Suricata IDPS and Linux kernel

É. Leblond, G. Longo

Stamus Networks

February 10, 2016

STAMUS NETWORKS

STAMUS NETWORKS

STAMUS NETWORKS

What is Suricata

- IDS and IPS engine
- Get it here:
  `http://www.suricata-ids.org`
- Open Source (GPLv2)
- Initially publicly funded now funded by consortium members
- Run by Open Information Security Foundation (OISF)
- More information about OISF at
  `http://www.openinfosecfoundation.org/`

STAMUS NETWORKS

Suricata Features

- High performance, scalable through multi threading
- Advanced Protocol handling
  - Protocol recognition
  - Protocol analysis: field extraction, filtering keywords
  - Transaction logging in extensible JSON format
- File identification, extraction, on the fly MD5 calculation
  - HTTP
  - SMTP
- TLS handshake analysis, detect/prevent things like Diginotar
- Lua scripting for detection
- Hardware acceleration support:
  - Endace
  - Napatech,
  - CUDA
  - PF_RING

STAMUS NETWORKS

Suricata capture modes

IDS

- pcap: multi OS capture
- af_packet: Linux high performance on vanilla kernel
- netmap: FreeBSD high performance
- NFLOG: Netfilter logging

IPS

- NFQUEUE: Using Netfilter on Linux
- ipfw: Use divert socket on FreeBSD
- af_packet: Level 2 software bridge

Offline analysis

- Pcap: Analyse pcap files
- Unix socket: Use Suricata for fast batch processing of pcap files

STAMUS NETWORKS

Evasion technics

Fooling detection

- Get your activity unnoticed
- Complete your attack and stay in place

Principle

- Signature-based IDS relay on packet content
- Modification of traffic could be used to avoid detection
- Without changing the impact of the attack

STAMUS NETWORKS

Play on interpretation issue

OS-based evasion

- All OS do not react the same
  - RFC are incomplete. Improvisations have been made.
  - Variation of traffic for a same flow is possible
- Overlapping Fragments

Application-based evasion

- Different servers can treat the same request differently.
- No web server are treating a twice used argument the same way.

STAMUS NETWORKS

Personnality

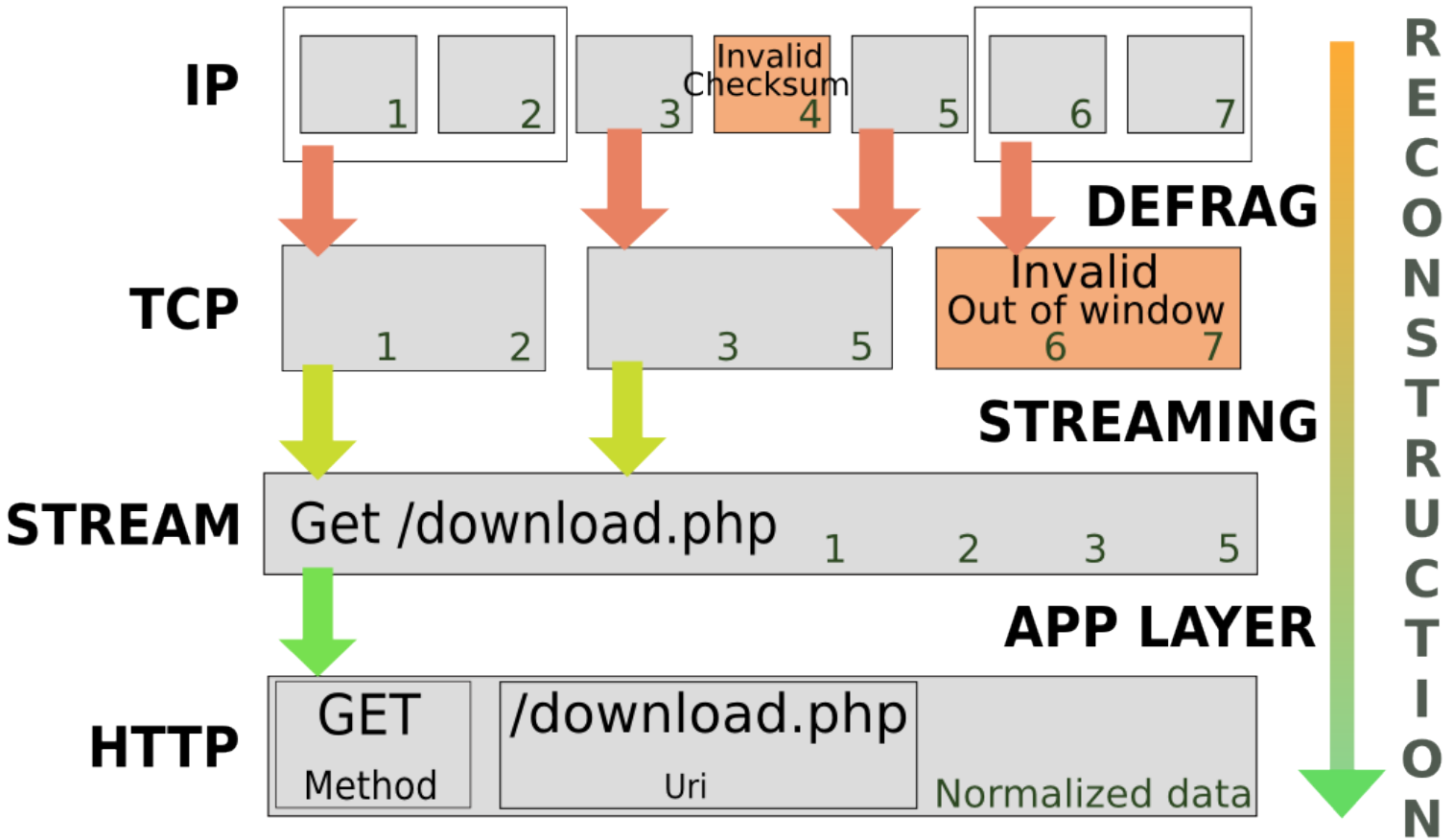
- IDS implements personnality
- It is possible to associate network and OS type
- For Suricata, HTTP servers can be personnified too.

Suricata configuration

```
host-os-policy:
  # Make the default policy windows.
  windows: [0.0.0.0/0]
  bsd: []
  bsd-right: []
  old-linux: []
  linux: [10.0.0.0/8]
```

STAMUS NETWORKS

Suricata reconstruction and normalization



STAMUS NETWORKS

STAMUS NETWORKS

A typical signature example

Signature example: Chat facebook

```
alert http $HOME_NET any -> $EXTERNAL_NET any \
(
 msg:"ET CHAT Facebook Chat (send message)"; \
 flow:established,to_server; content:"POST"; http_method; \
 content:"/ajax/chat/send.php"; http_uri; content:"facebook.com"; http_host; \
 content:"netdev"; http_client_body;
 reference:url,www.emergingthreats.net/cgi-bin/cvsweb.cgi/sigs/POLICY/POLICY_Facebook_Chat; \
 sid:2010784; rev:4; \
)
```

This signature tests:

- The HTTP method: *POST*
- The page: */ajax/chat/send.php*
- The domain: *facebook.com*
- The body content: *netdev*

STAMUS NETWORKS

No passthrough

All signatures are inspected

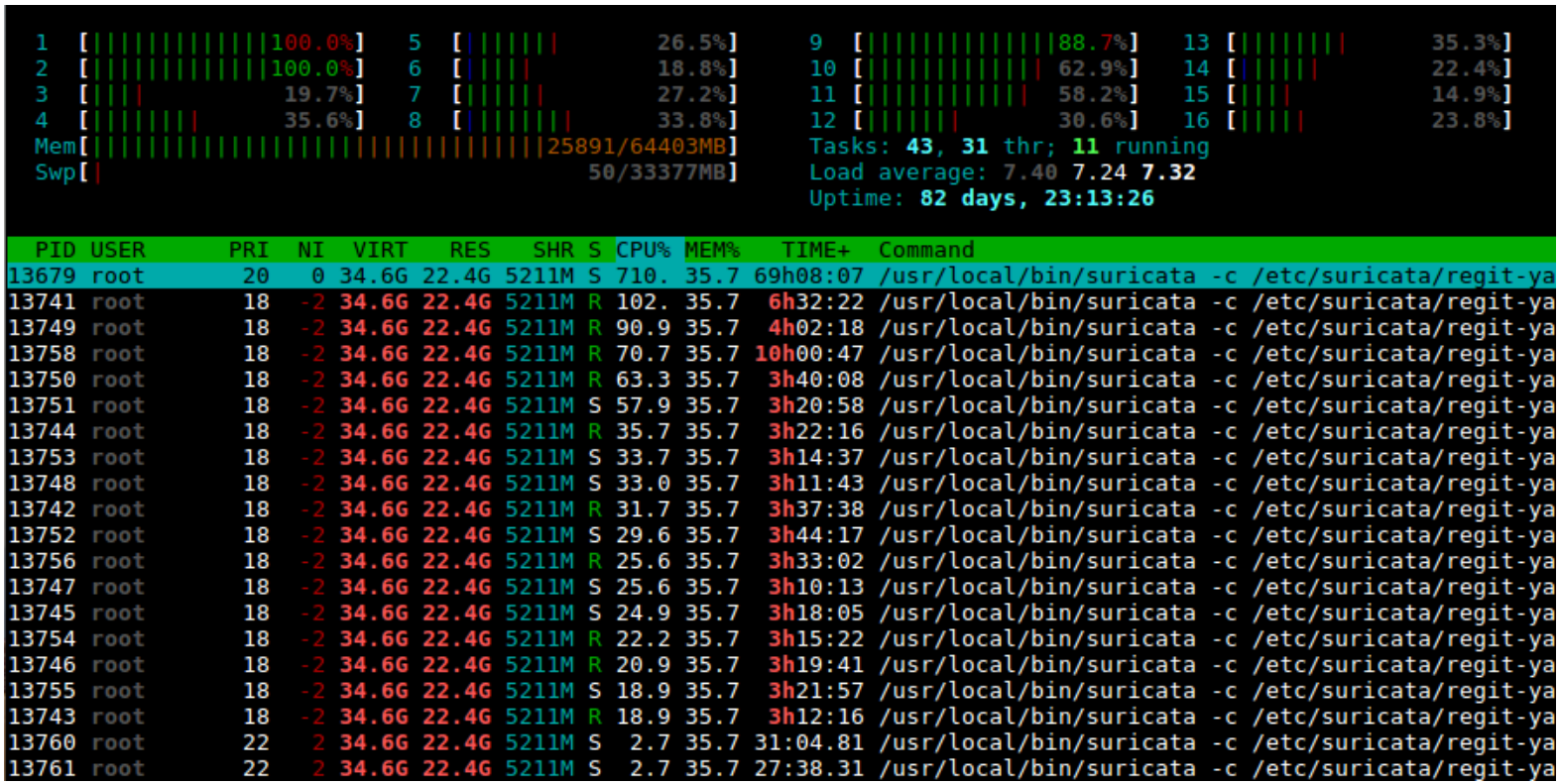
- Different from a firewall
- More than 15000 signatures in standard rulesets

Optimization on detection engine

- Tree pre filtering approach to limit the set of signatures to test
- Multi pattern matching on some buffers

STAMUS NETWORKS

CPU intensive



STAMUS NETWORKS

Perf top


```
Samples: 691K of event 'cycles', Event count (approx.): 256764876818
Overhead  Shared Object                 Symbol
  64.14%  suricata                      [.] SCACSearch
   3.20%  suricata                      [.] BoyerMoore
   1.16%  suricata                      [.] SigMatchSignatures
   0.90%  libc-2.19.so                  [.] memset
   0.87%  [kernel]                      [k] ixgbe_clean_rx_irq
   0.75%  suricata                      [.] IPOnlyMatchPacket
   0.68%  libpthread-2.19.so            [.] pthread_mutex_unlock
   0.64%  [kernel]                      [k] __netif_receive_skb_core
   0.62%  libpthread-2.19.so            [.] pthread_mutex_lock
   0.62%  suricata                      [.] AFPReadFromRing
   0.61%  [kernel]                      [k] irq_entries_start
   0.58%  [kernel]                      [k] tpacket_rcv
   0.55%  libc-2.19.so                  [.] __memcmp_sse4_1
   0.52%  [kernel]                      [k] memcpy
   0.42%  [kernel]                      [k] ixgbe_poll
   0.42%  [kernel]                      [k] menu_select
   0.40%  suricata                      [.] StreamTcpPacket
   0.36%  [kernel]                      [k] native_write_msr_safe
   0.35%  [kernel]                      [k] packet_lookup_frame.isra.56
```

STAMUS NETWORKS

- Bandwith per core is limited
  - From 150Mb/s
  - To 500Mb/s
- Scaling
  - Using RSS
  - Splitting load on workers

STAMUS NETWORKS

STAMUS NETWORKS

STAMUS NETWORKS

Linux raw socket

- Raw packet capture method
- Socket based or mmap based

STAMUS NETWORKS

Linux raw socket

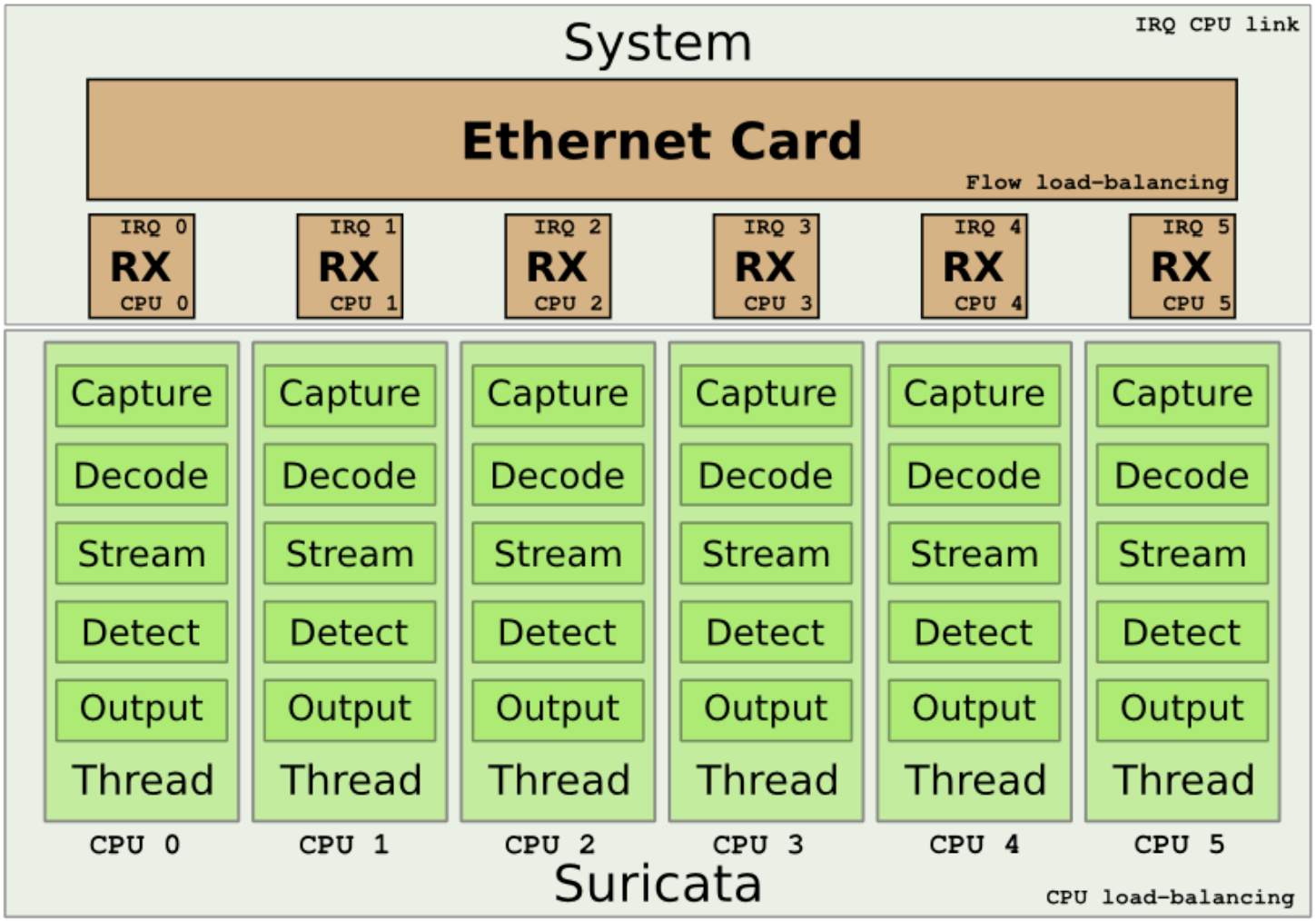
- Raw packet capture method
- Socket based or mmap based

Fanout mode

- Load balancing over multiple sockets
- Multiple load balancing functions
  - Flow based
  - CPU based
  - RSS based

STAMUS NETWORKS

Suricata workers mode



STAMUS NETWORKS

The rollover option

Concept

- Ring buffer can fill in burst or single flow
- Capture would gain of splitting single intensive flow
- Rollover mode switch to next socket when ring is full

STAMUS NETWORKS

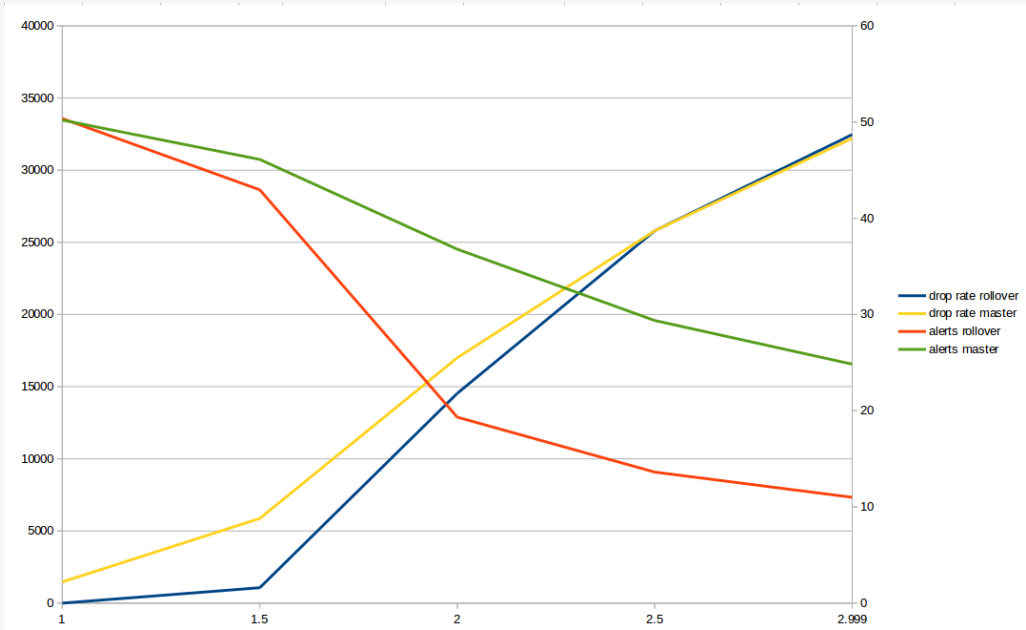
The rollover option

Concept

- Ring buffer can fill in burst or single flow
- Capture would gain of splitting single intensive flow
- Rollover mode switch to next socket when ring is full

Problem with Suricata

- Suricata reconstruct the stream
- Rollover mode causes reordering of stream
- Massive accuracy loss



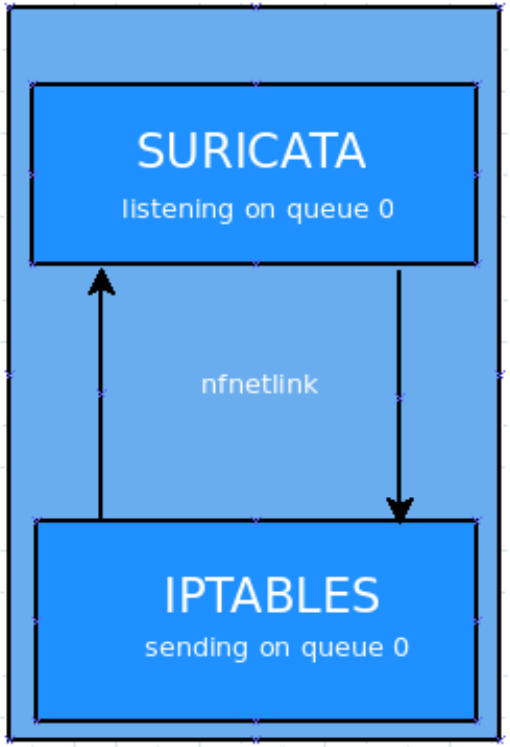
STAMUS NETWORKS

- It is used in Suricata to work in IPS mode, performing action like DROP or ACCEPT on the packets, permitting us to delegate the verdict on the packets.
- With NFQUEUE we are able to delegate the verdict on the packet to a userspace software.
- The following rules will ask a userspace software connected to queue 0 for a decision.
  - **nft add filter forward queue num 0**
  - **iptables -A FORWARD -j NFQUEUE –queue-num 0**

STAMUS NETWORKS

The following steps explains how NFQUEUE works with Suricata in IPS mode:

- Incoming packet matched by a rule is sent to Suricata through nfnetlink
- Suricata receives the packet and issues a verdict depending on our ruleset
- The packet is either trasmitted or rejected by kernel



STAMUS NETWORKS

- NFQUEUE number of packets per second on a single queue is limited due to the nature of nfnetlink communication.
- Batching verdict can help but without an efficient improvement.
- Starting Suricata with multiple queue could improve it:

```
suricata -c /etc/suricata/suricata.yaml -q 0 -q 1
```

STAMUS NETWORKS

STAMUS NETWORKS

STAMUS NETWORKS

Stream depth

Attacks characteristic

- In most cases attack is done at start of TCP session
- Generation of requests prior to attack is not common
- Multiple requests are often not even possible on same TCP session

Stream reassembly depth

- Suricata reassemble TCP sessions till `stream.reassembly.depth` bytes.
- Stream is not analyzed once limit is reached

STAMUS NETWORKS

Introducing offloading

Principle

- No need to get packet from kernel after stream depth is reached
- If there is
  - no file store
  - or other operation

Usage

Set `stream.offloading` option to `yes` in suricata config file to offload

STAMUS NETWORKS

STAMUS NETWORKS

Implementation

Suricata update

- Add callback function
- Capture method register itself and provide a callback
- Suricata calls callback when it wants to offload

STAMUS NETWORKS

Implementation

Suricata update

- Add callback function
- Capture method register itself and provide a callback
- Suricata calls callback when it wants to offload

Coded for NFQ

- Update capture register function
- Written callback function
  - Set a mark with respect to a mask on packet
  - Mark is set on packet when issuing the verdict

STAMUS NETWORKS

STAMUS NETWORKS

nftables ruleset

```
table ip filter {
        chain forward {
                type filter hook forward priority 0;
                # usual ruleset
        }

        chain ips {
                type filter hook forward priority 10;
                meta mark set ct mark
                mark 0x00000001 accept
                queue num 0
        }

        chain connmark_save {
                type filter hook forward priority 20;
                ct mark set mark
        }
}
```

STAMUS NETWORKS

Results of iperf3 tests

Local testing

```
eric@ice-age2:~/git/oisf (dev-offloading-v7)$ iperf3 -c  127.0.0.1 -p 6666
Connecting to host 127.0.0.1, port 6666
[  4] local 127.0.0.1 port 38532 connected to 127.0.0.1 port 6666
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  86.3 MBytes   724 Mbits/sec    0   1.03 MBytes
[  4]   1.00-2.00   sec  84.0 MBytes   705 Mbits/sec    0   1.03 MBytes
[  4]   2.00-3.00   sec  84.0 MBytes   705 Mbits/sec    0   1.03 MBytes
[  4]   3.00-4.00   sec  84.8 MBytes   712 Mbits/sec    0   1.03 MBytes
[  4]   4.00-5.00   sec  84.0 MBytes   705 Mbits/sec    0   1.03 MBytes
[  4]   5.00-6.00   sec  84.0 MBytes   705 Mbits/sec    0   1.03 MBytes
[  4]   6.00-7.00   sec  84.0 MBytes   705 Mbits/sec    0   1.03 MBytes
[  4]   7.00-8.00   sec  84.0 MBytes   705 Mbits/sec    0   1.03 MBytes
[  4]   8.00-9.00   sec  84.0 MBytes   705 Mbits/sec    0   1.03 MBytes
[  4]   9.00-10.00  sec  84.2 MBytes   706 Mbits/sec    0   1.03 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   844 MBytes   708 Mbits/sec    0             sender
[  4]   0.00-10.00  sec   842 MBytes   707 Mbits/sec                  receiver
```

STAMUS NETWORKS

Results of iperf3 tests

Local testing

```
eric@ice-age2:~/git/oisf (dev-offloading-v7)$ iperf3 -c  127.0.0.1 -p 6666
Connecting to host 127.0.0.1, port 6666
[  4] local 127.0.0.1 port 38532 connected to 127.0.0.1 port 6666
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  86.3 MBytes   724 Mbits/sec    0   1.03 MBytes
[  4]   1.00-2.00   sec  84.0 MBytes   705 Mbits/sec    0   1.03 MBytes
[  4]   2.00-3.00   sec  84.0 MBytes   705 Mbits/sec    0   1.03 MBytes
[  4]   3.00-4.00   sec  84.8 MBytes   712 Mbits/sec    0   1.03 MBytes
[  4]   4.00-5.00   sec  84.0 MBytes   705 Mbits/sec    0   1.03 MBytes
[  4]   5.00-6.00   sec  84.0 MBytes   705 Mbits/sec    0   1.03 MBytes
[  4]   6.00-7.00   sec  84.0 MBytes   705 Mbits/sec    0   1.03 MBytes
[  4]   7.00-8.00   sec  84.0 MBytes   705 Mbits/sec    0   1.03 MBytes
[  4]   8.00-9.00   sec  84.0 MBytes   705 Mbits/sec    0   1.03 MBytes
[  4]   9.00-10.00  sec  84.2 MBytes   706 Mbits/sec    0   1.03 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   844 MBytes   708 Mbits/sec    0             sender
[  4]   0.00-10.00  sec   842 MBytes   707 Mbits/sec                  receiver
```

<marketing>Local testing with offload is 90 times faster</marketing>

```
eric@ice-age2:~/git/oisf (dev-offloading-v7)$ iperf3 -c  127.0.0.1 -p 6666
Connecting to host 127.0.0.1, port 6666
[  4] local 127.0.0.1 port 38624 connected to 127.0.0.1 port 6666
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  7.42 GBytes  63.7 Gbits/sec    0   1.09 MBytes
[  4]   1.00-2.00   sec  7.04 GBytes  60.5 Gbits/sec    0   1.09 MBytes
[  4]   2.00-3.00   sec  7.06 GBytes  60.7 Gbits/sec    0   1.09 MBytes
[  4]   3.00-4.00   sec  7.30 GBytes  62.7 Gbits/sec    0   1.09 MBytes
[  4]   4.00-5.00   sec  8.35 GBytes  71.7 Gbits/sec    0   1.26 MBytes
[  4]   5.00-6.00   sec  8.45 GBytes  72.6 Gbits/sec    0   1.26 MBytes
[  4]   6.00-7.00   sec  7.22 GBytes  62.0 Gbits/sec    0   1.26 MBytes
[  4]   7.00-8.00   sec  6.99 GBytes  60.0 Gbits/sec    0   1.26 MBytes
[  4]   8.00-9.00   sec  6.98 GBytes  60.0 Gbits/sec    0   1.26 MBytes
[  4]   9.00-10.00  sec  7.05 GBytes  60.5 Gbits/sec    0   1.26 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  73.9 GBytes  63.4 Gbits/sec    0             sender
[  4]   0.00-10.00  sec  73.9 GBytes  63.4 Gbits/sec                  receiver
```

Selective offloading

Ignore some traffic

- Ignore intensive traffic like Netflix
- Can be done independently of stream depth
- Can be done using generic or custom signatures

STAMUS NETWORKS

Selective offloading

Ignore some traffic

- Ignore intensive traffic like Netflix
- Can be done independently of stream depth
- Can be done using generic or custom signatures

The offload keyword

- A new `offload` signature keyword
- Trigger offloading when signature match
- Example of signature

```
alert http any any -> any any (content:"netdevconf.org"; \\
        http_host; offload; sid:6666; rev:1;)
```

STAMUS NETWORKS

STAMUS NETWORKS

Implementation for other captures

Possibilities

- AF_PACKET
- Signaling Openvswitch
- Custom HW
- ...

Constraint

- Method needs to be fast
- It needs to handle
  - Huge amount of flow/items
  - Rapid change rate

STAMUS NETWORKS

STAMUS NETWORKS

Suricata and Linux

- A deep imbrication
- IDS constraint causes some generic features to fail
- Offloading looks promising

More information

- Suricata: `http://www.suricata-ids.org/`
- Netfilter: `http://www.netfilter.org/`
- Stamus Networks: `https://www.stamus-networks.com/`

STAMUS NETWORKS

Questions ?

Contact us

- Éric Leblond: eleblond@stamus-networks.com
- Giuseppe Longo: glongo@stamus-networks.com
- Twitter: @regiteric and @theglongo

- `https://www.stamus-networks.com/`

STAMUS NETWORKS