Graduation project

By: Alaa Kayed,

Abdelkareem Daghlas,

Rashid Abo Shamat

Supervised by:

Ahmad Awad

OS Fingerprinting Using Honeypot

Table Of Contents

01

Introduction

OS Fingerprinting

02

Problem

The motivation

03

Related works

DNS Traffic Analysis.
HTTP user agent.
Active OS fingerprinting tool.

04

Proposed Approach

Framework
Tools: T-pot, P0f, Filtration Algorithm, Splunk Configuration

05

Analysis & Results

Results of:
Framework
Filtration
Kibana
Splunk

06

Conclusion

Future Work

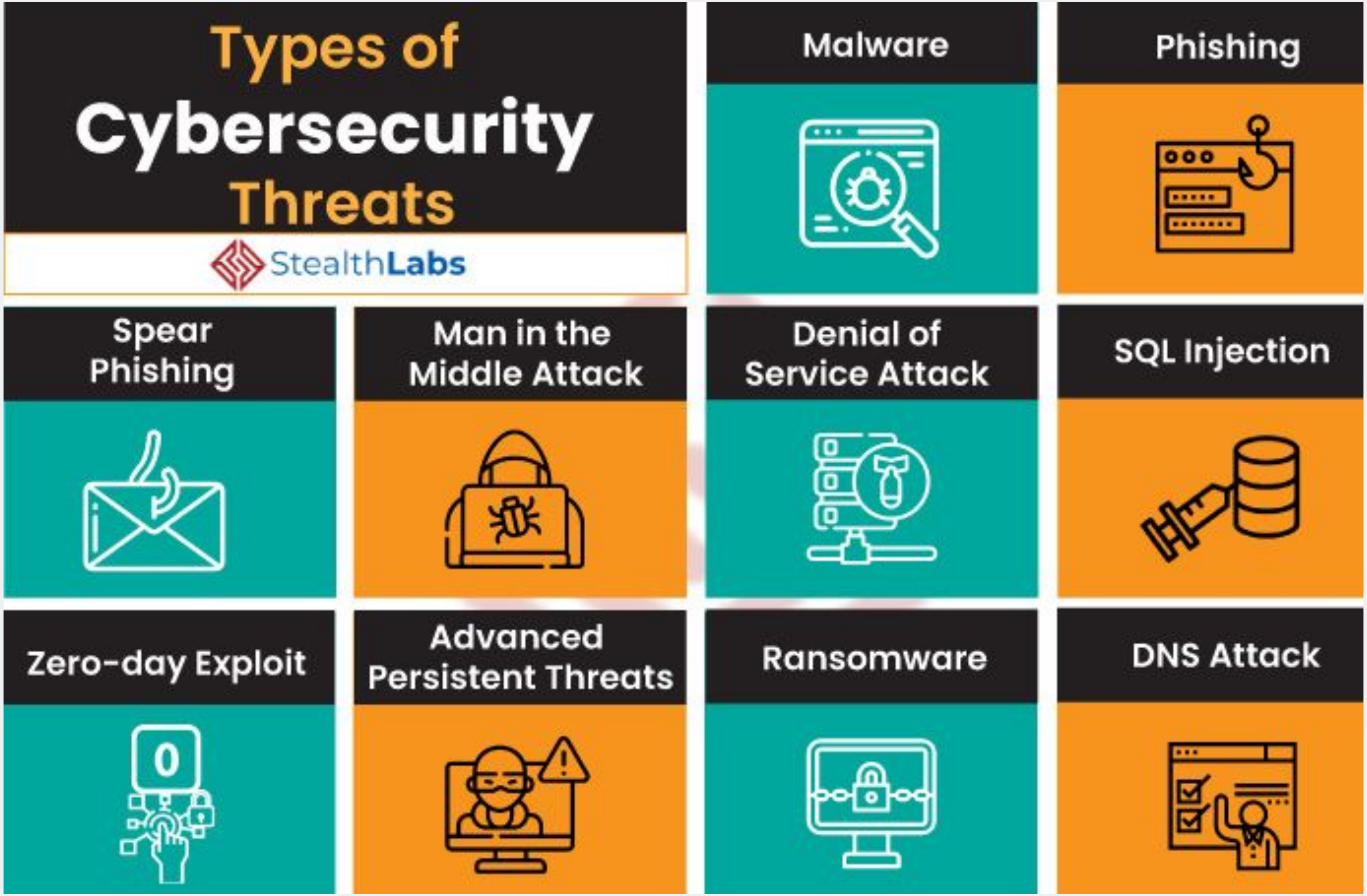
Introduction

OS Fingerprinting

OS Fingerprinting: detection of the operating system of an end-host by analyzing packets.

It is used by security professionals and hackers for mapping remote networks and determining which vulnerabilities might be present to exploit.

Motivation

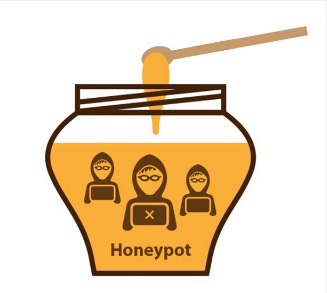


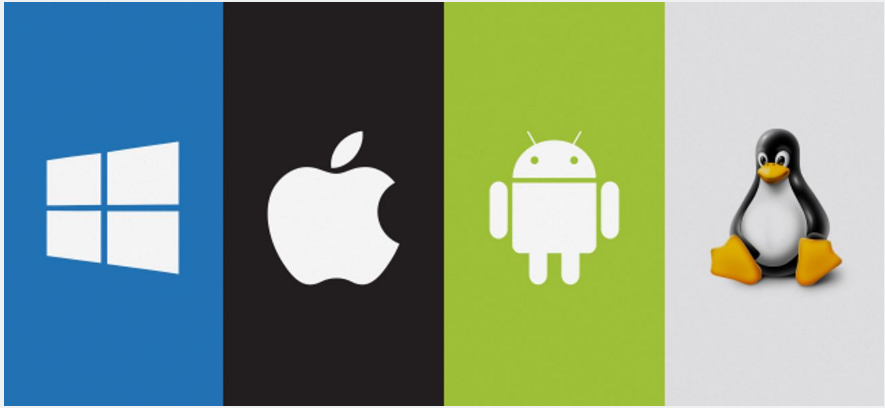
Types of Cybersecurity Threats

StealthLabs

- Malware
- Phishing
- Spear Phishing
- Man in the Middle Attack
- Denial of Service Attack
- SQL Injection
- Zero-day Exploit
- Advanced Persistent Threats
- Ransomware
- DNS Attack

Motivation





OS Fingerprinting in Cyber Security

Table Of Contents

01

Introduction

OS Fingerprinting

02

Problem

Why OS Fingerprinting
The motivation

03

Related works

DNS Traffic Analysis.
HTTP user agent.
Active OS fingerprinting tool.

04

Proposed Approach

Framework
Tools: T-pot, P0f, Filtration Algorithm, Splunk Configuration

05

Analysis & Results

Results of:
Framework
Filtration
Kibana
Splunk

06

Conclusion

Future Work

Related works

01

Using DNS Traffic Analysis

Done only on Android OS

(2016)

02

Using HTTP User agent

Less results than TCP\IP Header analysing technique

(2018)

03

Using active OS fingerprinting tool

Nmap being used in the fingerprinting

(2020)

Table Of Contents

01

Introduction

OS Fingerprinting

02

Problem

Why OS Fingerprinting
The motivation

03

Related works

DNS Traffic Analysis.
HTTP user agent.
Active OS fingerprinting tool.

04

Proposed Approach

Framework
Tools: T-pot, P0f, Filtration Algorithm, Splunk Configuration

05

Analysis & Results

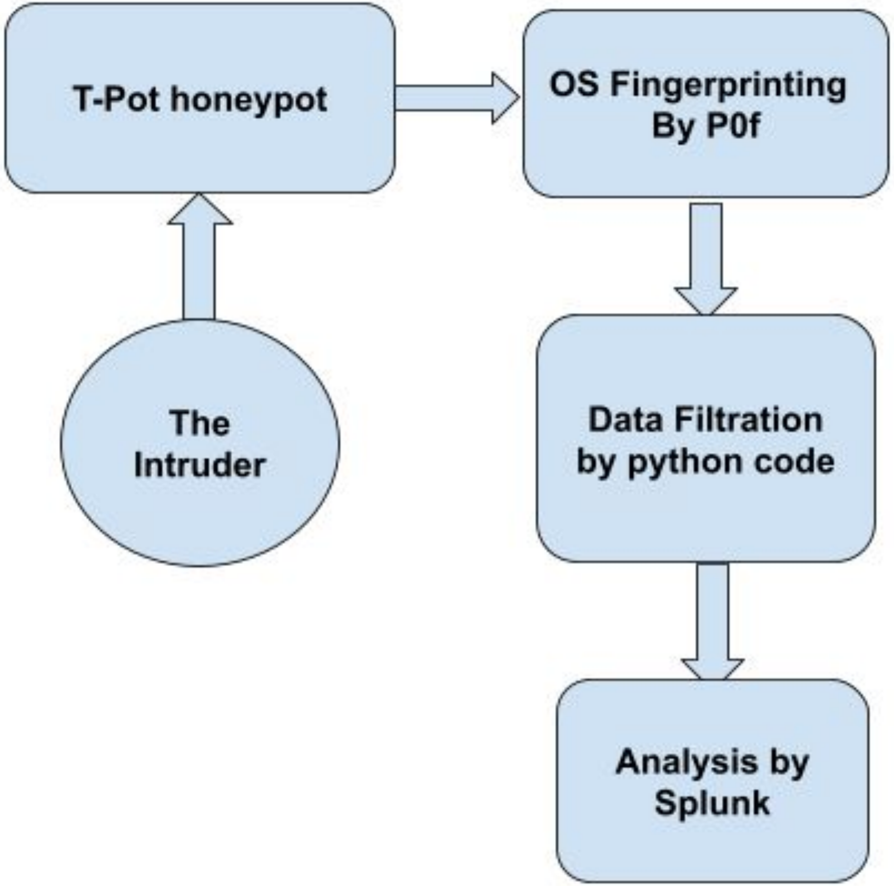
Results of:
Framework
Filtration
Kibana
Splunk

06

Conclusion

Future Work

The Framework



The Intruder

T-Pot honeypot

OS Fingerprinting By P0f

Data Filtration by python code

Analysis by Splunk

The framework

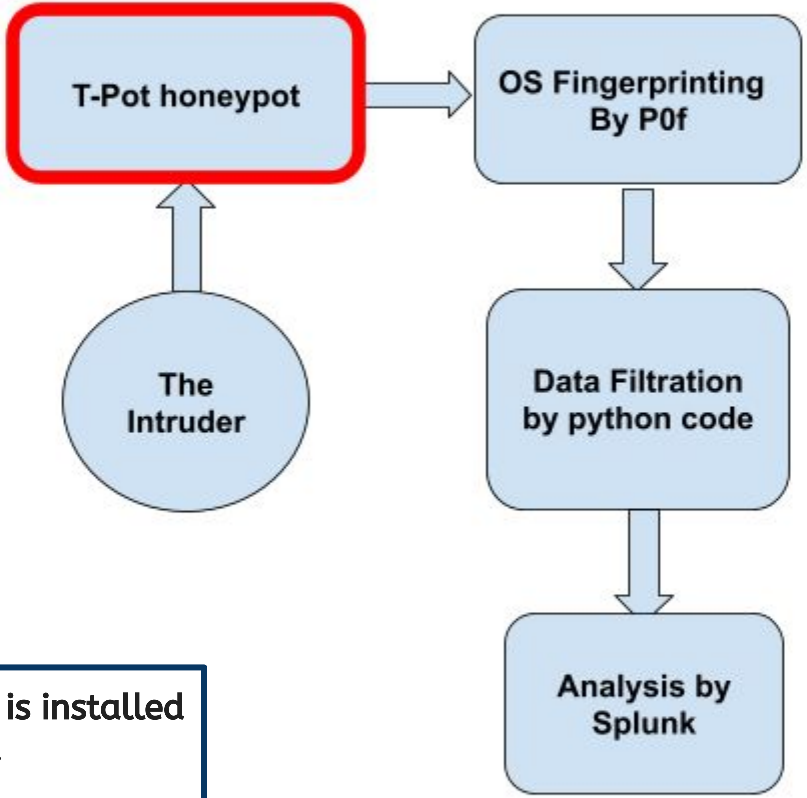
1. T-pot is multi-honeypot platform.

2. T-pot will take the form of an OS-system, to it will make a better decoy.

3. T-pot is easy to use and understand.

4. T-pot is updated to this day and it is not buggy neither wants a lot of maintenance.

5. T-pot has many unique tools like Cockpit, Elasticsearch, Kibana and even more.



T-pot as a honeypot is installed on a Stable linux OS.

- Debian OS 10

The framework

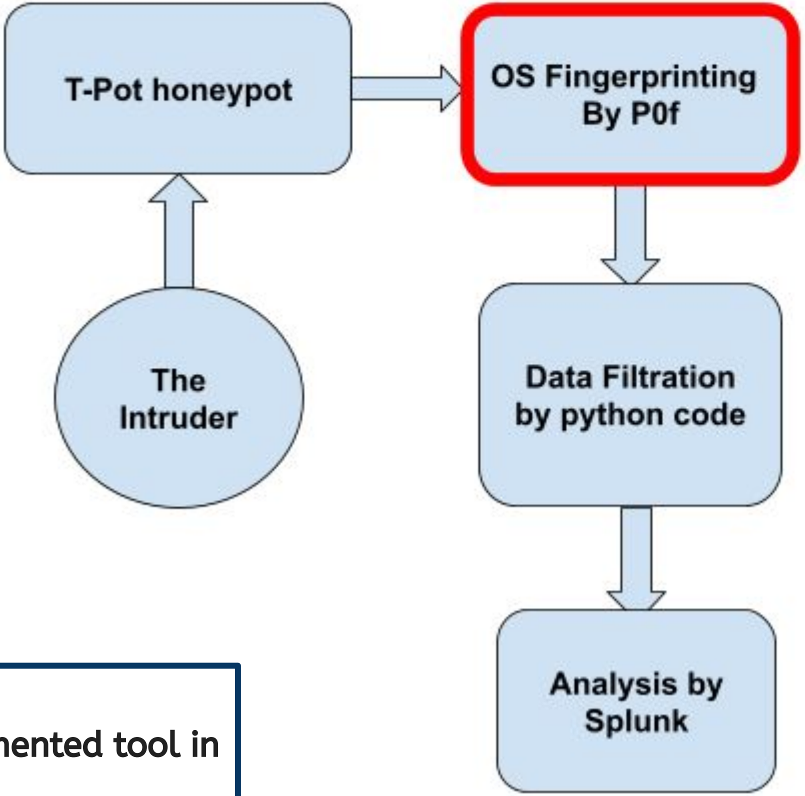
P0f : passive OS fingerprinting tool,
It identifies the OS of the machine in our system.

Other famous tools compared with P0f:

Ettercap: Active and Passive tool.

Satori: non-updated since 2014.

Zardax: this tool only shows the OS as a result.



P0f is a an implemented tool in T-pot

How does p0f work?

p0f analyses the packet fields that received from the network interface.

The TCP/IP packets fields are:

1. TTL (time to live)
2. Packet size
3. Windows size
4. Bit flag

It will recognize the OS, type of connection, the external IP and other info.

Data Filtration

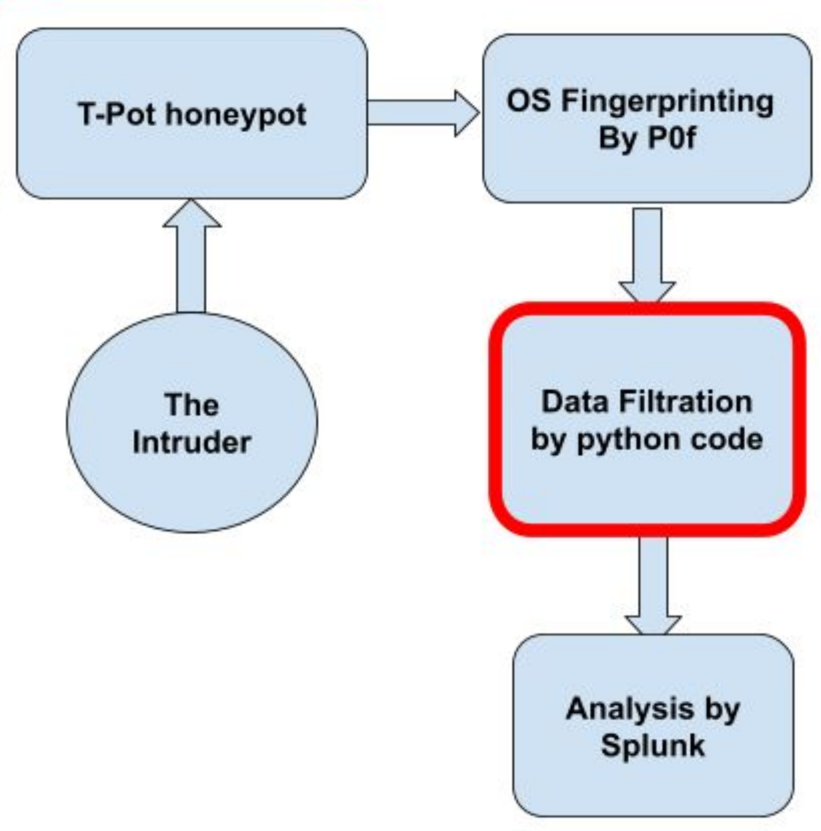
We created a filtration algorithm (Python).

Why filtering the data?

- Huge number of packets generated by P0f.
- Data is heavily duplicated.
- The analysis cost is enormous.

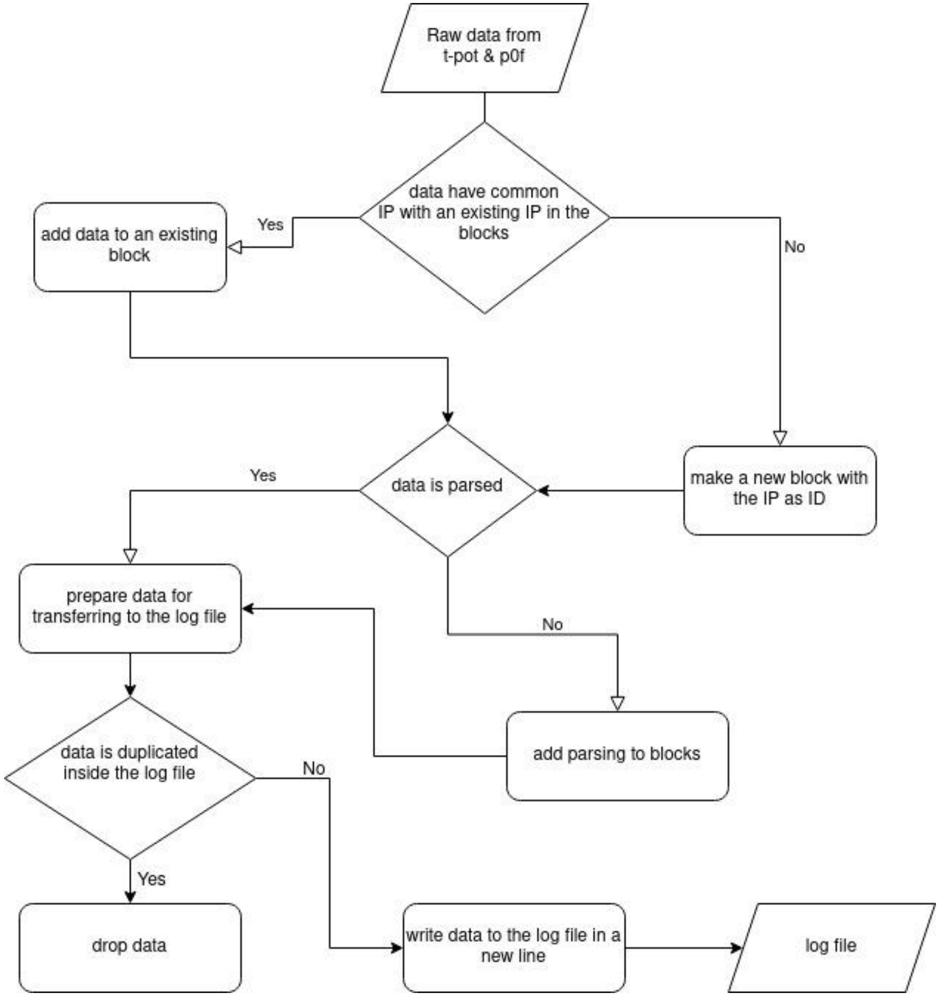
Advantages of filtration?

- Make the data easier for reading and logging.
- Non-duplicated results.
- Decrease the cost of analysis.



The framework

Filtration Algorithm



Raw data from t-pot & p0f
data have common IP with an existing IP in the blocks
Yes
add data to an existing block
No
make a new block with the IP as ID
Yes
data is parsed
prepare data for transferring to the log file
No
add parsing to blocks
data is duplicated inside the log file
No
Yes
drop data
write data to the log file in a new line
log file

Data Analysis

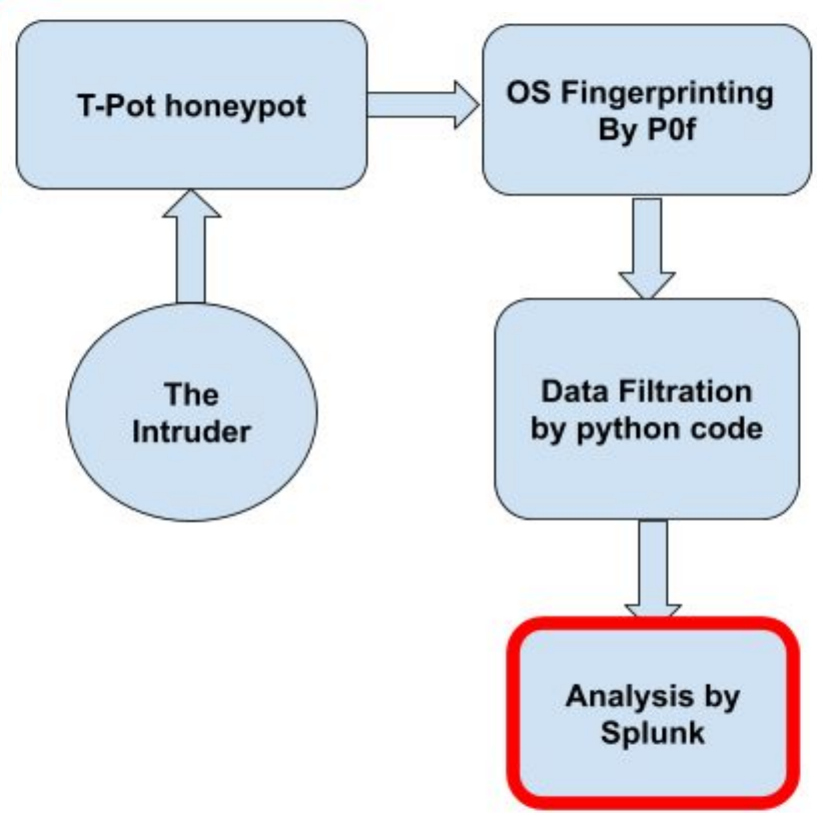
Analysis is done by Splunk.

Splunk is a SEIM tool for:

- Search.
- Data Analysation.
- Data Visualization.
- Indexes log file.

Why Splunk?

- Not all users have access to T-pot.
- Generating reports and statics.



Tools that was used in the framework

- T-Pot
- Cockpit, Performance monitoring.
- Kibana, analysis of all the honeytraps inside the t-pot.

- P0f

- Filtration algorithm (Python code)

- Splunk

Table Of Contents

01

Introduction

OS Fingerprinting

02

Problem

Why OS Fingerprinting
The motivation

03

Related works

DNS Traffic Analysis.
HTTP user agent.
Active OS fingerprinting tool.

04

Proposed Approach

Approach steps
Parameters & Simulation
Hypothesis
Flowchart & Pseudocode

05

Analysis & Results

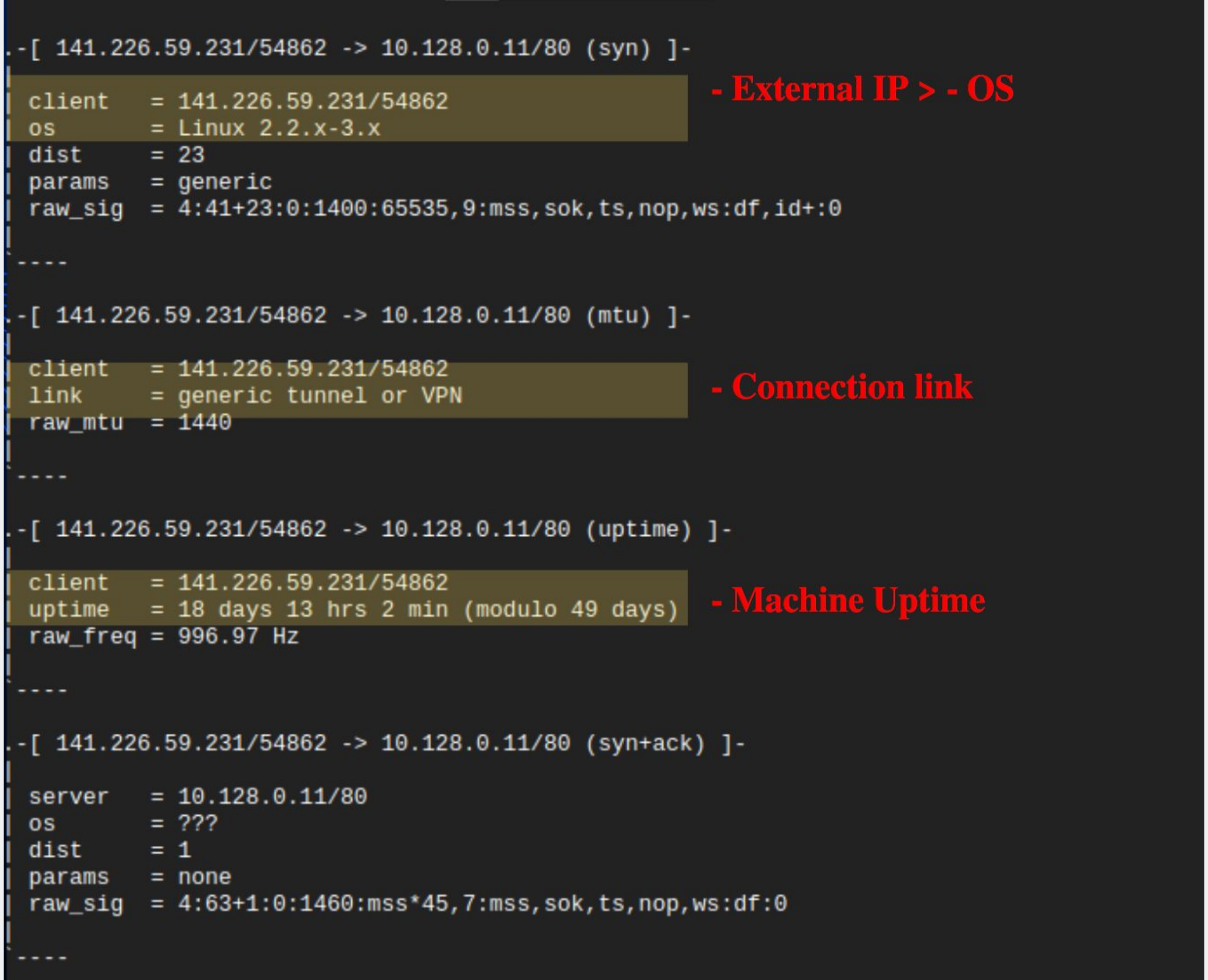
Results of:
Framework
Filtration
Kibana
Splunk

06

Conclusion

Future Work

P0f Linux results:



```
.-[ 141.226.59.231/54862 -> 10.128.0.11/80 (syn) ]-
|
| client   = 141.226.59.231/54862
| os       = Linux 2.2.x-3.x
| dist     = 23
| params   = generic
| raw_sig  = 4:41+23:0:1400:65535,9:mss,sok,ts,nop,ws:df,id+:0
|
`----

.-[ 141.226.59.231/54862 -> 10.128.0.11/80 (mtu) ]-
|
| client   = 141.226.59.231/54862
| link     = generic tunnel or VPN
| raw_mtu  = 1440
|
`----

.-[ 141.226.59.231/54862 -> 10.128.0.11/80 (uptime) ]-
|
| client   = 141.226.59.231/54862
| uptime   = 18 days 13 hrs 2 min (modulo 49 days)
| raw_freq = 996.97 Hz
|
`----

.-[ 141.226.59.231/54862 -> 10.128.0.11/80 (syn+ack) ]-
|
| server   = 10.128.0.11/80
| os       = ???
| dist     = 1
| params   = none
| raw_sig  = 4:63+1:0:1460:mss*45,7:mss,sok,ts,nop,ws:df:0
|
`----
```

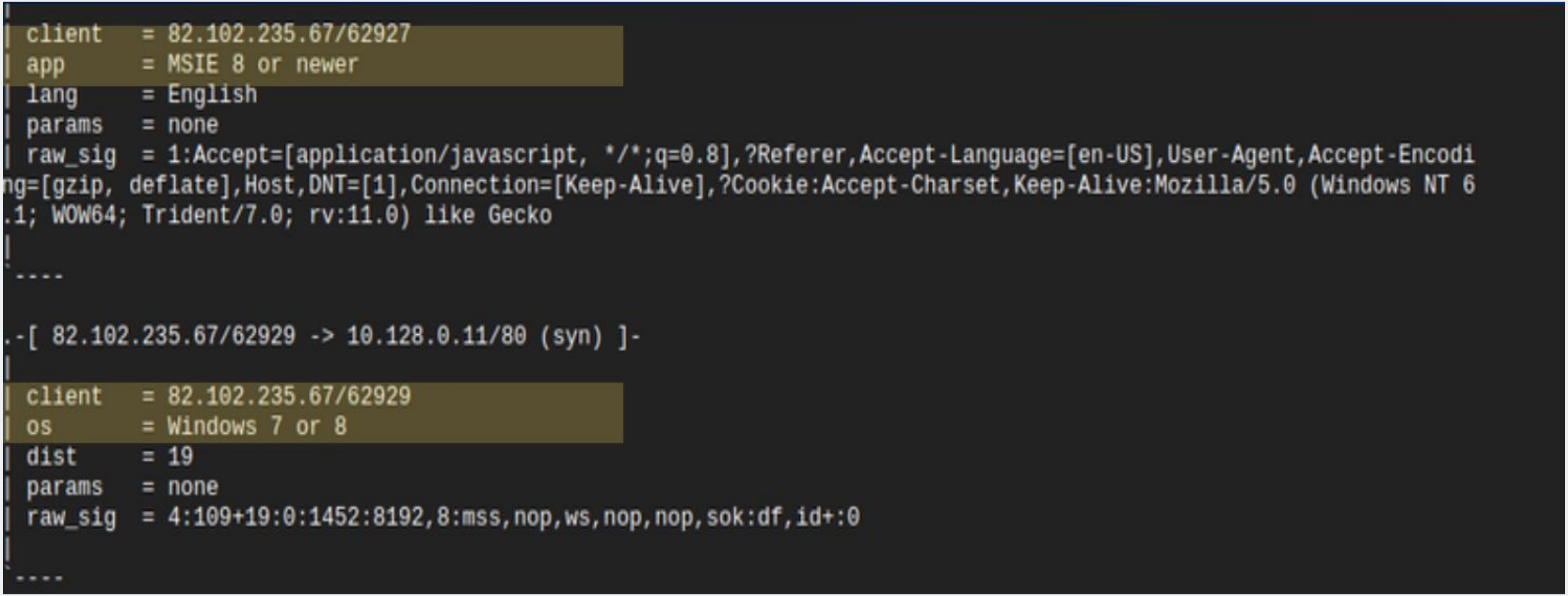
- External IP > - OS

- Connection link

- Machine Uptime

Results

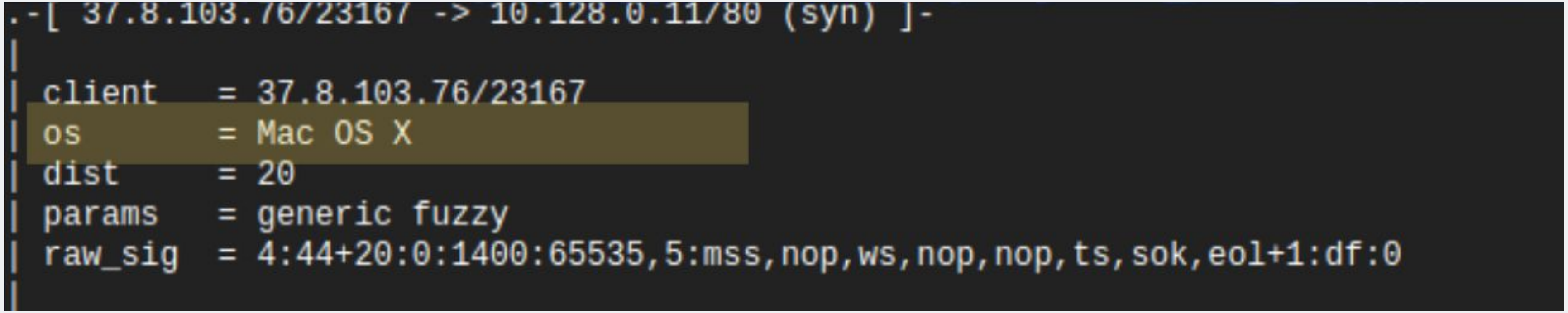
P0f Windows results:



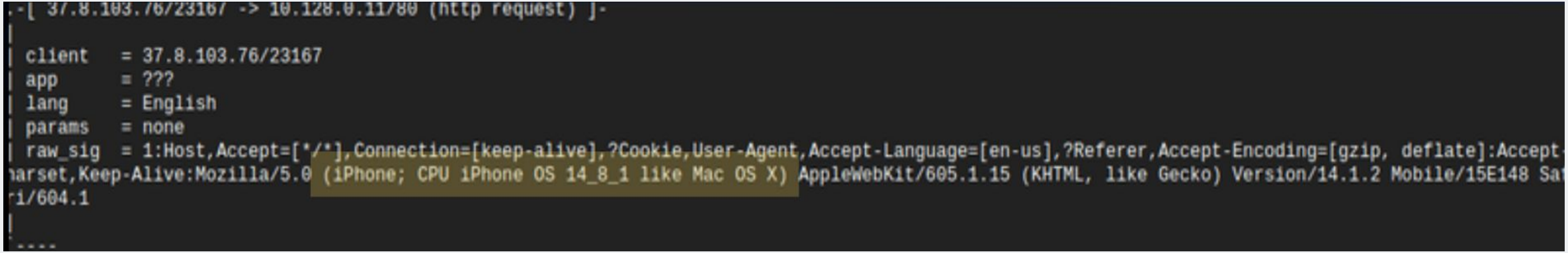
```
|
| client   = 82.102.235.67/62927
| app      = MSIE 8 or newer
| lang     = English
| params   = none
| raw_sig  = 1:Accept=[application/javascript, */*;q=0.8],?Referer,Accept-Language=[en-US],User-Agent,Accept-Encodi
ng=[gzip, deflate],Host,DNT=[1],Connection=[Keep-Alive],?Cookie:Accept-Charset,Keep-Alive:Mozilla/5.0 (Windows NT 6
.1; WOW64; Trident/7.0; rv:11.0) like Gecko
|
`----

.-[ 82.102.235.67/62929 -> 10.128.0.11/80 (syn) ]-
|
| client   = 82.102.235.67/62929
| os       = Windows 7 or 8
| dist     = 19
| params   = none
| raw_sig  = 4:109+19:0:1452:8192,8:mss,nop,ws,nop,nop,sok:df,id+:0
|
`----
```

P0f MAC results:



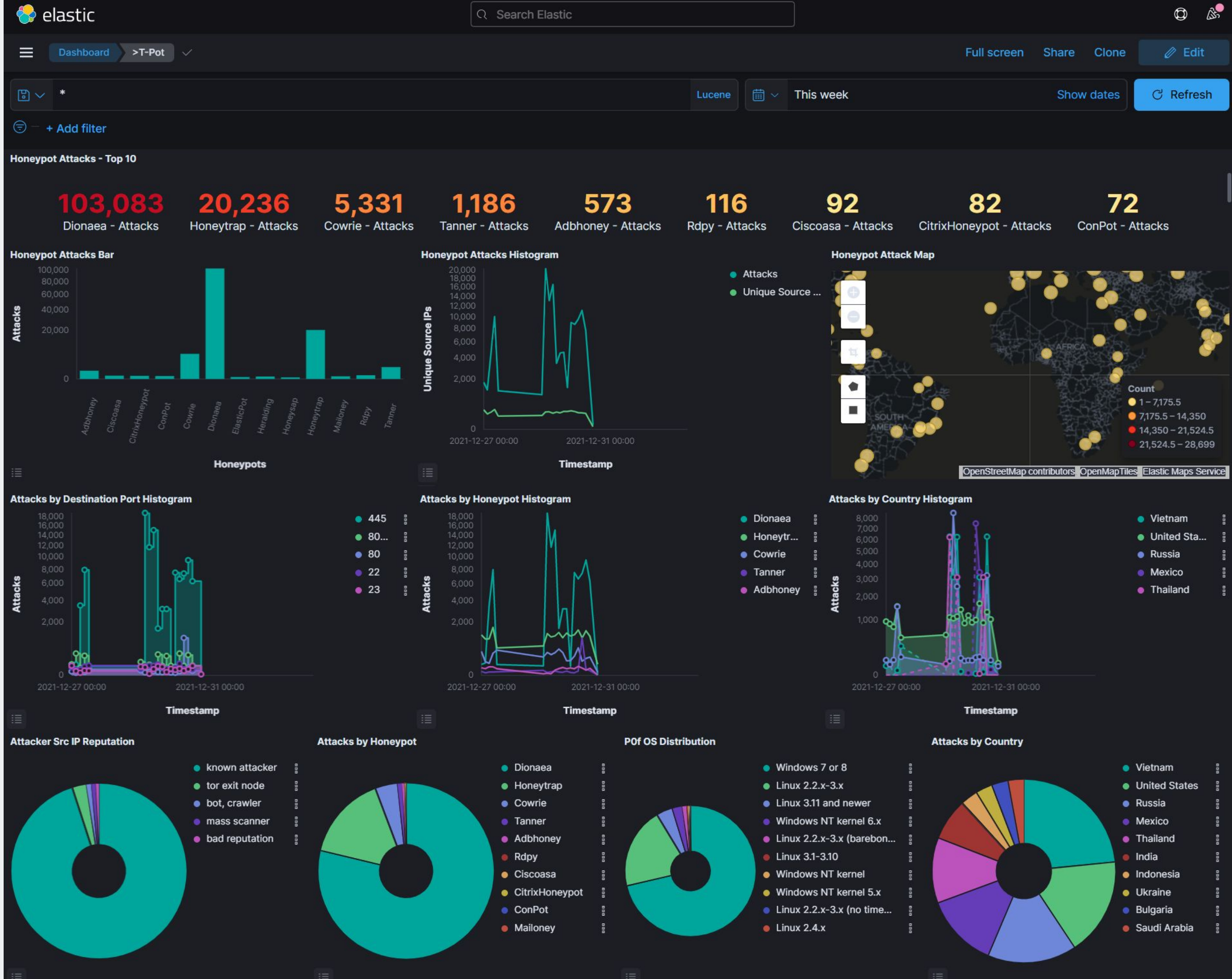
```
.-[ 37.8.103.76/23167 -> 10.128.0.11/80 (syn) ]-
|
| client   = 37.8.103.76/23167
| os       = Mac OS X
| dist     = 20
| params   = generic fuzzy
| raw_sig  = 4:44+20:0:1400:65535,5:mss,nop,ws,nop,nop,ts,sok,eol+1:df:0
|
```



```
.-[ 37.8.103.76/23167 -> 10.128.0.11/80 (http request) ]-
|
| client   = 37.8.103.76/23167
| app      = ???
| lang     = English
| params   = none
| raw_sig  = 1:Host,Accept=[*/*],Connection=[keep-alive],?Cookie,User-Agent,Accept-Language=[en-us],?Referer,Accept-Encoding=[gzip, deflate]:Accept-
harset,Keep-Alive:Mozilla/5.0 (iPhone; CPU iPhone OS 14_8_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.1.2 Mobile/15E148 Saf
ri/604.1
|
`----
```

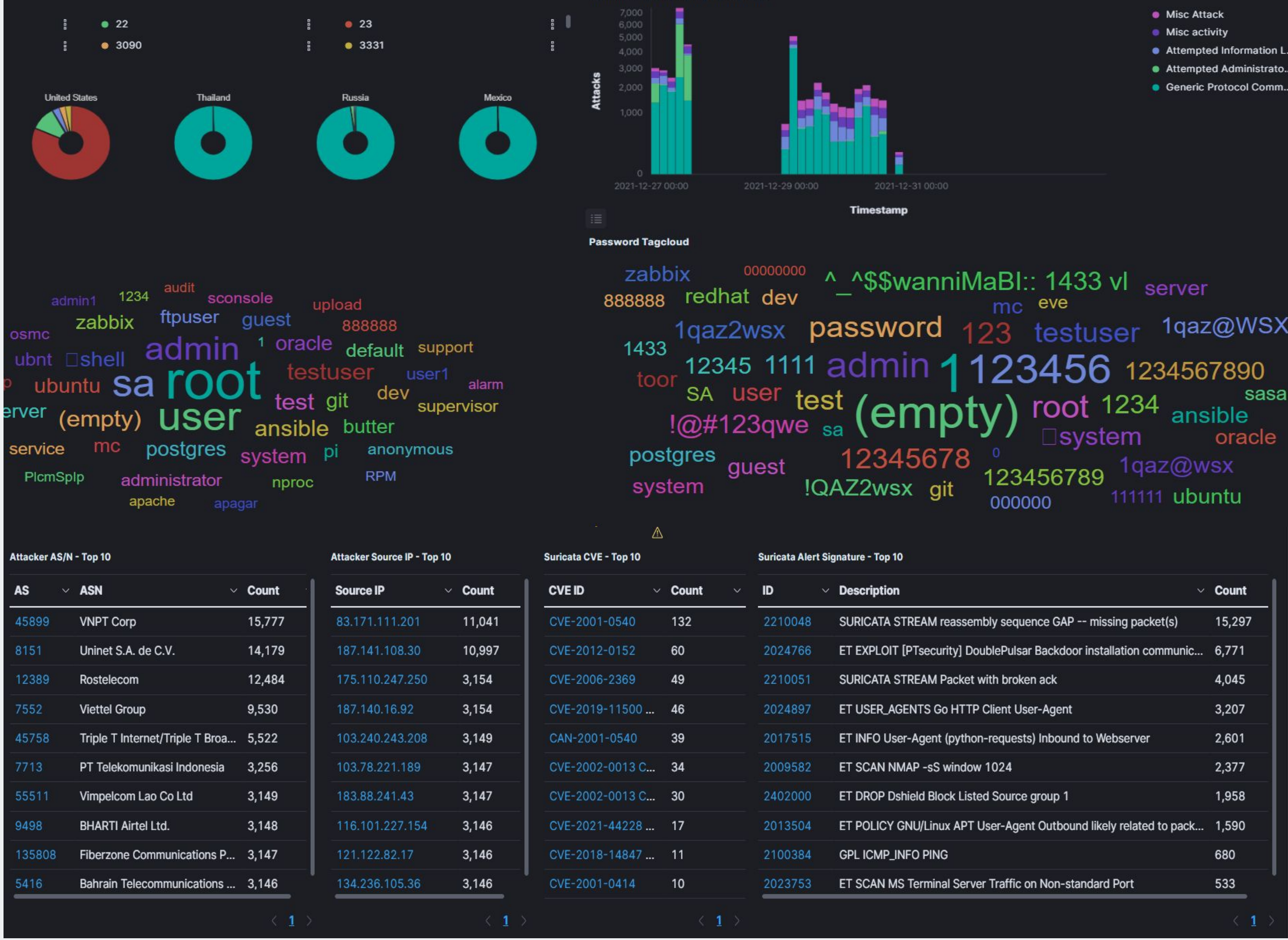
Results

Result of Kibana:



Results

Result of Kibana:


● 22
● 3090
● 23
● 3331
United States
Thailand
Russia
Mexico
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
Attacks
2021-12-27 00:00
2021-12-29 00:00
2021-12-31 00:00
Timestamp
● Misc Attack
● Misc activity
● Attempted Information L...
● Attempted Administrato...
● Generic Protocol Comm...
admin1 1234 audit sconsole upload
osmc zabbix ftpuser guest 888888
ubnt shell admin 1 oracle default support
ubuntu sa root testuser user1 alarm
erver (empty) user test git dev supervisor
ansible butter
service mc postgres system pi anonymous
PlcmSplp administrator nproc RPM
apache apagar
Password Tagcloud
zabbix 00000000 ^_^$$wanniMaBI:: 1433 vl server
888888 redhat dev mc eve
1qaz2wsx password 123 testuser 1qaz@WSX
1433 12345 1111 admin 1 123456 1234567890
toor SA user test root 1234 sasa
!@#123qwe sa (empty) ansible
system oracle
postgres 12345678 0 1qaz@wsx
guest 123456789
system !QAZ2wsx git 000000 111111 ubuntu

Attacker AS/N - Top 10

AS	ASN	Count
45899	VNPT Corp	15,777
8151	Uninet S.A. de C.V.	14,179
12389	Rostelecom	12,484
7552	Viettel Group	9,530
45758	Triple T Internet/Triple T Broa...	5,522
7713	PT Telekomunikasi Indonesia	3,256
55511	Vimpelcom Lao Co Ltd	3,149
9498	BHARTI Airtel Ltd.	3,148
135808	Fiberzone Communications P...	3,147
5416	Bahrain Telecommunications ...	3,146

‹ 1 ›

Attacker Source IP - Top 10

Source IP	Count
83.171.111.201	11,041
187.141.108.30	10,997
175.110.247.250	3,154
187.140.16.92	3,154
103.240.243.208	3,149
103.78.221.189	3,147
183.88.241.43	3,147
116.101.227.154	3,146
121.122.82.17	3,146
134.236.105.36	3,146

‹ 1 ›

Suricata CVE - Top 10

CVE ID	Count
CVE-2001-0540	132
CVE-2012-0152	60
CVE-2006-2369	49
CVE-2019-11500 ...	46
CAN-2001-0540	39
CVE-2002-0013 C...	34
CVE-2002-0013 C...	30
CVE-2021-44228 ...	17
CVE-2018-14847 ...	11
CVE-2001-0414	10

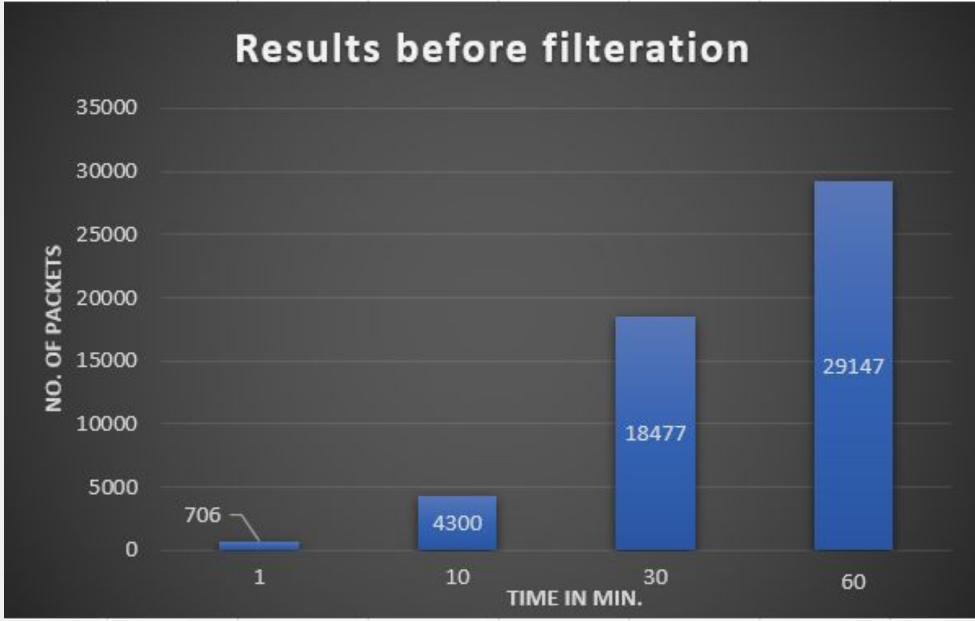
‹ 1 ›

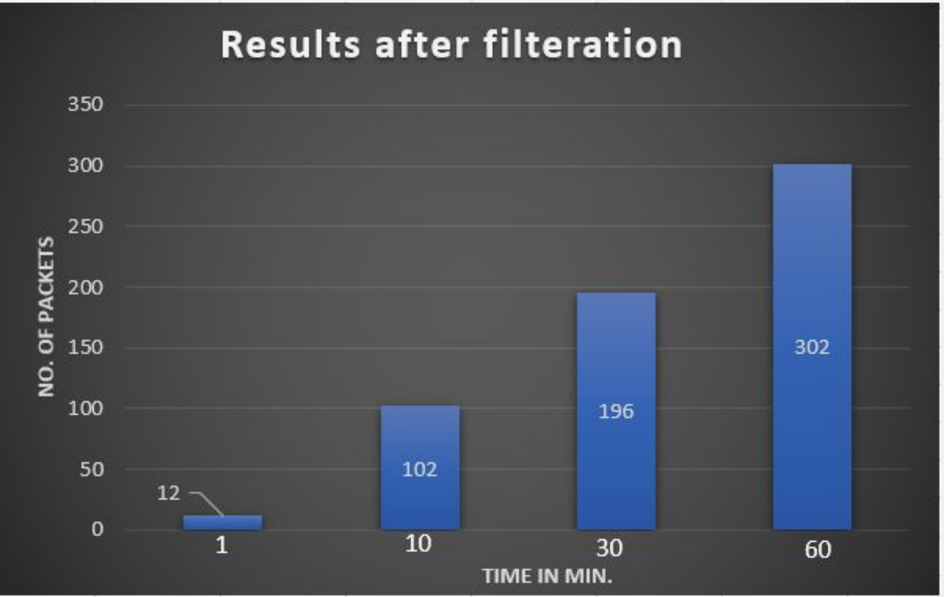
Suricata Alert Signature - Top 10

ID	Description	Count
2210048	SURICATA STREAM reassembly sequence GAP -- missing packet(s)	15,297
2024766	ET EXPLOIT [PTsecurity] DoublePulsar Backdoor installation communic...	6,771
2210051	SURICATA STREAM Packet with broken ack	4,045
2024897	ET USER_AGENTS Go HTTP Client User-Agent	3,207
2017515	ET INFO User-Agent (python-requests) Inbound to Webserver	2,601
2009582	ET SCAN NMAP -sS window 1024	2,377
2402000	ET DROP Dshield Block Listed Source group 1	1,958
2013504	ET POLICY GNU/Linux APT User-Agent Outbound likely related to pack...	1,590
2100384	GPL ICMP_INFO PING	680
2023753	ET SCAN MS Terminal Server Traffic on Non-standard Port	533

‹ 1 ›

Results after and before Filtration:

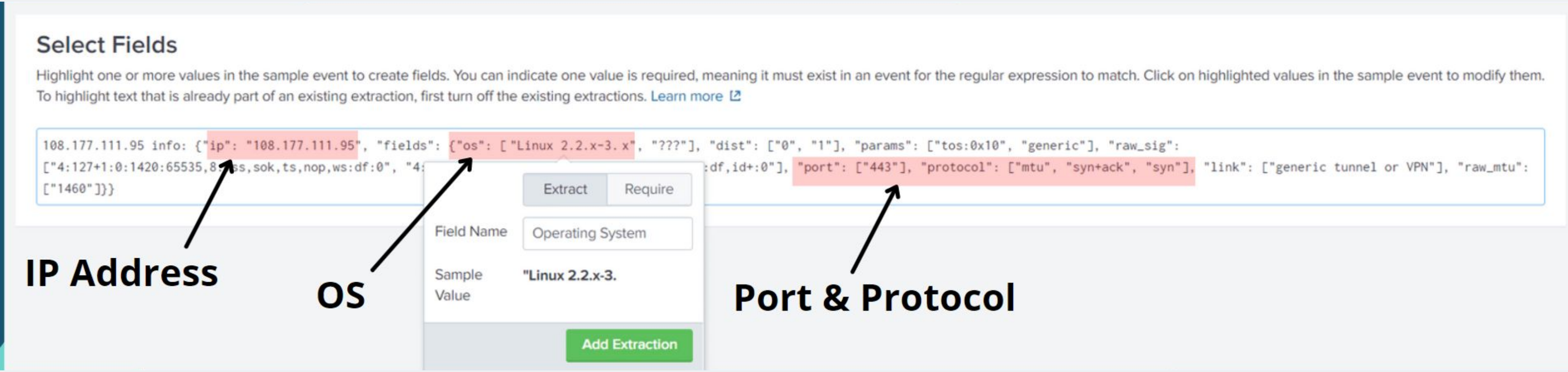




Results

Analysis by Splunk

Splunk was able to identify each field in the packet.



Select Fields

Highlight one or more values in the sample event to create fields. You can indicate one value is required, meaning it must exist in an event for the regular expression to match. Click on highlighted values in the sample event to modify them. To highlight text that is already part of an existing extraction, first turn off the existing extractions. Learn more

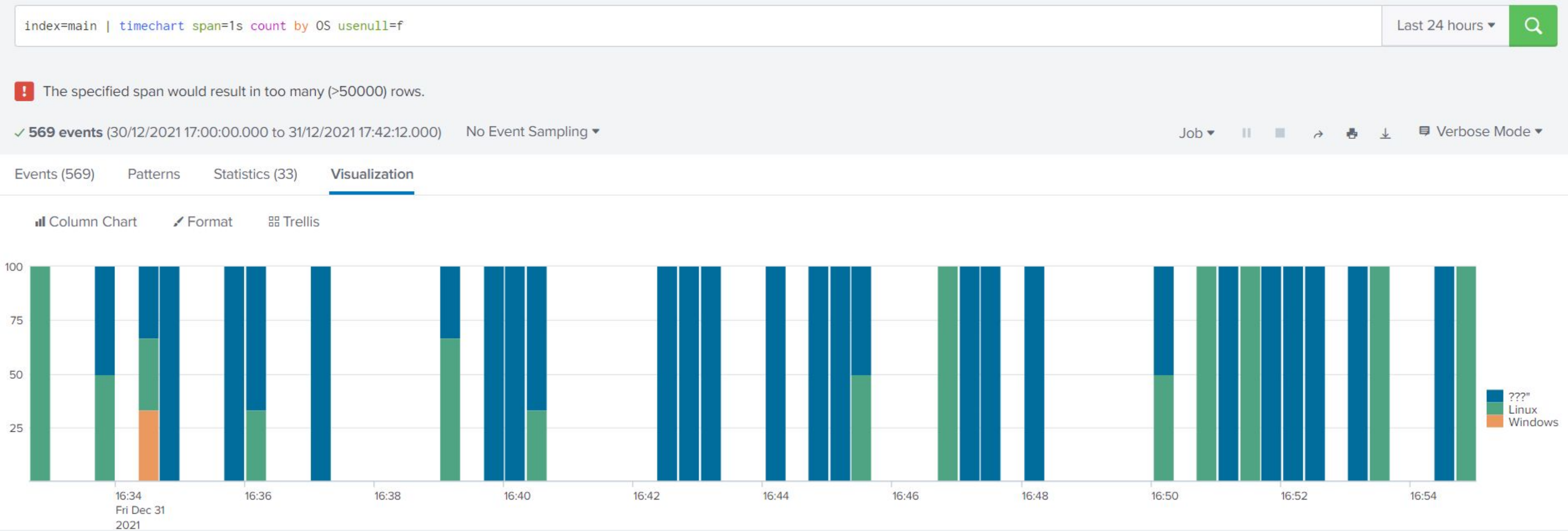
```
108.177.111.95 info: {"ip": "108.177.111.95", "fields": {"os": [ "Linux 2.2.x-3. x", "???"], "dist": ["0", "1"], "params": ["tos:0x10", "generic"], "raw_sig":
["4:127+1:0:1420:65535,8: ss,sok,ts,nop,ws:df:0", "4:                 :df,id+:0"], "port": ["443"], "protocol": ["mtu", "syn+ack", "syn"], "link": ["generic tunnel or VPN"], "raw_mtu":
["1460"]}}
```

Field Name: Operating System

Sample Value: "Linux 2.2.x-3.

Analysis by Splunk

Creating time charts by Splunk of the OS's Type:



```
index=main | timechart span=1s count by OS usenull=f
```

Last 24 hours ▾

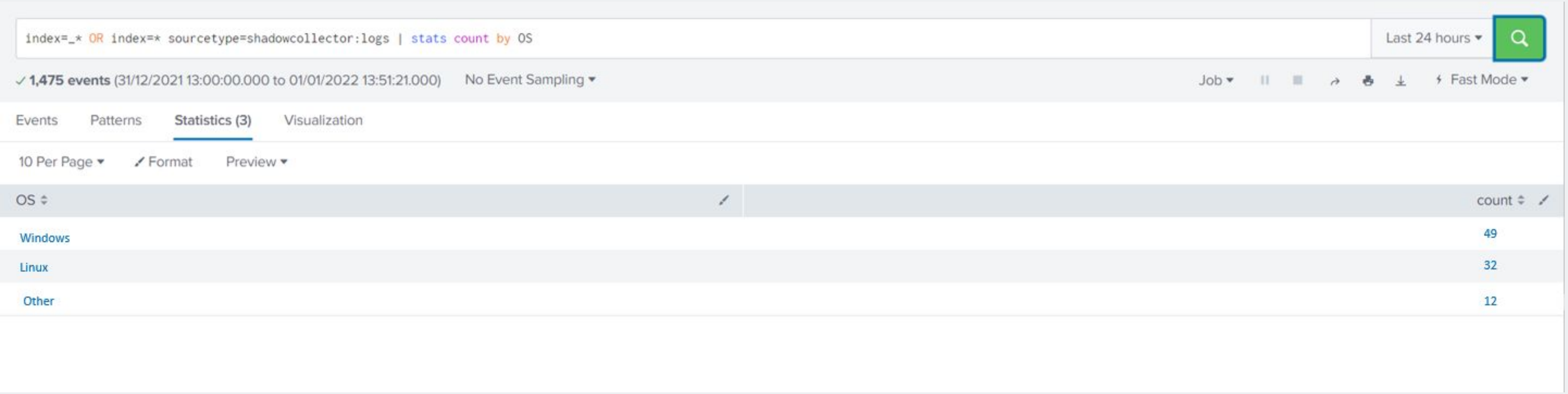
! The specified span would result in too many (>50000) rows.

✓ 569 events (30/12/2021 17:00:00.000 to 31/12/2021 17:42:12.000) No Event Sampling ▾

Results

Analysis by Splunk

Example of stats generated by Splunk



```
index=_* OR index=* sourcetype=shadowcollector:logs | stats count by OS
```

Last 24 hours

✓ 1,475 events (31/12/2021 13:00:00.000 to 01/01/2022 13:51:21.000)

No Event Sampling

Fast Mode

OS	count
Windows	49
Linux	32
Other	12

Performance Monitoring by Cockpit

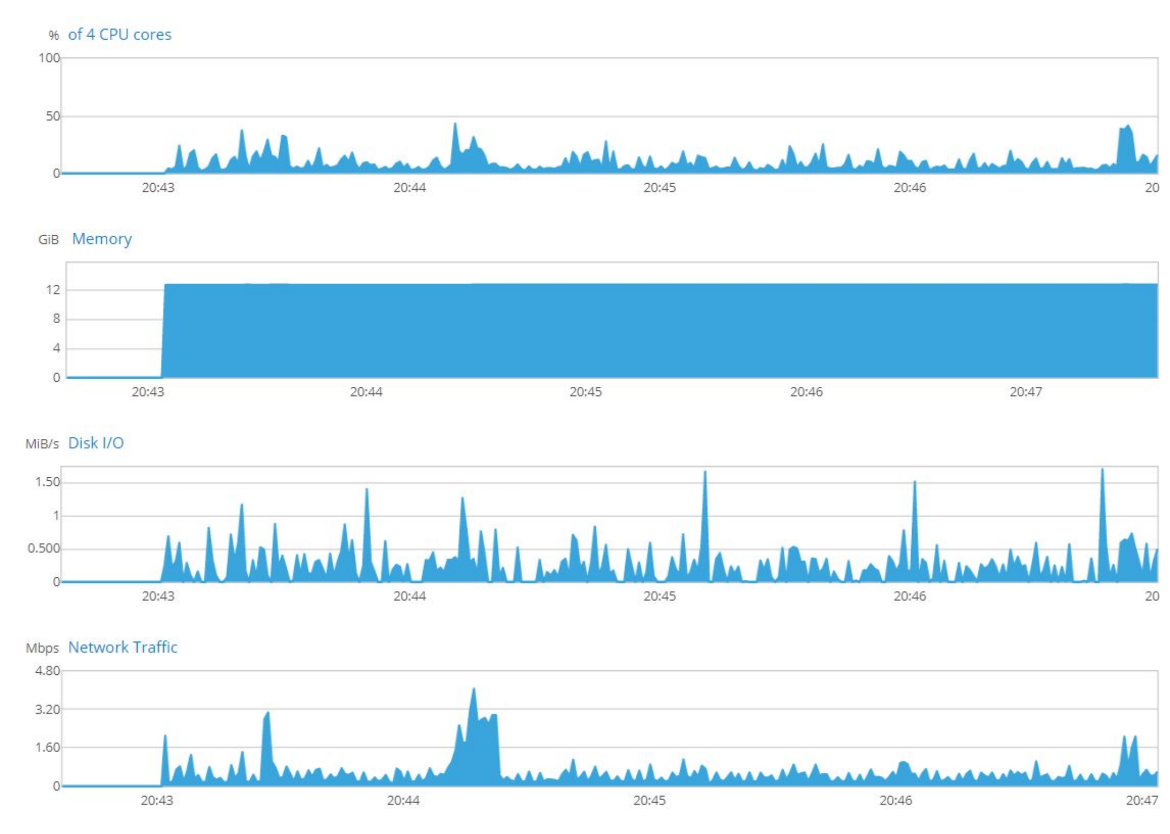


Table Of Contents

01

Introduction

OS Fingerprinting

02

Problem

Why OS Fingerprinting
The motivation

03

Related works

DNS Traffic Analysis.
HTTP user agent.
Active OS fingerprinting tool.

04

Proposed Approach

Approach steps
Parameters & Simulation
Hypothesis
Flowchart & Pseudocode

05

Analysis & Results

Results of:
Framework
Filtration
Kibana
Splunk

06

Conclusion

Future Work

Conclusion

Passive OS fingerprinting is crucial for successful network administration and cyber-attacks detection.

At this point of the project, we were able to:

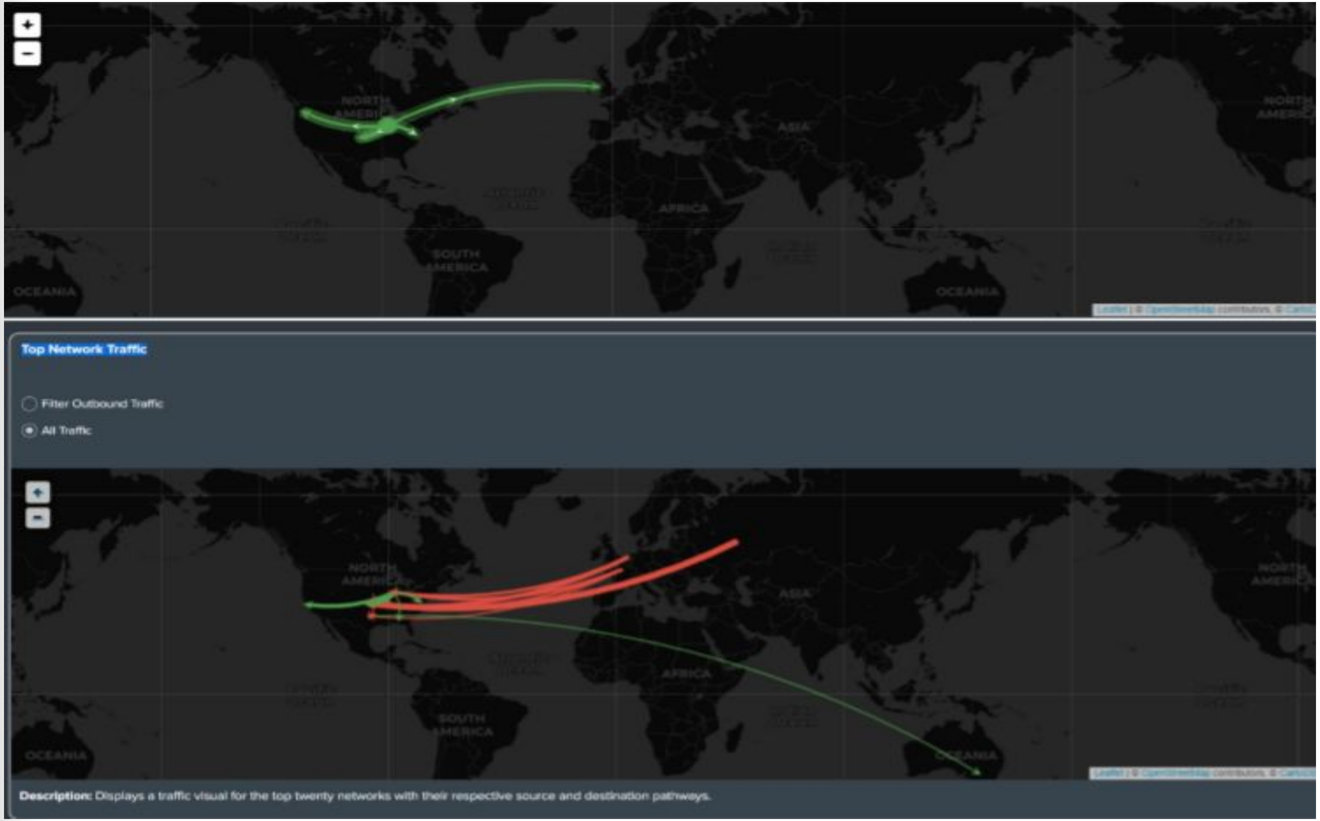
- Build a framework the contianes, T-Pot, P0f, filtration Algorithm, and analysis tool.
- Gather as much data as possible of the attackers trying to get inside a system.
- Analysing the gathered data and generating reports.

Future Work:

Working on more protocols

Machines Identification (ID) & Tracking

It's a part of a GREAT cybersecurity project !



Top Network Traffic

○ Filter Outbound Traffic

● All Traffic

Description: Displays a traffic visual for the top twenty networks with their respective source and destination pathways.

We did it in cooperation with CrossRealms !

An-Najah National University

Thank You!

Do you have any questions?

P0f tool limitations:

p0f is mainly depends on TCP/ IP header information to identify the operating system, some of operating systems have the same TCP/ IP stack implementation.

The attackers could hide their machine operating system using high level networking skills. Therefore, the framework is liable to show some false results.

Passive vs Active OS fingerprinting

Active OS fingerprinting : sends packets to the wanted machine and receives them this interaction will make the fingerprinting operation more precise and efficient, this will make the exposure of the fingerprinting more likely to happen.

Passive OS fingerprinting : analyzes only the received packets which will make it less efficient but the other end-host will not detect any fingerprinting attempts