Office suite and firewall

É. Leblond

Stamus Networks

2016 June 27

STAMUS NETWORKS

Firewall for managers

History

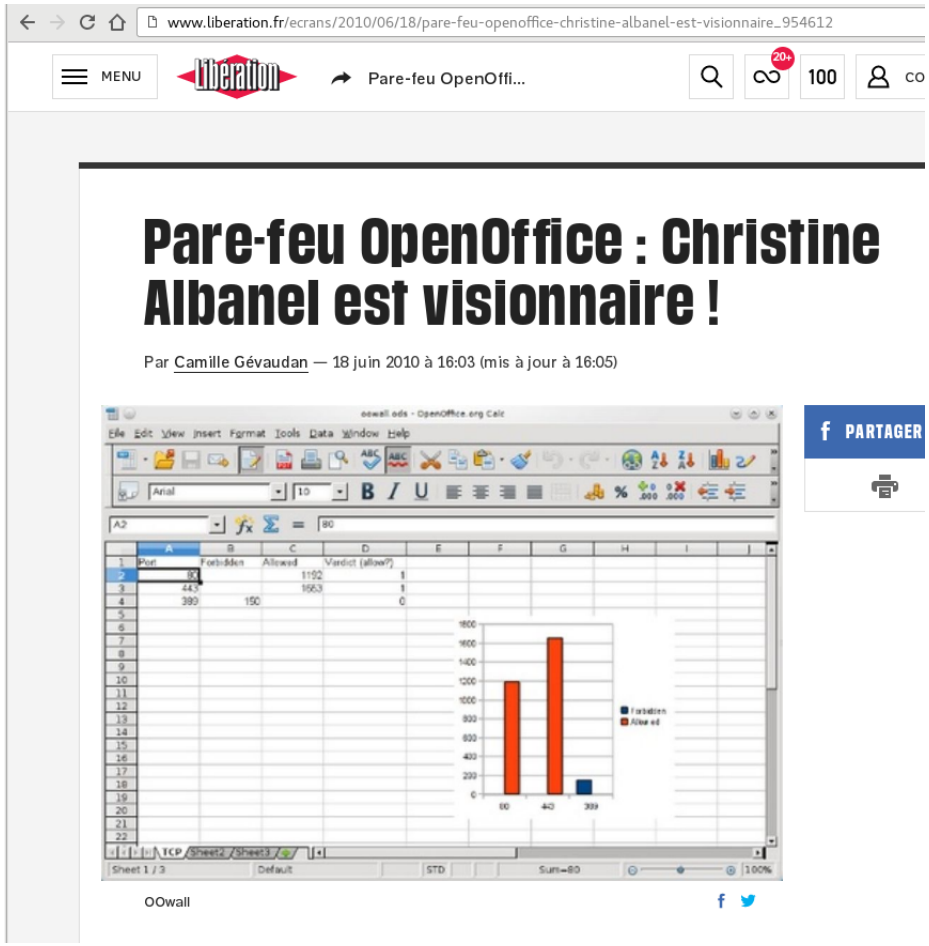
- C. Albanel minister of culture Culture in 2009 spoke about OpenOffice firewall
- Pierre Chifflier did code it in 2010

oowall: a manager compliant firewall

- Edit firewall policy from a spreadsheet
- Graph anything and get live refresh
- Available at `https://github.com/chifflier/oowall`

STAMUS NETWORKS

Demo



Video: `https://www.youtube.com/watch?v=9lxGBadTDCo&feature=youtu.be`

Architecture

Technologies

- Python binding for NFQUEUE
- XMLRPC
- Libreoffice API

Packet path

- Kernel send to queue
- Python script receives it
- Send it via XMLRPC to Libreoffice
- Get result based on spreadsheet
- Python script send it back to kernel

STAMUS NETWORKS

WTF: Word Termination Feature

Objective

- Fight against Word file transfer
- Because it is Office is heavy like hell
- And you even have to pay for it

Method

- Mark packet when a Word file is transferred
- Limit bandwith with Linux QoS

STAMUS NETWORKS

WTF: Waiting Transfer to Finish

STAMUS NETWORKS

Suricata configuration

The rule

```
alert http any any -> any any ( \
  msg: "Microsoft Word upload"; \
  nfq_set_mark:0x1/0x1; \
  filemagic:"Composite Document File V2 Document"; \
  sid:666 ; rev:1;)
```

Running suricata

```
suricata -q 0 -S word.rules
```

STAMUS NETWORKS

Netfilter configuration (1/2)

Queueing packets

```
table inet filter {
        chain forward {
                type filter hook forward priority 0; policy drop;
                ...
        }
        chain ips {
                type filter hook forward priority 10; policy accept;
                tcp dport 80 queue
                tcp sport 80 queue
        }
}
```

Analysing packets

- Suricata needs to get all packets
- Get all packets in both way
- NFQUEUE is a terminal target

Netfilter configuration (2/2)

Propagating the mark

- Mark is set on packet
- We want to mark all packet of a connection
- We need to propagate the mark

Using ct set

```
table inet filter {
        chain prerouting {
                type filter hook prerouting priority -150; policy accept;
                ct mark set mark
        }
        chain ips {
                type filter hook forward priority 10; policy accept;
                tcp dport 80 queue
                tcp sport 80 queue
        }
        chain postrouting {
                type filter hook postrouting priority -150; policy accept;
                meta mark set ct mark
        }
```

One slide of QoS

A diffserv implementation

- Controlling how packets are sent
  - Reordering the queue
  - Introducing delay
  - Dropping packets
- Different algorithm available
  - Queueless: fifo, prio
  - With queue: cbq, htb, ...

HTB example

- Split bandwith in different part
- Assign to part
  - Minimum guarantee bandwith
  - Maximum bandwith
  - Priority

Linux QoS configuration

Setting up QoS tree

```
tc qdisc add dev eth0 root \
        handle 1: htb default 0
tc class add dev eth0 parent 1: \
        classid 1:1 htb \
        rate 1kbps ceil 1kbps
```

Sending marked packets to their fate

```
tc filter add dev eth0 parent 1: \
        protocol ip prio 1 \
        handle 1 fw flowid 1:1
```

STAMUS NETWORKS

Evasion technique

What would you test to avoid this

- Change file extension
- Send compressed file

Filename extension change

- Most likely to happen
- Easy to spot in the IDS

STAMUS NETWORKS

Detecting evasion technique

Detecting the evasion

```
alert http any any -> any any ( \
  msg:"Tricky Microsoft Word upload"; \
  nfq_set_mark:0x2/0x2; \
  fileext:!"doc"; \
  filemagic:"Composite Document File V2 Document"; \
  filestore; \
  sid:667; rev:1;)
```

Being nice with clever people

```
tc class add dev eth0 parent 1: classid 1:2 htb \
        rate 10kbps ceil 10kbps
tc filter add dev eth0 parent 1: protocol ip \
        prio 1 handle 2 fw flowid 1:2
```

Watching the clever ones (1/2)

Watching the clever one from behind a PRISM

- Getting the most information possible about the clevers
- Storing in a pcap file all their trafic for a certain amount of time

STAMUS NETWORKS

Watching the clever ones (1/2)

Watching the clever one from behind a PRISM

- Getting the most information possible about the clevers
- Storing in a pcap file all their trafic for a certain amount of time

Difficulty

- We've got a mark on the connection and we want to keep all trafic
- We need a method to pass from connection to IP

STAMUS NETWORKS

Watching the clever ones (1/2)

Watching the clever one from behind a PRISM

- Getting the most information possible about the clevers
- Storing in a pcap file all their trafic for a certain amount of time

Difficulty

- We've got a mark on the connection and we want to keep all trafic
- We need a method to pass from connection to IP

A possible method: set feature + ulogd

- set allows set handling
- set can be list of IPs with timeout
- we can populate a set
- log all packets from the set to a pcap file with ulogd

Deny On Monitoring

Watch EVE file and respond

- Tail the log file
- Parse JSON message
- React when some motif is found

The code

```
file = open(args.file, 'r')
while 1:
    line = file.readline()
    event = json.loads(line)
    if event['event_type'] == 'alert':
        if event['alert']['signature_id'] == 667:
            call_add(args, event['src_ip'])
```

STAMUS NETWORKS

Watching the clever ones (2/2)

Using DOM to populate the set

```
nft add set inet filter cheaters { type ipv4_addr\; timeout 1h\; }
dom -n filter -s cheaters eve.json
```

Logging marked packets

```
nft add rule inet filter prerouting ip src @cheaters log group 1
nft add rule inet filter prerouting ip dst @cheaters log group 1
```

STAMUS NETWORKS

Ulogd to keep the trace

Ulogd2

- Netfilter logging daemon
- Inputs: NFLOG, NFCT, NFACCT, ...
- Outputs: syslog, file, DB, pcap, ...

STAMUS NETWORKS

Ulogd to keep the trace

Ulogd2

- Netfilter logging daemon
- Inputs: NFLOG, NFCT, NFACCT, ...
- Outputs: syslog, file, DB, pcap, ...

Configuring ulogd

- Ulogd will log packets to a pcap file
- We need to activate a stack in ulogd.conf:

```
plugin="/usr/local/lib/ulogd/ulogd_output_PCAP.so"
stack=log2:NFLOG,base1:BASE,pcap1:PCAP
```

STAMUS NETWORKS

Ulogd to keep the trace

Ulogd2

- Netfilter logging daemon
- Inputs: NFLOG, NFCT, NFACCT, ...
- Outputs: syslog, file, DB, pcap, ...

Configuring ulogd

- Ulogd will log packets to a pcap file
- We need to activate a stack in ulogd.conf:

```
plugin="/usr/local/lib/ulogd/ulogd_output_PCAP.so"
stack=log2:NFLOG,base1:BASE,pcap1:PCAP
```

Starting ulogd

```
ulogd -c ulogd.conf
```

Speeding up free Office suite

NFQ and performance

- Going via NFQ limit bandwidth
- Cost of queueing to userspace
- Even if possible to paralelize

Full bandwidth for free Office suite

- Not sending them to NFQ
- Use kernel Netfilter only

STAMUS NETWORKS

Selective shunting in Suricata

Ignore some traffic

- Ignore intensive traffic like Netflix
- Can be done using generic or custom signatures

STAMUS NETWORKS

Selective shunting in Suricata

Ignore some traffic

- Ignore intensive traffic like Netflix
- Can be done using generic or custom signatures

The offload keyword

- A new `offload` signature keyword
- Trigger offloading when signature match
- offloading OpenDocument:

```
alert http any any -> any any (filemagic:"OpenDocument"; \\
          offload; sid:6666; rev:1;)
alert smtp any any -> any any (filemagic:"OpenDocument"; \\
          offload; sid:6667; rev:1;)
```

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

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

Suricata shunting

Generic options

- Suricata don't inspect packets after stream depth
- Option added to shunt all flows once the limit is reached
- With limitation in case file storage
- Encrypted flows can be shunt too

Availability

- Offloading/shunting should be part of Suricata 3.2
- And extended to AF_PACKET via EBPF usage

STAMUS NETWORKS

Conclusion

Office suite

- You can help Microsoft office suck more
- This slides have been made using LaTeX beamer

More information

- Suricata: `http://www.suricata-ids.org/`
- Suricon: `http://suricon.net/`
- Suricata developer training: Paris, 12-16 Septembre
  `https://goo.gl/9tYbWP`
- Stamus Networks : `https://www.stamus-networks.com/`

STAMUS NETWORKS