

Configuration Manual

MSc Internship
Cybersecurity

Huma Sulthana
Student ID: X20190247

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Huma Sulthana

Student ID: X20190247

Programme: CyberSecurity **Year:** 2021-2022

Module: MSc Internship

Lecturer: Mr Vikas Sahni

Submission Due Date: 07/01/2022

Project Title: Controlling vulnerabilities in open-source libraries through different tools and techniques

Word Count: 526

Page Count: 9

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature: Huma Sulthana.....

Date: 7th January, 2022.....

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Controlling vulnerabilities in open-source libraries through different tools and techniques

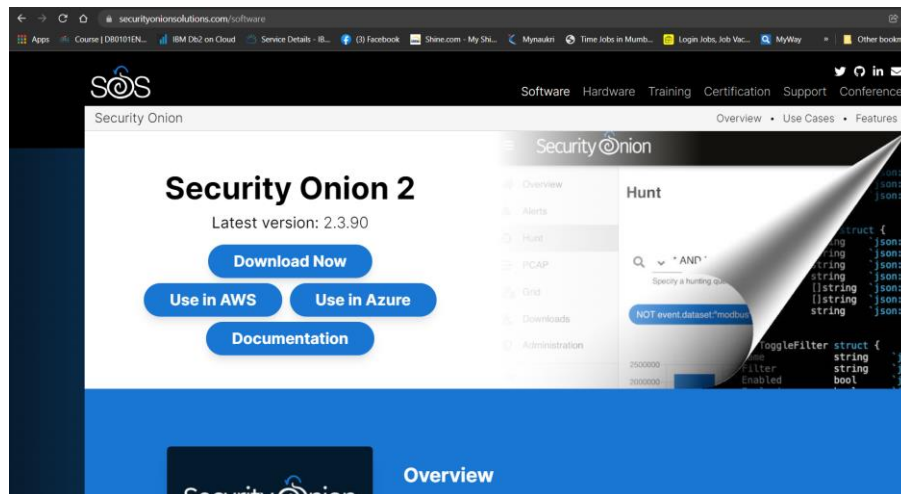
The features, tools, and capabilities implemented in this project are entailed in the configuration manual. This provides understanding of the experiments done in the project so far.

The stages below described the installation process in Windows 10 operating system.

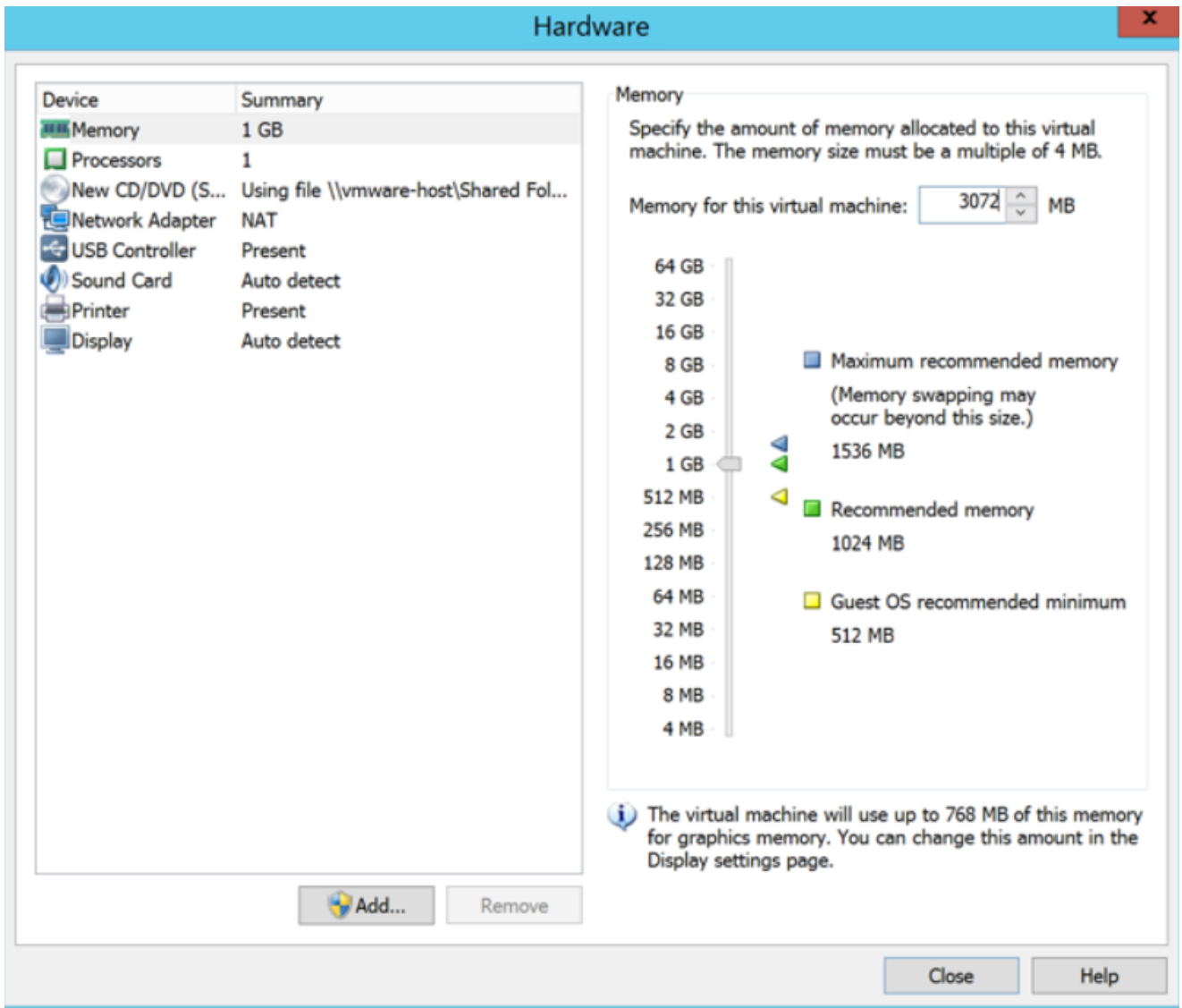
1. Preinstallation Security Onion ISO

In this section, the preinstallation process of Security Onion in Windows Operating system is discussed. Security onion include different tools for security such as intrusion detection, log management, and enterprise security monitoring.

1. At first, the latest version of Security Onion 2 (version: 2.3.90) is downloaded from <https://securityonionsolutions.com/software>

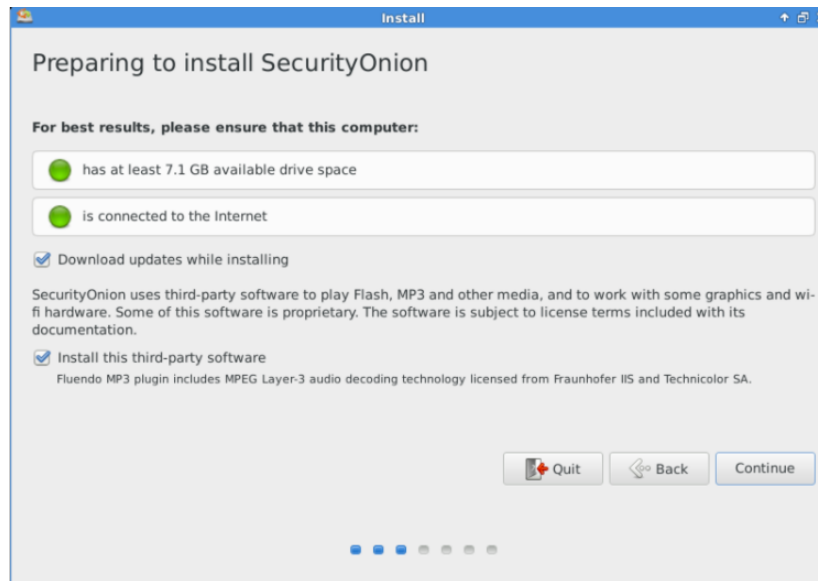


2. As it is installed in a Windows 10 OS, create a virtual machine for which download VM player from https://customerconnect.vmware.com/downloads/info/slug/desktop_end_user_computing/windows_os_optimization_tool_for_vmware_horizon/1_0
3. Play the virtual machine and check for the hardware requirement to install Security Onion as below.

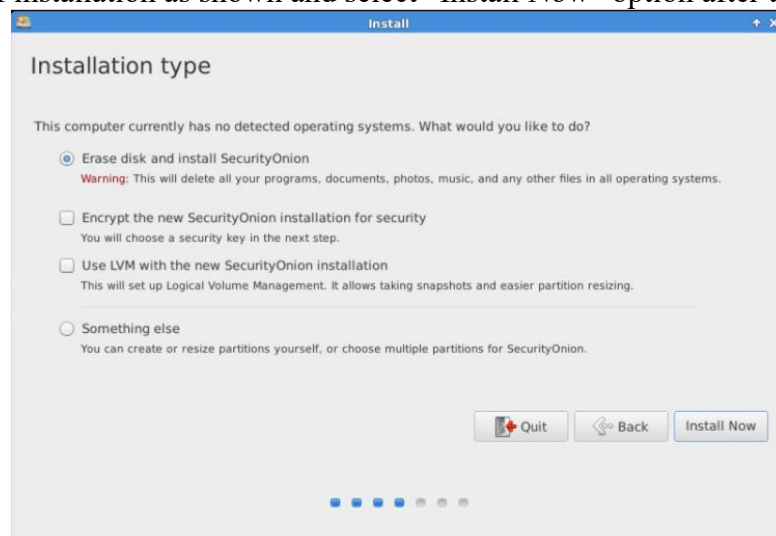


2. Installation and system update

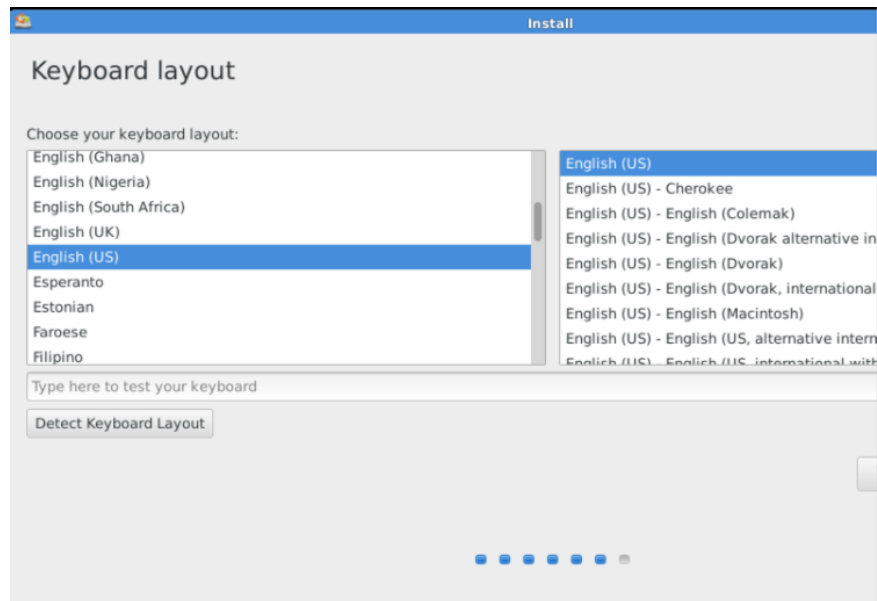
1. Booting up the VM player and the light blue screen of security onion is shown.
2. Check the boxes as shown below.



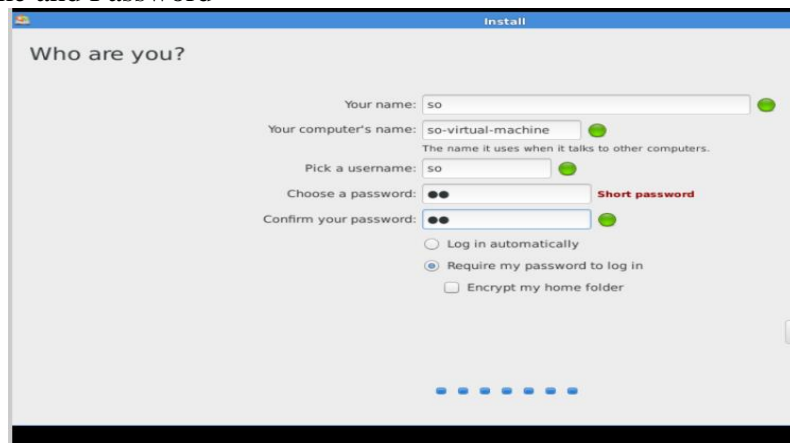
3. Select type of installation as shown and select “Install Now” option after that.



4. Conduct the change as per requirement like Keyboard changes and select continue.



5. Set Username and Password



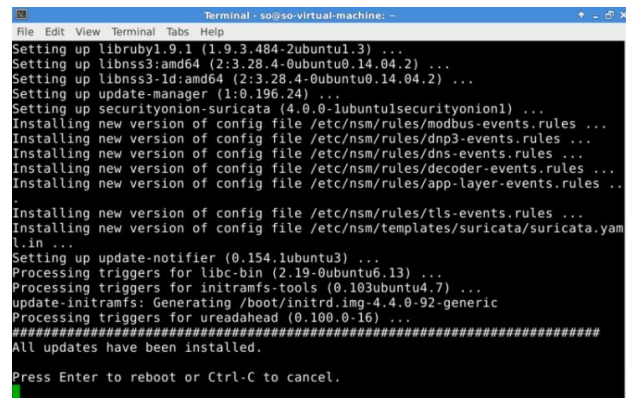
6. Login and Update the system as per requirement by writing command in the terminal emulator

```

Terminal - so@so-virtual-machine: ~
File Edit View Terminal Tabs Help
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gir1.2-json-1.0 gir1.2-timezonemap-1.0 gir1.2-xkl-1.0 libtimezonemap1
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  mysql-client-5.5 mysql-common
Suggested packages:
  tinycs mailx
The following packages will be upgraded:
  mysql-client-5.5 mysql-common mysql-server mysql-server-5.5
  mysql-server-core-5.5
5 upgraded, 0 newly installed, 0 to remove and 221 not upgraded.
Need to get 7,243 kB of archives.
After this operation, 42.0 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main mysql-common al
l 5.5.57-0ubuntu0.14.04.1 [13.0 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main mysql-server al
l 5.5.57-0ubuntu0.14.04.1 [11.3 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main mysql-server-5.
5 amd64 5.5.57-0ubuntu0.14.04.1 [1,866 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main mysql-client-5.
5 amd64 5.5.57-0ubuntu0.14.04.1 [1,588 kB]
37% [4 mysql-client-5.5 789 kB/1,588 kB 50%]

```

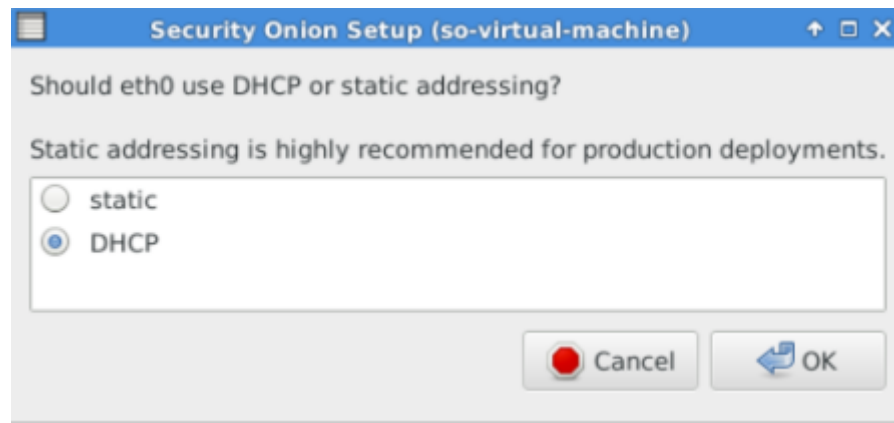
7. Provide password when asked and press enter when after unsuccessful upgrade or finish of updates



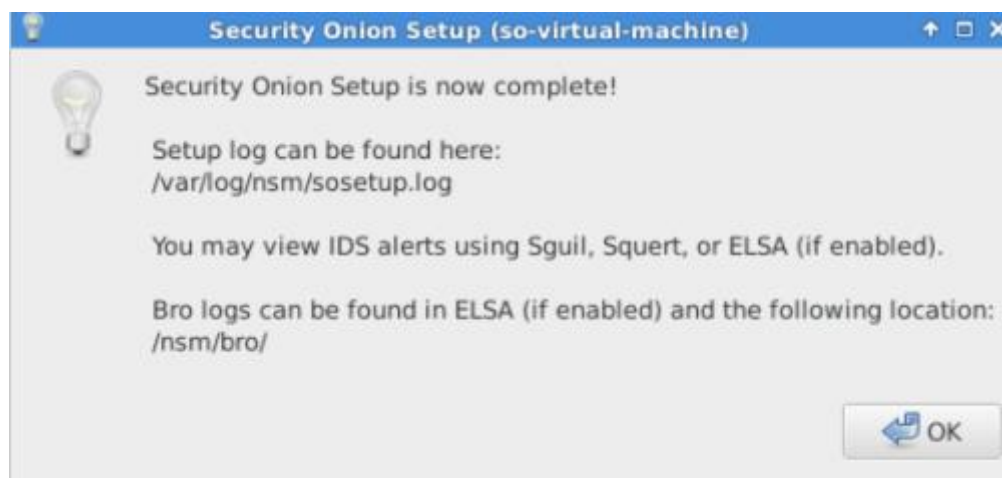
```
Terminal - so@so-virtual-machine: ~
File Edit View Terminal Tabs Help
Setting up libruby1.9.1 (1.9.3.484-2ubuntu1.3) ...
Setting up libnss3:amd64 (2:3.28.4-0ubuntu0.14.04.2) ...
Setting up libnss3-ld:amd64 (2:3.28.4-0ubuntu0.14.04.2) ...
Setting up update-manager (1:0.196.24) ...
Setting up securityonion-suricata (4.0.0-1ubuntu1securityonion1) ...
Installing new version of config file /etc/nsm/rules/modbus-events.rules ...
Installing new version of config file /etc/nsm/rules/dnp3-events.rules ...
Installing new version of config file /etc/nsm/rules/dns-events.rules ...
Installing new version of config file /etc/nsm/rules/decoder-events.rules ...
Installing new version of config file /etc/nsm/rules/app-layer-events.rules ...
Installing new version of config file /etc/nsm/rules/tls-events.rules ...
Installing new version of config file /etc/nsm/templates/suricata/suricata.yaml
l.in ...
Setting up update-notifier (0.154.1ubuntu3) ...
Processing triggers for libc-bin (2.19-0ubuntu6.13) ...
Processing triggers for initramfs-tools (0.103ubuntu4.7) ...
update-initramfs: Generating /boot/initrd.img-4.4.0-92-generic
Processing triggers for ureadahead (0.100.0-16) ...
#####
All updates have been installed.
Press Enter to reboot or Ctrl-C to cancel.
```

3. Configuration and setup completion

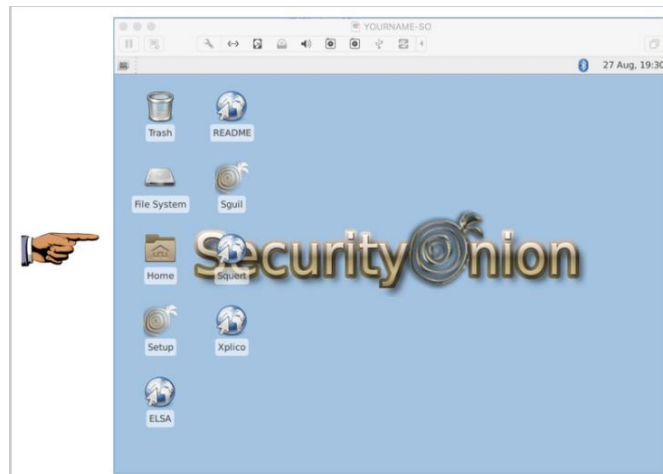
1. Open the software and click on setup
2. Configure the network interface by clicking on DHCP as shown below.



3. Create sequil username and password and the below message prompt on success



4. Click Ok till Sqert included in the system (five times)



5. Jupyter Notebook Installation

Jupyter Notebook is installed by following the documents from: <https://jupyter-docker-stacks.readthedocs.io/en/latest/index.html>

The Jupyter user is created from stack management from where the permission of a particular users can be set as shown in the figure below.

The image shows the Jupyter user management interface. At the top left is the Jupyter logo (an orange circle with a white 'j') and the word 'jupyter' in white. Below this, the 'Profile' section is titled 'Provide personal details.' and contains three input fields: 'Username' (with the value 'jupyter' and a note 'Username can't be changed once created.'), 'Full name', and 'Email address'. The 'Privileges' section is titled 'Assign roles to manage access and permissions.' and contains a 'Roles' dropdown menu with 'viewer' selected. Below the roles is a link 'Learn what privileges individual roles grant.' with an external link icon. At the bottom right are two buttons: 'Update user' (in blue) and 'Cancel'.

However, It could be done by writing commands as shown below.

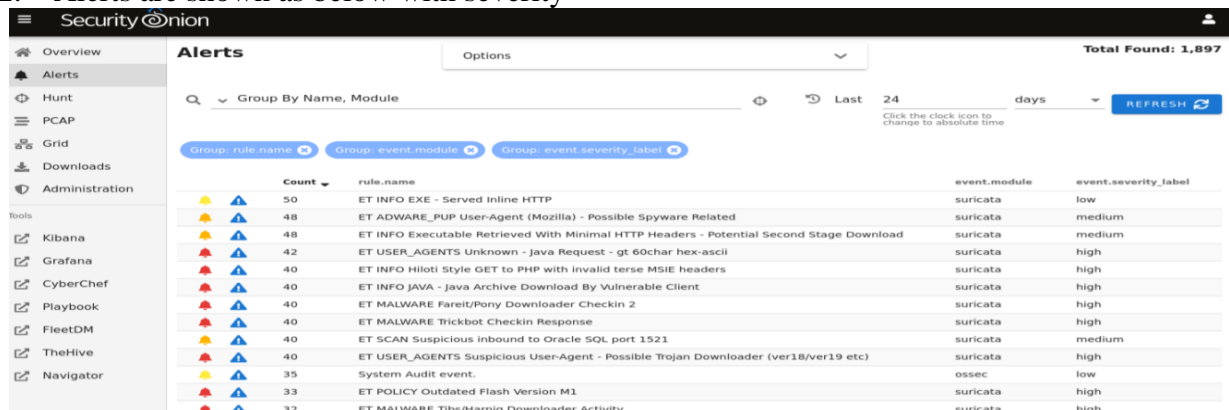
```
es = Elasticsearch(['https://192.168.6.100:9200'],
ca_certs=False,verify_certs=False, http_auth=('jupyter','password'))
searchContext = Search(using=es, index='*:so-*', doc_type='doc')
```

6. Analysis

1. Following commands are ran to import packet capture

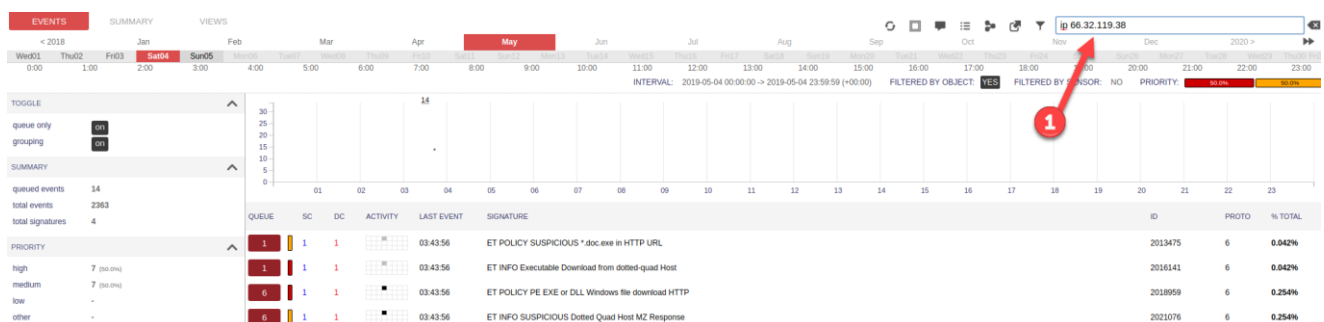
```
apnic@apnic-virtual-machine:~$ cd /opt/samples
apnic@apnic-virtual-machine:/opt/samples$ ls
10k.pcap          evidence03.pcap    ip-fragment-attack.pcap
4in6.pcap         example.com-1.pcap markofu
6to4.pcap         example.com-3.pcap mta
best_malware_protection.pcap example.com-4.pcap readme.txt
bredolab-sample.pcap example.com-5.pcap shellshock
bro               example.com-6.pcap zeus-sample-1.pcap
ConfickerB9hrs.pcap example.com-7.pcap zeus-sample-2.pcap
emerging-all.pcap fake_av.pcap       zeus-sample-3.pcap
```

2. Alerts are shown as below with severity



Count	rule.name	event.module	event.severity_label
50	ET INFO EXE - Served Inline HTTP	suricata	low
48	ET ADWARE_PUP User-Agent (Mozilla) - Possible Spyware Related	suricata	medium
48	ET INFO Executable Retrieved With Minimal HTTP Headers - Potential Second Stage Download	suricata	medium
42	ET USER_AGENTS Unknown - Java Request - gt 60char hex-ascii	suricata	high
40	ET INFO Hiloti Style GET to PHP with invalid terse MSIE headers	suricata	high
40	ET INFO JAVA - Java Archive Download By Vulnerable Client	suricata	high
40	ET MALWARE Fareit/Pony Downloader Checkin 2	suricata	high
40	ET MALWARE Trickbot Checkin Response	suricata	high
40	ET SCAN Suspicious Inbound to Oracle SQL port 1521	suricata	medium
40	ET USER_AGENTS Suspicious User-Agent - Possible Trojan Downloader (ver18/ver19 etc)	suricata	high
35	System Audit event.	ossec	low
33	ET POLICY Outdated Flash Version M1	suricata	high
32	ET MALWARE Tibbs/Harnig Downloader Activity	suricata	high

3. In the Squert's filter can be set for specific IP addresses



4. Jupyter Note helps to apply machine learning and python libraries to analyse the results better.

5. For example, the code below converts the results into python dict.

```
response = s.execute()
if response.success():
    df = pd.DataFrame([d.to_dict() for d in s.scan()])
df
```

6. The result of this code is given below.

	process	winlog	tags	@timestamp	file	@version	event	user
0	{'pid': 3956, 'entity_id': 'EBE732EE-504F-61A5...	{'execution': {'ThreadID': 4400, 'ProcessID': ...}}	velociraptor	2021-11-30T04:22:45.357Z	{'target': 'C:\Program Files\WindowsApps\Micro...	1	{'code': '11', 'module': 'sysmon', 'category': ...}	NaN
1	{'pid': 3956, 'entity_id': 'EBE732EE-504F-61A5...	{'execution': {'ThreadID': 4400, 'ProcessID': ...}}	velociraptor	2021-11-30T04:22:45.357Z	{'target': 'C:\Program Files\WindowsApps\Micro...	1	{'code': '11', 'module': 'sysmon', 'category': ...}	NaN
2	{'pid': 3956, 'entity_id': 'EBE732EE-504F-61A5...	{'execution': {'ThreadID': 4400, 'ProcessID': ...}}	velociraptor	2021-11-30T04:22:45.357Z	{'target': 'C:\Program Files\WindowsApps\Micro...	1	{'code': '11', 'module': 'sysmon', 'category': ...}	NaN
3	{'pid': 3956, 'entity_id': 'EBE732EE-504F-61A5...	{'execution': {'ThreadID': 4400, 'ProcessID': ...}}	velociraptor	2021-11-30T04:22:45.357Z	{'target': 'C:\Program Files\WindowsApps\Micro...	1	{'code': '11', 'module': 'sysmon', 'category': ...}	NaN
4	{'pid': 3956, 'entity_id': 'EBE732EE-504F-61A5...	{'execution': {'ThreadID': 4400, 'ProcessID': ...}}	velociraptor	2021-11-30T04:22:45.357Z	{'target': 'C:\Program Files\WindowsApps\Micro...	1	{'code': '11', 'module': 'sysmon', 'category': ...}	NaN
...
3190	{'pid': 3224, 'entity_id': 'EBE732EE-6DD6-61A5...	{'execution': {'ThreadID': 4400, 'ProcessID': ...}}	velociraptor	2021-11-30T01:00:55.162Z	{'target': 'C:\Windows\SoftwareDistribution\Do...	1	{'code': '', 'module': 'sysmon', 'category': ...}	NaN
3191	{'parent': {'entity_id': 'EBE732EE-611E-61A5-9...	{'execution': {'ThreadID': 4400, 'ProcessID': ...}}	velociraptor	2021-11-30T01:00:55.162Z	NaN	1	{'code': '', 'module': 'sysmon', 'category': ...}	{'name': 'NT AUTHORITY\SYSTEM'}

7. However, this result can be simplified by selecting the cells of interest using the following code.

```
response = s.execute()
if response.success():
    df = pd.DataFrame([d['event']['dataset'], d['process']['executable'], d['file']
    ['target']] for d in s))
df.columns=['Dataset', 'Executable', 'Target']
df
```

8. After applying these commands, the results look like:

	Dataset	Executable	Target
0	file_create	C:\WINDOWS\system32\svchost.exe	C:\Program Files\WindowsApps\Microsoft.YourPhone_1.21102.134.0_x64__8wekyb3d8bbwe\YourPhoneServer\Microsoft.AspNetCore.SignalR.Protocols.Json.dll
1	file_create	C:\WINDOWS\system32\svchost.exe	C:\Program Files\WindowsApps\Microsoft.YourPhone_1.21102.134.0_x64__8wekyb3d8bbwe\YourPhoneServer\Microsoft.Bcl.AsyncInterfaces.dll
2	file_create	C:\WINDOWS\system32\svchost.exe	C:\Program Files\WindowsApps\Microsoft.YourPhone_1.21102.134.0_x64__8wekyb3d8bbwe\YourPhoneServer\Microsoft.AspNetCore.SignalR.Protocols.MessagePack.dll
3	file_create	C:\WINDOWS\system32\svchost.exe	C:\Program Files\WindowsApps\Microsoft.YourPhone_1.21102.134.0_x64__8wekyb3d8bbwe\YourPhoneServer\Microsoft.Extensions.Caching.Abstractions.dll
4	file_create	C:\WINDOWS\system32\svchost.exe	C:\Program Files\WindowsApps\Microsoft.YourPhone_1.21102.134.0_x64__8wekyb3d8bbwe\YourPhoneServer\Microsoft.AspNetCore.SignalR.Client.dll
5	file_create	C:\WINDOWS\system32\svchost.exe	C:\Program Files\WindowsApps\Microsoft.YourPhone_1.21102.134.0_x64__8wekyb3d8bbwe\YourPhoneServer\Microsoft.AspNetCore.SignalR.Common.dll
6	file_create	C:\WINDOWS\system32\svchost.exe	C:\Program Files\WindowsApps\Microsoft.YourPhone_1.21102.134.0_x64__8wekyb3d8bbwe\YourPhoneServer\Microsoft.Extensions.Caching.Memory.dll
7	file_create	C:\WINDOWS\system32\svchost.exe	C:\Program Files\WindowsApps\Microsoft.YourPhone_1.21102.134.0_x64__8wekyb3d8bbwe\YourPhoneServer\Microsoft.AspNetCore.SignalR.Client.Core.dll
8	file_create	C:\WINDOWS\system32\svchost.exe	C:\Program Files\WindowsApps\Microsoft.YourPhone_1.21102.134.0_x64__8wekyb3d8bbwe\YourPhoneServer\Microsoft.AspNetCore.Http.Features.dll
9	file_create	C:\WINDOWS\system32\svchost.exe	C:\Program Files\WindowsApps\Microsoft.YourPhone_1.21102.134.0_x64__8wekyb3d8bbwe\YourPhoneServer\Humanizer.dll

This showed the target file dataset and executable file in the system director as well.

9. NIDS/HIDS method

1. The first network-based Intrusion detection method is NIDS-1 for which Suricata is used in this case. This helps to identify the malicious traffic and fingerprint anomalies. The event module column showed it

Count	rule.name	event.module	event.severity_label
50	ET INFO EXE - Served Inline HTTP	suricata	low
48	ET ADWARE_PUP User-Agent (Mozilla) - Possible Spyware Related	suricata	medium
48	ET INFO Executable Retrieved With Minimal HTTP Headers - Potential Second Stage Download	suricata	medium
42	ET USER_AGENTS Unknown - Java Request - gt 60char hex-ascii	suricata	high
40	ET INFO Hiloti Style GET to PHP with invalid terse MSIE headers	suricata	high
40	ET INFO JAVA - Java Archive Download By Vulnerable Client	suricata	high
40	ET MALWARE Fareit/Pony Downloader Checkin 2	suricata	high
40	ET MALWARE Trickbot Checkin Response	suricata	high
40	ET SCAN Suspicious inbound to Oracle SQL port 1521	suricata	medium
40	ET USER_AGENTS Suspicious User-Agent - Possible Trojan Downloader (ver18/ver19 etc)	suricata	high
35	System Audit event.	ossec	low
33	ET POLICY Outdated Flash Version M1	suricata	high
32	ET MALWARE Tibs/Harnig Downloader Activity	suricata	high

2. The second NIDS method is analysis-driven which used bro system especially for Zeek alert

so-bro - REQ - Zeek app

3. Host Intrusion Detection System or HIDS is built by Wazuh which performs log activities, integrity checking of files, real-time alerts, and detection of rootkits. The process to conduct this is shown below.

First, find the existing rule in `/opt/so/rules/hids/ruleset/rules/`.

Copy the rule to `/opt/so/rules/hids/local_rules.xml`.

Put the rule inside `<group>` `</group>` tags and give it a name.

Update the `<rule>` section to include `noalert="1"` along with `overwrite="yes"`.

Finally, restart Wazuh with `sudo so-wazuh-restart`.

References

[1] <https://securityonionsolutions.com/software/>

[2] <https://samsclass.info/50/proj/p1so-pc.htm>

[3] <https://jupyter-docker-stacks.readthedocs.io/en/latest/index.html>