



Implementation, configuration, and investigation of a SOC using Open-Source Tools

**Abel Melinte
B00137882**

*Department of Informatics,
School of Informatics and Engineering,
Technological University Dublin,
Dublin 15*

[Page Count: 30

Word Count: 3258]

**Network Security Analytics
Digital Forensics and Cyber Security
05/12/2023**



***Department of Informatics,
School of Informatics and Engineering,
Technological University Dublin,
Dublin 15***

Declaration On Plagiarism

I declare that the work we are submitting for assessment by the Institute examiner(s) is entirely our own work, except where the author or source has been duly referenced and attributed.

We confirm that this material has not been previously submitted for a degree or any other qualification at TUD or any at other institution.

We further confirm that we have read and understood the Institute policy on plagiarism in assignments and examinations (3AS08.doc) and that we are not, as far as we are aware, in breach of any of these regulations.

Names: Abel Melinte

Student IDs: B00137882

Signed: Abel Melinte

Date: 05/12/2023

Abstract

This paper describes how to create and run an extensive Security Operations Centre (SOC) with only open-source tools. The study centres on the identification and utilisation of open-source technologies that are customised for blue teams. These tools are then integrated into a running system to strengthen defences and proactively address cyber threats.

Using the SOC framework, the study applies and incorporates several open-source tools. In order to verify the system's responsiveness, it investigates several attack techniques and carries out at least three attacks. The focus is on showcasing how these assaults can be used to provide precise and useful alerts.

Table of Contents

Abstract.....	3
1. Introduction	1
1.1 What is a Security Operation Center (SOC)	1
1.2 Objective	1
1.3 Research of Open-Source Tools for Blue Team	1
1.3.1 Snort.....	1
1.3.2 Splunk Universal Forwarder	1
1.3.3 Splunk Enterprise	1
1.3.4 UFW Firewall	2
1.3.5 Wazuh	2
1.3.6 Pfsense	2
2. Implementation and Integration of Tools.....	2
2.1 Operating Systems & Network Used	2
2.2 Open-Source Tools Installation	2
2.2.1 Snort Installation	2
2.2.2 Splunk Enterprise Installation	4
2.2.3 Splunk Universal Forwarder	6
3. Snort Rules Configuration	7
4. Research of Attack Methods.....	9
4.1 What is Hydra? [Brute Force Attack]	9
4.2 What is HPing3? [Denial Of Service Attack]	9
4.3 Cross-Site Scripting	9
4.4 What is Nikto? [Website Vulnerability Scanner]	10
4.5 Metasploit	10
5. Implementation of Attack Scenarios.....	10
5.1 Hydra Attack Procedure	10
5.2 HPing3 Attack Procedure	14

5.3 Nikto Attack Procedure.....	16
6. Pivoting from Alerts to Attack Details.....	19
6.1.1 Pivoting Hydra Alert.....	19
6.1.2 Pivoting Nikto Attack	19
6.1.3 Pivoting HPing3	20
7. Conclusion.....	20
8. References	22

Figure 1 Installing Snort	3
Figure 2 Snort Network Assign.....	3
Figure 3 Enable Promiscuous Mode	3
Figure 4 Adding Victim Machine	3
Figure 5 Downloading Splunk	4
Figure 6 Installing Splunk	4
Figure 7 Starting Splunk	5
Figure 8 Splunk Login Localhost.....	5
Figure 9 Splunk Settings.....	6
Figure 10 Outputs.conf	7
Figure 11 Inputs.conf	7
Figure 12 Snorpy 2.0	8
Figure 13 Alerts	8
Figure 14 Hydra SSH.....	11
Figure 15 Hydra Password	11
Figure 16 Output Result.....	12
Figure 17 Splunk Results	12
Figure 18 Hydra FTP Setting.....	13
Figure 19 Hydra Telnet.....	13
Figure 20 Hydra Telnet Setting	13
Figure 21 Telnet Result	14
Figure 22 Hydra SMTP.....	14
Figure 23 Splunk SMTP.....	14
Figure 24 HPing3 Attack TCP.....	15
Figure 25 Alerts for HPing	15
Figure 26 Results in Splunk	15
Figure 27 HPing3 Command UDP.....	15
Figure 28 HPing3 Results in Splunk.....	16
Figure 29 Nikto Command	17
Figure 30 Nikto Alert	17
Figure 31 Splunk Dashboard	17
Figure 32 XSS Attempt	18
Figure 33 XSS Working	18
Figure 34 Wireshark Result Hydra	19

Figure 35 Wireshark	19
Figure 36Wireshark.....	20
Figure 37Wireshark.....	20

1. Introduction

1.1 What is a Security Operation Center (SOC)

A Security Operation Center is a portion within an organization, its full purpose is to be responsible for monitoring, detecting, analysing, and responding to any malicious traffic which is discovered by them. Within a SOC, it is important to have a high number of well-experienced employees as they have the ability to prevent any attack as they would be extremely strategic after having all the experience built up.

1.2 Objective

In this assignment, the main purpose of it was to build and have a fully functionable open-source SOC, this was done by researching and using open-source tools for blue teams, implementing a minimum of three attacks, creating alerts within the open-source tool that detects each incoming attack and finally pivoting the alerts to get a more in-depth information on the attack.

1.3 Research of Open-Source Tools for Blue Team

1.3.1 Snort

Snort has been used throughout this assignment, its main purpose is to focus on the network and to alert the user whenever there is suspicious network activity occurring within the SOC. Snort can alert rules from any receiving connection or attempts, all what is needed to be done is to edit the rules list within the folder where in that location you set alerts for specific incoming traffic such as connection from a continues suspicious IP or even just assign a port to alert the user.

1.3.2 Splunk Universal Forwarder

Splunk Universal Forwarder was also used within this operation, its main objective is to forward any alerts that get alerted in snort towards the main Splunk enterprise so the data can be viewed at a more in-depth detail than just an alert itself.

1.3.3 Splunk Enterprise

Splunk Enterprise was the main tool, which was used throughout this procedure, its overall purpose was to receive incoming alerts from Snort (Snort -> Splunk Universal Forwarder -> Splunk Enterprise) and have the ability to view the data in a more in-depth detail than just seeing it as an alert in the Snort console.

1.3.4 UFW Firewall

Throughout this process, it was not a task to mitigate each attack, but in case of a real-life scenario, UFW Firewall was used towards the end just to show an example of preventing on receiving attacks from specific ports. It was done by an addition of rules being imported via the command-line interface.

1.3.5 Wazuh

Wazuh is a security monitoring platform that is suitable for enterprise use. It is an open-source, free platform used for integrity monitoring, incident response, and threat detection. Workloads in cloud-based, virtualized, on-premises, and containerised environments can all be safeguarded by Wazuh.

1.3.6 Pfsense

PfSense is an operating system for routers and firewalls that is free and open source. The majority of commodity hardware, including outdated PCs and embedded systems, can run PfSense. PfSense is commonly configured and managed via an intuitive web interface, which simplifies administration for individuals with no networking expertise as well. In most cases, configuring a router doesn't require using a terminal or editing configuration files. From the web UI, software updates can also be performed.

2. Implementation and Integration of Tools

2.1 Operating Systems & Network Used

Victim – Ubuntu 22.04.3 – 192.168.27.136

Attack – Kali Linux 2023.4 – 192.168.1.130

Snort & Splunk Forwarder – Ubuntu 22.04.3 - 192.168.27.106

Splunk Enterprise - Ubuntu 22.04.3 - 192.168.1.153

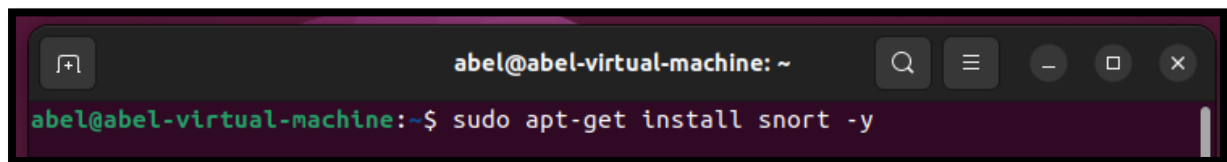
2.2 Open-Source Tools Installation

In this section, it will be shown how each tool has been installed and configured on each machine so it can gain the ability to receive incoming traffic and also be able to alert any suspicious traffic such as repetitive pinging frequently, specific ports etc.

2.2.1 Snort Installation

On one of the Ubuntu Machines, Snort was installed and configured, here are the steps of the installation and configurations done.

Implementation, configuration, and investigation of a SOC using Open-Source Tools



```
abel@abel-virtual-machine: ~  
abel@abel-virtual-machine:~$ sudo apt-get install snort -y
```

Figure 1 Installing Snort

Here shows how Snort package was installed via CLI, the -y at the end of the command is just to skip any agreement that needs to be accepted.

Once downloaded, it forwards over to the next step which was to assign the address of the local network 192.168.27.0 was assigned as a local network as each machine has an IP of 192.168.27.*.

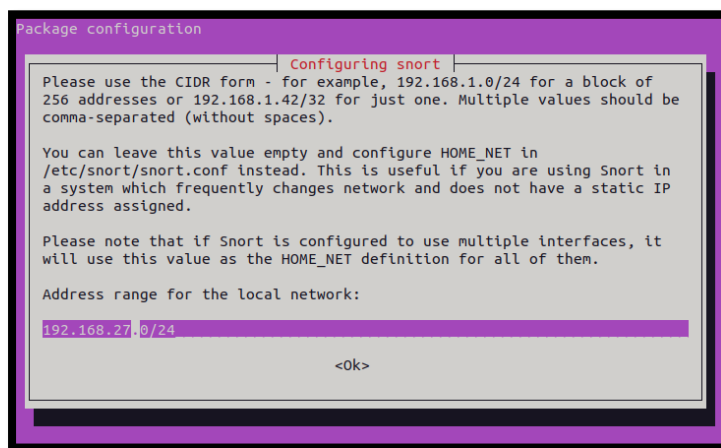
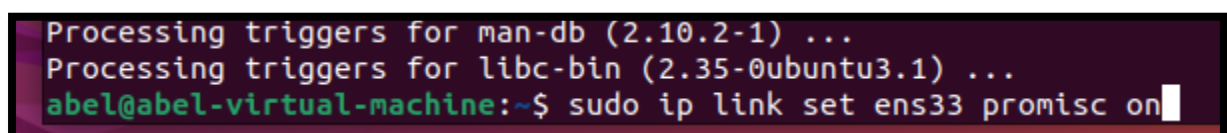


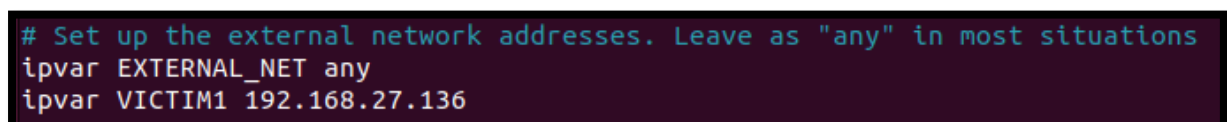
Figure 2 Snort Network Assign



```
Processing triggers for man-db (2.10.2-1) ...  
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...  
abel@abel-virtual-machine:~$ sudo ip link set ens33 promisc on
```

Figure 3 Enable Promiscuous Mode

After the network was assigned, it was needed to assign the card name which in this case was “ens33” and enable promiscuous mode. When promiscuous is enabled, it allows the machine to read each network packet that arrives.



```
# Set up the external network addresses. Leave as "any" in most situations  
ipvar EXTERNAL_NET any  
ipvar VICTIM1 192.168.27.136
```

Figure 4 Adding Victim Machine

Implementation, configuration, and investigation of a SOC using Open-Source Tools

In the `/etc/snort/snort.conf`, the file had to be configured, the image above shows that an external IP has been added which in our case it is the Victim machine which will be attacked so when the Victims receives attacks, Snort will be able to generate alerts from that IP. That is Snort successfully installed and functioning, the next step will be assigning rules for alert detection but that will be further explained in the document.

2.2.2 Splunk Enterprise Installation

In this section, a step-by-step guide will show how Splunk Enterprise was installed and configured on the Ubuntu machine. First what was needed to be done in order to download Splunk Enterprise, was to create an account for Splunk, which will then give us access to the Trial Download.

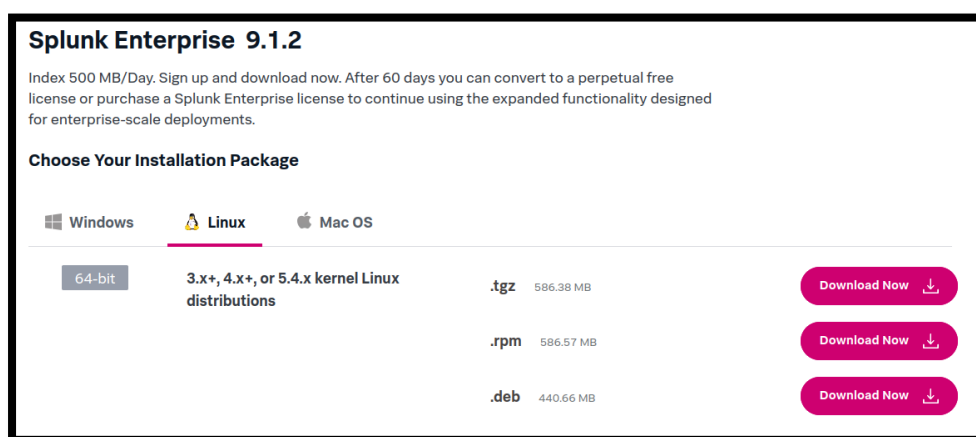


Figure 5 Downloading Splunk

While on the download page, the .tgz file is meant to be downloaded.

```
splunk/etc/deployment-apps/  
splunk/etc/deployment-apps/README  
abel@abel-virtual-machine:~/Desktop$ sudo tar xvzf splunk-9.1.2-b6b9c8185839-Linux-x86_64.tgz -C /opt/
```

Figure 6 Installing Splunk

The command to install the .tgz file was executed and in the command, it was specified to download the specific file in the `/opt` directory. Once the file has been fully installed in the `/opt` directory, all what was needed was to start Splunk, the command for that was “`sudo ./splunk start`” the command would only work if you were in the specific directory.

Implementation, configuration, and investigation of a SOC using Open-Source Tools

```
abel@abel-virtual-machine:~/Desktop$ cd /opt/splunk/bin/  
abel@abel-virtual-machine:/opt/splunk/bin$ sudo ./splunk start
```

Figure 7 Starting Splunk

After Splunk was started, all that was needed was to navigate to your localhost with the 8000 port which would look like this “localhost:8000”.

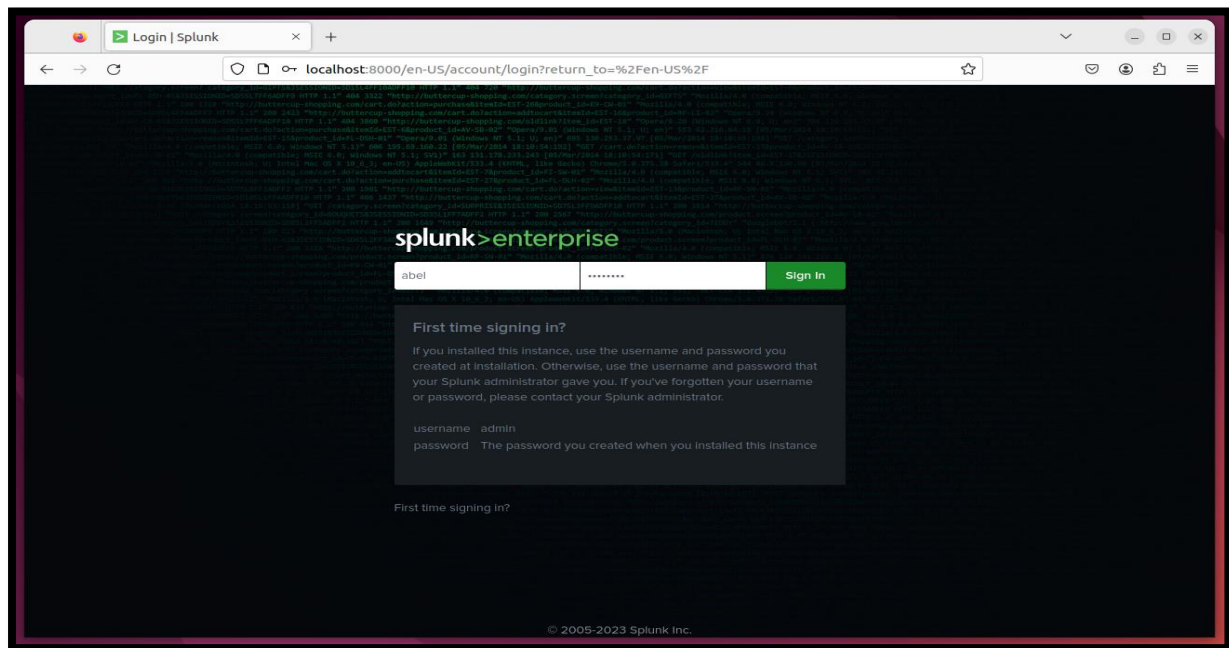


Figure 8 Splunk Login Localhost

After logging in, a dashboard has opened where all the configuration was done.

Implementation, configuration, and investigation of a SOC using Open-Source Tools

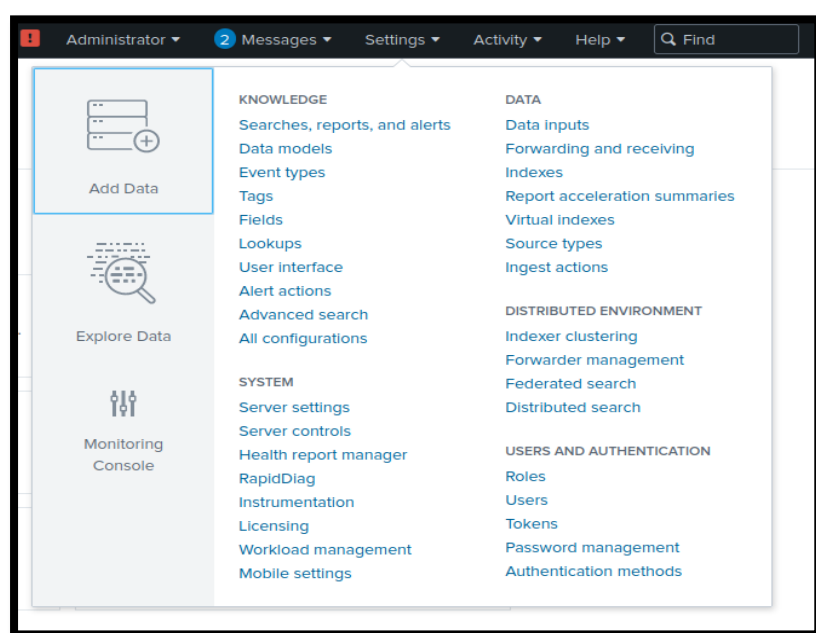


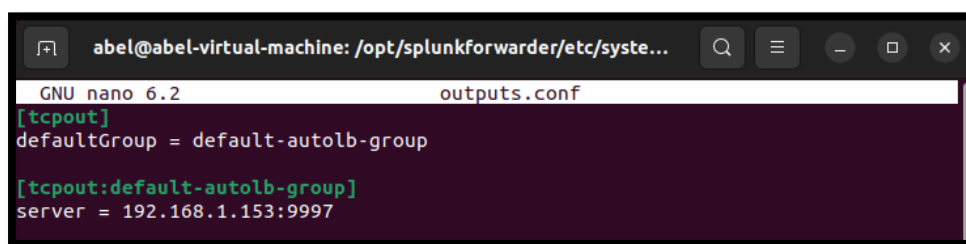
Figure 9 Splunk Settings

In the top NAV bar, you navigate over to “Settings” and select “indexes”, Create a “New Index”, and assign the name “snort” all lowercase, it will be explained why this specific name had to be assigned, then everything was left default and next. Headed back to the settings and selected Forwarding and Receiving, scrolled down to where receive data was located and a new data was added, the port that was entered was port 9997.

2.2.3 Splunk Universal Forwarder

The final tool in relation to data forwarding was Splunk Universal Forwarder, here is a step-by-step process on how it was installed and also configured.

Splunk Universal was again downloaded from the Splunk official website as a .tgz file. Before downloading it, it had to be installed on the machine where Snort was installed as if Splunk Forwarder was installed on the same machine where the Enterprise is located, it would not be able to download as they would be using the same ports which will prevent the installation from happening. The process is the exact same as the Enterprise so there isn’t a need of explanation, the only difference is the different file, which is downloaded apart from that, the same commands were used. One thing that had to be added was an **outputs.conf** and **inputs.conf** file, this is how it was done. Navigated to `/opt/splunkforwarder/etc/system/local` and then ran the “`sudo nano output.conf`” and created a file with these assigned rules.

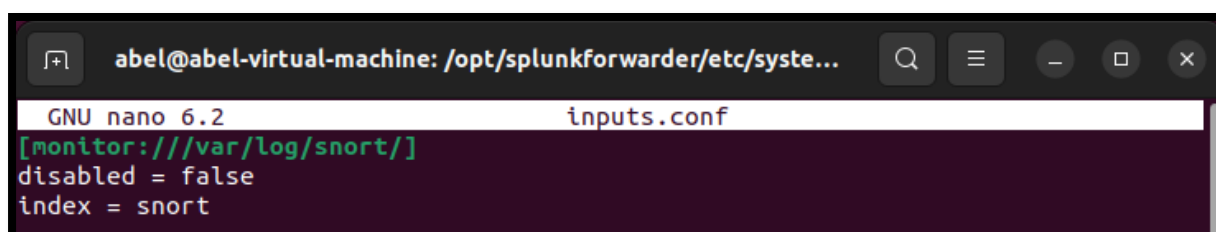


```
abel@abel-virtual-machine: /opt/splunkforwarder/etc/syste...
GNU nano 6.2 outputs.conf
[tcpout]
defaultGroup = default-autolb-group

[tcpout:default-autolb-group]
server = 192.168.1.153:9997
```

Figure 10 Outputs.conf

This configuration just allows the data to be forwarded to the machine which Splunk Enterprise is installed on and on the port which was assigned in Splunk Enterprise dashboard. Then an input.conf was created, this was just to assign the index so when you navigate to the Splunk dashboard, you just search for “index=snort” and the alerts will display.



```
abel@abel-virtual-machine: /opt/splunkforwarder/etc/syste...
GNU nano 6.2 inputs.conf
[monitor:///var/log/snort/]
disabled = false
index = snort
```

Figure 11 Inputs.conf

3. Snort Rules Configuration

Snort does not generate any alerts unless it's configured to do so, navigating to the Snort Rules directory contains a file called local.rules, this file a rules file that allows you to enter your own custom rules for specific attacks on the local network. Creating rules could be a little bit challenging, so the tool that was used to help out with the situation was Snorpy 2.0, this tool allows you to just fill in the needed information and it generates a rule which you then just implement into the local.rules. Here is an example if I want to create an alert for just receiving pings.

Implementation, configuration, and investigation of a SOC using Open-Source Tools

The screenshot shows the SNORPY web interface, titled "A Web Based Snort Rule Creator / Maker for Building Simple Snort Rules". The interface includes a rule configuration section with the following fields: "alert" (dropdown), "icmp" (dropdown), "any" (text), "source port" (text), "any" (text), "dest port" (text), "10001" (text), and "1" (text). Below these is a "Pings Incoming" text field, a "Class-Type" dropdown, a "Priority" dropdown, and a "gid" text field. The "ICMP" section has "ICMP TYPE" and "ICMP CODE" dropdowns. The "Data Size" and "Reference" sections have dropdowns and text input fields. The "Threshold Tracking Type" section has a dropdown, "TRK BY" dropdown, "Count #", and "Seconds" text input fields. On the right, there are "Add Content Match" and "Add Regex Match" buttons with green plus icons. At the bottom, a preview box shows the generated rule: "alert icmp any any -> any any (msg:'Pings Incoming'; sid:10001; rev:1;)".

Figure 12 Snorpy 2.0

Generated Alert : alert icmp any any -> any any (msg:"Pings Incoming"; sid:10001; rev:1;)

Here is the local.conf after being configured to alert HPing3 Attack, Hydra Brute Force Attack and

```
abel@abel-virtual-machine: /etc/snort/rules
GNU nano 6.2 local.rules *
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures.  Put your local
# additions here.

#HPING3
alert icmp 192.168.27.130 any -> any 80 (msg:"HPing3 Detected"; sid:100003; rev:1;)
alert udp 192.168.27.130 any -> any 80 (msg:"HPing3 Detected"; sid:100004; rev:1;)
alert tcp 192.168.27.130 any -> any 80 (msg:"HPing3 Detected"; sid:100005; rev:1;)

#HPING3 TEST
#alert tcp 192.168.27.130 any -> 192.168.27.136 80 ( msg:"HPing3 Detected"; dsize:>64; sid:10001; rev:1; )
#alert icmp 192.168.27.130 any -> 192.168.27.136 80 ( msg:"HPing3 Detected"; dsize:>64; sid:10002; rev:1; )
#alert udp 192.168.27.130 any -> 192.168.27.136 80 ( msg:"HPing3 Detected"; dsize:>64; sid:10003; rev:1; )

#HYDRA
alert tcp 192.168.27.130 any -> 192.168.27.136 22 (msg:"Hydra Detected [TCP]!"; sid:10004; rev:1;)
alert tcp 192.168.27.130 any -> 192.168.27.136 21 (msg:"Hydra Detected[FTP]!"; sid:10005; rev:1;)
alert tcp 192.168.27.130 any -> 192.168.27.136 23 (msg:"Hydra Detected[TELNET]!"; sid:10006; rev:1;)
alert tcp 192.168.27.130 any -> 192.168.27.136 25 (msg:"Hydra Detected[SMTP]!"; sid:10007; rev:1;)

#NIKTO
alert tcp 192.168.27.130 any -> 192.168.27.136 80 (msg:"Nikto Vulnerability Scanner Detected!"; sid:10008; rev:1;)
```

Figure 13 Alerts

The tools which the alerts are planned on detecting are HPing3, Hydra, and Nikto.

4. Research of Attack Methods

After each alert was set, the next plan was to navigate to the tools which I want to use to perform the attacks related to the alerts. Here is what attacks were executed.

Tool	Port	Attack Type
Hydra Graphical	22 / 21 / 23 / 25	Brute Force Attack
HPing3	80	Denial Of Service Attack
Nikto	80	Web Vulnerability Scanner

These were the tools which have been used throughout the attack process as they were the most recommended tools to use for a local network attack, the downside to the tools is that they are very well made and straight forward to use which can lead anyone new who is unaware of they are doing to perform a malicious attack.

4.1 What is Hydra? [Brute Force Attack]

Hydra is a tool that comes built-in with Kali Linux Operating System. The tool is extremely powerful, and its main purpose is to crack usernames and passwords within couple seconds depending on how difficult the username and password is, it is able to attack across 47 protocols but throughout this assignment the protocols which were chosen were SSH, UDP, TELNET, SMTP. If you are using a machine that doesn't include Hydra, it has a simple installation process regardless. The attacks that Hydra execute are call Brute Force Attacks, which also means cracking password etc.

4.2 What is HPing3? [Denial Of Service Attack]

HPing3 is a command-line tool which comes built in with Kali Linux. Its mainly known to be used as a ping flood tool also known as Denial-of-Service attack. Its main features are packet manipulation, flood attack, simulations, and security auditing. Hping3 provides an indispensable toolkit for comprehensive network analysis, enhancing understanding and fortifying network defences of course while being used within the legal boundaries.

4.3 Cross-Site Scripting

Injection attacks known as Cross-Site Scripting (XSS) occur when malicious scripts are inserted into websites that are otherwise trustworthy and safe. XSS attacks happen when a hacker sends malicious code—typically in the form of a browser side script—to a separate end user through an online application. The vulnerabilities that facilitate the success of these attacks are relatively

common and arise whenever a web application incorporates user input without verifying or encoding it in the output it produces.

4.4 What is Nikto? [Website Vulnerability Scanner]

Nikto is an essential component of Kali Linux that is used to check web servers for security flaws. In order to find known vulnerabilities, out-of-date software, and misconfigurations that could leave systems vulnerable to exploitation, Nikto conducts thorough scans on servers. Its capabilities include examining SSL/TLS setups, spotting possible vulnerabilities, and producing detailed reports that summarise issues that are found.

4.5 Metasploit

The most popular open-source penetration testing framework in the world, Metasploit is used by security experts as a platform for developing security tools and exploits as well as for penetration testing. Hacking is made easy by the architecture for both attackers and defences. A user can configure an exploit module, combine it with a payload, point at a target, and fire it against the target system using the many Metasploit tools, libraries, user interfaces, and modules. Hundreds of exploits and a variety of payload options are stored in Metasploit's vast and comprehensive database.

5. Implementation of Attack Scenarios

After the tools were decided on which do use and the alerts were created etc, here will be a demonstration of how each tool has been used to perform the malicious attack towards the victim computer.

5.1 Hydra Attack Procedure

The advantage of using Kali Linux is that the entire OS comes with every tool already installed so there is no need for any installation of tools, its all there to use straight away. When Hydra is open, we are instantly given an attack page where it is required to fill in information where needed.

Implementation, configuration, and investigation of a SOC using Open-Source Tools

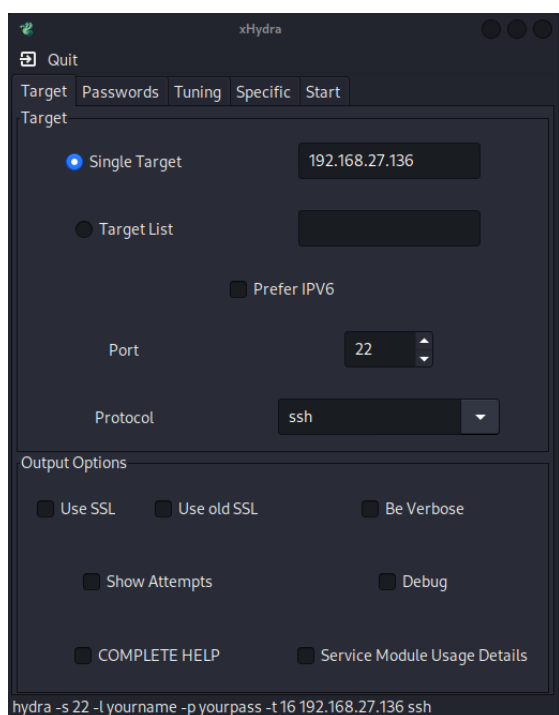


Figure 14 Hydra SSH

Here shows the image of Hydra with the victim machine IP implemented, basically it is telling the machine to attack that specific IP on port 22 with the ssh protocol, moving onto the password tab,

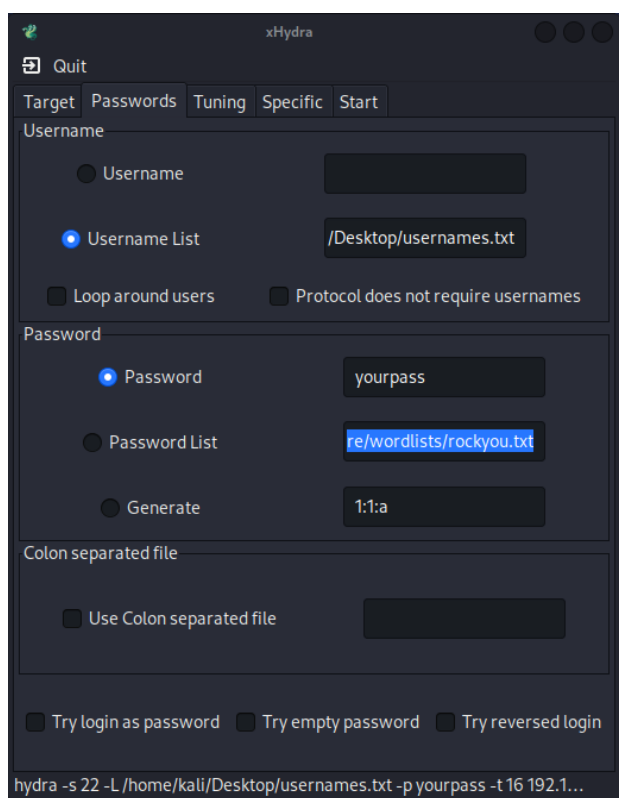
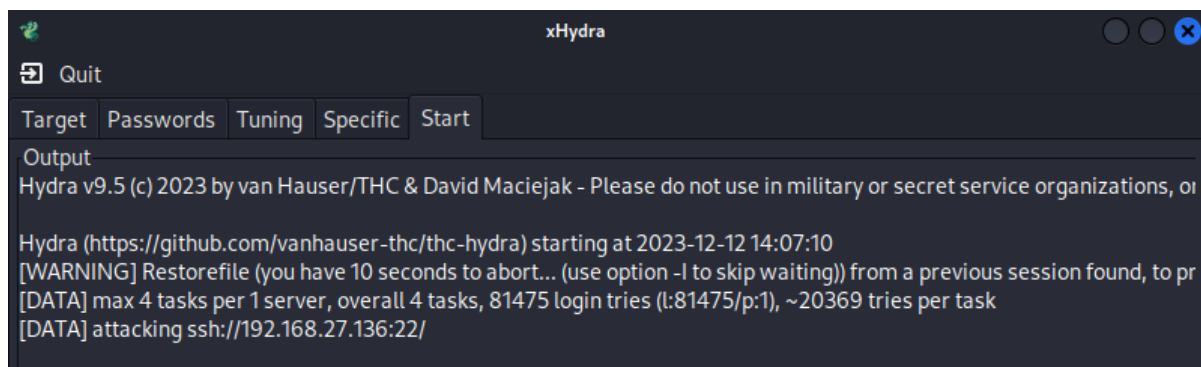


Figure 15 Hydra Password

Implementation, configuration, and investigation of a SOC using Open-Source Tools

Username list has been checked which the file will then be used to crack the username of the machine and for the password it is telling it to use the rockyou.txt to crack the password. Once this information is filled in, the attack is ready to go.

Here is the output from Hydra.



```
xHydra

Quit

Target Passwords Tuning Specific Start

Output
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-12-12 14:07:10
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to pr
[DATA] max 4 tasks per 1 server, overall 4 tasks, 81475 login tries (l:81475/p:1), ~20369 tries per task
[DATA] attacking ssh://192.168.27.136:22/
```

Figure 16 Output Result

So basically, what's happening now is the attack is going towards the victims IP which passes through the rules in Snort which then the alerts should be generate on the Splunk Dashboard on the separate Ubuntu Machine.

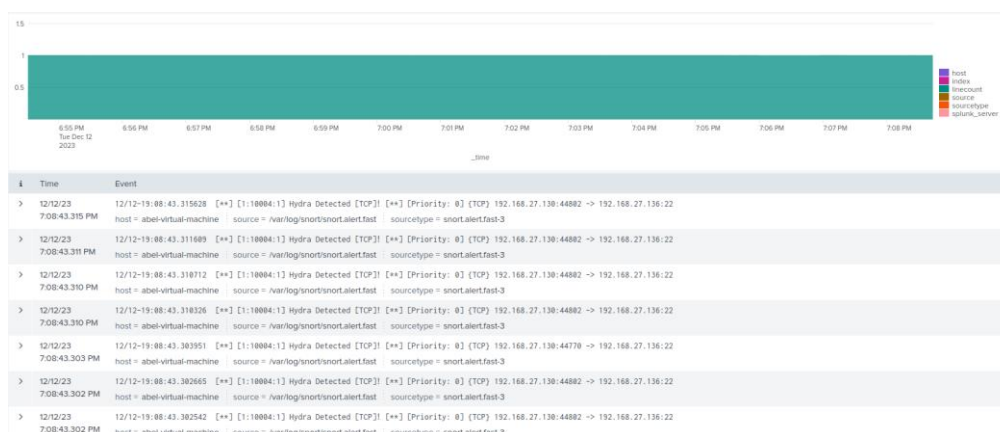


Figure 17 Splunk Results

Now that the attack has been done on the TCP port, the next move is to perform the attack on the other 3 ports just to prove that it alerts detect for ports also.

Implementation, configuration, and investigation of a SOC using Open-Source Tools



Figure 18 Hydra FTP Setting

In the image above it just shows that the port and protocol is being changed everything else stays the same. Here are the results in Splunk after running this attack.



Figure 19 Hydra Telnet

Next is Hydra Telnet Attack



Figure 20 Hydra Telnet Setting

Changing the port and protocol while the rest remains the same.

Implementation, configuration, and investigation of a SOC using Open-Source Tools



Figure 21 Telnet Result

Finally, the final result test for Hydra attack is on port 25 which is SMTP.



Figure 22 Hydra SMTP

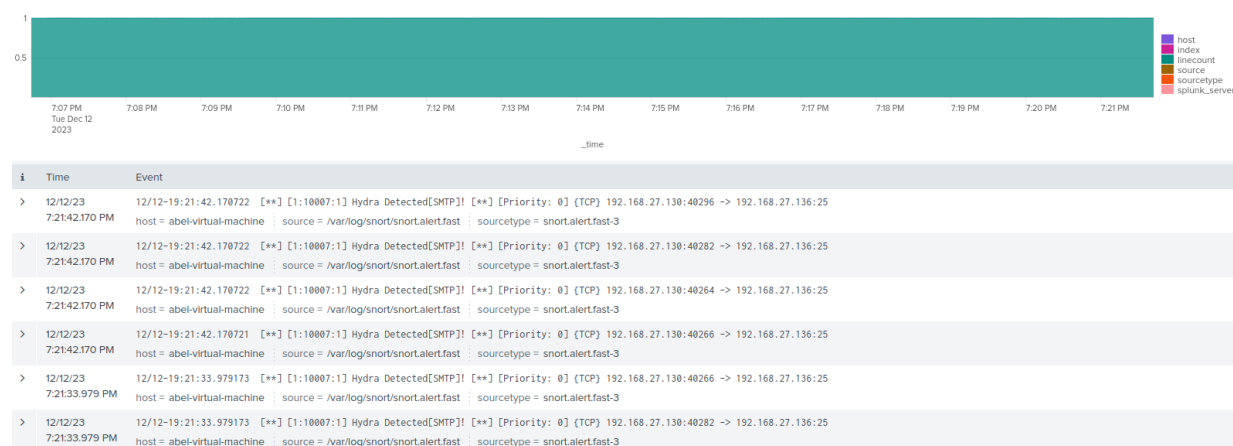


Figure 23 Splunk SMTP

5.2 HPing3 Attack Procedure

The next attack, HPing3 was used to perform Denial of Service attacks. While using the help that is provided to you automatically when HPing3 is opened, an attack was created which is ready to be execute and flood the victim machine. Here is what the command looks like.

Implementation, configuration, and investigation of a SOC using Open-Source Tools

```
(kali㉿kali)-[~]  
$ sudo hping3 -c 1 -d 64 -p 80 --flood 192.168.27.136
```

Figure 24 HPing3 Attack TCP

Here is a little breakdown of what the command sends to the victim, when it's attacking its set to only send one packet per execution with a data size of 64, and to be direct to port 80, also declared a `--flood` which will then make Hping3 send packets faster than normal. Here is what alert gets selected when this command is executed:

```
#HPING3  
alert icmp 192.168.27.130 any -> any 80 (msg:"HPing3 Detected"; sid:100003; rev:1;)  
alert udp 192.168.27.130 any -> any 80 (msg:"HPing3 Detected"; sid:100004; rev:1;)  
alert tcp 192.168.27.130 any -> any 80 (msg:"HPing3 Detected"; sid:100005; rev:1;)
```

Figure 25 Alerts for HPing

Here are the results in Splunk when this attack is executed in the console. (TCP)

Time	Event
12/12/23 7:59:37.832 PM	host = abel-virtual-machine : source = /var/log/snort/snort.alert.fast : sourcetype = snort.alert.fast-3
12/12/23 7:59:37.832 PM	host = abel-virtual-machine : source = /var/log/snort/snort.alert.fast : sourcetype = snort.alert.fast-3
12/12/23 7:59:37.831 PM	host = abel-virtual-machine : source = /var/log/snort/snort.alert.fast : sourcetype = snort.alert.fast-3
12/12/23 7:59:37.831 PM	host = abel-virtual-machine : source = /var/log/snort/snort.alert.fast : sourcetype = snort.alert.fast-3
12/12/23 7:59:37.831 PM	host = abel-virtual-machine : source = /var/log/snort/snort.alert.fast : sourcetype = snort.alert.fast-3

Figure 26 Results in Splunk

Command to execute the same command but UDP this time.

```
(kali㉿kali)-[~]  
$ sudo hping3 -c 1 -d 64 -p 80 --flood 192.168.27.136 --udp
```

Figure 27 HPing3 Command UDP

Implementation, configuration, and investigation of a SOC using Open-Source Tools

Here are result after running the command for the UDP flood.

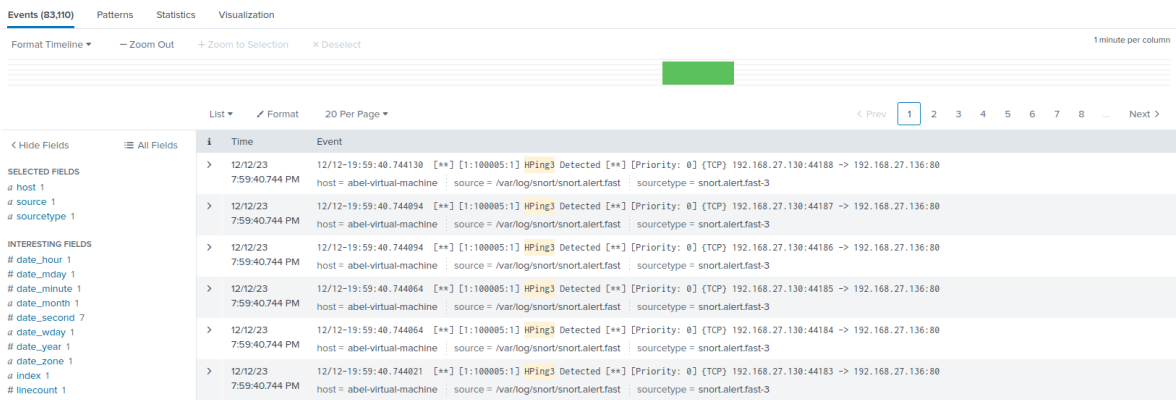
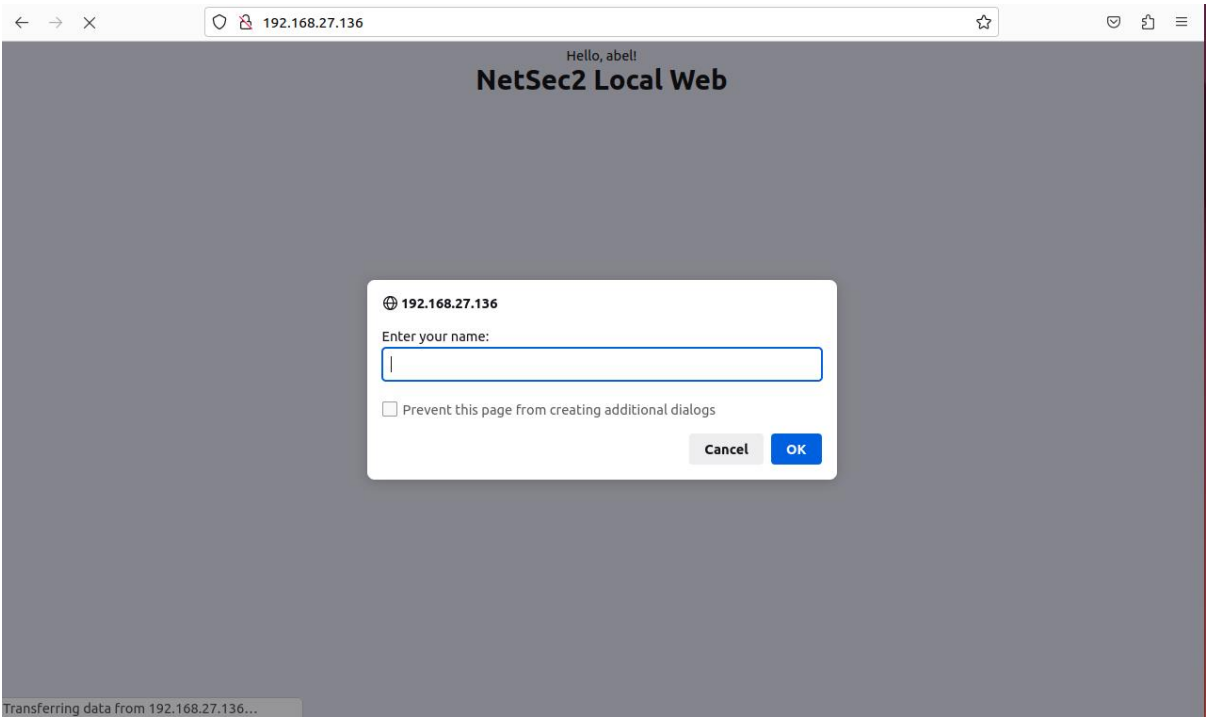


Figure 28 HPing3 Results in Splunk

5.3 Nikto Attack Procedure

The following attack is using Nikto to scan possible vulnerabilities within a webpage on a machine. Having a webpage created on the victim machine.



Here is the result when the command has been executed via Kali Linux.

Implementation, configuration, and investigation of a SOC using Open-Source Tools

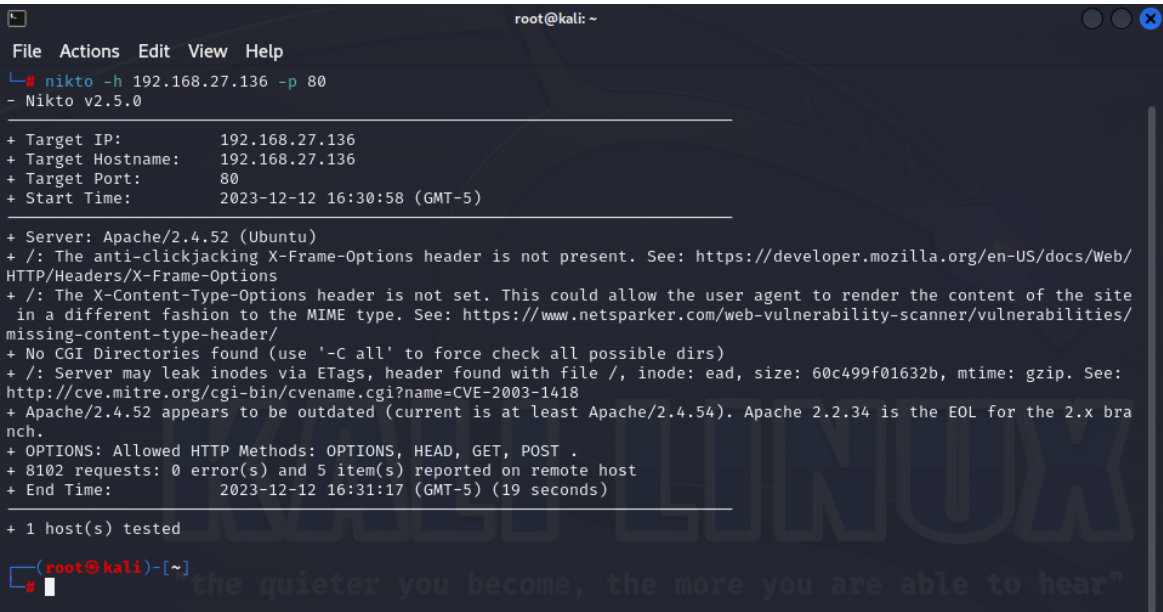


Figure 29 Nikto Command

Snort rule that creates this alert:

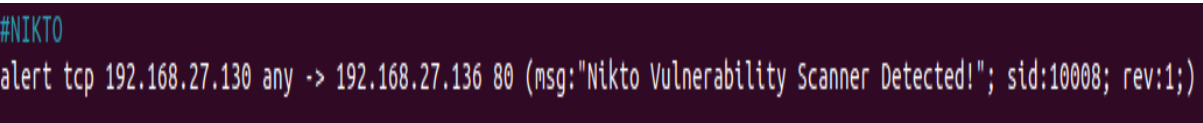


Figure 30 Nikto Alert

Also here is the Splunk result from the Nikto attack which has occurred:

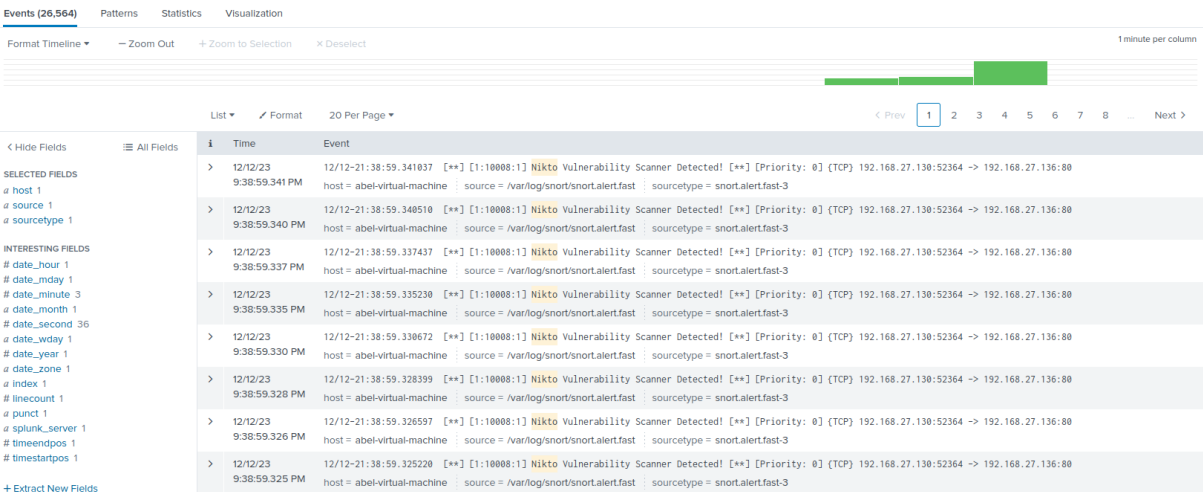


Figure 31 Splunk Dashboard

Also, an XSS script has been attempted but unfortunately it does not forward towards Splunk Dashboard.

Implementation, configuration, and investigation of a SOC using Open-Source Tools

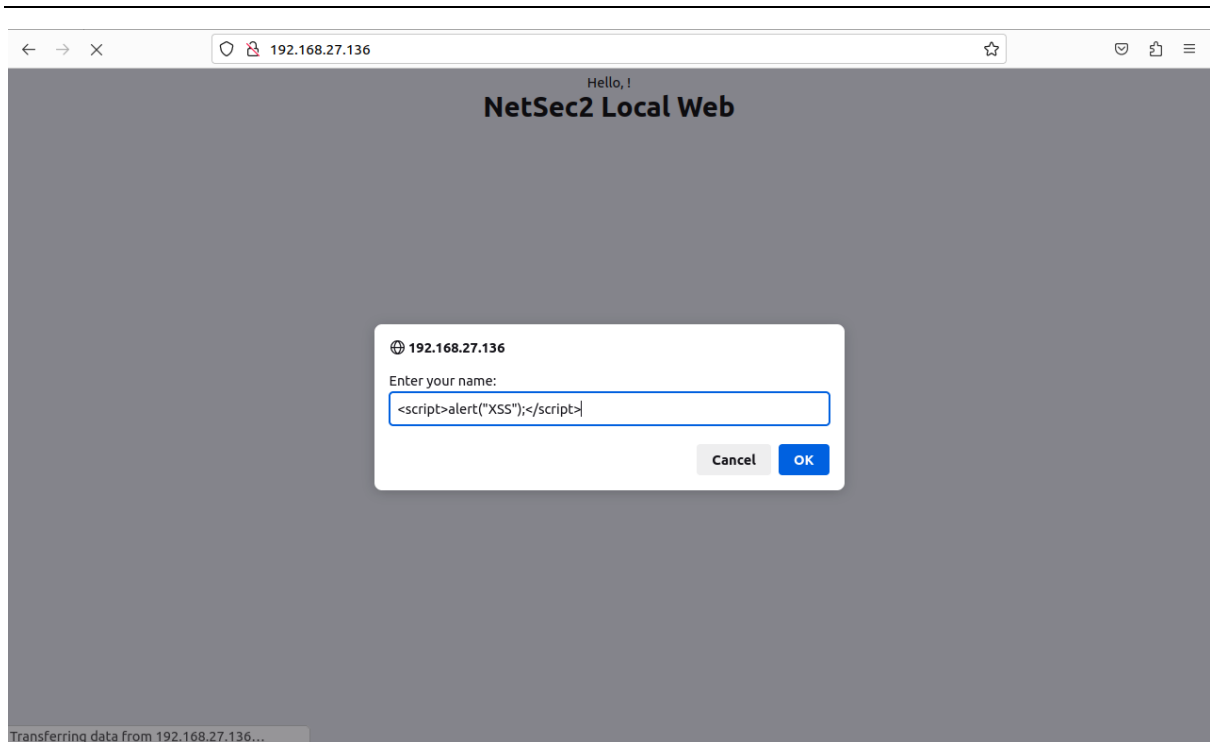


Figure 32 XSS Attempt

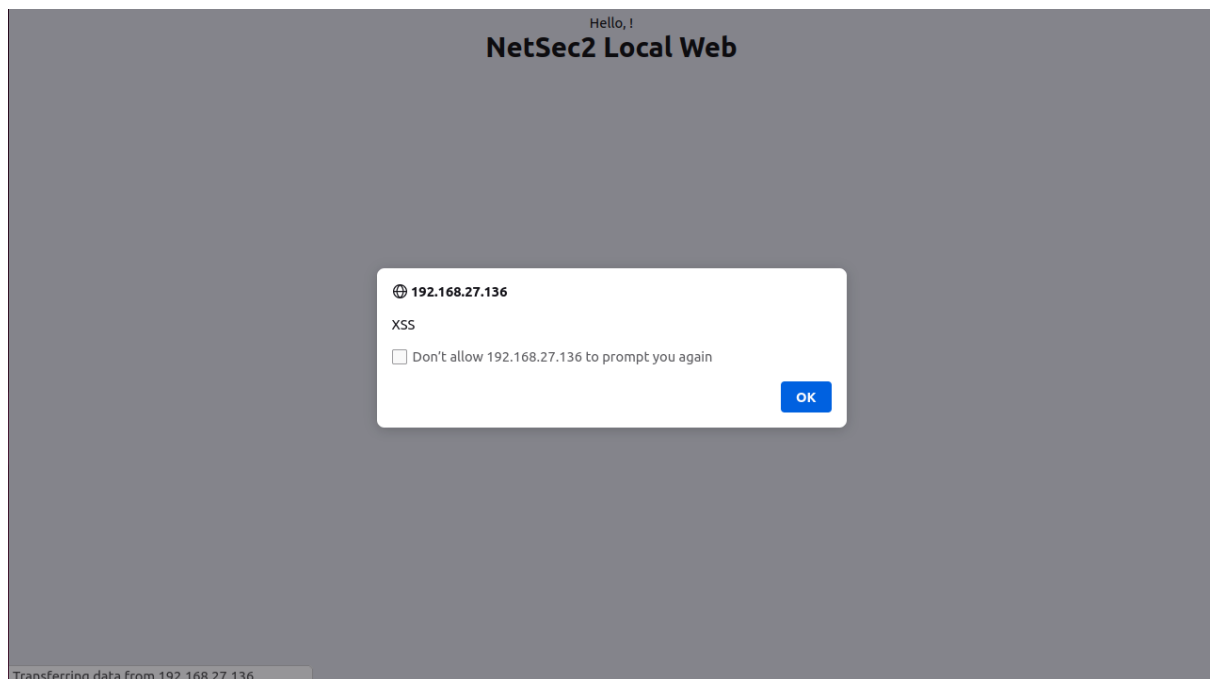


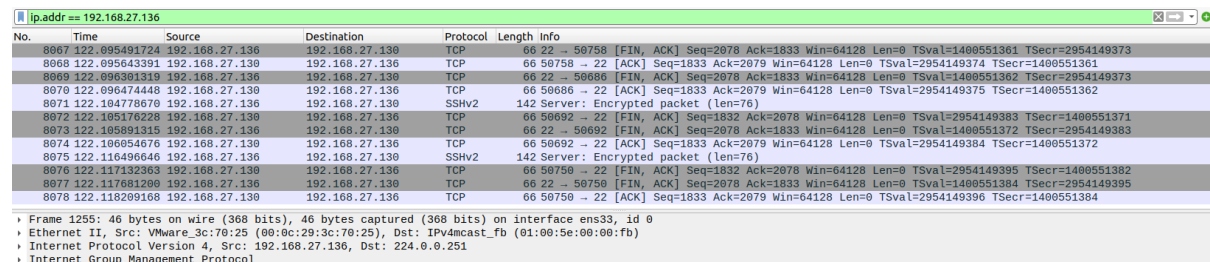
Figure 33 XSS Working

It was known that the website had XSS vulnerability as it shows in Nikto result that the option says, "Allowed HTTP Methods: OPTIONS, HEAD, GET, POST".

6. Pivoting from Alerts to Attack Details

To get a more in-depth view on alerts that were created by Snort, the method which that was done was by implement the Snort traffic forward into Wireshark and viewing the receiving packets after the execution of each attack. Here is what packets have been received in Wireshark.

6.1.1 Pivoting Hydra Alert

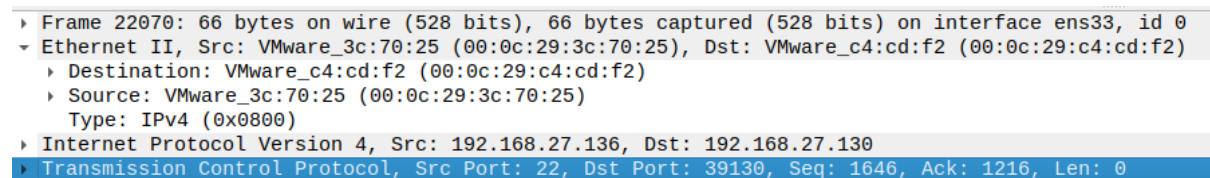


No.	Time	Source	Destination	Protocol	Length	Info
8067	122.095491724	192.168.27.136	192.168.27.136	TCP	66	22 → 50758 [FIN, ACK] Seq=2078 Ack=1833 Win=64128 Len=0 TSval=1400551361 TSecr=2954149373
8068	122.095643391	192.168.27.136	192.168.27.136	TCP	66	50758 → 22 [ACK] Seq=1833 Ack=2079 Win=64128 Len=0 TSval=2954149374 TSecr=1400551361
8069	122.096391319	192.168.27.136	192.168.27.136	TCP	66	22 → 50606 [FIN, ACK] Seq=2078 Ack=1833 Win=64128 Len=0 TSval=1400551362 TSecr=2954149373
8070	122.096474440	192.168.27.136	192.168.27.136	TCP	66	50606 → 22 [ACK] Seq=1833 Ack=2079 Win=64128 Len=0 TSval=2954149375 TSecr=1400551362
8071	122.104778670	192.168.27.136	192.168.27.136	SSHv2	142	Server: Encrypted packet (len=76)
8072	122.105176228	192.168.27.136	192.168.27.136	TCP	66	50692 → 22 [FIN, ACK] Seq=1832 Ack=2078 Win=64128 Len=0 TSval=2954149383 TSecr=1400551371
8073	122.105891315	192.168.27.136	192.168.27.136	TCP	66	22 → 50692 [FIN, ACK] Seq=2078 Ack=1833 Win=64128 Len=0 TSval=1400551372 TSecr=2954149383
8074	122.106054676	192.168.27.136	192.168.27.136	TCP	66	50692 → 22 [ACK] Seq=1833 Ack=2079 Win=64128 Len=0 TSval=2954149384 TSecr=1400551372
8075	122.116490646	192.168.27.136	192.168.27.136	SSHv2	142	Server: Encrypted packet (len=76)
8076	122.117132363	192.168.27.136	192.168.27.136	TCP	66	50750 → 22 [FIN, ACK] Seq=1832 Ack=2078 Win=64128 Len=0 TSval=2954149395 TSecr=1400551382
8077	122.117891200	192.168.27.136	192.168.27.136	TCP	66	22 → 50750 [FIN, ACK] Seq=2078 Ack=1833 Win=64128 Len=0 TSval=1400551384 TSecr=2954149395
8078	122.118209168	192.168.27.136	192.168.27.136	TCP	66	50750 → 22 [ACK] Seq=1833 Ack=2079 Win=64128 Len=0 TSval=2954149396 TSecr=1400551384

Frame 1255: 46 bytes on wire (368 bits), 46 bytes captured (368 bits) on interface ens33, id 0
Ethernet II, Src: VMware_3c:70:25 (00:0c:29:3c:70:25), Dst: IPv4mcast_fb (01:00:5e:00:00:fb)
Internet Protocol Version 4, Src: 192.168.27.136, Dst: 224.0.0.251
Internet Group Management Protocol

Figure 34 Wireshark Result Hydra

After researching the alerts it comes back that most of the packets which are received are encrypted, all that shows us is what version SSH uses and that's mostly it. On the decrypted packets it shows some more details but not too much, it shows here which machine the attack is coming from,



Frame 22070: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface ens33, id 0
Ethernet II, Src: VMware_3c:70:25 (00:0c:29:3c:70:25), Dst: VMware_c4:cd:f2 (00:0c:29:c4:cd:f2)
Destination: VMware_c4:cd:f2 (00:0c:29:c4:cd:f2)
Source: VMware_3c:70:25 (00:0c:29:3c:70:25)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.27.136, Dst: 192.168.27.130
Transmission Control Protocol, Src Port: 22, Dst Port: 39130, Seq: 1646, Ack: 1216, Len: 0

Figure 35 Wireshark

6.1.2 Pivoting Nikto Attack

Finally for Nikto attack, in Wireshark it displays the packets but this time they aren't encrypted, here is an image of how the packets look.

Implementation, configuration, and investigation of a SOC using Open-Source Tools

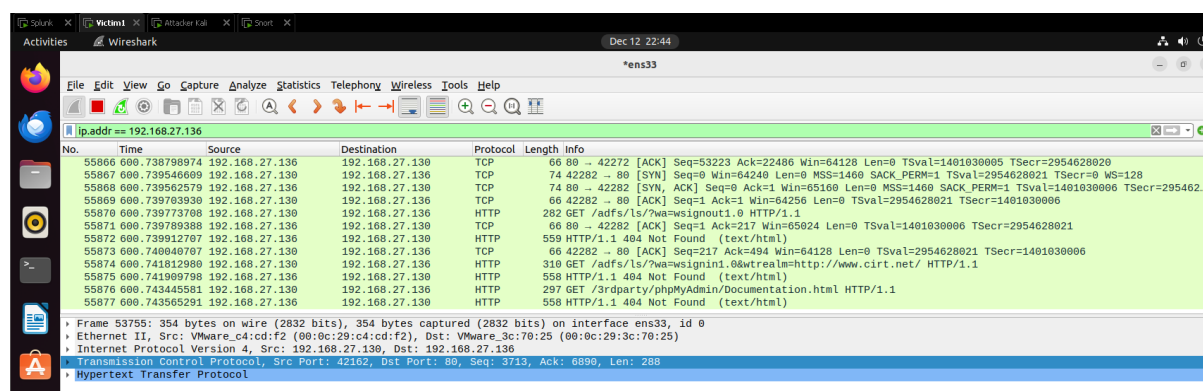


Figure 36Wireshark

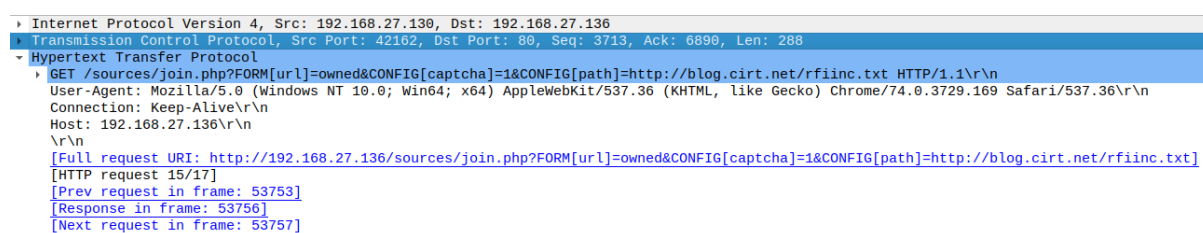


Figure 37Wireshark

It also shows us the username which has executed this attack.

6.1.3 Pivoting HPing3

Unfortunately Wireshark was not able to receive packets from the HPing3 attack as it flooded Splunk Enterprise and Wireshark at the same time which caused the machine to boot offline, and attempts has been tried with Wireshark on by itself but it was still the same result.

7. Conclusion

To sum up, the process of putting in place an open-source Security Operations Centre (SOC) that is fully operational has been instructive and priceless. By carefully investigating and incorporating many open-source tools designed specifically for blue teams, a strong defence system has been created. In addition to strengthening the security posture, the thorough installation and smooth integration of these tools have improved the SOC's responsiveness and agility. Essentially, this open-source SOC's successful deployment and demonstration demonstrate not only its effectiveness but also the value of utilising a variety of integrated tools to strengthen defences, precisely identify threats, and take prompt action to protect against ever-evolving cyber threats. This assignment is

Implementation, configuration, and investigation of a SOC using Open-Source Tools

evidence of the effectiveness of open-source software and the thoughtful combination of tools to protect digital environments against contemporary attacks.

8. References

- Cyavatar (2022). Writing Snort Rules with Examples and Cheat Sheet. [online] CYVATAR.AI. Available at: <https://cyavatar.ai/write-configure-snort-rules/>. [Accessed: 05/12/2023]
- snorpy.cyb3rs3c.net. (n.d.). Snorpy 2.0 - Web Based Snort Rule Creator. [online] Available at: <http://snorpy.cyb3rs3c.net/>. [Accessed: 05/12/2023]
- Wazuh (2018). Wazuh. [online] Wazuh. Available at: <https://wazuh.com/>. [Accessed: 05/12/2023]
- docs.splunk.com. (n.d.). Configure forwarding with outputs.conf - Splunk Documentation. [online] Available at: <https://docs.splunk.com/Documentation/Forwarder/9.1.2/Forwarder/Configureforwardingwithoutoutputs.conf> [Accessed: 06/12/2023].
- community.splunk.com. (2016). Can i tcpout to multiple servers with output.conf file? [online] Available at: <https://community.splunk.com/t5/Getting-Data-In/Can-i-tcpout-to-multiple-servers-with-output-conf-file/m-p/234346> [Accessed: 06/12/2023].
- Trent, R. (2023). Using Kali Linux and Hydra for Attack Testing and Alert Generation. [online] Rod's Blog. Available at: <https://rodtrent.substack.com/p/using-kali-linux-and-hydra-for-attack> [Accessed: 09/12/2023].
- Die.net. (2010). hping3(8) - Linux man page. [online] Available at: <https://linux.die.net/man/8/hping3>. [Accessed: 09/12/23]
- PeerSpot. (n.d.). Wazuh Reviews, Competitors and Pricing. [online] Available at: <https://www.peerspot.com/products/wazuh-reviews#:~:text=Wazuh%20is%20an%20enterprise-ready> [Accessed: 09/12/23].
- teklager.se. (n.d.). what is pfSense - introduction to open source router/firewall operating system. [online] Available at: <https://teklager.se/en/pfsense-introduction-open-source-router-firewall/>. [Accessed: 10/12/23]
- Simplilearn.com. (2021). What is Metasploit: Overview, Framework, and How is it Used | Simplilearn. [online] Available at: <https://www.simplilearn.com/what-is-metasploit-article#:~:text=Metasploit%20is%20one%20of%20the>. [Accessed: 08/12/23]
- Kali Linux. (n.d.). Get Kali. [online] Available at: <https://www.kali.org/get-kali/#kali-virtual-machines>. [Accessed: 11/12/23]

Implementation, configuration, and investigation of a SOC using Open-Source Tools

Anon, (2023). Install Elastic (ELK) Stack 8.x on Ubuntu 22.04 LTS. [online] Available at: <https://portforwarded.com/install-elastic-elk-stack-8-x-on-ubuntu-22-04-lts/> [Accessed: 05/12/23]

Splunk. (2016). Download Splunk Enterprise for Free | Splunk. [online] Available at: https://www.splunk.com/en_us/download/splunk-enterprise.html. [Accessed: 05/12/23]

Splunk. (n.d.). Universal Forwarder for Remote Data Collection. [online] Available at: https://www.splunk.com/en_us/download/universal-forwarder.html [Accessed: 06/12/23]