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)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
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.	☐

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

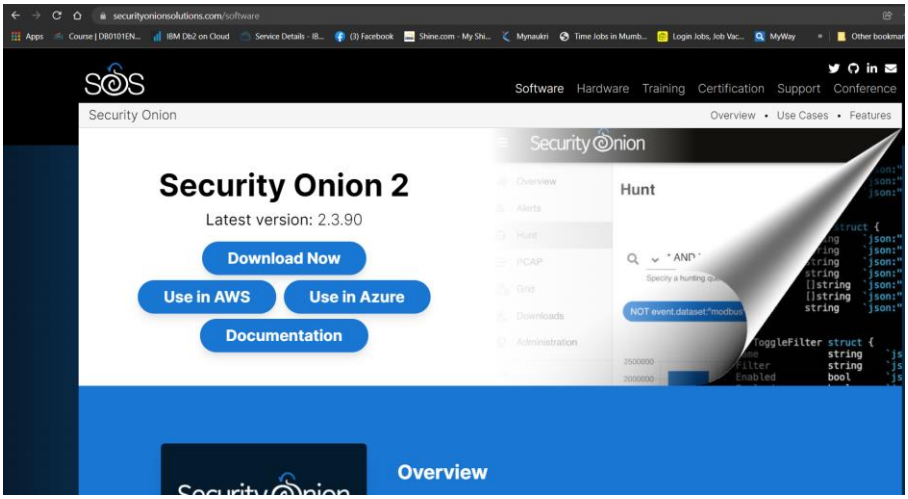
The features, tools, and capabilities implemented in this project are entailed in the configuration manual. This provides understanding of the experiments done in the project so far.

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

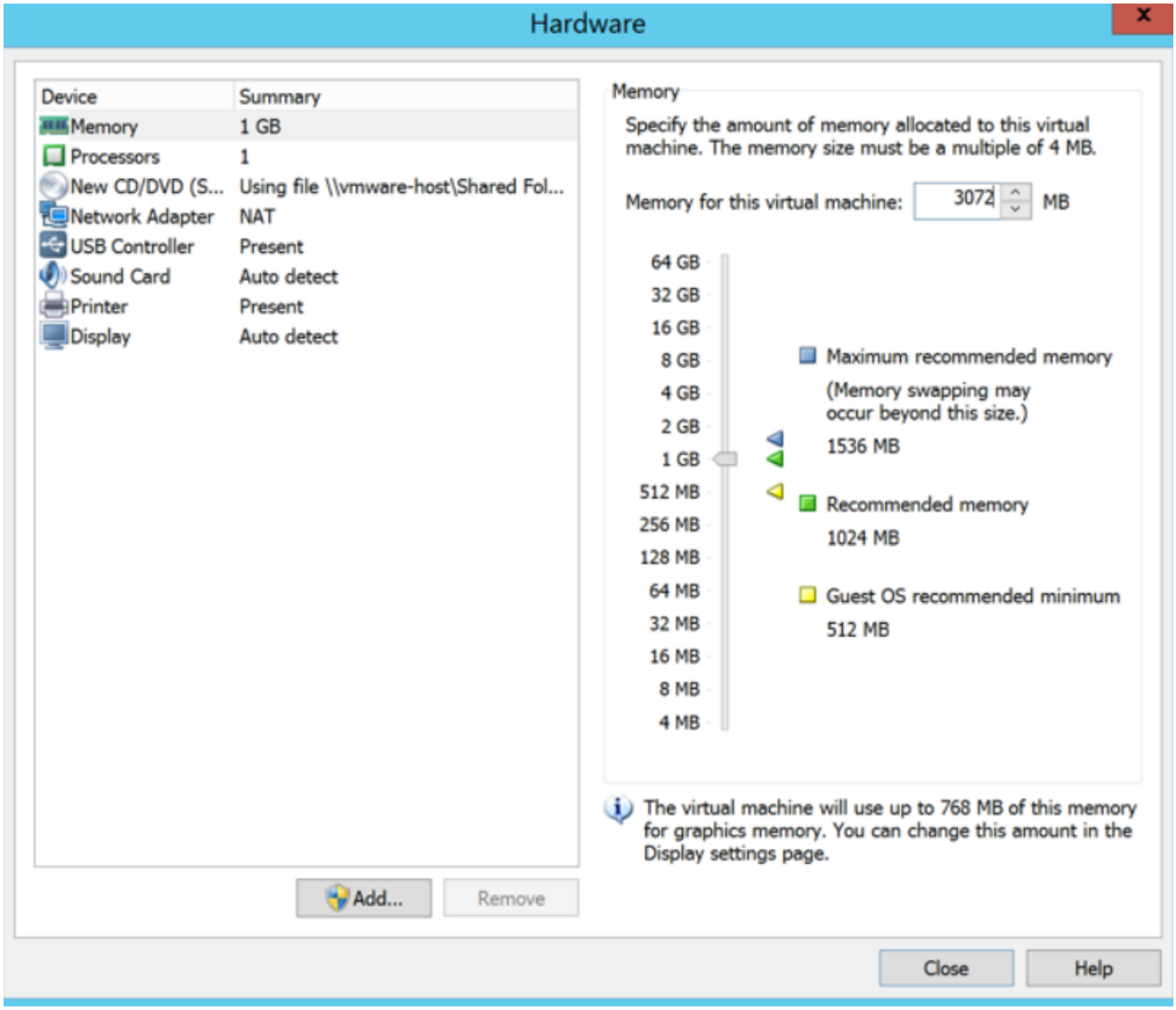
1. Preinstallation Security Onion ISO

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

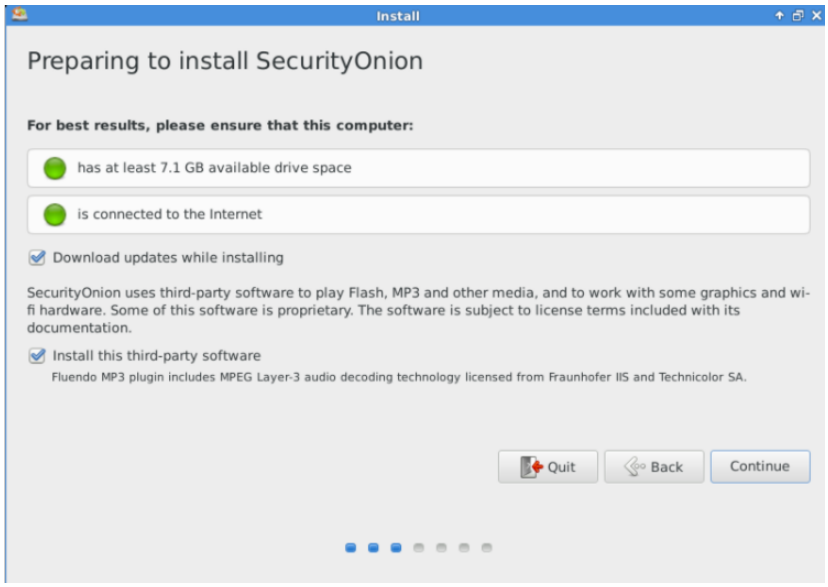


2. As it is installed in a Windows 10 OS, create a virtual machine for which download VM player from https://customerconnect.vmware.com/downloads/info/slug/desktop_end_user_computing/windows_os_optimization_tool_for_vmware_horizon/1_0
3. Play the virtual machine and check for the hardware requirement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

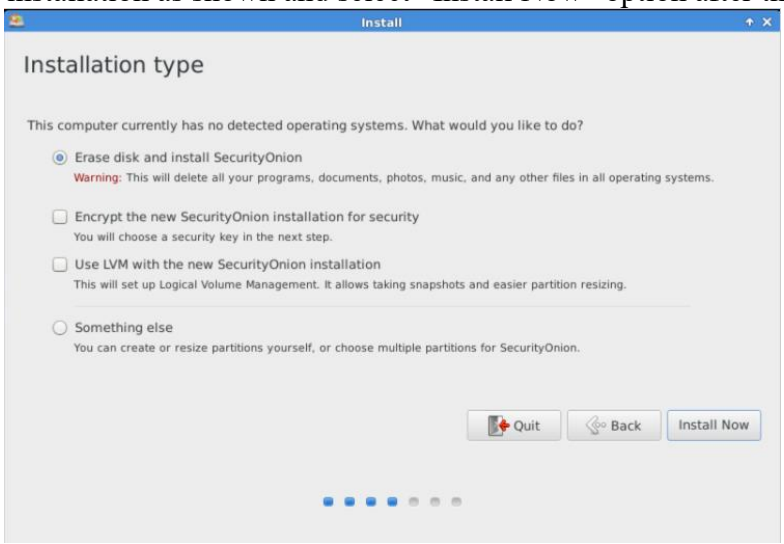


2. Installation and system update

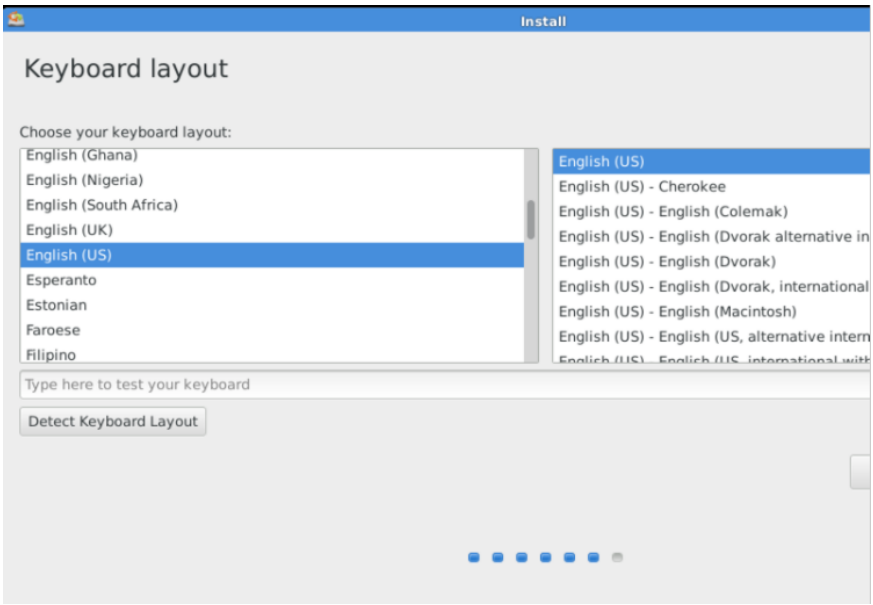
1. Booting up the VM player and the light blue screen of security onion is shown.
2. Check the boxes as shown below.



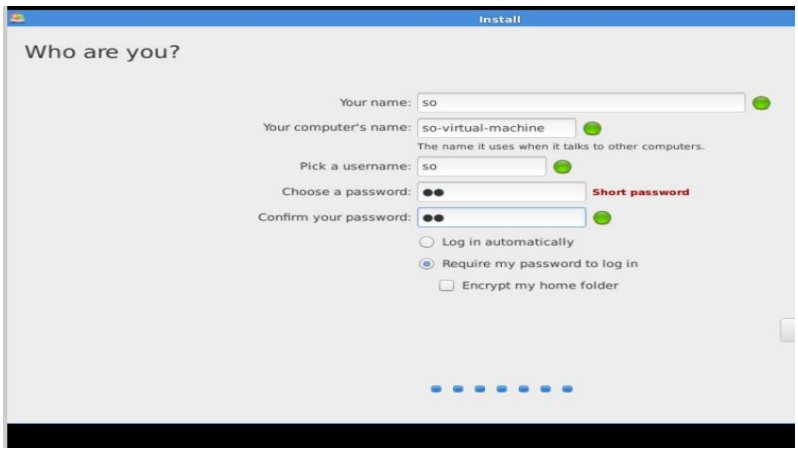
3. Select type of installation as shown and select “Install Now” option after that.



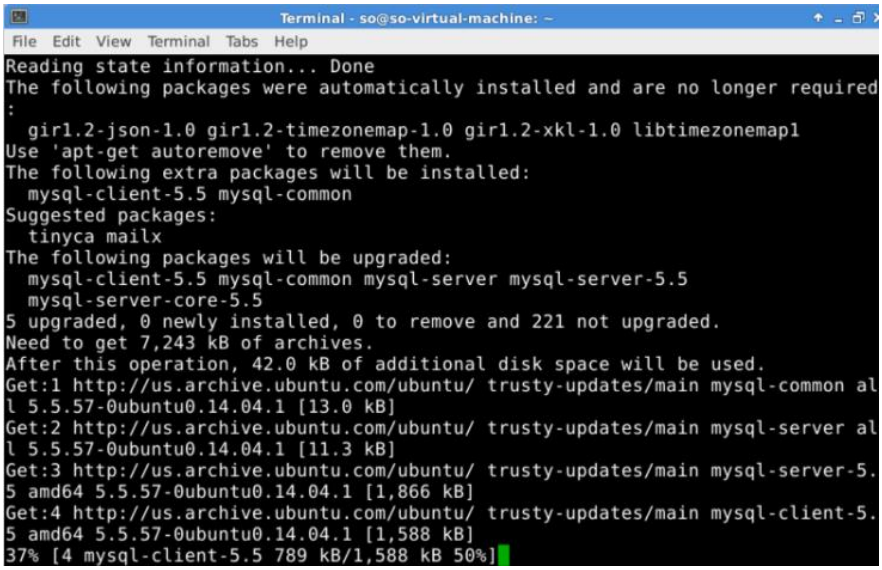
4. Conduct the change as per requirement like Keyboard changes and select continue.



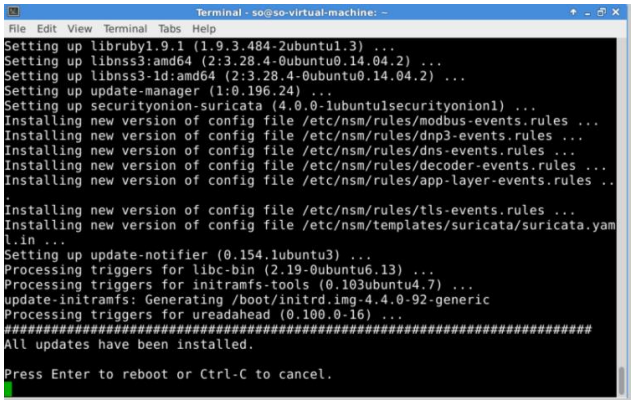
5. Set Username and Password



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

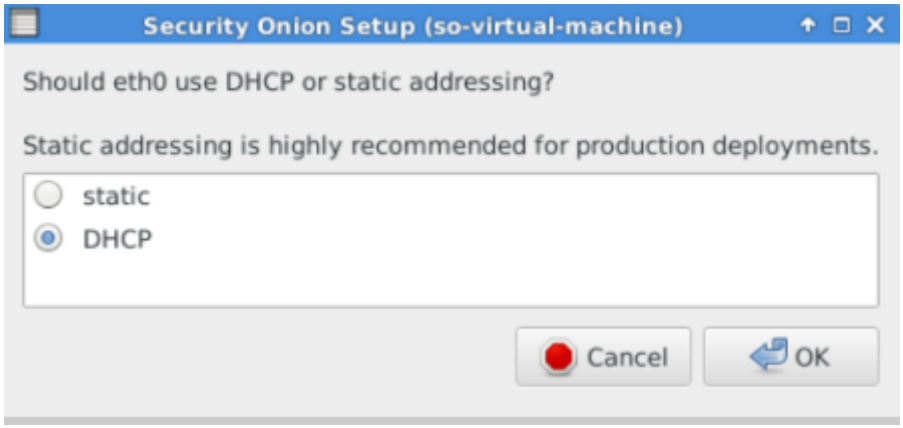


7. Provide password when asked and press enter when after unsuccessful upgrade or finish of updates

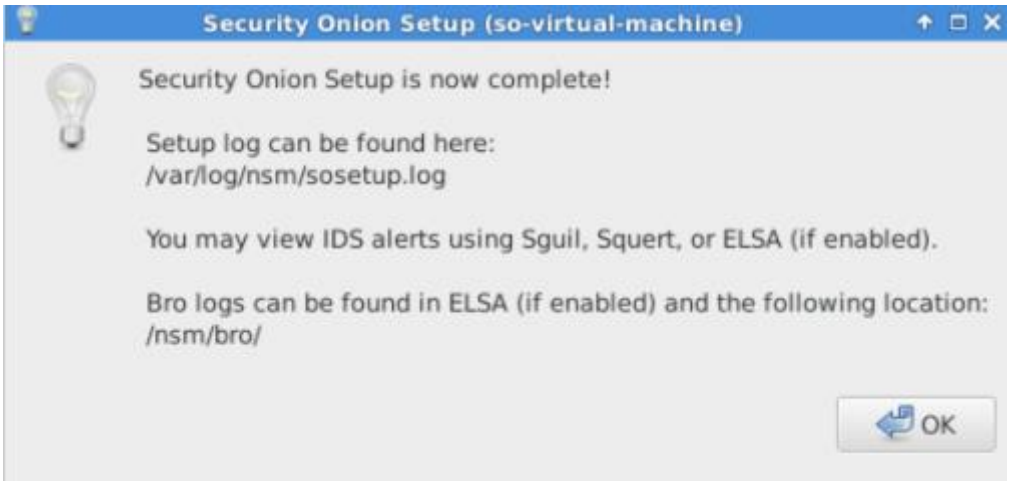


3. Configuration and setup completion

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



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

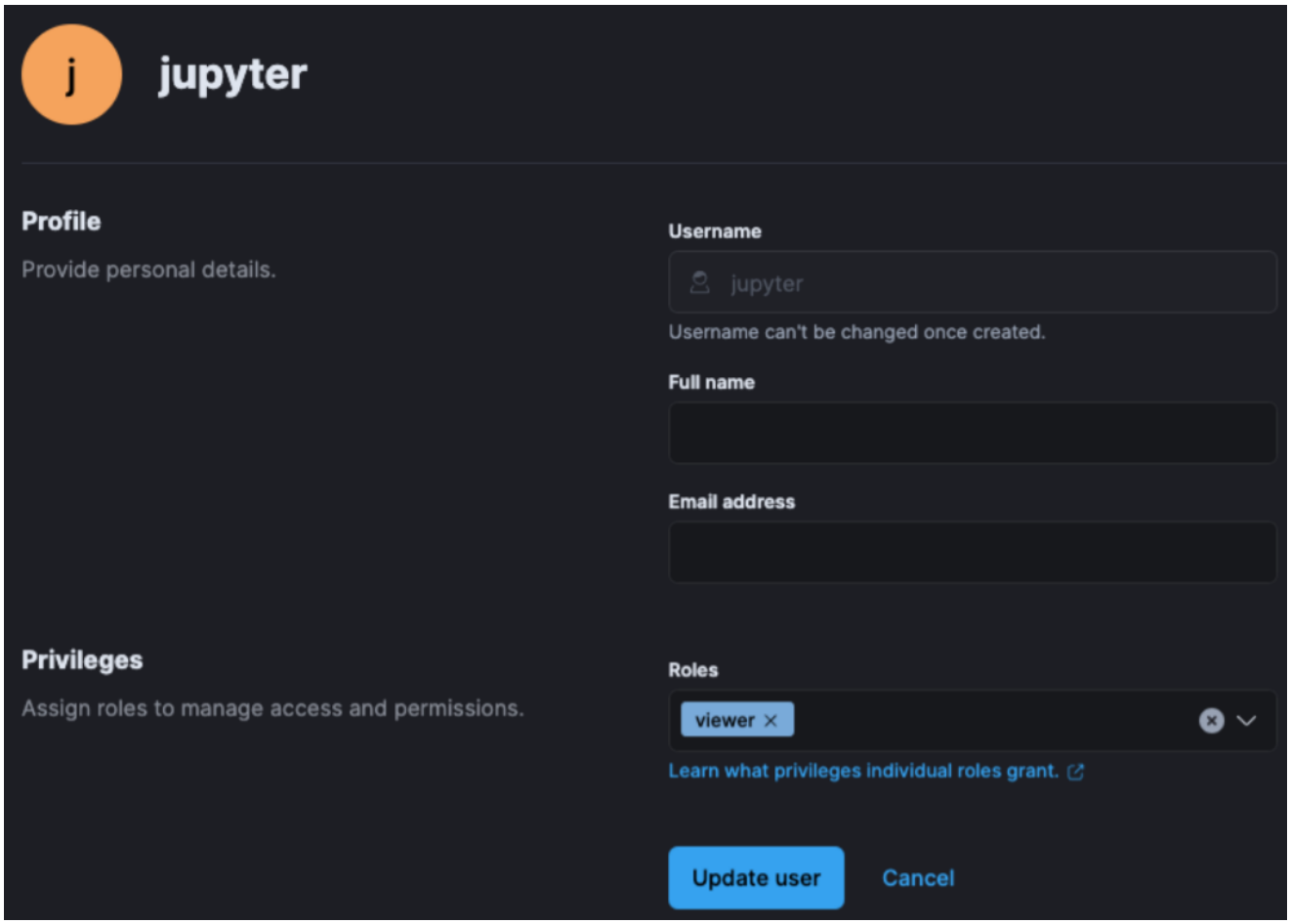


4. Click Ok till Sqert included in the system (five times)

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.



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

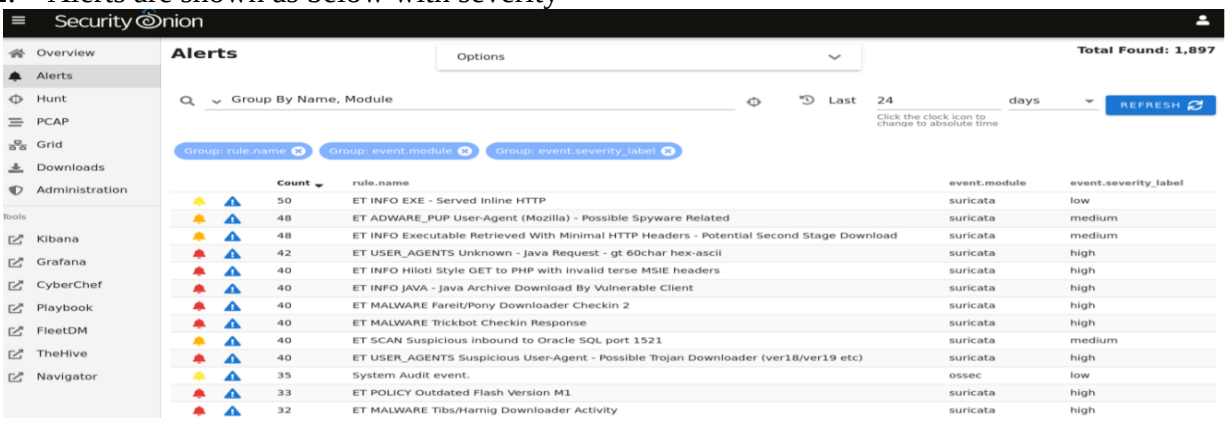
```
es = Elasticsearch(['https://192.168.6.100:9200'],
ca_certs=False,verify_certs=False, http_auth=('jupyter','password'))
searchContext = Search(using=es, index='*:so-*', doc_type='doc')
```

6. Analysis

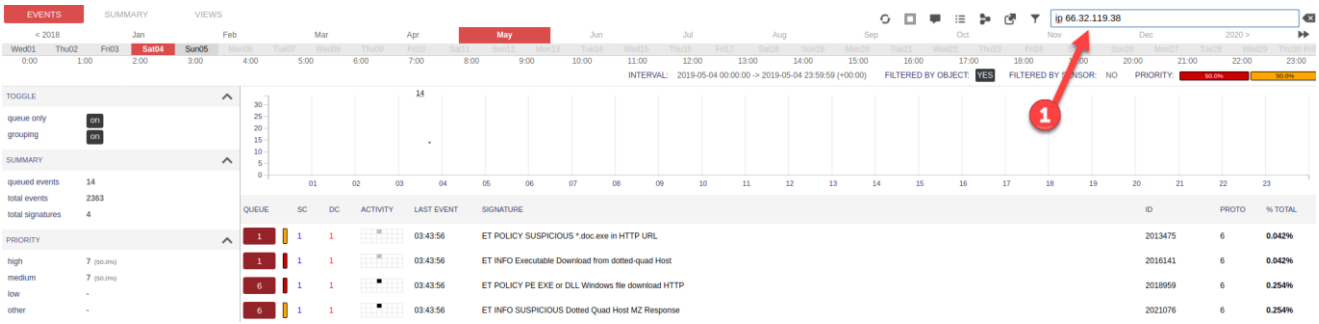
1. Following commands are ran to import packet capture

```
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



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-511E-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
```

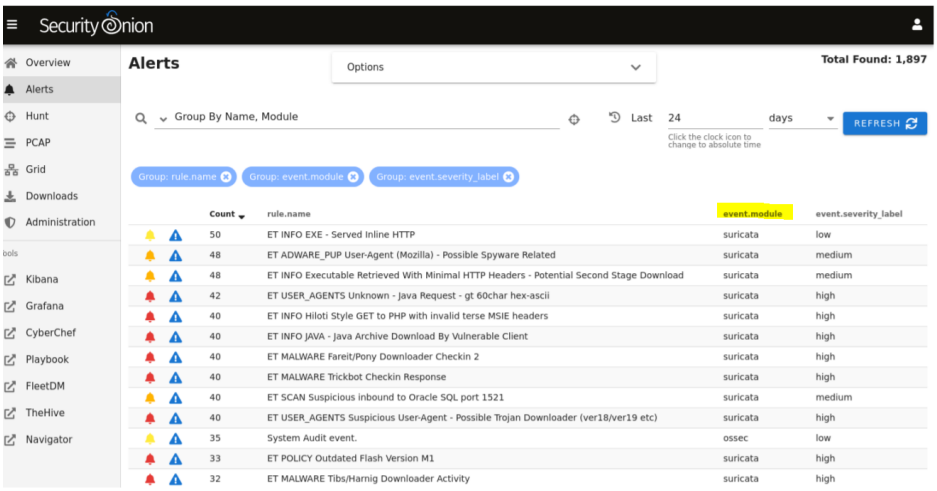
8. After applying these commands, the results look like:

	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



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