ebtables sucks, how can we make it suck less?

(Open Discussion)

Jiri Benc, Daniel Borkmann, Jesper Dangaard Brouer
<{jbenc,dborkman,jbrouer}@redhat.com>

Netfilter Workshop, Copenhagen, March 7, 2013

Status Quo

A wise hacker once said:

"I do not consider it wise to create more, rather then fewer, users of ebtables.

It is one of the most poorly constructed subsystems in the entire networking.

Just my $0.02"

Some of ebtables' Code Smells

- Data structures are shared between user space and kernel, thus not extendable, no room for refactoring or optimizations on structures
- Structures are transferred between address spaces as a huge blob array with jump offsets
- Parsing this blob is very complicated and needs hacks to distinguish which kind of structure follows etc.
- For each packet, it performs string comparisons for network device name matching rules, which slows things horribly down
- It even implements it's own strcmp, going through single bytes in order to match wildcard device names as well
- ... likely there's even more to beef about ;-)

Who is actually using it?

- What some users in the wild are doing, simplified:
  - 2 Machines connected via 10Gbit/s with *lots* of guest VMs each
  - Each has one bridge device, vnet devices conected to bridge
  - Stop guest's traffic if its considered "inappropriate"
  - Traffic must not reach local guests or outside world in that case
  - Long ebt rule lists à la `-A PREROUTING -i vnetX -j chain-vnetX`
  - In chain `chain-vnetX`: checks relevant to the VM

Example

- Machines connected over 10Gbit/s Ethernet
- `netperf -t UDP_STREAM -- -m 1024`, average of 9-10 runs

	Mbits/sec	
noebt	4,093.02	0.00%
rules_0	3,951.75	-3.45%
proto_rules_50[1]	3,776.36	-7.74%
proto_rules_100	3,597.95	-12.10%
iface_rules_50[2]	3,453.15	-15.63%
iface_rules_100	2,880.33	-29.63%

[1] 50 rules in PREROUTING chain in nat table comparing protocol field in frames.

[2] 50 rules in PREROUTING chain in nat table comparing interface name.

Open Discussion

- **Big picture:** Integrate `ebtables` into `iptables`?
- **Smaller picture:**
  - Convert `ebt_table->lock` to RCU
  - Converting interface names to ifindices
    - Must be network namespace aware
    - Wildcard must be handled separately
    - Need to use shadow structures
  - Parsing the userland structures and design new, more efficient and extendable structures
    - Would require refactoring most of ebtables
  - Introduction of "template jumps" à la `-A PREROUTING --template chain-%i -j tempjump`
    - Template is used to construct the name of the chain
    - Jump to a user-defined chain if template matched