Achievement unlocked:

No central conntrack lock

Jesper Dangaard Brouer
(Eric Dumazet)
(Florian Westphal)

Red Hat inc.

Netfilter Workshop, July 2014

Background

- Already have:
  - Parallel conntrack lookups RCU based
    - (Approx) Since around 2007 / kernel 2.6.21
- Issue:
  - Insert and delete conntracks took central lock

History

- Eric Dumazet proposed first patch (May 2013)
  - Jesper tests patch
- Jesper D. Brouer takes over patch (Dec 2013)
  - Transform into 5 set patchset
- Patchset V3 accepted (March 2014)
- Available in kernel v.3.15
  - Minor fixes for conntrack-tools in v.3.16
    - By Florian and Pablo

Changes: struct netns_ct

- Struct netns_ct
  - adjusted elements cache-line placement

Changes: special lists

- The special lists:
  - dying, unconfirmed/template
  - detached from central lock
- Lists now per CPU, with per CPU spinlock
  - e.g. see helper functions
    - nf_ct_add_to_dying_list()
    - nf_ct_add_to_unconfirmed_list()

Changes: expectations code

- Netfilter expectations were protected
  - with the same lock as conntrack entries (nf_conntrack_lock)

- Split out expectations locking
  - Own "central" lock (nf_conntrack_expect_lock).
  - Involved fixing race conditions
    - for exp->master conntrack ptr

Achievement unlocked: No central nf conntrack lock

Changes: Remove: central nf_conntrack_lock

- Array of hashed spinlocks
  - to protect insert/delete of conntracks into hash
    - 1024 spinlocks, minimal cost (4KB memory)
    - lockdep support: 1024 becomes 8 (if CONFIG_LOCKDEP=y)
- Locking both directions: nf_conntrack_double_lock()
  - correct lock order by
    - simply locking smallest hash value first
- Hash resize tricky
  - Need to take all locks in the array
  - Uses seqcount_t to synchronize
    - hash table users with the resizing process

Achievement unlocked: No central nf conntrack lock

Performance improvement

- SYN-flood attack tested on a 24-core E5-2695v2(ES)
  - with 10Gbit/s ixgbe (with tool trafgen):

- Base kernel: 810.405 new conntrack/sec
- After patch: 2.233.876 new conntrack/sec

Achievement unlocked: No central nf conntrack lock

Benefit / use-case

- Conntrack can be used in DDoS scenarios
- Invalid connection can be dropped
  - like floods attack (SYN+ACK or ACK)
  - easily be deflected using:

```
# iptables -A INPUT -m state --state INVALID -j DROP
# sysctl -w net/netfilter/nf_conntrack_tcp_loose=0
```


What is left

- Still have central atomic counter for conntracks
  - cause CPU cache-line bounce for each connection

- Kept central locking for expectations
  - Should not be too important
    - default max 256 expectations allowed

Achievement unlocked: No central nf conntrack lock

The End

- Thanks to
  - Eric Dumazet
    - This is really the fruit of his work
  - Florian Westphal
    - For helping me solve race conditions
    - For fixing conntrack-tools fallout bugs

Achievement unlocked: No central nf conntrack lock

Extra

- Extra slides