Suricata Extreme Performance Tuning

With Incredible Courage

Mark II

- Michal Purzynski (@MichalPurzynski)
  - Threat Management, Mozilla
    Intrusion detection
    Digital Forensics
    Incident response
- Peter Manev (@pevma)
  - Suricata Core Team
  - Lead QA and training instructor
  - Stamus Networks
  - Mobster evangelist

For the brother and sister mobsters

SEPTun is (hopefully at least) yearly series of articles for using extreme and/or experimental techniques to deploy Suricata on high speed networks.

This SuriCon we are going to talk about :

- xDP+eBPF NIC driver bypass
- ...and some surprises ...

eXpress Data Path

- Upsides
  - ✓ No specific HW requirements
  - ✓ Bare metal packet processing
  - ✓ Integrated fast path in the kernel stack
  - ✓ Not a kernel bypass
  - ✓ Programmable
- Downsides
  - ✓ Newborn baby(first mailing list - April 2017)
  - ✓ Introduced in kernel 4.12

NICs with native driver XDP support

- Broadcom
- Cavium/Qlogic
- Cavium
- Intel: ixgbe + i40e
- Mellanox
- Netronome
- Virtio-net

eBPF

- Upsides
  - ✓ Improvement over classical BPF
  - ✓ Allows for user hooks/programs
  - ✓ Extends and improves performance of Suricata
  - ✓ Opens more kernel space possibilities
    - Elephant flow bypass
- Downsides
  - ✓ Write your own eBPF “hooks”
  - ✓ Requires skilled programing expertise

XDP & eBPF & kernel

- XDP uses eBPF scripts -
  - “New era in user programmable networking”- **Jesper Brouer**
- XDP eBPF program returns a simple verdict:
  - XDP_DROP
  - XDP_PASS
  - XDP_TX
  - XDP_ABORT
  - XDP_REDIRECT (likely huge boost for IPS)

XDP & eBPF & kernel

- ...amd64/include/uapi/linux/if_link.h:892:#define XDP_FLAGS_SKB_MODE
- ...amd64/include/uapi/linux/if_link.h:893:#define XDP_FLAGS_DRV_MODE
- ...amd64/include/uapi/linux/if_link.h:894:#define XDP_FLAGS_HW_MODE

XDP & Suricata

- Code work available in a PR by Regit (Eric Leblond)
- Packet arrives to FIFO. Card writes it back to queue. No SKB yet.
- eBPF program is run per packet
  - (eBPF-JIT)Transformed into CPU native assembly instructions during the eBPF kernel JIT loading stage
  - x86_64, arm64, ppc64, mips64, sparc64, s390x

```
- interface: eth3
  threads: 8
  cluster-id: 97
  cluster-type: cluster_qm # symmetric hashing  is a must!
  defrag: yes
  # eBPF file containing a 'loadbalancer' function that will be inserted into the
  # kernel and used as load balancing function
  #ebpf-lb-file:  /etc/suricata/lb.bpf
  # eBPF file containing a 'filter' function that will be inserted into the
  # kernel and used as packet filter function
  # eBPF file containing a 'xdp' function that will be inserted into the
  # kernel and used as XDP packet filter function
  #ebpf-filter-file:  /etc/suricata/filter.bpf
  # Xdp mode, "soft" for skb based version, "driver" for network card based
  # and "hw" for card supporting eBPF.
  xdp-mode: driver
  xdp-filter-file:  /etc/suricata/xdp_filter.bpf
  # if the ebpf filter implements a bypass function, you can set 'bypass' to
  # yes and benefit from these feature
  bypass: yes
  use-mmap: yes
  ring-size: 200000
```

—

…
[19012] 31/10/2017 -- 18:50:41 - (runmode-af-packet.c:220) <Config> (ParseAFPConfig) -- Enabling locked memory for mmap on iface eth3
[19012] 31/10/2017 -- 18:50:41 - (runmode-af-packet.c:231) <Config> (ParseAFPConfig) -- Enabling tpacket v3 capture on iface eth3
[19012] 31/10/2017 -- 18:50:41 - (runmode-af-packet.c:326) <Config> (ParseAFPConfig) -- Using queue based cluster mode for AF_PACKET (iface eth3)
[19012] 31/10/2017 -- 18:50:41 - (runmode-af-packet.c:424) <Info> (ParseAFPConfig) -- af-packet will use '/etc/suricata/xdp_filter.bpf' as XDP filter file…

Tested with:

- Kernel 4.13.10/ 4.14RC
- Intel NIC x510/520/710
- ixgbe , i40e

Use in tree kernel modules for the NIC!

—

- Using XDP on driver level can achieve huge IDPS offloading
- Principle - do not send traffic to Suricata that you don’t want to be inspected
- Low level - do not even make kernel work on it
- Allocating SKB takes time - only to trash it later?

–

- Current implementation of the XDP eBPF filter doesn’t cover UDP – only TCP
- Needs more NICs driver/HW level implementation
- Current XDP on HW level is done only by Mellanox
- Needs more IDPS testing

–

There is one more

thing…..

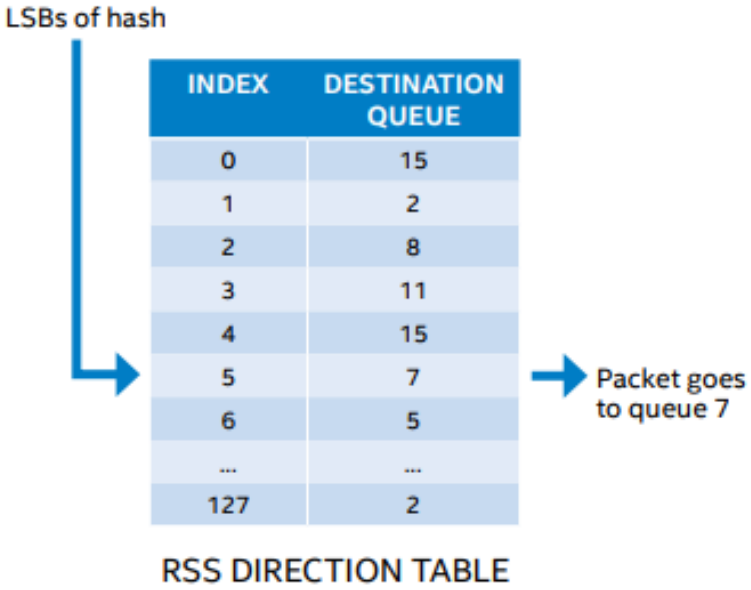
RSS strikes back

- *“A future work might show how to still use RSS and multiqueue, with careful process pinning, after changing the hash. We believe and briefly tested it with Bro IDS - symmetric hash (can be set with ethtool), processes pinned gave us no visible signs of packet reordering”*

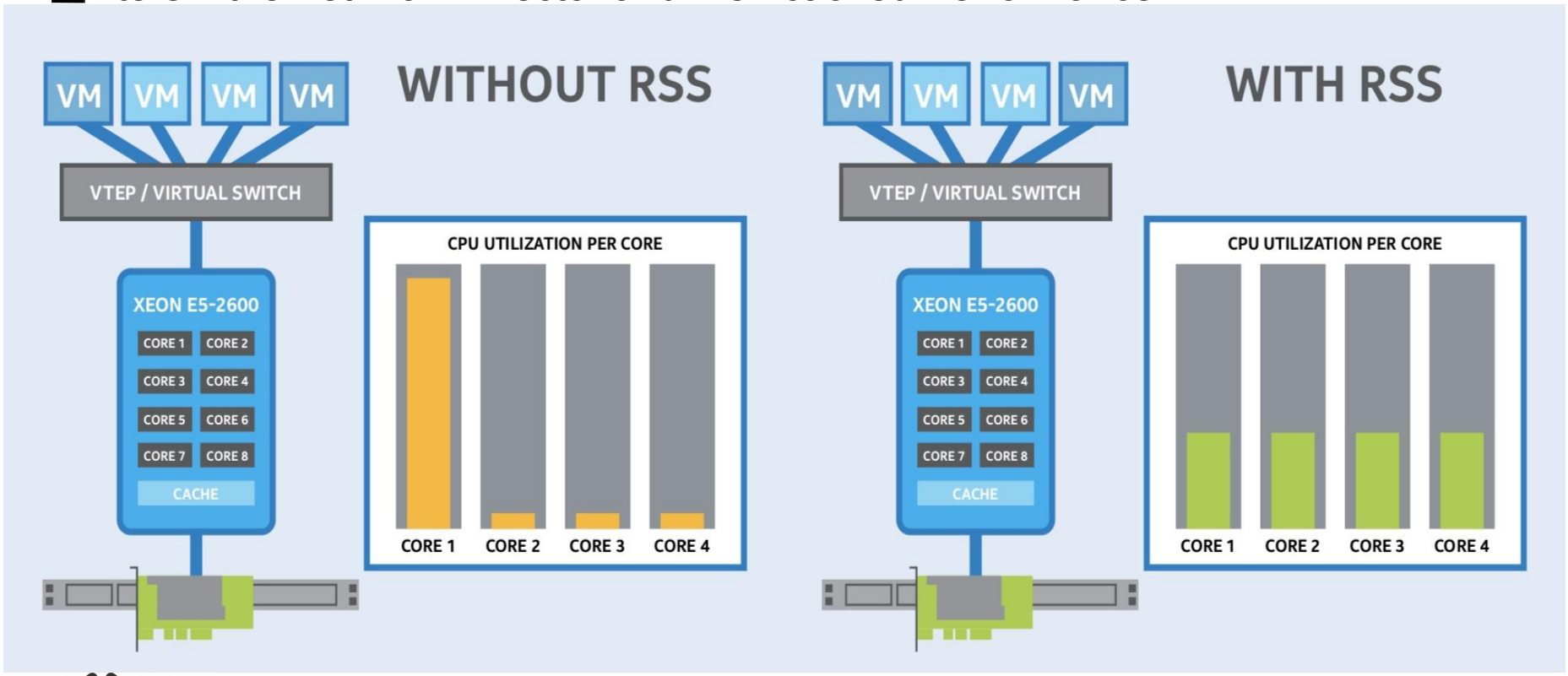
SepTun Mark I, 2016

What is RSS

- Technology defined by Microsoft
  - Load balancing network data over multiple cores/cpus



_Intel® Ethernet Flow Director and Memcached Performance



SURICON PRAGUE 2017

And DA problem is

The hash of:
Ipsrc=1.1.1.1,ipdst=2.2.2.2,sport=11111,dport=22222

And DA problem is

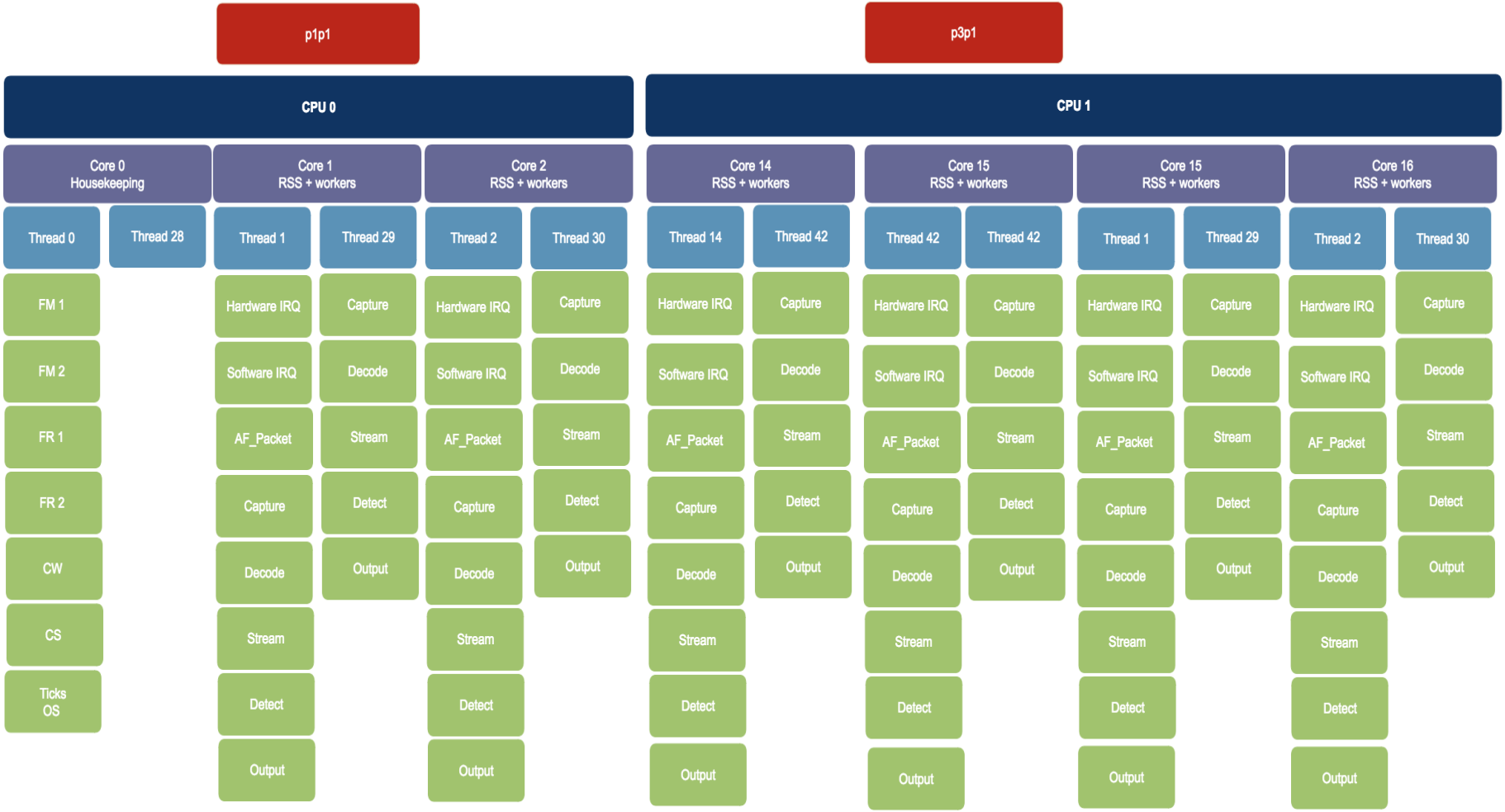
The hash of:
ipsrc=1.1.1.1,ipdst=2.2.2.2,sport=11111,dport=22222

Is **NOT** the same as the hash of:
ipsrc=2.2.2.2,ipdst=1.1.1.1,sport=22222,dport=11111

….so not going to the same queue/CPU!

```
root@suricata:~# ethtool -x eth3
RX flow hash indirection table for eth3 with 16 RX ring(s):
    0:      0     1     2     3     4     5     6     7
    8:      8     9    10    11    12    13    14    15
   16:      0     1     2     3     4     5     6     7
   24:      8     9    10    11    12    13    14    15
   32:      0     1     2     3     4     5     6     7
   40:      8     9    10    11    12    13    14    15
   48:      0     1     2     3     4     5     6     7
   56:      8     9    10    11    12    13    14    15
   64:      0     1     2     3     4     5     6     7
   72:      8     9    10    11    12    13    14    15
   80:      0     1     2     3     4     5     6     7
   88:      8     9    10    11    12    13    14    15
   96:      0     1     2     3     4     5     6     7
  104:      8     9    10    11    12    13    14    15
  112:      0     1     2     3     4     5     6     7
  120:      8     9    10    11    12    13    14    15
RSS hash key:
6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a:6d:5a
```

Suricata with RSS AF_Packet on NUMA



You can handle the packets…..

Thanks to

- Mozilla (time, traffic, hardware)
- Eric Leblond (@Regit – XDP port to Suri)
- Kernel dev team!
- SuriCon 2017 !!

SURICON PRAGUE 2017

External links

- http://people.netfilter.org/hawk/presentations/driving-IT2017/driving-IT-2017_XDP_eBPF_technology_Jesper_Brouer.pdf
- https://prototype-kernel.readthedocs.io/en/latest/networking/XDP/index.html
- http://www.ndsl.kaist.edu/~kyoungsoo/papers/TR-symRSS.pdf
- http://www.ran-lifshitz.com/2014/08/28/symmetric-rss-receive-side-scaling/
- https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/intel-ethernet-flow-director.pdf
- https://github.com/regit/suricata/tree/ebpf-3.18