eBPF and XDP in Suricata reloaded

É. Leblond/P. Manev

Stamus Networks

Nov. 1, 2019

SURICON

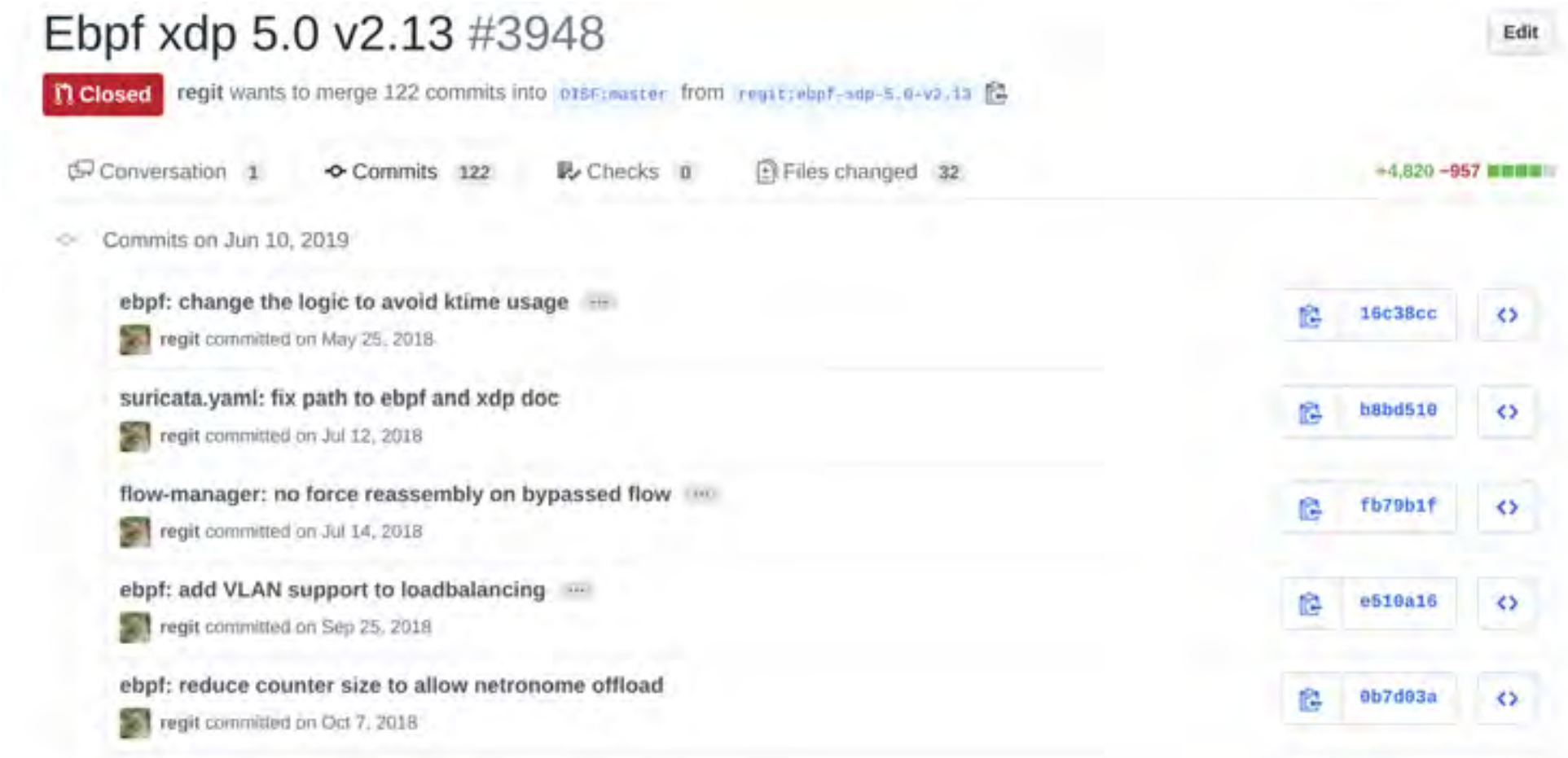
What, why them again? it's not Vancouver here

What, why them again? it's not Vancouver here

Vancouver was just the beginning of the trip

What, why them again? it's not Vancouver here

Vancouver was just the beginning of the trip



SURICON

SURICON

SURICON

Impact of loosing packets

Methodology

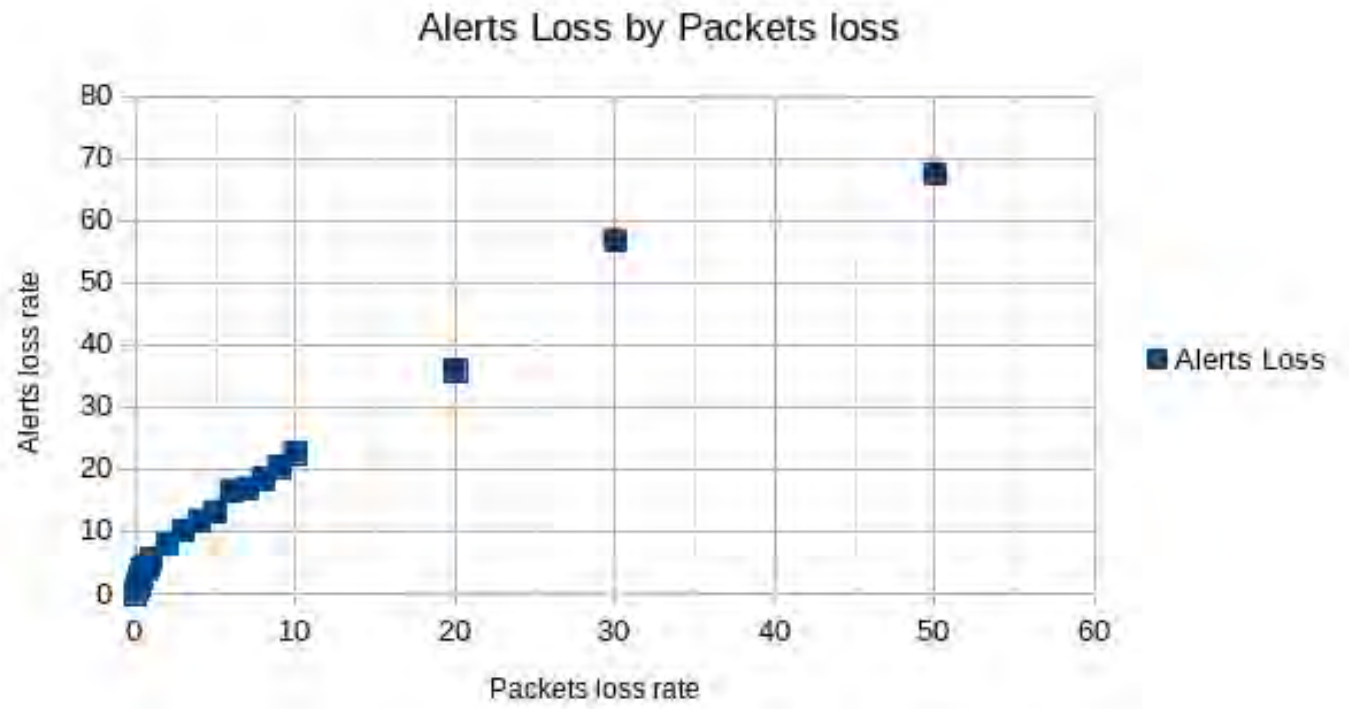
- Use a sample traffic
- Modify the pcap file to have specified random packet loss
- Do it 3 times par packet loss
- Get graph out of that

Test data

- Using a test pcap of 445Mo.
- Real traffic but lot of malicious behaviors
- Traffic is a bit old

SURICON

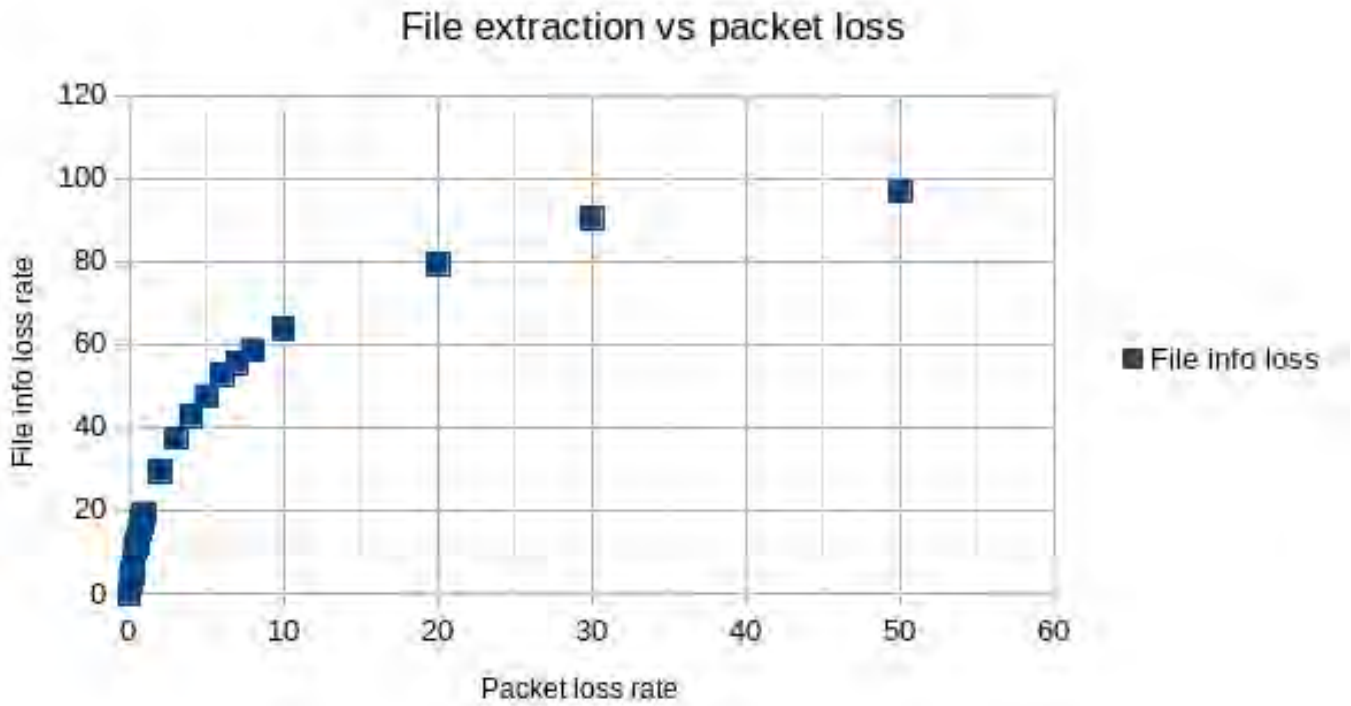
Alert loss by packet loss



Some numbers

- 10% missed alerts with 3% packets loss
- 50% missed alerts with 25% packets loss

The case of file extraction

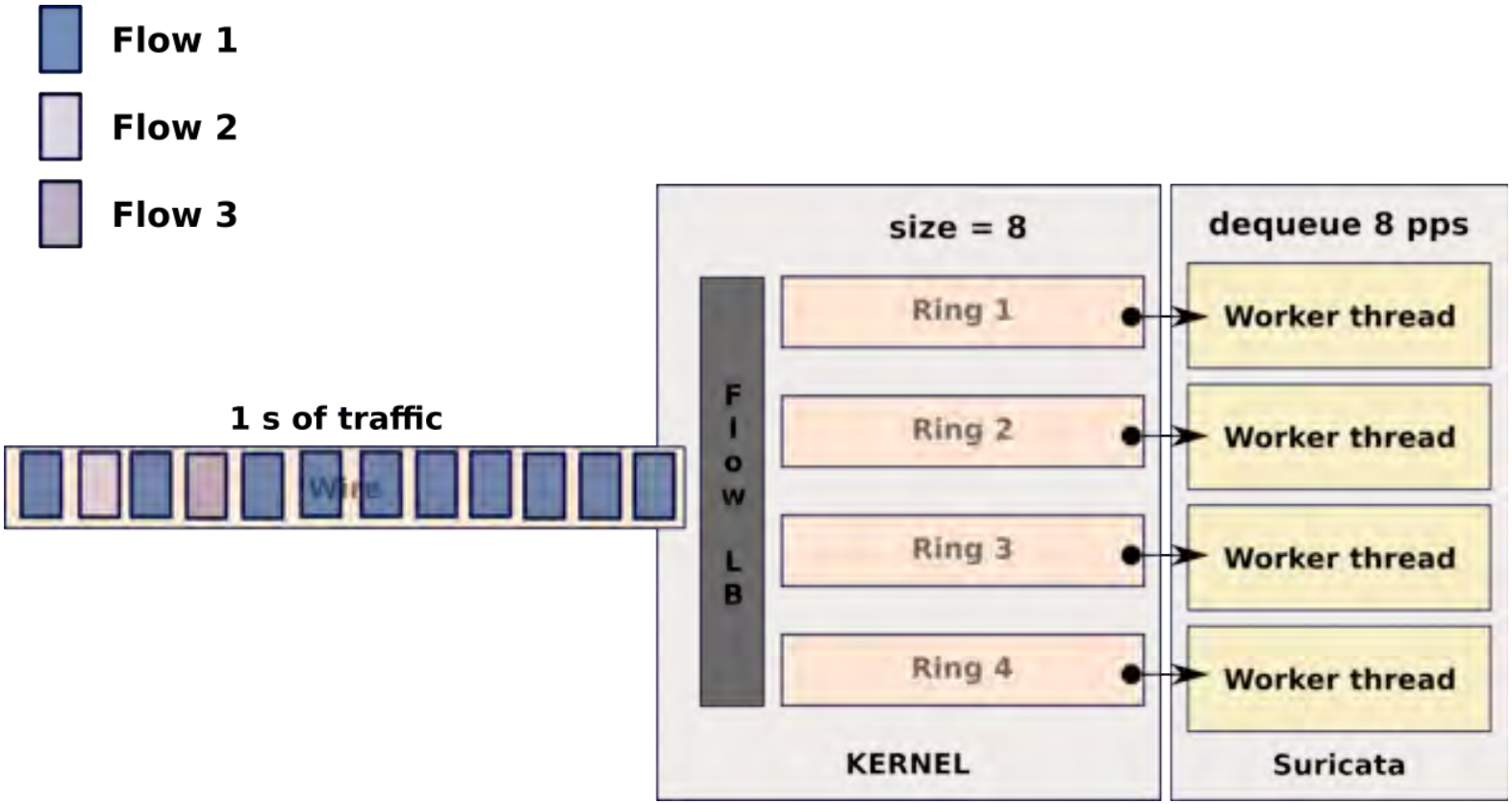


Some numbers

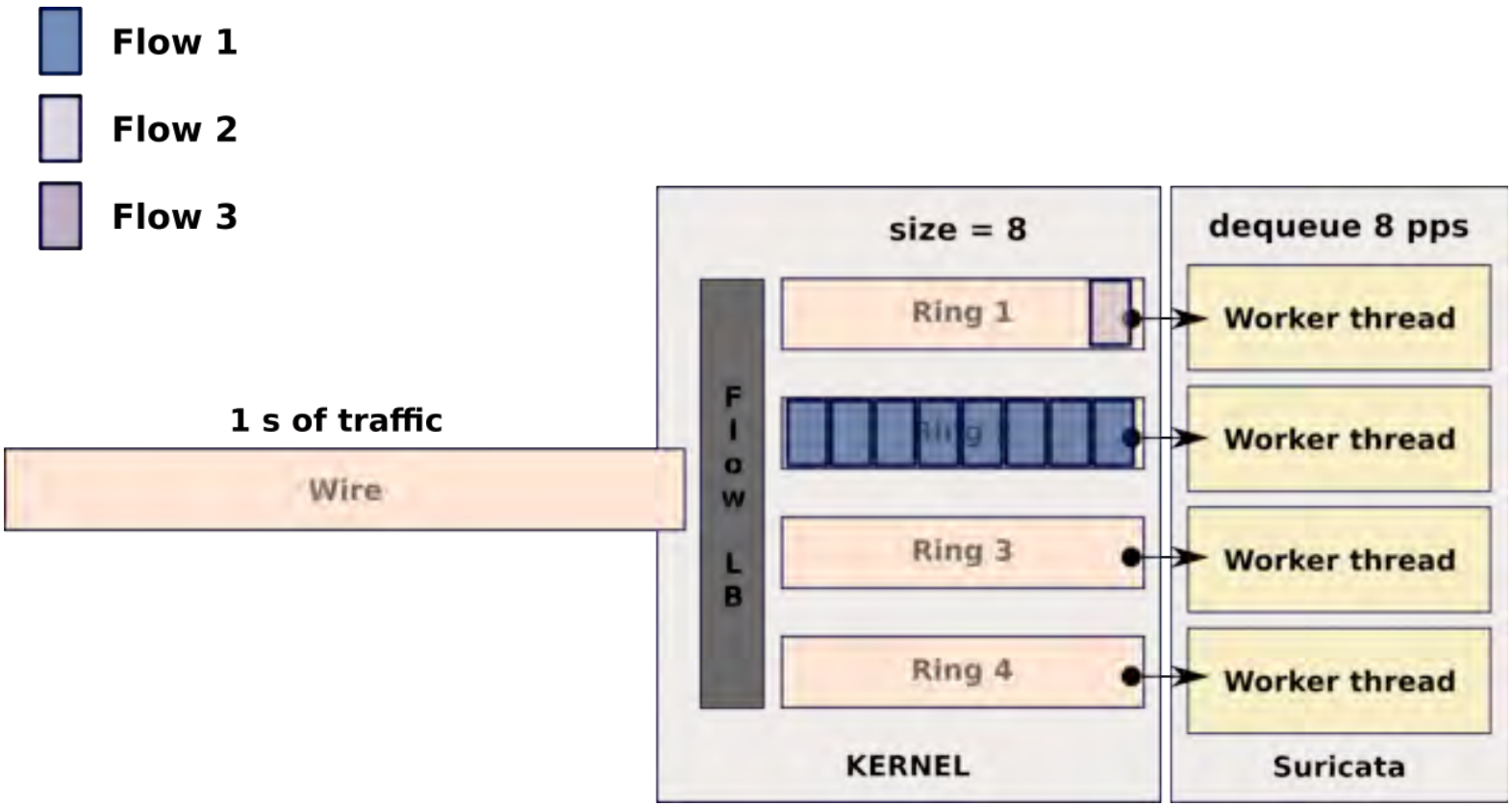
- 10% failed file extraction with 0.4% packets loss
- 50% failed file extraction with 5.5% packets loss

SURICON

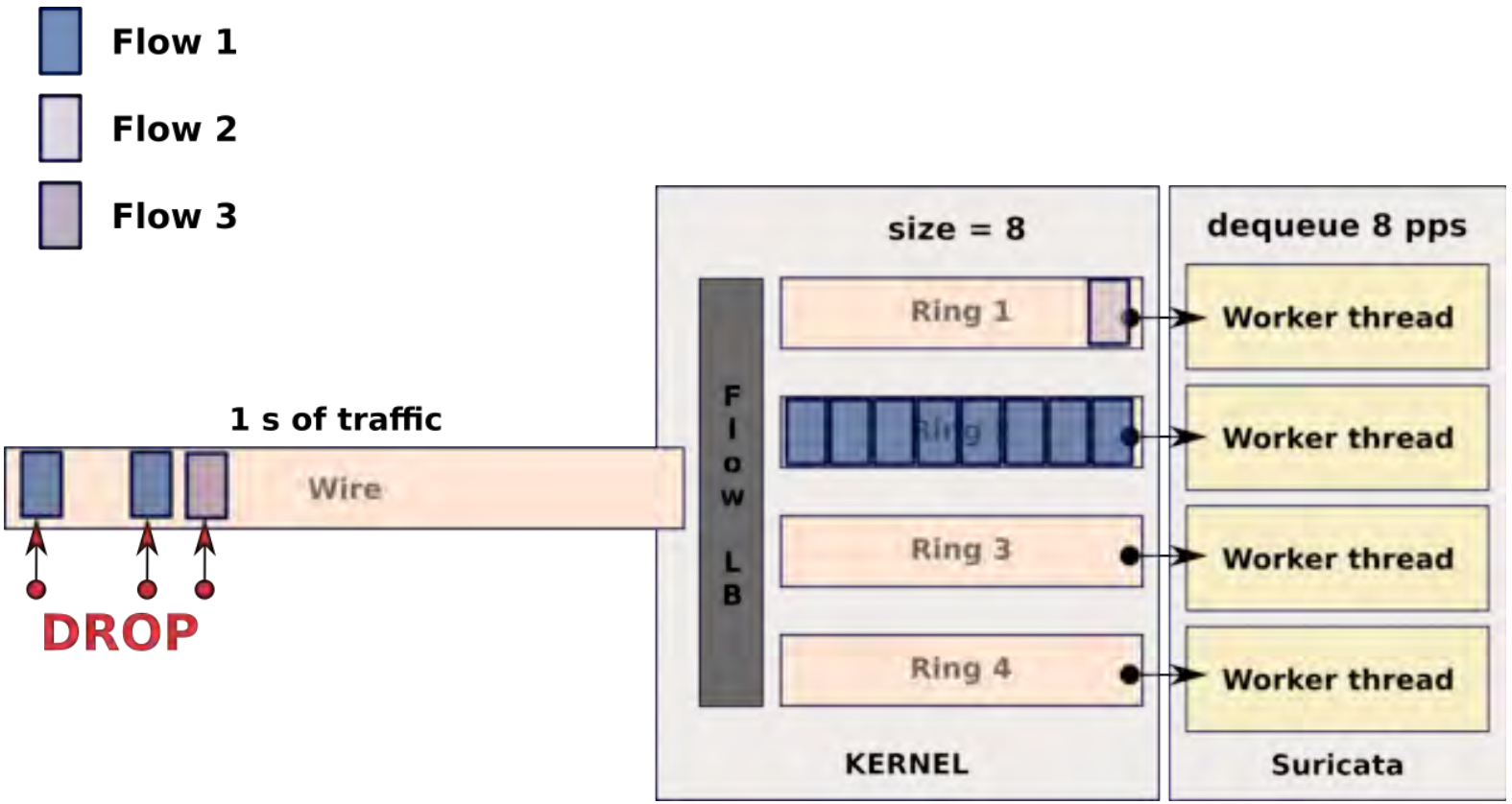
The elephant flow problem (1/2)



The elephant flow problem (1/2)



The elephant flow problem (1/2)



SURICON

The elephant flow problem (2/2)

Ring buffer overrun

- Limited sized ring buffer
- Overrun cause packets loss
- that cause streaming malfunction

Ring size increase

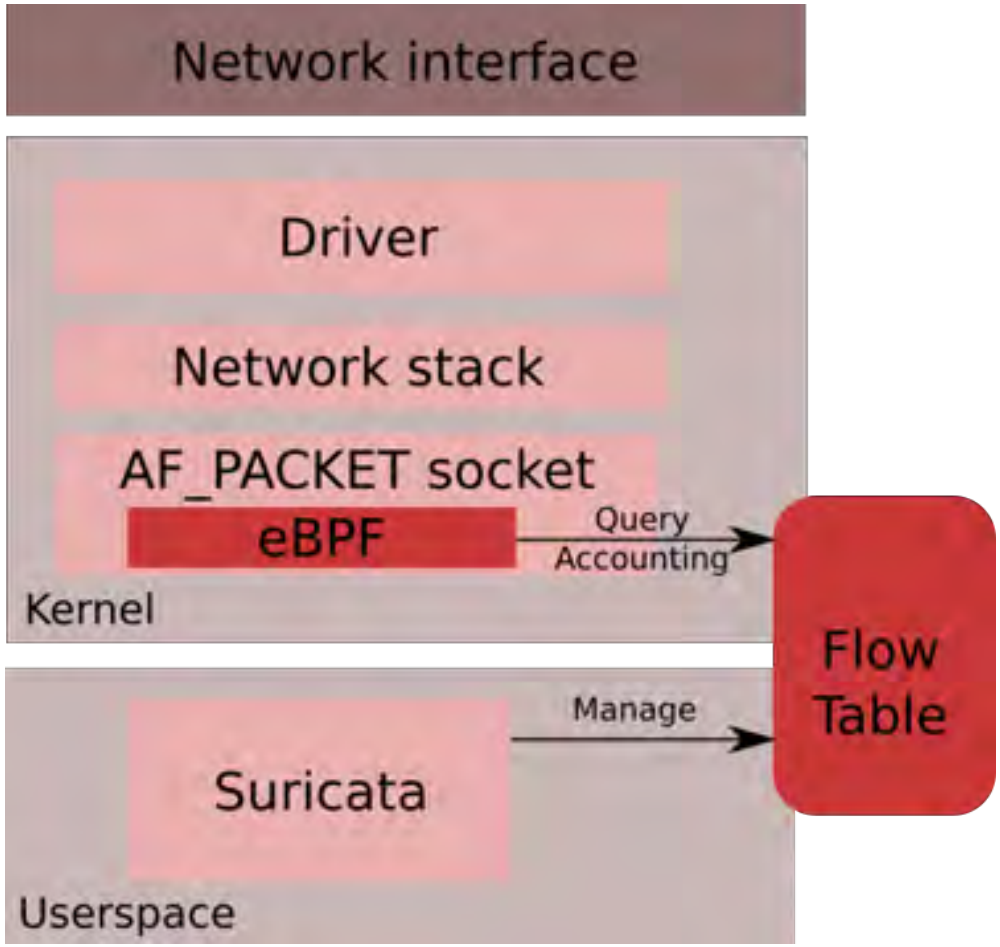
- Work around
- Use memory
- Fail for non burst
  - Dequeue at N
  - Queue at speed N+M

SURICON

SURICON

SURICON

Suricata eBPF bypass architecture



Bypass

eBPF bypass

- Suricata specialized filter
- Flow tables for IPv4 and IPv6
- Bypass function add entry to flow table

Flow handling

- At timeout of flow, fetch the corresponding entries in the table
- Compare counters, remove entries if no update or update counters

SURICON

SURICON

eXtreme Data Path

Reaching bare metal performance

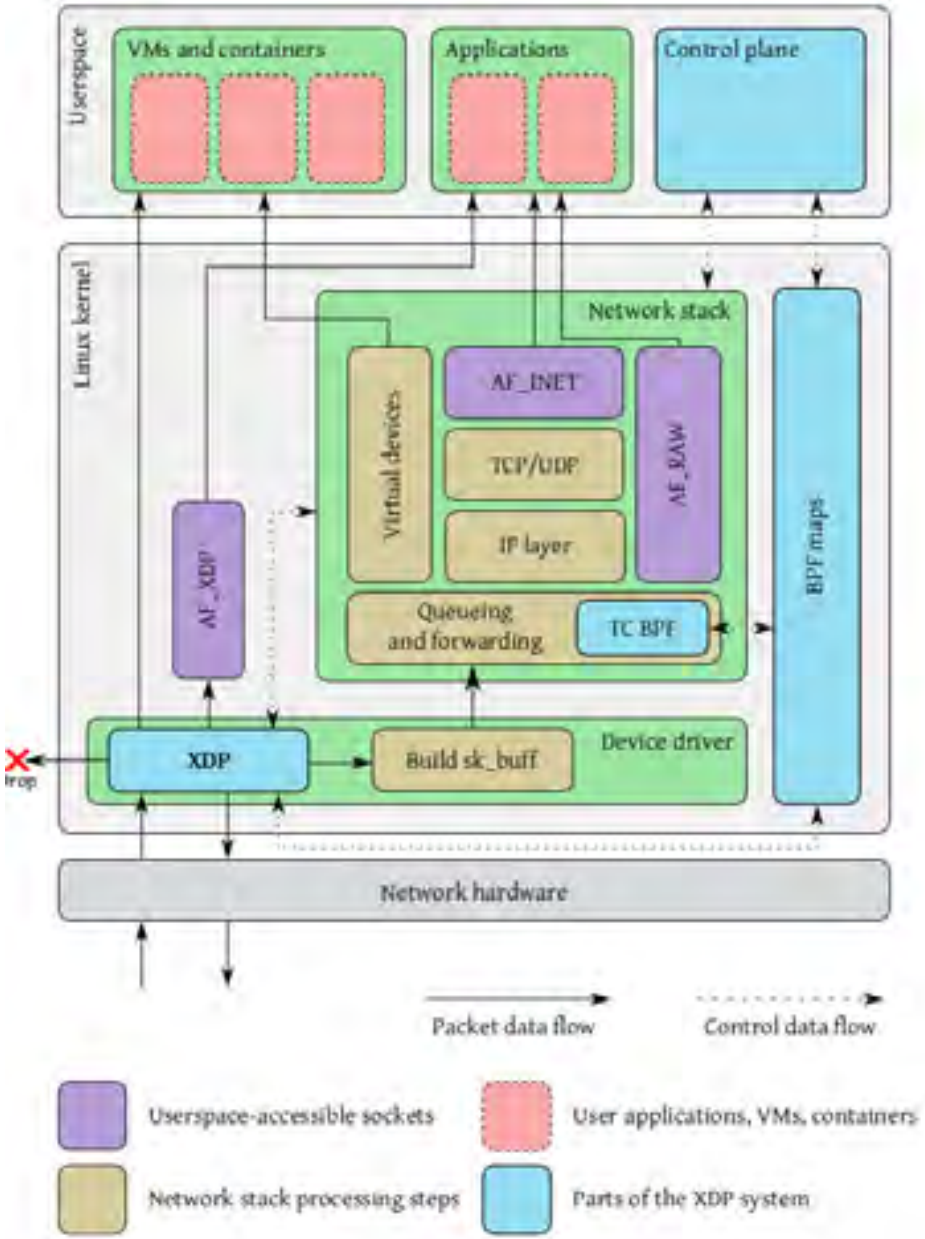
- Answer to high performance need
  - DDoS fight
  - Custom protocol implementation
- Run userspace code
- When Linux network stack do too much

Motivation

- Avoid cost of skb creation
- "Kill" DPDK
  - Universal solution and APIs
  - Avoid non Linux application on Linux

SURICON

XDP explained



A recent Linux kernel feature

Run a eBPF code the earliest possible

- in the driver
- in the card
- before the regular kernel path

Act on data

- Drop packet (eXtreme Drop Performance)
- Transmit to kernel
- Rewrite and transmit packet to kernel
- Redirect to another interface
- CPU load balance

SURICON

Implementation in Suricata

Similar to eBPF filter

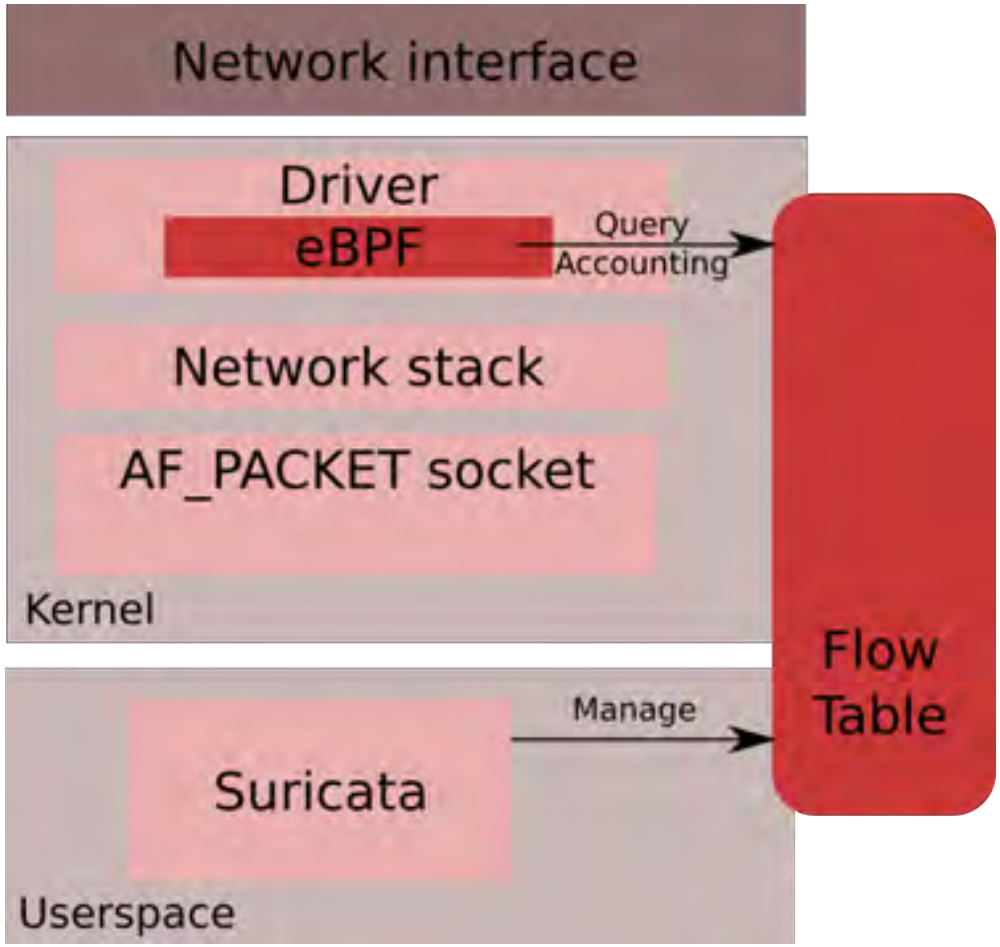
- Same logic for bypass
- Only verdict logic is different

But annoying difference

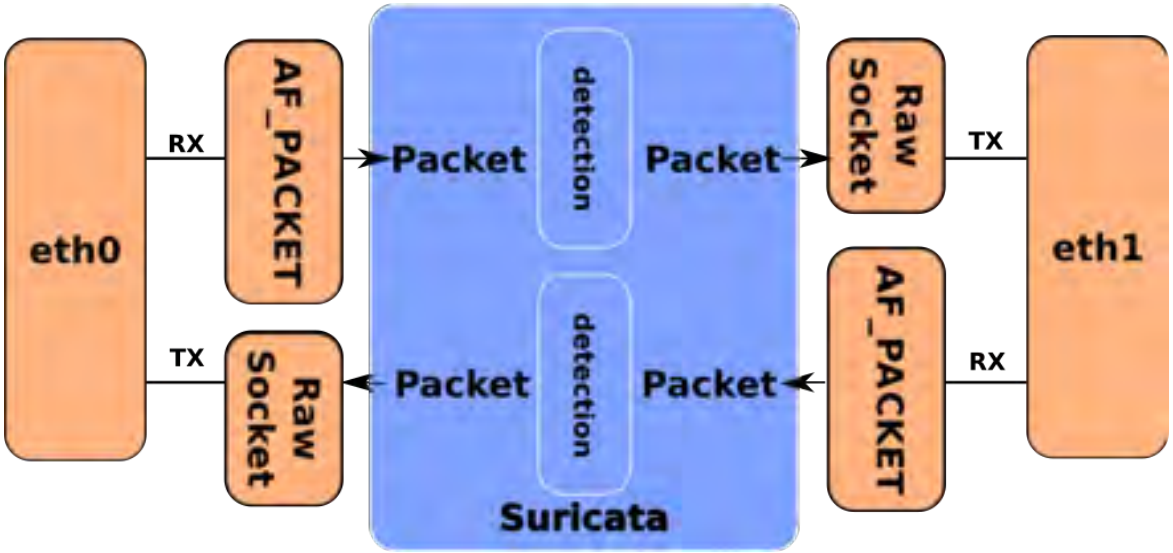
- eBPF code does the parsing
- Need to bind to an interface

SURICON

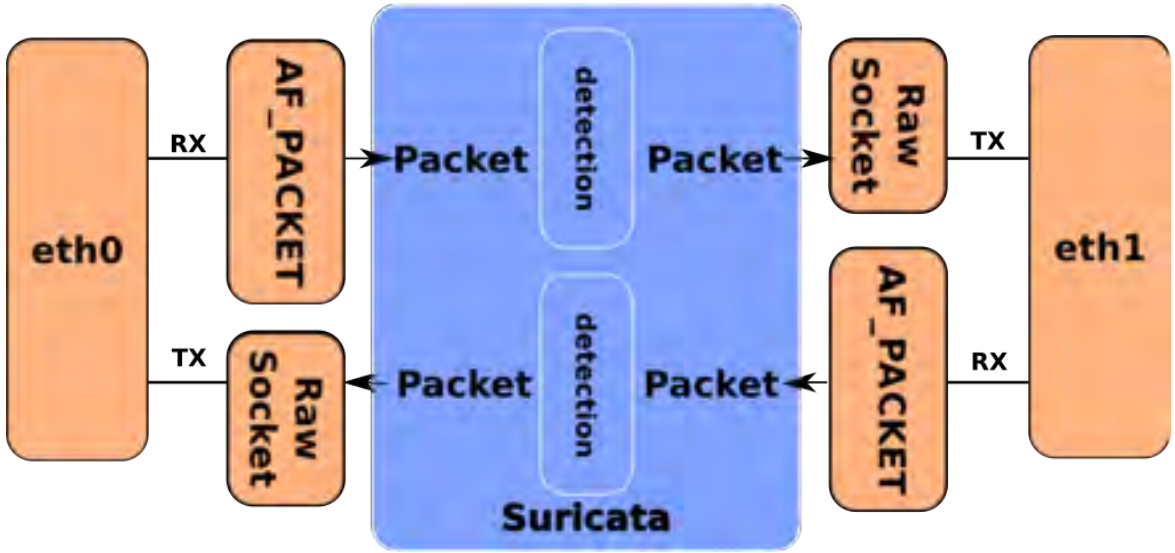
Suricata XDP architecture



AF_PACKET IPS mode



AF_PACKET IPS mode



IPS and bypass

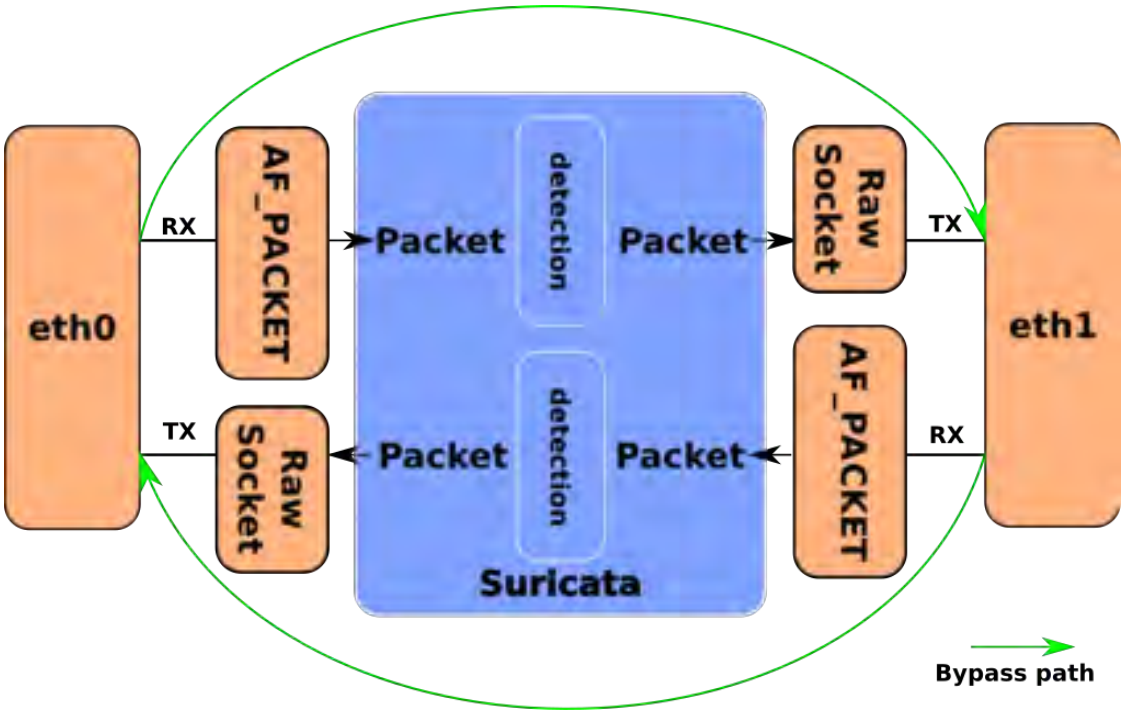
- Can’t drop packet at bypass stage
- Need to forward from one iface to another

SURICON

XDP and IPS mode: bypass

Use TX_REDIRECT

- Direct copy from interface to interface



Results

Direct NIC to NIC transfer

- Skip all kernel task
- Wire speed copy
- If eBPF code is fast enough

Obtained performance

TODO: Ask Brad Woodberg to update his tests

SURICON

TLS bypass

- Bypass counter shows a lot of TLS is not capture bypassed
- Problem with short session that are already in buffer at bypass time
- Bypass is done too late

SURICON

XDP bypass of TLS

Conditional XDP bypass of encrypted traffic

```
#if ENCRYPTED_TLS_BYPASS
    /* Packet to or from port 443 */
    if ((dport == __constant_ntohs(443)) ||
        (sport == __constant_ntohs(443))) {
        __u8 *app_data;
        /* Let's jump to data */
        nh_off += sizeof(struct iphdr) + sizeof(struct tcphdr);
        /* Please eBPF verifier and Victor with defensive code */
        if (data_end > data + nh_off + 4) {
            app_data = data + nh_off;
            /* Drop application data for tls 1.2 */
            if (app_data[0] == 0x17 &&
                app_data[1] == 0x3 && app_data[2] == 0x3) {
                return XDP_DROP;
```

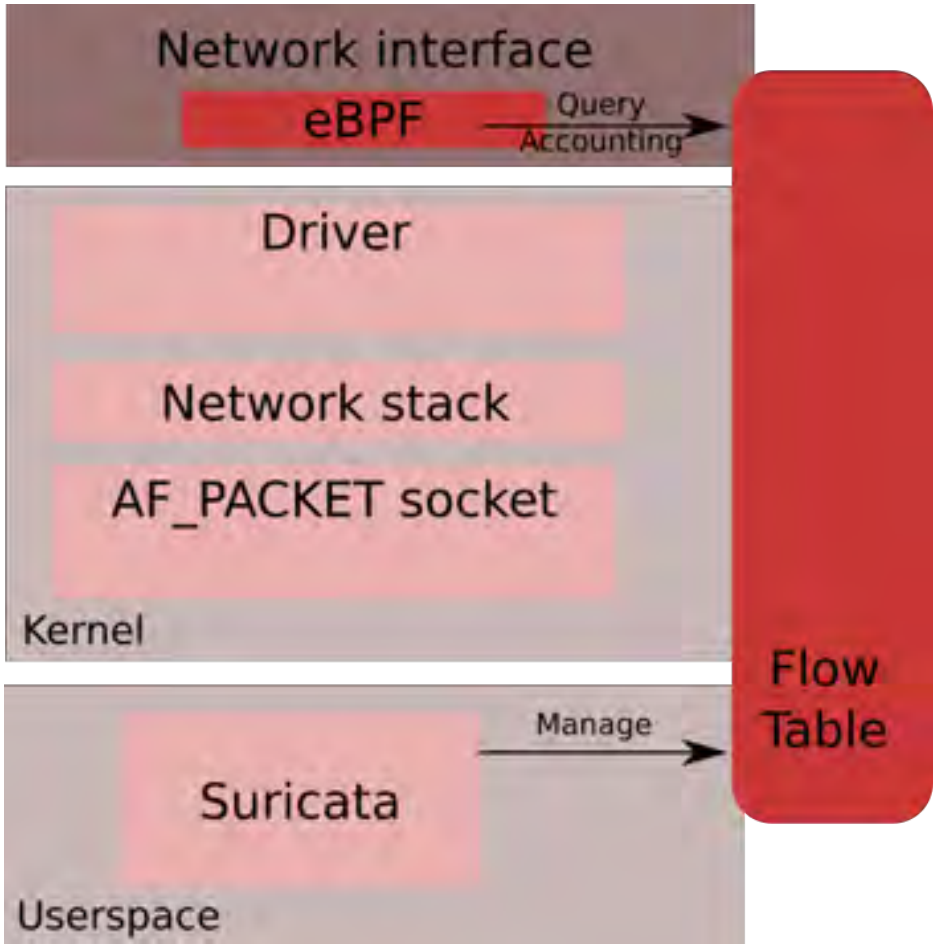
SURICON

HW offload in Netronome (quoting Simon Horman)

- The Netronome CX SmartNICs feature a network flow processor (NFP, or more commonly NPU).
- BPF programs is JITed to the instruction set of the NFP, which is analogous to JITing programs to the host instruction set, say x86_64 or aarch64.
- The JITed program is then loaded onto the NFP where it runs naively.

SURICON

Suricata XDP HW architecture



RSS queue assignement in eBPF code

Programmable RSS load balancing

- Netronome cards allow you to set target RSS queues in eBPF
- Let's fix the wrong thread problem (`https://redmine.openinfosecfoundation.org/issues/2725`)
- I was ready to suffer but a few lines later

SURICON

RSS queue assignement in eBPF code

Programmable RSS load balancing

- Netronome cards allow you to set target RSS queues in eBPF
- Let's fix the wrong thread problem (`https://redmine.openinfosecfoundation.org/issues/2725`)
- I was ready to suffer but a few lines later

```
/* IP-pairs + protocol (UDP/TCP/ICMP) hit same CPU */
__u32 xdp_hash = tuple.src + tuple.dst;
xdp_hash = SuperFastHash((char *)&xdp_hash, 4,
                         INITVAL + iph->protocol);
ctx->rx_queue_index = xdp_hash % RSS_QUEUE_NUMBERS;
```

SURICON

This talk...

- Is only possible with community and vendor help !
- Thank you!
  - LRZ
  - Netronome
  - Napatech
  - Accolade
- The effort achieved major break through at 1/2/3am...
- CpI PhDs-wise time effort worth so far

SURICON

Setup

- 7.5 million pps
- Nasty network
  - Asyn/public/private IPs
  - One way traffic
  - Not complete 3 way handshakes
  - Elephant flows
  - Go ahead make my day..
- Close to 40Gbps university traffic
- XDP in HW / XDP in SW

SURICON

Traffic

```
bwm-ng v0.6.1 (probing every 0.500s), press 'h' for help
input: /proc/net/dev type: rate
-         iface                   Rx                   Tx                Total
==============================================================================
           eno1:          868.53  B/s          772.91  B/s            1.60 KB/s
     ens2np0np0:            4.52 GB/s            0.00  B/s            4.52 GB/s
             lo:            0.00  B/s            0.00  B/s            0.00  B/s
------------------------------------------------------------------------------
          total:            4.52 GB/s          772.91  B/s            4.52 GB/s
```

Lscpu

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                112
On-line CPU(s) list:   0-111
Thread(s) per core:    2
Core(s) per socket:    28
Socket(s):             2
NUMA node(s):          2
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 85
Model name:            Intel(R) Xeon(R) Platinum 8176 CPU @ 2.10GHz
Stepping:              4
CPU MHz:               1000.052
BogoMIPS:              4200.00
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              1024K
L3 cache:              39424K
```

Just RTFM and start tuning

https://suricata.readthedocs.io/en/suricata-5.0.0/capture-hardware/ebpf-xdp.html#hardware-bypass-with-netronome

Just RTFM and start tuning

`https://suricata.readthedocs.io/en/suricata-5.0.0/capture-hardware/ebpf-xdp.html#hardware-bypass-with-netronome`

The truth

In fact not, we did write the code and the doc while doing the test

Increase flow tables size for IPv4 and IPv6

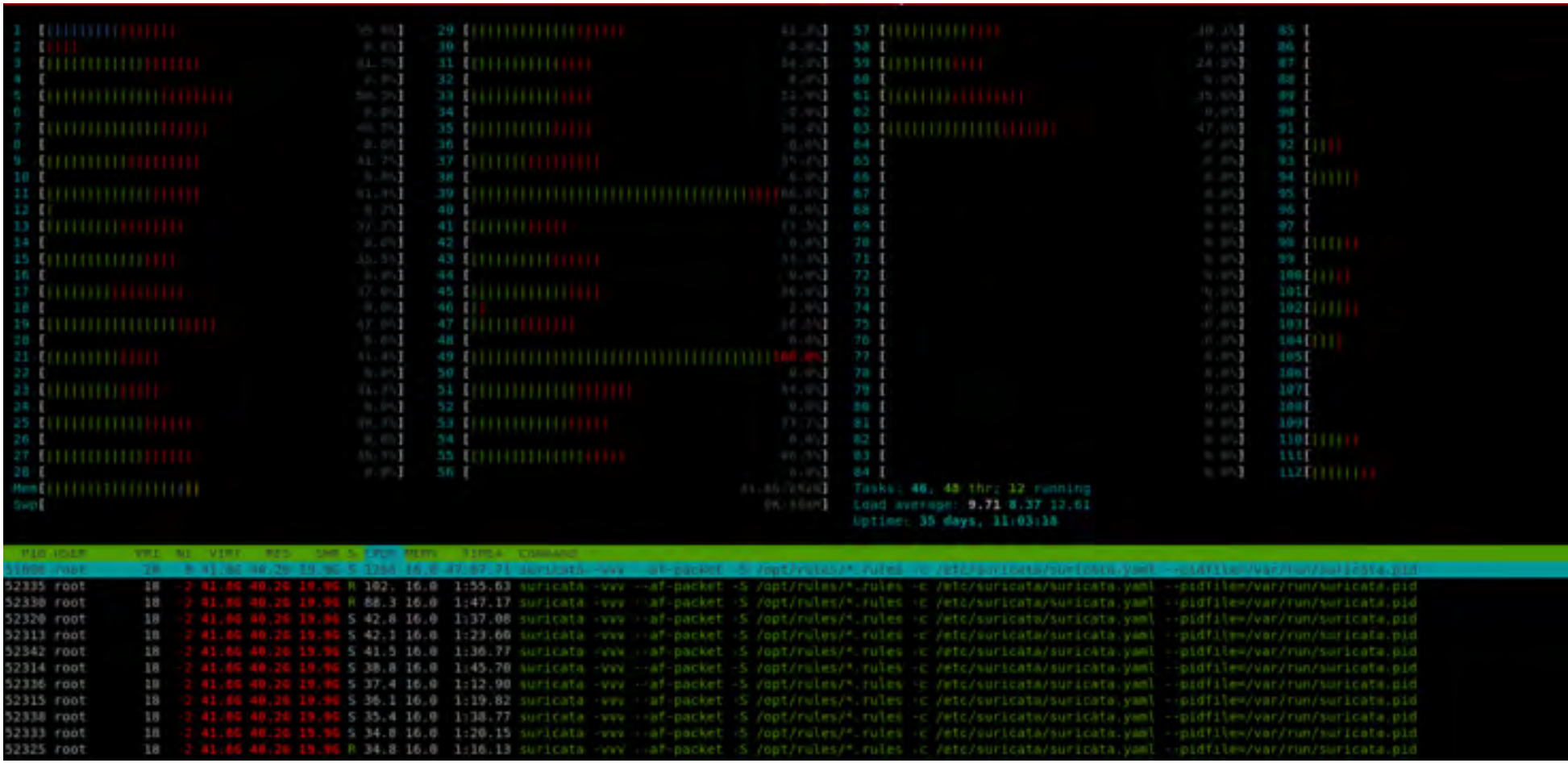
```
struct bpf_map_def SEC("maps") flow_table_v4 = {
#if USE_PERCPU_HASH
    .type = BPF_MAP_TYPE_PERCPU_HASH,
#else
    .type = BPF_MAP_TYPE_HASH,
#endif
    .key_size = sizeof(struct flowv4_keys),
    .value_size = sizeof(struct pair),
    .max_entries = 627680,
};

struct bpf_map_def SEC("maps") flow_table_v6 = {
#if USE_PERCPU_HASH
    .type = BPF_MAP_TYPE_PERCPU_HASH,
#else
    .type = BPF_MAP_TYPE_HASH,
#endif
    .key_size = sizeof(struct flowv6_keys),
    .value_size = sizeof(struct pair),
    .max_entries = 632768,
};
```

SURICON

Htop with XDP driver mode on Mellanox

Htop with XDP hardware mode on Netronome



SURICON

perf top with XDP driver mode on Mellanox

```
Samples: 1M of event 'cycles:ppp', 4000 Hz, Event count (approx.): 55
Overhead  Shared Object          Symbol
  20.09%  [kernel]               [k] __bpf_prog_run
   4.87%  libhs.so.5.1.0         [.] 0x0000000000739ee1
   3.92%  [kernel]               [k] __indirect_thunk_start
   2.40%  [kernel]               [k] tpacket_rcv
   2.33%  [kernel]               [k] memcpy_erms
   1.62%  suricata               [.] IPOnlyMatchPacket
   1.59%  libhs.so.5.1.0         [.] 0x0000000000739f32
   1.34%  suricata               [.] DetectRun
   1.33%  [kernel]               [k] mlx5e_skb_from_cqe_mpwrq_linear
   1.01%  suricata               [.] SCHSMatchEvent
   0.78%  suricata               [.] FlowManager
   0.69%  libpthread-2.28.so     [.] __pthread_mutex_lock
   0.65%  libhs.so.5.1.0         [.] 0x0000000000739ef4
   0.64%  suricata               [.] PrefilterPacketFlagsMatch
   0.61%  libhs.so.5.1.0         [.] 0x0000000000680350
   0.60%  [kernel]               [k] native_queued_spin_lock_slowpath
   0.59%  suricata               [.] QuickSortSigIntId
   0.57%  [kernel]               [k] __netif_receive_skb_core
   0.55%  suricata               [.] SCRadixFindKey
   0.55%  libhs.so.5.1.0         [.] 0x0000000000739f70
   0.52%  libhs.so.5.1.0         [.] avx512_hs_scan
   0.52%  suricata               [.] AFPReadFromRingV3
```

SURICON

perf top with XDP hardware mode on Netronome

```
Samples: 1M of event 'cycles:ppp', 4000 Hz, Event count (approx.): 275247456010
Overhead  Shared Object          Symbol
   5.75%  suricata               [.] IPOnlyMatchPacket
   5.08%  suricata               [.] rs_ntp_state_get_tx
   4.12%  suricata               [.] FlowManager
   2.93%  suricata               [.] rs_ikev2_state_get_tx
   2.84%  libc-2.28.so           [.] _int_malloc
   2.83%  libhs.so.5.1.0         [.] 0x0000000000739ee1
   2.43%  suricata               [.] SCRadixFindKey
   2.00%  libpthread-2.28.so     [.] __pthread_mutex_lock
   1.94%  suricata               [.] DetectPortLookupGroup
   1.93%  [kernel]               [k] __bpf_prog_run
   1.85%  libpthread-2.28.so     [.] __pthread_mutex_unlock_usercnt
   1.74%  suricata               [.] DetectRun
   1.68%  libhs.so.5.1.0         [.] 0x0000000000739f32
   1.24%  libc-2.28.so           [.] cfree@GLIBC_2.2.5
   1.15%  libc-2.28.so           [.] _int_free
   1.08%  [kernel]               [k] do_syscall_64
   1.07%  suricata               [.] FlowCompare
   0.88%  libhs.so.5.1.0         [.] avx512_hs_scan
   0.84%  libc-2.28.so           [.] malloc_consolidate
   0.80%  [kernel]               [k] native_queued_spin_lock_slowpath
```

Tools

- bpftool (starting to be standard packaged in Eoan)
- Quentin Monnet's twitter doc `https://twitter.com/qeole/status/1103688642701217794?lang=en`
- Brendan Gregg `http://www.brendangregg.com/blog/2019-07-15/bpf-performance-tools-book.html`
- perf top/stat/record
- HW Datasheets (CPU/NIC/etc)
- Dedicated HW for a long period of time (BIOS access is helpful too)
- Advanced knowledge of the setup
- Change one variable at a time and run a test

Netronome ethtool stats

```
ethtool -S ens2np0np0 |grep -i bpf
     bpf_pass_pkts: 226127085188
     bpf_pass_bytes: 237847884307747
     bpf_app1_pkts: 0
     bpf_app1_bytes: 0
     bpf_app2_pkts: 0
     bpf_app2_bytes: 0
     bpf_app3_pkts: 0
     bpf_app3_bytes: 0
```

Compile bpftool

```
:/opt/linux-source-4.19# make -C tools/bpf/bpftool/
make: Entering directory '/opt/linux-source-4.19/tools/bpf/bpftool'

Auto-detecting system features:
...                         libbfd: [ on  ]
...        disassembler-four-args: [ on  ]

  CC       map_perf_ring.o
  CC       skeleton.o
  CC       perf.o
  CC       prog.o
  CC       btf_dumper.o
  CC       common.o
  CC       cgroup.o
  CC       main.o
  CC       json_writer.o
  CC       cfg.o
  CC       map.o
  CC       jit_disasm.o
  CC       disasm.o
make[1]: Entering directory '/opt/linux-source-4.19/tools/lib/bpf'

Auto-detecting system features:
...                        libelf: [ on  ]
...                          bpf: [ on  ]

Warning: Kernel ABI header at 'tools/include/uapi/linux/bpf.h' differs from latest version at 'include/uapi/linux/bpf.h'
  CC       libbpf.o
  CC       bpf.o
  CC       nlattr.o
  CC       btf.o
  CC       libbpf_errno.o
  CC       str_error.o
  LD       libbpf-in.o
  LINK     libbpf.a
make[1]: Leaving directory '/opt/linux-source-4.19/tools/lib/bpf'
  LINK     bpftool
make: Leaving directory '/opt/linux-source-4.19/tools/bpf/bpftool'

make install
```

SURICON

bpftool and jq

Bfpftool show program

```
root@suricata-ng02:/opt/linux-source-4.19/tools/bpf/bpftool# bpftool prog show
54: cgroup_skb  tag 7be49e3934a125ba  gpl
        loaded_at 2019-10-15T02:24:47+0200  uid 0
        xlated 296B  not jited  memlock 4096B  map_ids 52,53
55: cgroup_skb  tag 2a142ef67aaad174  gpl
        loaded_at 2019-10-15T02:24:47+0200  uid 0
        xlated 296B  not jited  memlock 4096B  map_ids 52,53
56: cgroup_skb  tag 7be49e3934a125ba  gpl
        loaded_at 2019-10-15T02:24:47+0200  uid 0
        xlated 296B  not jited  memlock 4096B  map_ids 54,55
57: cgroup_skb  tag 2a142ef67aaad174  gpl
        loaded_at 2019-10-15T02:24:47+0200  uid 0
        xlated 296B  not jited  memlock 4096B  map_ids 54,55
58: cgroup_skb  tag 7be49e3934a125ba  gpl
        loaded_at 2019-10-15T02:24:47+0200  uid 0
        xlated 296B  not jited  memlock 4096B  map_ids 56,57
59: cgroup_skb  tag 2a142ef67aaad174  gpl
        loaded_at 2019-10-15T02:24:47+0200  uid 0
        xlated 296B  not jited  memlock 4096B  map_ids 56,57
104: xdp  name xdp_hashfilter  tag dea6ef69443d3c07 dev ens2np0np0  gpl
        loaded_at 2019-10-30T15:15:19+0100  uid 0
        xlated 1720B  jited 4088B  memlock 4096B  map_ids 92,93
root@suricata-ng02:/opt/linux-source-4.19/tools/bpf/bpftool# bpftool prog show id 104
104: xdp  name xdp_hashfilter  tag dea6ef69443d3c07 dev ens2np0np0  gpl
        loaded_at 2019-10-30T15:15:19+0100  uid 0
        xlated 1720B  jited 4088B  memlock 4096B  map_ids 92,93
```

SURICON

The good news

- 32 cores are dealing with 35Gbps University traffic with full ETPRO (50k sigs)
- Basically 1 core per 1 Gbps for a CPU
- Intel(R) Xeon(R) Platinum 8176 CPU @ 2.10GHz (28 core/socket)
- 32 cores/32 RSS/32 suricata worker threads/AFPv3
- Netronome 40Gbps Agilio/eBPF enabled firmware
- XDP offer huge flexibility via eBPF
- pinned maps save state cross Suricata restarts (major boost for IPS as well)

It's not all roses (just yet)

Suricata

- Some counters seem off (including in Suricata)
- Suricata's config/setup on higher speeds becomes more and more complex
- At higher speeds log writing seems to get a lot of log file lock contention - directly affecting drops

OS

- Though to profile and pinpoint/ investigate the end-to-end path/reason (tooling)
- Needs more investigation of the performance 'hot spots'

SURICON

It's not all roses (just yet)

XDP

- RSS/threads need to be power of 2 (32/64...)
- MTU 3000B , "one frame cannot exceed a page"

Test/QA can benefit from

- Defined/repeatable test framework/log formats etc
- Troubleshooting and pin pointing hot spots requires common effort
- Cold War style environment - switch off chats/phones/satellite dishes/TV and Radio sets and go underground

SURICON

LLC Load misses

```
Performance counter stats for 'system wide':

      392,774.11 msec cpu-clock                 #  111.587 CPUs utilized
         339,698      context-switches          #  864.869 M/sec
               2      cpu-migrations            #    0.005 M/sec
          45,951      page-faults               #  116.991 M/sec
 154,433,950,183      cycles                    # 393187.813 GHz                    (31.02%)
 156,262,379,619      instructions              #    1.01  insn per cycle           (38.61%)
  25,741,114,192      branches                  # 65536706.075 M/sec                (38.58%)
     771,702,366      branch-misses             #    3.00% of all branches          (38.61%)
  53,945,558,779      L1-dcache-loads           # 137345035.005 M/sec               (38.62%)
   6,197,895,690      L1-dcache-load-misses     #   11.49% of all L1-dcache hits    (38.56%)
     649,449,405      LLC-loads                 # 1653493.879 M/sec                 (30.88%)
     130,440,232      LLC-load-misses           #   20.08% of all LL-cache hits     (30.90%)
<not supported>       L1-icache-loads
   2,462,379,891      L1-icache-load-misses                                       (30.91%)
  53,996,118,167      dTLB-loads                # 137473758.872 M/sec               (30.94%)
     101,991,140      dTLB-load-misses          #    0.19% of all dTLB cache hits   (30.95%)
     112,723,570      iTLB-loads                # 286993.462 M/sec                  (30.94%)
      16,287,652      iTLB-load-misses          #   14.45% of all iTLB cache hits   (30.86%)
```

Busy thread

```
Samples: 809K of event 'cycles:ppp', 4000 Hz, Event count (approx.): 245062134018
Overhead  Shared Object          Symbol
   8.00%  suricata               [.] IPOnlyMatchPacket
   5.96%  libhs.so.5.1.0         [.] 0x0000000000739ee1
   3.41%  suricata               [.] FlowManager
   2.90%  [kernel]               [k] __bpf_prog_run
   2.31%  suricata               [.] DetectPortLookupGroup
   2.14%  libc-2.28.so           [.] _int_malloc
   1.96%  suricata               [.] DetectRun
   1.91%  libpthread-2.28.so     [.] __pthread_mutex_lock
   1.74%  suricata               [.] SCRadixFindKey
   1.69%  libhs.so.5.1.0         [.] 0x0000000000739f32
   1.62%  [kernel]               [k] tpacket_rcv
   1.43%  libpthread-2.28.so     [.] __pthread_mutex_unlock_usercnt
   1.39%  suricata               [.] AFPReadFromRingV3
   1.14%  [kernel]               [k] delay_tsc
   0.93%  libc-2.28.so           [.] malloc_consolidate
   0.83%  libhs.so.5.1.0         [.] avx512_hs_scan
   0.83%  [kernel]               [k] memcpy_erms
   0.77%  suricata               [.] FlowCompare
   0.75%  libc-2.28.so           [.] _int_free
   0.74%  libhs.so.5.1.0         [.] 0x0000000000739ef4
   0.72%  [kernel]               [k] nfp_net_poll
```

SURICON

Some top level sigs

```
6466833 ET INFO Observed DNS Query to .biz TLD
4103564 GPL ICMP_INFO PING BSDtype
3423095 ET SCAN Suspicious inbound to MSSQL port 1433
1624002 ET INFO Observed DNS Query to .work TLD
1229814 ET POLICY RDP connection confirm
1082937 ET INFO Observed DNS Query to .cloud TLD
 798053 GPL ICMP_INFO PING *NIX
 689702 ET INFO Observed DNS Query to .world TLD
 662795 ET INFO Observed DNS Query to .life TLD
 612491 ET JA3 Hash - Possible Malware - Banking Phish
 545210 ET DNS Query to a .tk domain - Likely Hostile
 530073 ET JA3 Hash - Possible Malware - Fake Firefox Font Update
 314599 ET DNS Query for .su TLD (Soviet Union) Often Malware Related
 285551 GPL SNMP public access udp
 246999 ET DNS Query for .cc TLD
 157486 ET SCAN MS Terminal Server Traffic on Non-standard Port
 144811 ETPRO POLICY Android Device Connectivity Check
 130202 ET INFO Session Traversal Utilities for NAT (STUN Binding Request)
 111448 ET INFO Session Traversal Utilities for NAT (STUN Binding Response)
  97521 ET DNS Query for .to TLD
  95590 ET INFO Observed DNS Query to .fit TLD
  92904 ET USER_AGENTS Microsoft Device Metadata Retrieval Client User-Agent
  88907 ET DNS Query to a *.pw domain - Likely Hostile
  87299 ET SCAN Potential SSH Scan
  84543 ET JA3 Hash - [Abuse.ch] Possible Adware
  79593 GPL RPC xdmcp info query
  77983 ET POLICY MS Remote Desktop Administrator Login Request
  50801 ETPRO HUNTING Suspicious Registrar Nameservers in DNS Response (internet .bs)
  46769 ET POLICY GNU\/Linux APT User-Agent Outbound likely related to package management
  45233 ET POLICY DNS Query For XXX Adult Site Top Level Domain
  42272 ET JA3 Hash - [Abuse.ch] Possible Tofsee
  27191 GPL DNS named version attempt
  26652 ET POLICY Python-urllib\/ Suspicious User Agent
  25734 ET SCAN Suspicious inbound to mySQL port 3306
  24974 ET USER_AGENTS Go HTTP Client User-Agent
  24969 ET JA3 Hash - [Abuse.ch] Possible Adwind
  22793 ET SCAN Behavioral Unusually fast Terminal Server Traffic Potential Scan or Infection (Inbound)
  22709 ETPRO POLICY Python Requests Suspicious User Agent
  19109 ET INFO WinHttp AutoProxy Request wpad.dat Possible BadTunnel
  17322 ET POLICY HTTP traffic on port 443 (POST)
  16766 ET POLICY curl User-Agent Outbound
  13583 GPL ICMP_INFO PING BayRS Router
  13545 ET SCAN Suspicious inbound to PostgreSQL port 5432
  13464 GPL ICMP_INFO PING Flowpoint2200 or Network Management Software
  12547 ET HUNTING Suspicious User-Agent (1 space)
  12261 ET INFO Observed DNS Query to .okinawa TLD
  12253 ET INFO Observed DNS Query to .desi TLD
  12072 ETPRO POLICY Empty User-Agent Header
  10975 GPL WEB_SERVER 403 Forbidden
  10321 ET SCAN Suspicious inbound to Oracle SQL port 1521
   9307 ET INFO DNS Query for Suspicious .gdn Domain
```

SURICON

Worker threads still pegged

```
top - 13:31:08 up 36 days,  7:13,  6 users,  load average: 12.16, 11.98, 12.17
Threads:  44 total,  14 running,  30 sleeping,   0 stopped,   0 zombie
%Cpu(s):  8.8 us,  0.5 sy,  0.2 ni, 85.6 id,  0.0 wa,  0.0 hi,  5.0 si,  0.0 st
MiB Mem : 257629.0 total, 184485.3 free,  66633.5 used,   6510.2 buff/cache
MiB Swap:    488.0 total,    488.0 free,      0.0 used. 188959.0 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
98851 root      18  -2   65.2g  63.3g  39.7g R  98.7  25.2   9:45.66 W#19-ens2np0np0
98864 root      18  -2   65.2g  63.3g  39.7g R  92.7  25.2   8:10.51 W#32-ens2np0np0
98836 root      18  -2   65.2g  63.3g  39.7g R  49.5  25.2   7:18.00 W#04-ens2np0np0
98852 root      18  -2   65.2g  63.3g  39.7g R  47.5  25.2   5:15.05 W#20-ens2np0np0
98837 root      18  -2   65.2g  63.3g  39.7g S  46.5  25.2   8:18.28 W#05-ens2np0np0
98860 root      18  -2   65.2g  63.3g  39.7g R  46.2  25.2   6:28.44 W#28-ens2np0np0
98857 root      18  -2   65.2g  63.3g  39.7g R  43.9  25.2   7:02.90 W#25-ens2np0np0
98846 root      18  -2   65.2g  63.3g  39.7g R  43.5  25.2   9:52.99 W#14-ens2np0np0
98853 root      18  -2   65.2g  63.3g  39.7g S  40.5  25.2   5:20.55 W#21-ens2np0np0
98842 root      18  -2   65.2g  63.3g  39.7g R  39.5  25.2   4:40.68 W#10-ens2np0np0
98839 root      18  -2   65.2g  63.3g  39.7g S  37.9  25.2   5:51.56 W#07-ens2np0np0
98861 root      18  -2   65.2g  63.3g  39.7g S  36.2  25.2   5:50.05 W#29-ens2np0np0
98843 root      18  -2   65.2g  63.3g  39.7g S  35.9  25.2   5:11.50 W#11-ens2np0np0
98850 root      18  -2   65.2g  63.3g  39.7g S  35.5  25.2   5:07.60 W#18-ens2np0np0
98835 root      18  -2   65.2g  63.3g  39.7g S  34.6  25.2   5:25.28 W#03-ens2np0np0
98858 root      18  -2   65.2g  63.3g  39.7g S  34.2  25.2   5:15.75 W#26-ens2np0np0
98838 root      18  -2   65.2g  63.3g  39.7g R  33.6  25.2   5:00.52 W#06-ens2np0np0
98841 root      18  -2   65.2g  63.3g  39.7g S  33.6  25.2   4:38.37 W#09-ens2np0np0
98847 root      18  -2   65.2g  63.3g  39.7g S  32.9  25.2   4:52.59 W#15-ens2np0np0
98848 root      18  -2   65.2g  63.3g  39.7g S  32.9  25.2   5:00.20 W#16-ens2np0np0
98840 root      18  -2   65.2g  63.3g  39.7g S  32.2  25.2   5:18.45 W#08-ens2np0np0
98862 root      18  -2   65.2g  63.3g  39.7g S  32.2  25.2   4:59.88 W#30-ens2np0np0
98849 root      18  -2   65.2g  63.3g  39.7g R  31.9  25.2   5:06.70 W#17-ens2np0np0
98863 root      18  -2   65.2g  63.3g  39.7g R  30.6  25.2   4:52.30 W#31-ens2np0np0
98856 root      18  -2   65.2g  63.3g  39.7g S  30.2  25.2   4:48.02 W#24-ens2np0np0
98855 root      18  -2   65.2g  63.3g  39.7g R  29.6  25.2   5:16.89 W#23-ens2np0np0
98831 root      18  -2   65.2g  63.3g  39.7g S  29.2  25.2   5:45.53 W#02-ens2np0np0
98854 root      18  -2   65.2g  63.3g  39.7g S  29.2  25.2   5:42.19 W#22-ens2np0np0
98844 root      18  -2   65.2g  63.3g  39.7g S  28.6  25.2   4:27.12 W#12-ens2np0np0
98830 root      22   2   65.2g  63.3g  39.7g S  28.2  25.2   5:13.31 W#01-ens2np0np0
98845 root      18  -2   65.2g  63.3g  39.7g R  28.2  25.2   5:16.02 W#13-ens2np0np0
98859 root      18  -2   65.2g  63.3g  39.7g R  27.6  25.2   6:29.37 W#27-ens2np0np0
98869 root      20   0   65.2g  63.3g  39.7g S  17.6  25.2   2:43.10 FR#01
98871 root      20   0   65.2g  63.3g  39.7g S  17.6  25.2   2:42.61 FR#03
98865 root      20   0   65.2g  63.3g  39.7g S  16.3  25.2   3:00.42 FM#01
98866 root      20   0   65.2g  63.3g  39.7g S  15.3  25.2   2:31.49 FM#02
98868 root      20   0   65.2g  63.3g  39.7g S  15.0  25.2   2:32.11 FM#04
98867 root      20   0   65.2g  63.3g  39.7g S  14.6  25.2   2:31.91 FM#03
98870 root      20   0   65.2g  63.3g  39.7g S  11.6  25.2   2:41.31 FR#02
98872 root      20   0   65.2g  63.3g  39.7g S  11.6  25.2   2:43.60 FR#04
98618 root      20   0   65.2g  63.3g  39.7g S   0.3  25.2   1:22.08 Suricata-Main
98873 root      20   0   65.2g  63.3g  39.7g S   0.0  25.2   0:00.00 CW
98874 root      20   0   65.2g  63.3g  39.7g S   0.0  25.2   0:00.06 CS
98875 root      20   0   65.2g  63.3g  39.7g S   0.0  25.2   0:00.08 US
```

SURICON

SURICON

Upcoming

- Full test and configuration data release (2020)
- Napatech NIC testing and added to the mix (just released new driver/code)
- Doc update for Suricata
- SEPTun III planning around Feb 2020 (cc @MichalPurzynski)

SURICON

Future dev work

AF_XDP

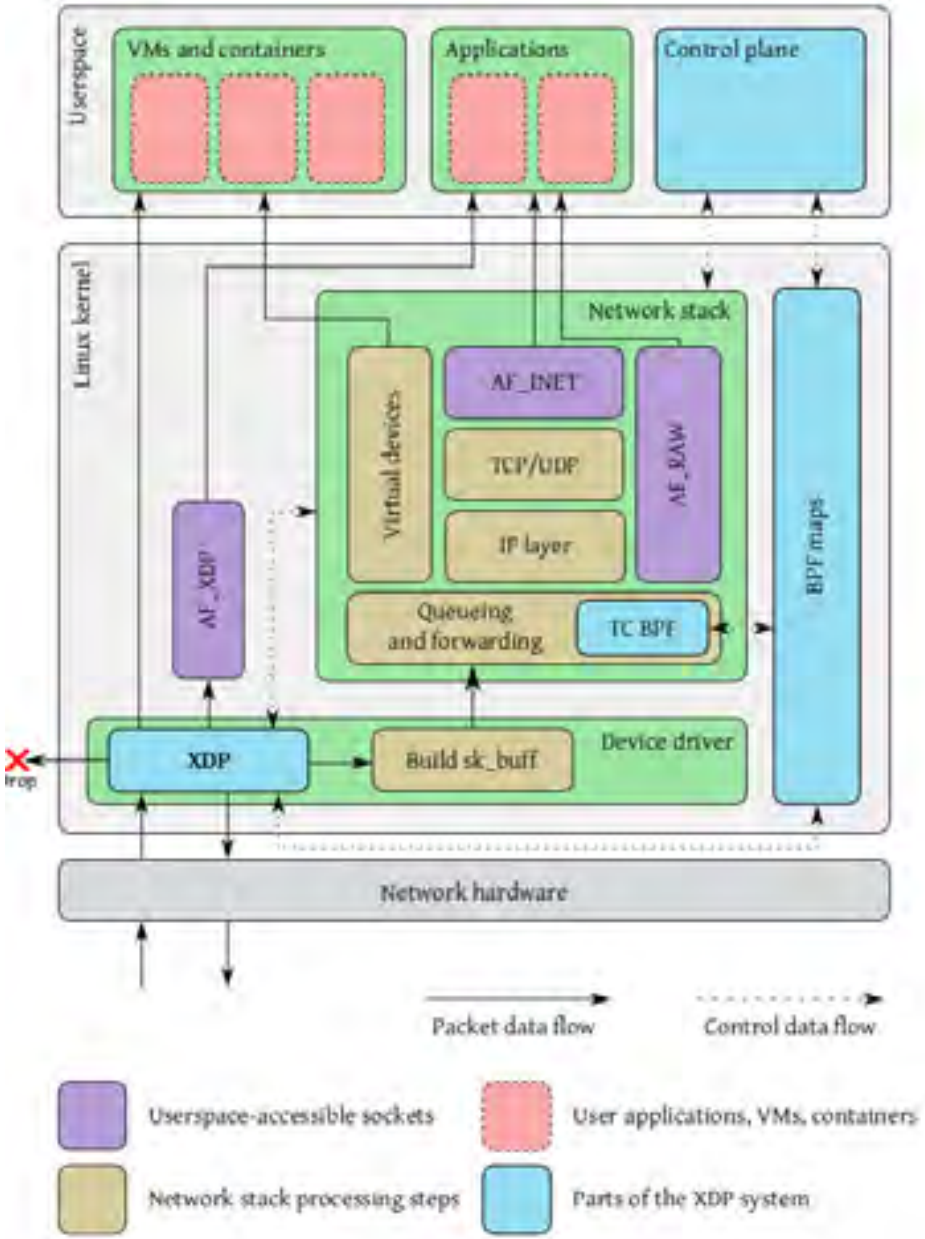
- AF_XDP capture is now in Linux vanilla
- eBPF code send data to userspace
- Kernel work is skipped

Some missing features

- Timestamp are missing

SURICON

XDP explained: AF_XDP



Conclusion

Suricata, eBPF and XDP

- Available in Suricata 5, need Linux 4.16
- Network card bypass for Netronome

More information

- Suricata doc: `http://suricata.readthedocs.io/en/latest/capture-hardware/ebpf-xdp.html`
- XDP whitepaper: `https://www.stamus-networks.com/2019/07/16/whitepaper-introduction-to-ebpf-and-xdp-support-in`

SURICON

Questions ?

Thanks to

- Jesper Dangaard Brouer
- Alexei Starovoitov
- Daniel Borkmann

Contact Us

- Twitter: @pevma @regiteric

SURICON