

Full Packet Capture for the Masses



” 🍷 🥖 🧀 ” Edition - Pass The Salt 2018 - Xavier Mertens

```
<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
  ]]>
</profile>
```

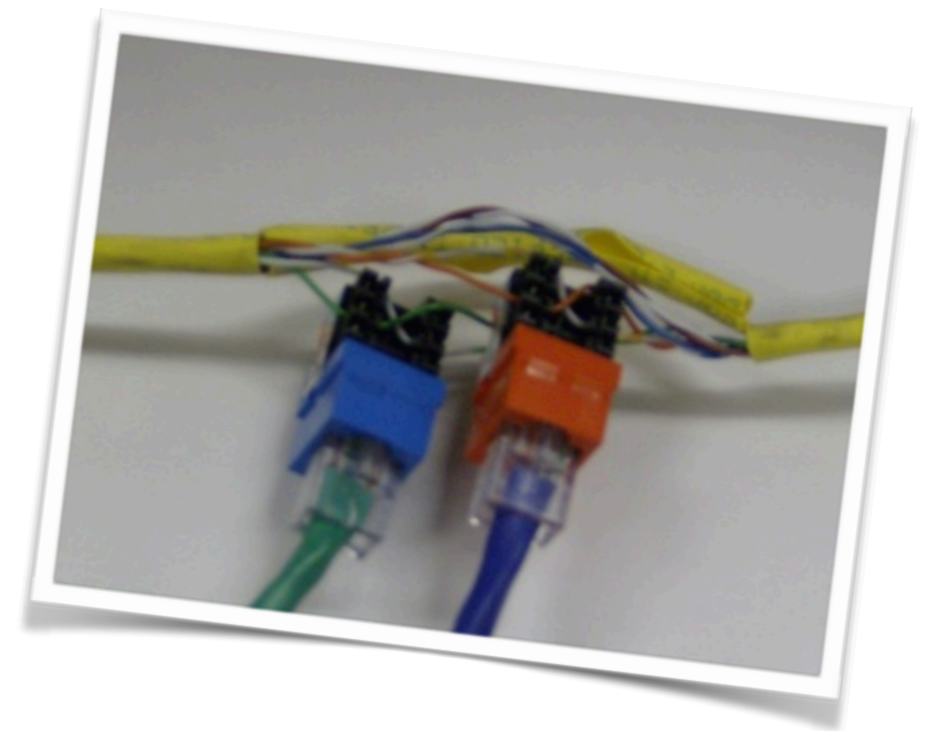


Follow
me!

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...

i	Time	Event
>	6/7/18 9:00:03.000 PM	31.18.50.35 - - [07/Jun/2018:12:00:03 -0700] "POST //xmlrpc.php HTTP/1.1" 418 731 "-" .36" host = rick source = /var/tmp/rick/access.log.2018-06-07 sourcetype = access_combined
>	6/7/18 8:59:47.000 PM	31.18.50.35 - - [07/Jun/2018:11:59:47 -0700] "POST //xmlrpc.php HTTP/1.1" 200 594 "-" .36" host = rick source = /var/tmp/rick/access.log.2018-06-07 sourcetype = access_combined
>	6/7/18 8:59:33.000 PM	31.18.50.35 - - [07/Jun/2018:11:59:33 -0700] "POST //xmlrpc.php HTTP/1.1" 200 594 "-" .36" host = rick source = /var/tmp/rick/access.log.2018-06-07 sourcetype = access_combined
>	6/7/18 8:59:16.000 PM	31.18.50.35 - - [07/Jun/2018:11:59:16 -0700] "POST //xmlrpc.php HTTP/1.1" 200 594 "-" .36" host = rick source = /var/tmp/rick/access.log.2018-06-07 sourcetype = access_combined
>	6/7/18 8:59:01.000 PM	31.18.50.35 - - [07/Jun/2018:11:59:01 -0700] "POST //xmlrpc.php HTTP/1.1" 200 594 "-" .36" host = rick source = /var/tmp/rick/access.log.2018-06-07 sourcetype = access_combined
>	6/7/18 8:58:50.000 PM	31.18.50.35 - - [07/Jun/2018:11:58:50 -0700] "POST //xmlrpc.php HTTP/1.1" 200 594 "-" .36" host = rick source = /var/tmp/rick/access.log.2018-06-07 sourcetype = access_combined

L3 or L7?

Layer 3	timestamp:src_ip:src_port:dst_ip:dst_port	Firewall Logs, Netflow, Basic Packet Capture
Layer 7	timestamp:src_ip:src_port:dst_ip:dst_port + headers, payloads	“NG” Firewall Logs, Full Packet Capture

L3 or L7?

	Pro	Con
Flows	Easy setup Optimized storage	“Lack of visibility”
FPC	“Full view” Extract artefacts Replay Evidences	Retention (storage) Privacy Performance Sensors required

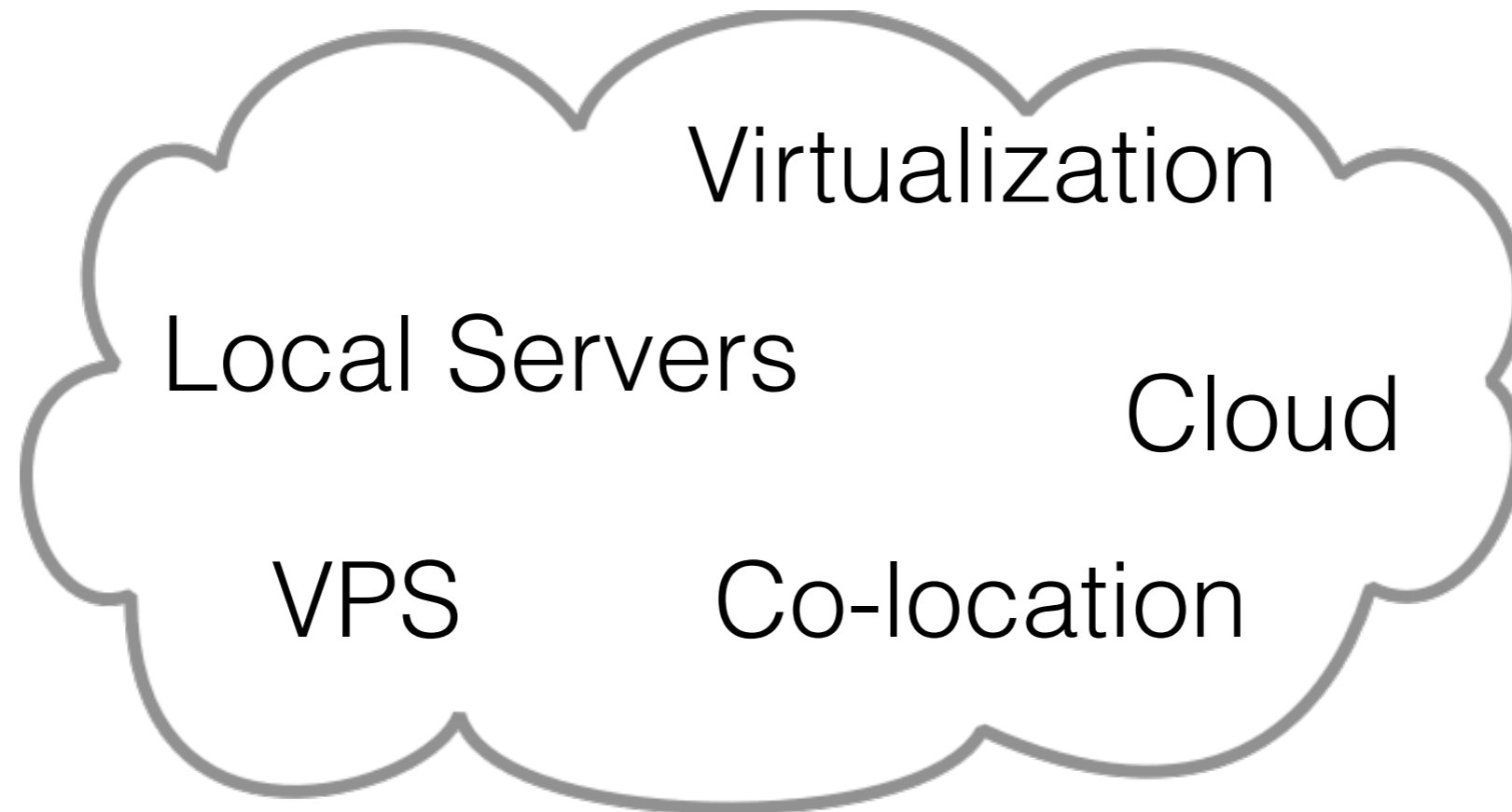
⚠️ Warning ⚠️



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



Solution?

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

Requirements

Must be ~~free~~ open

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>

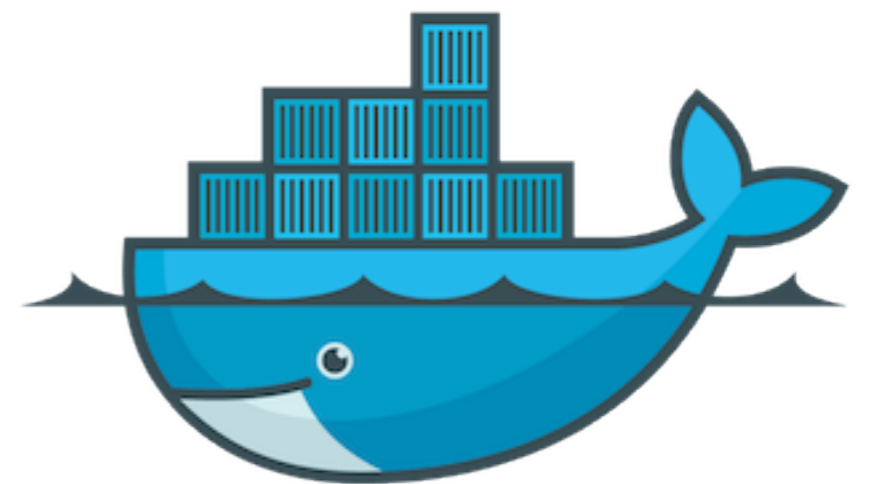
Moloch

The screenshot displays the Moloch web interface for viewing network sessions. The top navigation bar includes options like Sessions, SPI View, SPI Graph, Connections, Files, Stats, Settings, and Users. The current view is 'Sessions', showing a time range from 2018/01/13 12:50:20 to 2018/01/13 13:50:20. A bar chart shows session activity over time, and a world map highlights the geographic distribution of sessions. Below these is a table of session details.

	Start Time	Stop Time	Src IP / Country	Src Port	Dst IP / Country	Dst Port	Packets	Databytes / Bytes	Moloch Node	Info	Tags
+	2018/01/13 12:51:21	2018/01/13 12:51:21	fe80:0:0:0:ec4:7aff:fe c6:2f0c	546	ff02:0:0:0:0:1:2	547	1	86 / 94	moloch_capture		
+	2018/01/13 12:51:21	2018/01/13 12:51:21	fe80:0:0:0:ec4:7aff:fe c6:2eff	546	ff02:0:0:0:0:1:2	547	1	86 / 94	moloch_capture		
+	2018/01/13 12:51:21	2018/01/13 12:51:21	fe80:0:0:0:ec4:7aff:fe c5:c796	546	ff02:0:0:0:0:1:2	547	1	86 / 94	moloch_capture		
+	2018/01/13 12:51:22	2018/01/13 12:51:22	fe80:0:0:0:ec4:7aff:fe c5:c7b0	546	ff02:0:0:0:0:1:2	547	1	86 / 94	moloch_capture		
+	2018/01/13 12:51:22	2018/01/13 12:51:22	201.26.25.125 BRA	52196	193.200.43.13 FRA	23	3	0 / 198	moloch_capture		
+	2018/01/13 12:51:22	2018/01/13 12:51:43	201.26.25.125 BRA	52220	193.200.43.13 FRA	23	17	100 / 1,238	moloch_capture		
+	2018/01/13 12:51:22	2018/01/13 12:51:22	fe80:0:0:0:ec4:7aff:fe c6:2f0b	546	ff02:0:0:0:0:1:2	547	1	86 / 94	moloch_capture		
+	2018/01/13 12:51:23	2018/01/13 12:51:23	109.248.9.15 RUS	49889	163.172.30.221 GBR	9002	1	0 / 60	moloch_capture		
+	2018/01/13 12:51:23	2018/01/13 12:51:23	fe80:0:0:0:ec4:7aff:fe c6:2e0d	546	ff02:0:0:0:0:1:2	547	1	86 / 94	moloch_capture		
+	2018/01/13 12:51:23	2018/01/13 12:51:24	191.113.82.119 CHL	39380	193.200.43.13 FRA	23	3	0 / 198	moloch_capture		
+	2018/01/13 12:51:23	2018/01/13 12:51:45	191.113.82.119 CHL	39416	193.200.43.13 FRA	23	17	101 / 1,239	moloch_capture		
+	2018/01/13 12:51:25	2018/01/13 12:51:25	fe80:0:0:0:ec4:7aff:fe c6:2ea2	546	ff02:0:0:0:0:1:2	547	1	86 / 94	moloch_capture		
+	2018/01/13 12:51:25	2018/01/13 12:51:25	fe80:0:0:0:ec4:7aff:fe c6:2fc2	546	ff02:0:0:0:0:1:2	547	1	86 / 94	moloch_capture		

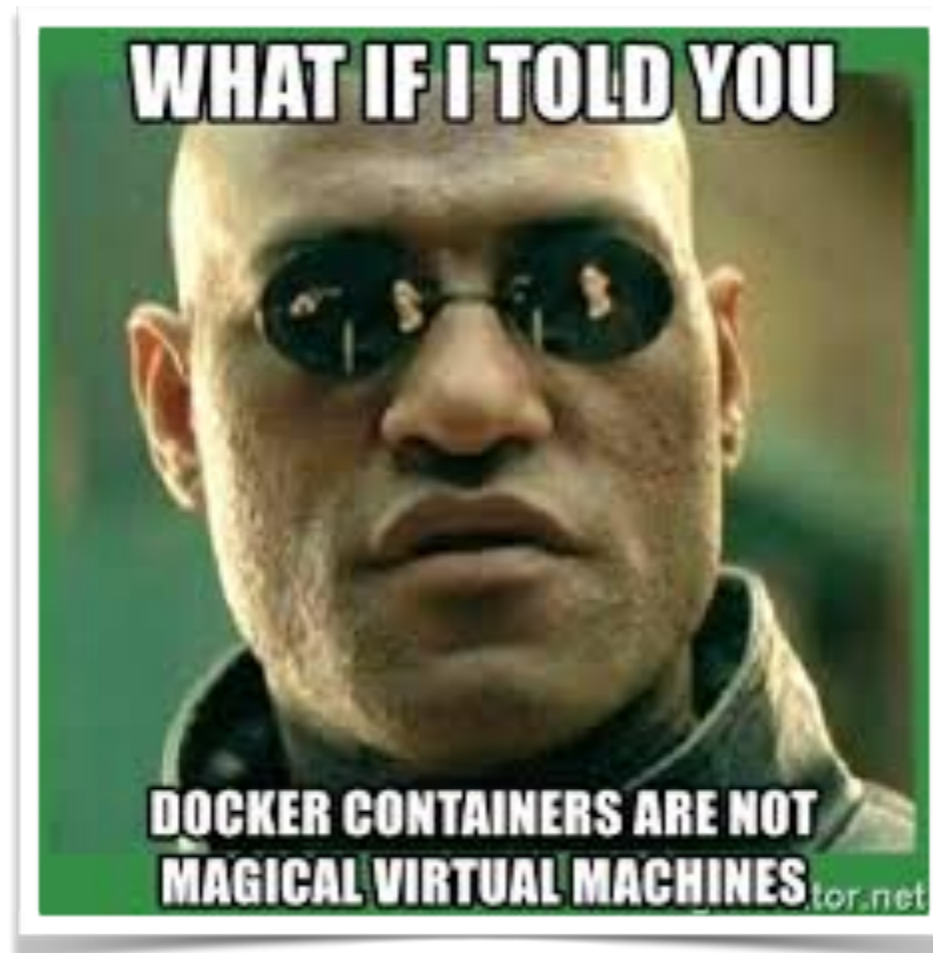
Docker

Easy way to deploy software across multiple platforms



docker

⚠ Warning ⚠



Got an Idea... Google it!

GIAC Gold Paper by Mauricio Espinosa Gomez

<https://www.sans.org/reading-room/whitepapers/cloud/full-packet-capture-infrastructure-based-docker-containers-36977>

1st Approach

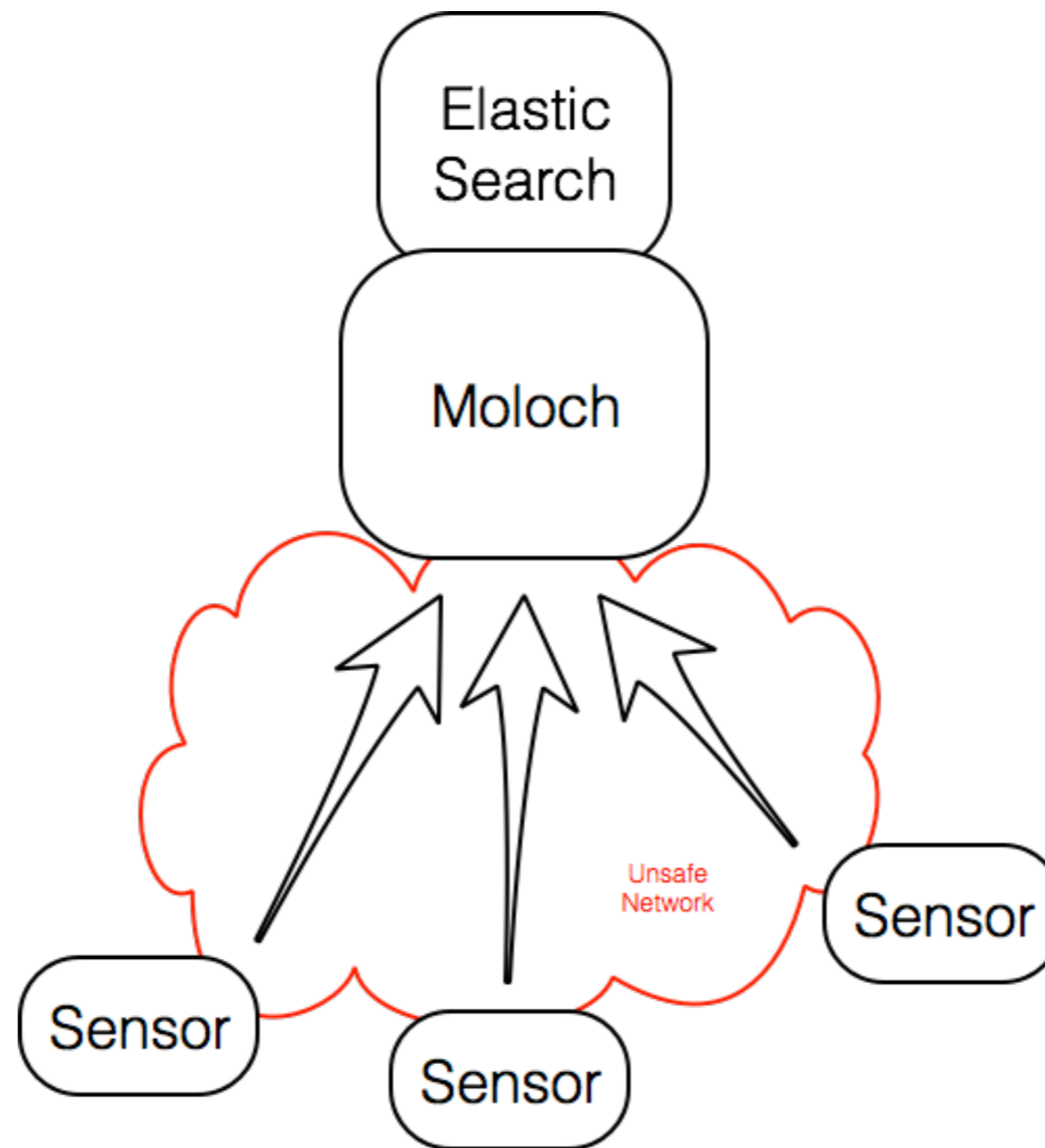
Pro	Con
Full automated deployment via Puppet Multiple nodes in Moloch Real-time indexing Good for internal networks	Multiple Moloch instances deployed ElasticSearch must be reachable from sensors

My Approach

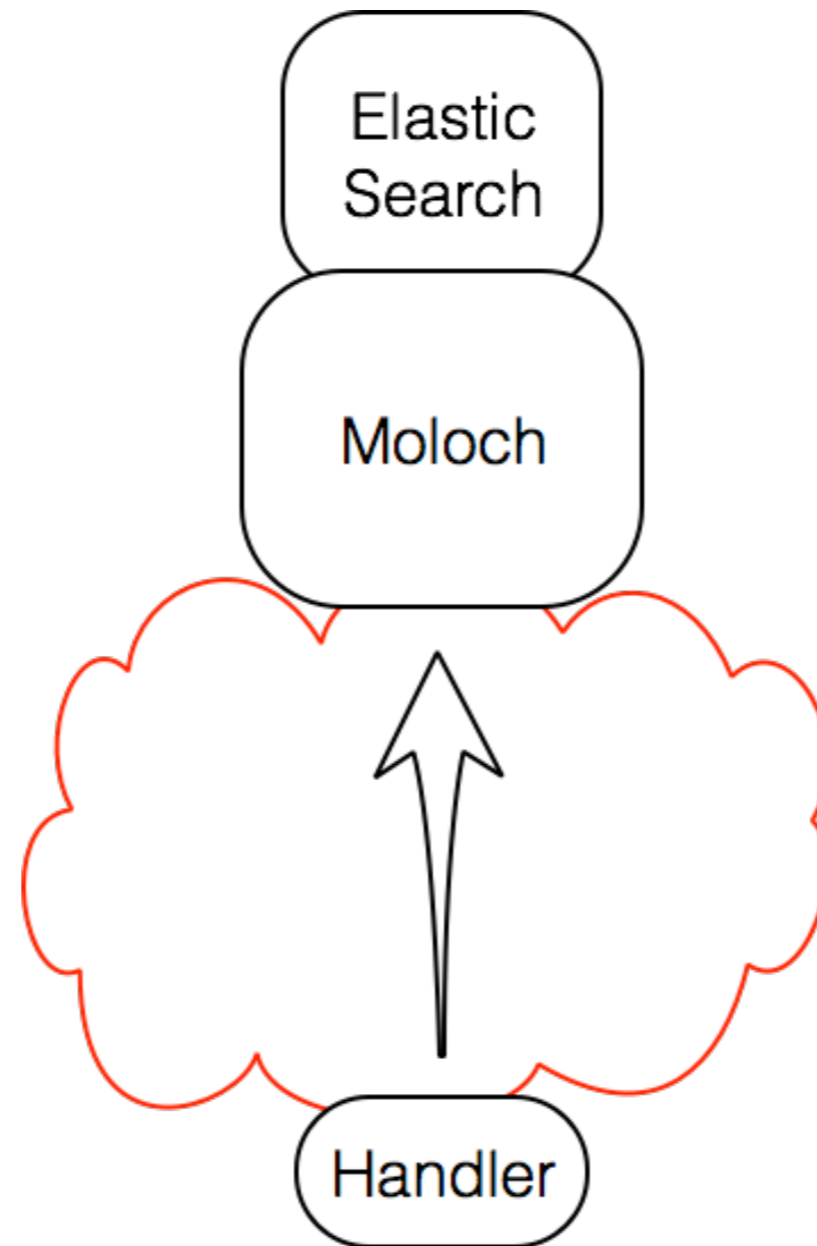
Deploy a very small Docker container as sensor
(only based on tcpdump & scp)

PCAP files are synchronised with Moloch at regular intervals

My Approach



My Approach



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
```

Moloch Server

```
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:rw
```

```
      - /data/moloch/core/raw:/data/moloch/raw:rw
```

```
      - /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 \  
  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 transfert via scp:
--- Cut Here ---
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQDdBKW+43IJ5 ... 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_tcpdump' 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_tcpdump 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? ^WTip!

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?

Pro	Con
<p>No footprint on the sensor Runs on any system SSH transfer is safe Easy to tune / adapt to your \$ENV</p>	<p>Not realtime processing Small risk of broken flows Cannot search packets based on the node</p>

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