()
  - Release transaction object via call_rcu()
    - Add struct rcu_head in struct nft_trans
    - Release object via call_rcu
  - Problem:
    - Anonymous sets:
      - Released when no more references from rules to set.
      - Destroy rhashtable trigger may sleep splat.
  - Solution:
    - Add function to destroy rhashtable from atomic context
    - We have guarantees no packets are walking on this structure anymore
      - First synchronize_rcu() guarantees this.

Loading nf_tables ruleset (5)

- Removing first synchronize_rcu()
  - This one is harder.
- Move generation mask away from struct nft_rule
  - Add struct nft_rule_head arrays, store generation mask here.
  - Keep two arrays, one for each generation
  - Packets walk either of the two arrays versions
- Problem:
  - Array structure adds more complexity:
    - Array shift in case of rule insertion.
    - Array needs to be expanded.
- Proof-of-concept patch shows no performance impact.