Full Packet Capture for the Masses
<profile>
  <name>Xavier Mertens</name>
  <nick>xme</nick>
  <jobs>
    <day>Security Freelancer</day>
    <night>Blogger, ISC Handler, Hacker</night>
  </jobs>
<![CDATA[
    https://xavier.mertens.consulting
    https://blog.rootshell.be
    https://isc.sans.edu
    https://www.brucon.org
  ]]>
Who Never Had This Issue?

“Who’s talking?”

Knowing who’s talking on your network is a key requirement when you have to investigate a security incident.
Logging S*cks...

<table>
<thead>
<tr>
<th>i</th>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6/7/18 9:00:03.000 PM</td>
<td>31.18.50.35 - - [07/Jun/2018:12:00:03 -0700] &quot;POST //xmlrpc.php HTTP/1.1&quot; 418 731 &quot;-&quot; .36&quot; host = rick source = /var/tmp/rick/access.log.2018-06-07 sourcetype = access_combined</td>
</tr>
<tr>
<td>5</td>
<td>6/7/18 8:59:01.000 PM</td>
<td>31.18.50.35 - - [07/Jul/2018:11:59:01 -0700] &quot;POST //xmlrpc.php HTTP/1.1&quot; 200 594 &quot;-&quot; .36&quot; host = rick source = /var/tmp/rick/access.log.2018-06-07 sourcetype = access_combined</td>
</tr>
</tbody>
</table>
## L3 or L7?

<table>
<thead>
<tr>
<th>Layer 3</th>
<th>timestamp:src_ip:src_port:dst_ip:dst_port</th>
<th>Firewall Logs, Netflow, Basic Packet Capture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 7</td>
<td>timestamp:src_ip:src_port:dst_ip:dst_port + headers, payloads</td>
<td>“NG” Firewall Logs, Full Packet Capture</td>
</tr>
<tr>
<td></td>
<td>Pro</td>
<td>Con</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td><strong>Flows</strong></td>
<td>Easy setup</td>
<td>“Lack of visibility”</td>
</tr>
<tr>
<td></td>
<td>Optimized storage</td>
<td></td>
</tr>
<tr>
<td><strong>FPC</strong></td>
<td>“Full view”</td>
<td>Retention (storage)</td>
</tr>
<tr>
<td></td>
<td>Extract artefacts</td>
<td>Privacy</td>
</tr>
<tr>
<td></td>
<td>Replay</td>
<td>Performance</td>
</tr>
<tr>
<td></td>
<td>Evidences</td>
<td>Sensors required</td>
</tr>
</tbody>
</table>
⚠️ Warning ⚠️

I DON'T KNOW HOW TO PUT THIS BUT...

GDPR IS KIND OF A BIG DEAL
Full Packet Capture

```
/usr/sbin/tcpdump \
  -n -Z nobody \
  -i eth0 \ 
  -s 0 \ 
  -C 500 \ 
  -G 3600 -W 48 \ 
  -w /data/dump-%Y%m%d%H%M.pcap \ 
not port 22 and not port 1194
```
Modern Infrastructure

- Virtualization
- Local Servers
- Cloud
- VPS
- Co-location
Solution?

Collect data from multiple locations and centralise all data for better retention
Requirements

Must be free
Easy to deploy on different OS
Can be deployed on devices not directly connected to the central repository (easy data transfer)
Moloch

Moloch is (IMHO) the best complete FPC framework. Developed by Andy Wick & Eoin Miller (AOL CERT). Powerful, Scalable.
Moloch

Components:

• Capturer (online / offline)
• DB (ElasticSearch)
• Viewer (Web GUI)

Multiple architecture available (*)

(*) https://github.com/aol/moloch/wiki/Architecture
Moloch
Docker

Easy way to deploy software across multiple platforms
⚠️ Warning ⚠️

WHAT IF I TOLD YOU

DOCKER CONTAINERS ARE NOT MAGICAL VIRTUAL MACHINES

tor.net
Got an Idea… Google it!

GIAC Gold Paper by Mauricio Espinosa Gomez

1st Approach

<table>
<thead>
<tr>
<th>Pro</th>
<th>Con</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full automated deployment via Puppet</td>
<td>Multiple Moloch instances deployed</td>
</tr>
<tr>
<td>Multiple nodes in Moloch</td>
<td>ElasticSearch must be reachable from sensors</td>
</tr>
<tr>
<td>Real-time indexing</td>
<td></td>
</tr>
<tr>
<td>Good for internal networks</td>
<td></td>
</tr>
</tbody>
</table>
My Approach

Deploy a very small Docker container as sensor (only based on tcpdump & scp)
PCAP files are synchronised with Moloch at regular intervals
My Approach
My Approach

- Elastic Search
- Moloch
- Handler
Moloch Server

There exist good Docker containers but without good data persistence support!

Forked one(*) and added some stuff:

- persistence: config & raw data
- automatic indexing of PCAP files (no live mode)

(*) https://hub.docker.com/r/danielguerra/docker-moloch/
Moloch Server

/data/moloch/bin/moloch-capture \
  -m \ 
  -R /data/pcap \ 
  --copy \ 
  --delete \ 
  --flush
version: "2"

services:
  moloch_elasticsearch:
    image: elasticsearch:5.2.2-alpine
    restart: always
    hostname: moloch_elasticsearch
    container_name: moloch_elasticsearch
    volumes:
      - /etc/localtime:/etc/localtime:ro
      - /data/moloch/elasticsearch:/usr/share/elasticsearch/data
    network_mode: bridge

  moloch_capture:
    build: ./docker-moloch
    image: danielguerra/docker-moloch
    restart: always
    hostname: moloch_capture
    container_name: moloch_capture
    depends_on:
      - moloch_elasticsearch
    links:
      - moloch_elasticsearch:elasticsearch
    volumes:
      - /etc/localtime:/etc/localtime:ro
      - /data/moloch/core/etc:/data/moloch/etc:ro
      - /data/moloch/core/raw:/data/moloch/raw:ro
      - /data/tcpdump:/data/pcap:rw
    ports:
      - '8005:8005'
    network_mode: bridge
Moloch Server

docker-compose up

https://moloch:8005
Sensor

Run a tcpdump to dump packets to files
Scp files to moloch
Sensor Deployment

# git clone \n  https://github.com/xme/moloch/sensor.git
# cd sensor
# docker build -t sensor .
Sensor Deployment

PCAP_INTERFACE=eth0
PCAP_CAPTURE_SIZE=0
PCAP_FILE_SIZE=50
PCAP_FILE_ROTATE=100
PCAP_BPF_FILTER=not port 22 and not port 1194
PCAP_SENSOR_NAME=boogey
SCP_TARGET=xavier@moloch:/data/tcpdump
SCP_ARGUMENTS=-P 65522 -o
StrictHostKeyChecking=no
# Sensor Kick Off

```
# docker run -d --rm --env-file=env.txt --net=host --name sensor sensor1
Please use this key to allow PCAP files transfer via scp:
--- Cut Here ---
ssh-rsa AAAAB3NzaC1yc2EAAAADQABAAACAAQDdBKWh43I75 ... MUKOKMyQ== root@sensor1
--- Cut Here ---
2018-01-12 09:55:10,026 CRIT Supervisor running as root (no user in config file)
2018-01-12 09:55:10,034 INFO supervisord started with pid 14
2018-01-12 09:55:11,037 INFO spawned: 'pcap_cron' with pid 17
2018-01-12 09:55:11,039 INFO spawned: 'pcap_tcapdump' with pid 18
2018-01-12 09:55:12,111 INFO success: pcap_cron entered RUNNING state, process has
stayed up for > than 1 seconds (startsecs)
2018-01-12 09:55:12,112 INFO success: pcap_tcapdump entered RUNNING state, process
has stayed up for > than 1 seconds (startsecs)
```
--net=host

⚠️ WARNING ⚠️
To allow the container access to the interfaces, docker network isolation is disabled
Tips

Use BPF filters to reduce the noise!

Moloch has an interesting feature:

dontSaveBPFs=port 22:10;port 65522:10;port 65523:10;port 1194:10
Bug?\textsuperscript{^W}Tip!

Do NOT use ‘any’ interface in the tcpdump config!

$ file *.pcap
cooked_sample.pcap: tcpdump capture file (little-endian) - version 2.4 (Linux "cooked", capture length 262144)
sample.pcap: tcpdump capture file (little-endian) - version 2.4 (Ethernet, capture length 262144)
So?

<table>
<thead>
<tr>
<th>Pro</th>
<th>Con</th>
</tr>
</thead>
<tbody>
<tr>
<td>No footprint on the sensor</td>
<td>Not realtime processing</td>
</tr>
<tr>
<td>Runs on any system</td>
<td>Small risk of broken flows</td>
</tr>
<tr>
<td>SSH transfer is safe</td>
<td>Cannot search packets based on the node</td>
</tr>
<tr>
<td>Easy to tune / adapt to your $ENV</td>
<td></td>
</tr>
</tbody>
</table>
Wanna Test?

https://github.com/xme/fpc/
10:55:17.578190 00:00:00:00:00:00 > 00:00:00:00:00:00, ethertype IPv4 (0x0800),
length 77: 127.0.0.1.38048 > 127.0.0.1.7777: Flags [P.], seq 1:12, ack 1, win 342, options [nop,nop,TS val 1437796971 ecr 1437795587], length 11
 0x0000:  4500 003f 189c 4000 4006 241b 7f00 0001  E..?..@.@.$....
 0x0010:  7f00 0001 94a0 1e61 97cd 1d9a b8d8 37b8  .......a......7.
 0x0020:  8018 0156 fe33 0000 0101 080a 55b3 0a6b  ...V.3.......U..k
 0x0030:  55b3 0503 5468 616e 6b20 596f 7521 0a    U...Thank.You!

@xme | xavier@rootshell.be