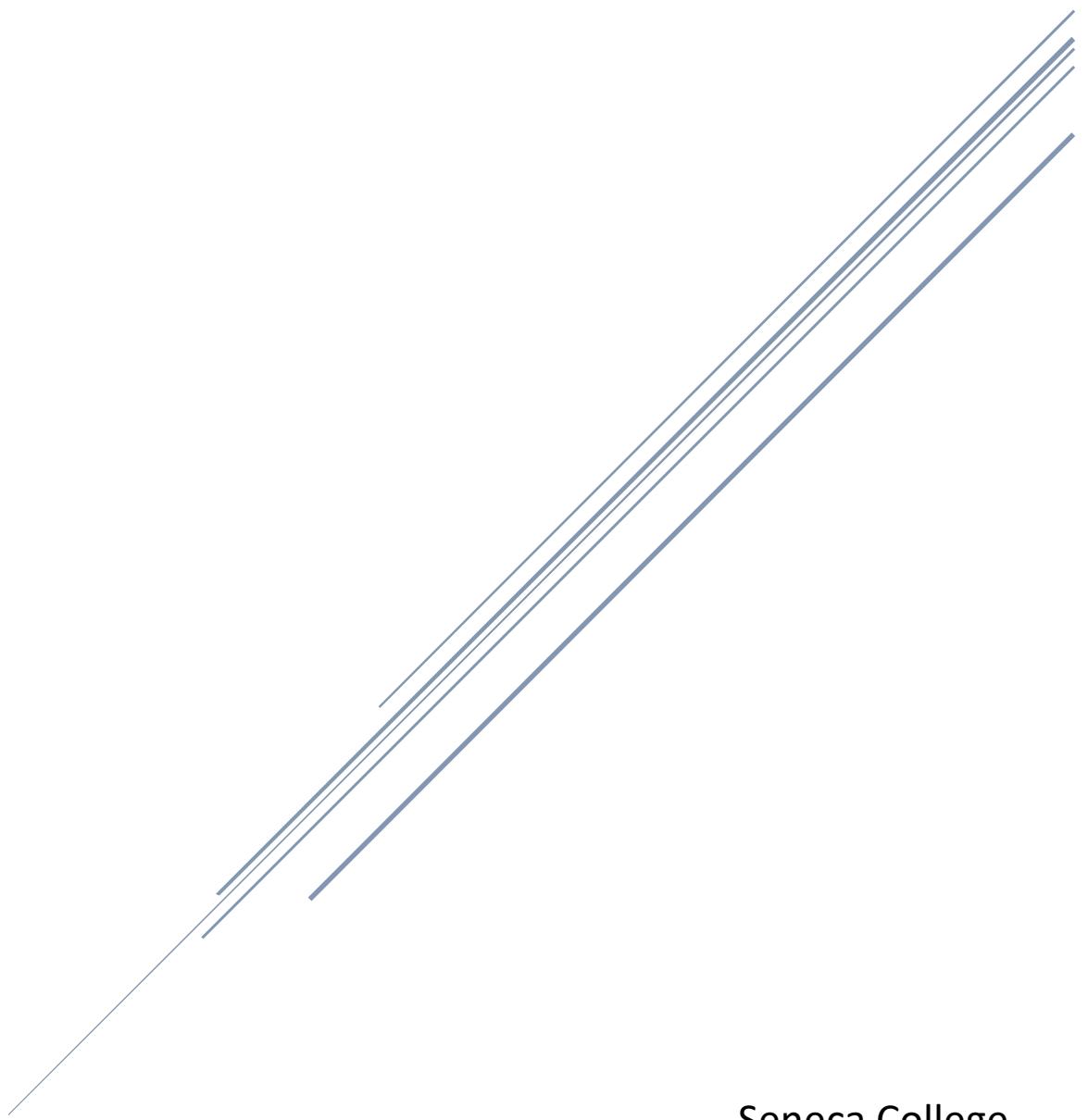


SRT 411 PROJECT PHASE 2

Using Snort to collect all the necessary logs



Seneca College

Group Members Associated with Configuring Project Phase 1

Professor Asma Paracha

Members

Ali Abdulkarim: 150706166

Murad Okasa: 108741208

Buwaneka Hettiarachchi: 104376165

College: Seneca College

Table of Contents

Introduction	4
NMAP scans of the Attack Network.....	4
Machine Connectivity	6
Greenbone OpenVAS network scan	7
Snort.....	10
Attacks	12
Brute force (ssh).....	13
Brute force (FTP).....	14
DDoS Session.....	15
Any Nmap Scan	16
The Security Goal on this Log collection	17
Vulnerabilities and Threats	17
How the collected Data would help me reach the goal.....	18
The Collected Data and Data Cleaning Techniques	18
Collected CSV files.....	18
DDoS Attack CSV	19
FTP Brute force	19
SSH Brute force Attack.....	20
NMAP scan	20
References	21

Figure 1: Nmap Scan of the open ports on Metasploitable 2 machine	4
Figure 2: Nmap Open Port scan on the Centos Machine 3.....	5
Figure 3: Nmap Open Port Scans on the Centos Machine 4.....	5
Figure 4: Showing the Open ports in the Kali Linux Machine	6
Figure 5: Machine 2 showing connectivity to the machines present in the attack network.....	6
Figure 6: Vulnerability scan on each machine and how malicious each machine is	7
Figure 7: Vulnerabilities that are present in the IP address 172.20.21.1	7
Figure 8: Vulnerabilities that are present in the IP address 172.20.21.2	8
Figure 9: Vulnerability scan report on the IP address 172.20.21.3.....	8
Figure 10: Information on the Open ports of the first machine	9
Figure 11: Open ports for the second machine	9
Figure 12: Open ports for the third machine.....	9
Figure 13: The available CVEs in machine 1.....	10
Figure 14: Available CVEs for machine 2.....	10

Figure 15: Available CVEs for machine 3.....	10
Figure 16: Editing the Snort configuration to add details regarding log directory	11
Figure 17: Output for log file for the PCAP files being collected	11
Figure 18: Using the snort configure command to ensure whether snort is running without issues	12
Figure 19: Rules that we generated to collect the logs	12
Figure 20: Performing a brute force attack using Hydra to attack Metasploitable 2 machine	13
Figure 21:Snort Collecting the logs of the attack.....	13
Figure 22: Alerts showing the SID and the generated alert log	14
Figure 23: Created the FTP Brute force session on the Metasploitable 2 machine	14
Figure 24: Generated Alerts from the FTP Brute force session	15
Figure 25: Performing an hping DDoS attack on Machine 4.....	15
Figure 26: The Generated Logs of the DDoS attack	16
Figure 27: Performing an Nmap Scan on the Network.....	16
Figure 28: Performed the log collection for the Nmap scans	17
Figure 29: All the generated Alerts, OpenVAS CVEs, and the Snort Log files are generated	18
Figure 30: DDoS CSV	19
Figure 31: FTP Brute-Force Attack	19
Figure 32: SSH-Brute Force attack	20
Figure 33: NMAP Scan.....	20

Introduction

This Phase focuses on the collection of the various logs within our system Environment. This is a continuation of the 1st phase of the Project, where our group requires to identify and collect the various different logs. We, then collect the logs to generate data from, data can be both malicious and normal traffic, data should be collected using snort, and later we need to analyze the collected data.

NMAP scans of the Attack Network

We started this phase by first scanning an Nmap scan on the Network, in this scenario, we were scanning the Attack network (172.20.21.0/24). This is the network where we will be able to discover the open ports in all the necessary machines such as the Matasploitable2 machine, Centos Machine 3 and 4, and the Kali Linux machine.

Figure 1: Nmap Scan of the open ports on Metasploitable 2 machine

Figure 2: Nmap Open Port scan on the Centos Machine 3

Figure 3: Nmap Open Port Scans on the Centos Machine 4

```

Nmap scan report for 172.20.21.10
Host is up (0.000023s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.51 ((Debian))
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (4 hosts up) scanned in 137.01 seconds

```

(kali㉿kali-g4-s5) [~/nessus]

\$

Figure 4: Showing the Open ports in the Kali Linux Machine

Machine Connectivity

The following images show that the machines can successfully ping all the other machines present within the Network. We need to ensure that all the packets are working fine in order to perform the scans.

```

PS C:\Users\srt411> ping 172.20.21.2
Pinging 172.20.21.2 with 32 bytes of data:
Reply from 172.20.21.2: bytes=32 time<1ms TTL=64
Reply from 172.20.21.2: bytes=32 time<1ms TTL=64
Reply from 172.20.21.2: bytes=32 time<1ms TTL=64

Ping statistics for 172.20.21.2:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
PS C:\Users\srt411> ping 172.20.21.3

Pinging 172.20.21.3 with 32 bytes of data:
Reply from 172.20.21.3: bytes=32 time<1ms TTL=64

Ping statistics for 172.20.21.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Users\srt411> ping 172.20.21.10

Pinging 172.20.21.10 with 32 bytes of data:
Reply from 172.20.21.10: bytes=32 time<1ms TTL=64

Ping statistics for 172.20.21.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

Figure 5: Machine 2 showing connectivity to the machines present in the attack network

Greenbone OpenVAS network scan

The Greenbone OpenVAS is used to detect the vulnerabilities that are present within the scanned machines and give out a rating on how severe each vulnerability that is present within its environment with a vulnerability scale. In this scenario, the IP addresses that I am focusing on are 172.20.21.1, 172.20.21.2, 172.20.21.3.

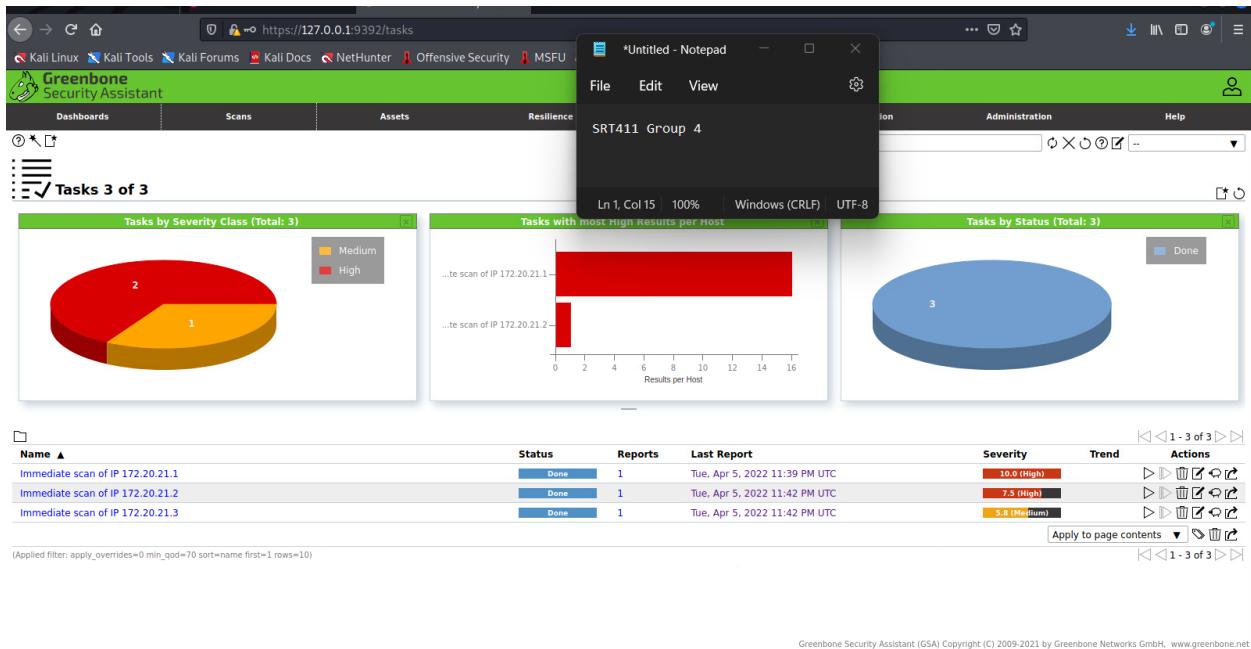


Figure 6: Vulnerability scan on each machine and how malicious each machine is

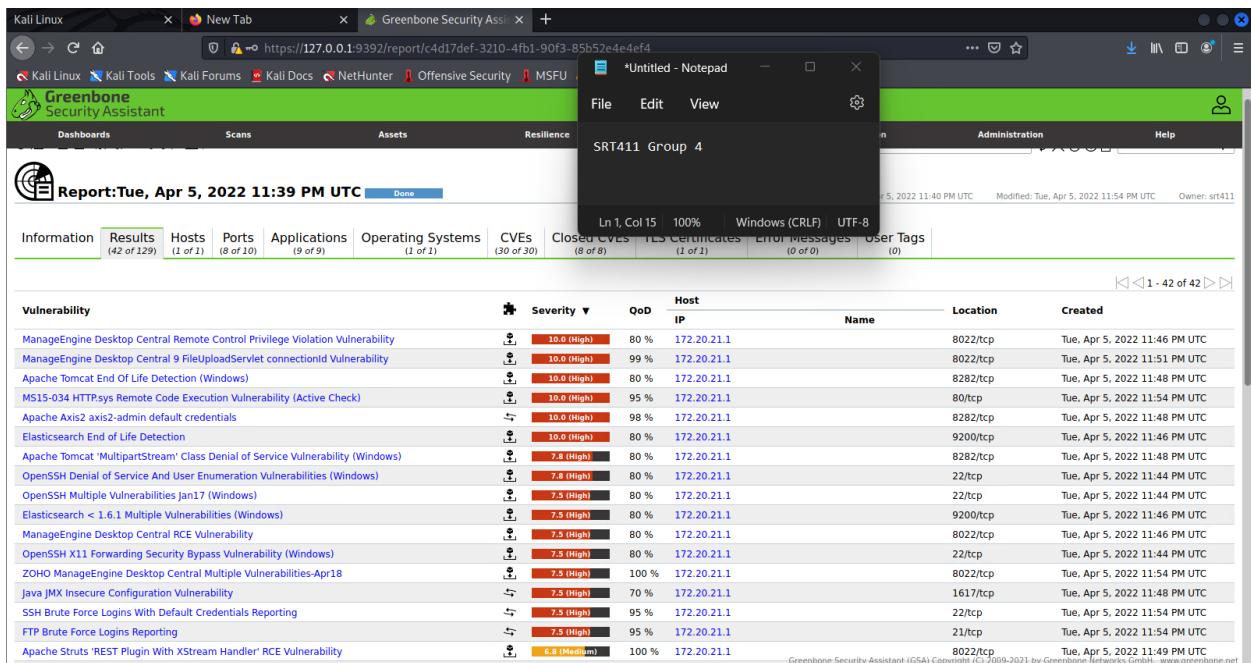


Figure 7: Vulnerabilities that are present in the IP address 172.20.21.1

The screenshot shows a Kali Linux desktop environment with several windows open. In the center, a Firefox browser displays a Greenbone Security Assistant report for the IP address 172.20.21.2. The report is dated Tuesday, April 5, 2022, at 11:42 PM UTC. The main content area shows a table of vulnerabilities:

Vulnerability	Severity	QoD	Host IP	Name	Location	Created
phpinfo() output Reporting	7.5 (High)	80 %	172.20.21.2		80/tcp	Tue, Apr 5, 2022 11:46 PM UTC
HTTP Debugging Methods (TRACE/TRACK) Enabled	5.8 (Medium)	99 %	172.20.21.2		80/tcp	Tue, Apr 5, 2022 11:46 PM UTC
SSH Weak Encryption Algorithms Supported	4.3 (Medium)	95 %	172.20.21.2		22/tcp	Tue, Apr 5, 2022 11:45 PM UTC
TCP timestamps	2.6 (Low)	80 %	172.20.21.2		general/tcp	Tue, Apr 5, 2022 11:45 PM UTC

(Applied filter: apply_overrides=0 levels=html rows=100 min_qod=70 first=1 sort-reverse=severity)

Figure 8: Vulnerabilities that are present in the IP address 172.20.21.2

The screenshot shows a Kali Linux desktop environment with several windows open. In the center, a Firefox browser displays a Greenbone Security Assistant report for the IP address 172.20.21.3. The report is dated Wednesday, April 6, 2022, at 12:00 AM UTC. The main content area shows a table of vulnerabilities:

Vulnerability	Severity	QoD	Host IP	Name	Location	Created
HTTP Debugging Methods (TRACE/TRACK) Enabled	5.8 (Medium)	99 %	172.20.21.3		8080/tcp	Tue, Apr 5, 2022 11:48 PM UTC
HTTP Debugging Methods (TRACE/TRACK) Enabled	5.8 (Medium)	99 %	172.20.21.3		80/tcp	Tue, Apr 5, 2022 11:48 PM UTC
WordPress User IDs and User Names Disclosure	5.8 (Medium)	99 %	172.20.21.3		8080/tcp	Tue, Apr 5, 2022 11:55 PM UTC
HTTP Debugging Methods (TRACE/TRACK) Enabled	5.8 (Medium)	99 %	172.20.21.3		443/tcp	Tue, Apr 5, 2022 11:48 PM UTC
Missing 'HttpOnly' Cookie Attribute	5.0 (Medium)	80 %	172.20.21.3		80/tcp	Tue, Apr 5, 2022 11:48 PM UTC
Cleartext Transmission of Sensitive Information via HTTP	4.8 (Medium)	80 %	172.20.21.3		8080/tcp	Tue, Apr 5, 2022 11:47 PM UTC
SSH Weak Encryption Algorithms Supported	4.3 (Medium)	95 %	172.20.21.3		22/tcp	Tue, Apr 5, 2022 11:47 PM UTC
TCP timestamps	2.6 (Low)	80 %	172.20.21.3		general/tcp	Tue, Apr 5, 2022 11:47 PM UTC

(Applied filter: apply_overrides=0 levels=html rows=100 min_qod=70 first=1 sort-reverse=severity)

Figure 9: Vulnerability scan report on the IP address 172.20.21.3

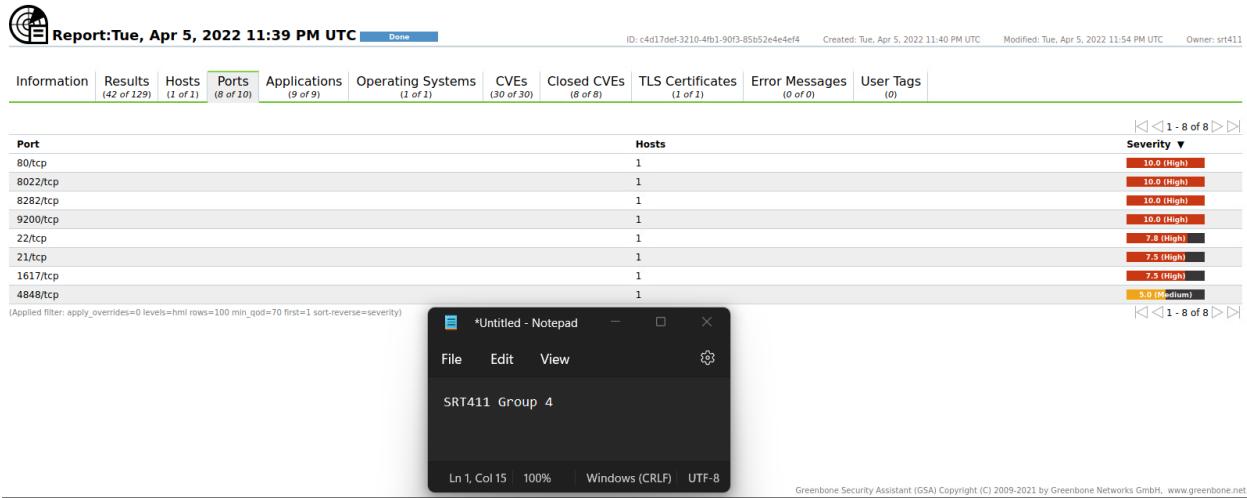


Figure 10: Information on the Open ports of the first machine

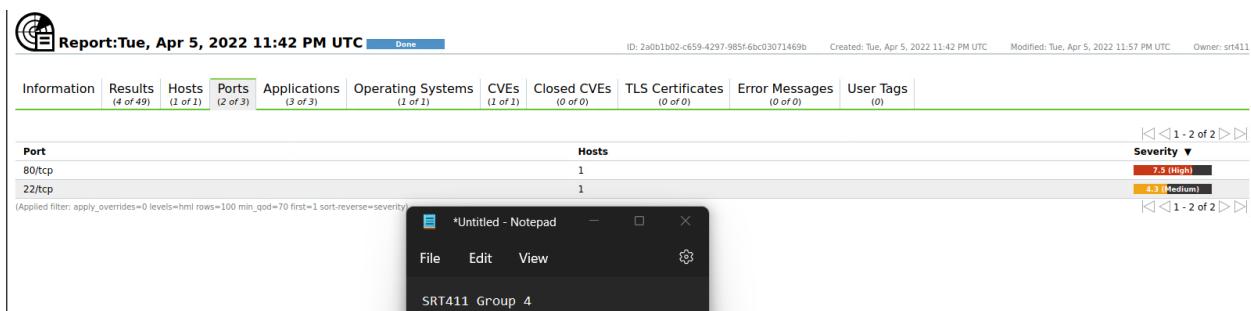


Figure 11: Open ports for the second machine

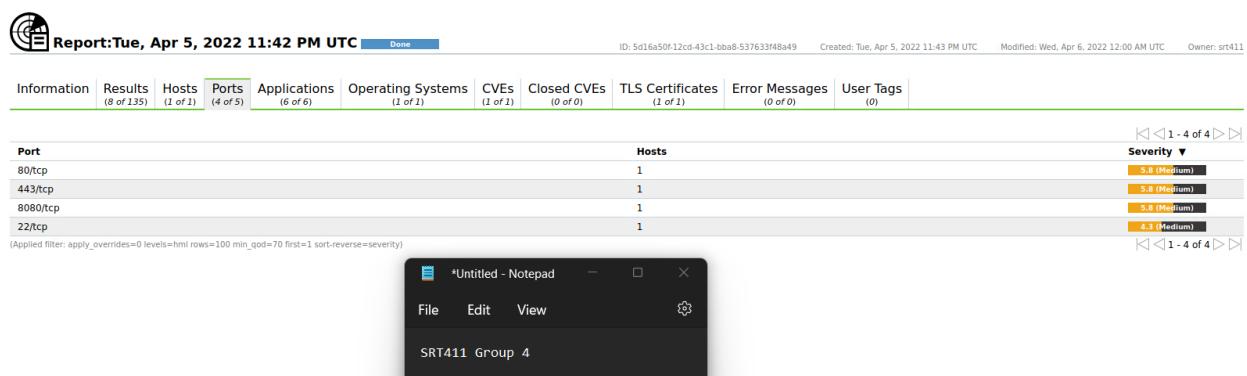


Figure 12: Open ports for the third machine

Figure 13: The available CVEs in machine 1

Figure 14: Available CVEs for machine 2

Figure 15: Available CVEs for machine 3

Snort

Snort is an open-source tool that is used for purposes such as Intrusion detection and other activities such as log collection and log generation. Our main goal within this phase is to collect the logs using snort, for snort to successfully collect the logs, we need to create specific rules and these rules are used to detect the certain activities that are present in the environment.

```

GNU nano 2.9.8                               snort.conf

#
# Configure default bpf_file to use for filtering what traffic reaches snort. For more information see snort -h command line option
# config bpf_file:
#
# Configure default log directory for snort to log to. For more information see README.event_queue
# config logdir: /var/log/snort

#####
# Step #3: Configure the base detection engine. For more information see README.rules
#####

# Configure PCRE match limitations
config pcre.match_limit: 3500
config pcre.match_limit_recursion: 1500

# Configure the detection engine See the Snort Manual, Configuring Snort - Includes - Config
config detection: search-method ac-split search-optimize max-pattern-len 20

# Configure the event queue. For more information, see README.event_queue
config event_queue: max_queue 8 log 5 order_events content_length

#####
## Configure GTP if it is to be used.
## For more information, see README.GTP
#####

```

Figure 16: Editing the Snort configuration to add details regarding log directory

```

#####
# Step #6: Configure output plugins
# For more information, see Snort Manual, Configuring Snort - Output Modules
#####

# unified2
# Recommended for most installs
# output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types

# Additional configuration for specific types of installs
# output alert_unified2: filename snort.alert, limit 128, nostamp
# output log_unified2: filename snort.log, limit 128, nostamp

# syslog
# output alert_syslog: LOG_AUTH LOG_ALERT

# pcap
output pcap: /home/srt411/snort_src/tcpdump.log

# metadata reference data. do not modify these lines
include classification.config
include reference.config

#####
# Step #7: Customize your rule set
# For more information, see Snort Manual, Writing Snort Rules
#
# NOTE: All categories are enabled in this conf file
#####

# site specific rules
include $RULE_PATH/local.rules

include $RULE_PATH/app-detect.rules
include $RULE_PATH/attack-responses.rules

```

Figure 17: Output for log file for the PCAP files being collected

```

I memory-cap : 1048576 bytes
+-----[event-filter-global]-----
+-----[event-filter-local]-----
I none
+-----[suppression]-----
I none

Rule application order: pass->drop->sdrop->reject->alert->log
Verifying Preprocessor Configurations!

[ Port Based Pattern Matching Memory ]
[ Number of patterns truncated to 28 bytes: 0 ]
peap DaQ configured to passive.
Acquiring network traffic from "ens192".
Reload thread starting...
Reload thread started, thread 0x7fd2e0ad0700 (2576?) Decoding Ethernet

==== Initialization Complete ====
->> Snort! <-
o'''-' Version 2.9.19 GRE (Build 85)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2021 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.9.1 (with TPACKET_V3)
Using PCRE version: 8.42 2018-03-28
Using ZLIB version: 1.2.11

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 3.2 <Build 1>
Preprocessor Object: appid Version 1.1 <Build 5>
Preprocessor Object: SF_SCOMPLUS Version 1.0 <Build 1>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>

Commencing packet processing (pid=25766)

```

Figure 18: Using the snort configure command to ensure whether snort is running without issues

Generated Snort Rulesets

```

GNU nano 2.9.8          /etc/snort/rules/local.rules

alert tcp any any -> $HOME_NET 80 (flags: S; msg:"DDoS encounter"; sid:1000001; flow:stateless; detection_filter:track_by_dst, $)
alert tcp any any -> $HOME_NET 22 (msg:"Possible SSH brute forcing!"; flags: S; detection_filter:track_by_src, count 5, seconds 5)
alert tcp any any -> $HOME_NET $HTTP_PORTS (msg:"Metasploit Meterpreter"; flow:to_server,established; content:"RECV"; http_client)
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"Possible FTP brute force!"; metadata:service ftp-data; session:binary; sid:109)
alert tcp any any -> $HOME_NET any (msg:"Nmap Scan is happening"; sid:1000004; rev:1)

```

Figure 19: Rules that we generated to collect the logs

Attacks

In this process, we are working on four different log collections within our home network, two attacks are going to be related to brute force attacks where one attack is performed as a ssh attack while the other is performed as an ftp attack. The reason why we performed two separate attacks is to see if snort is able to differentiate the brute force attacks. Then, we performed a

DDoS attack using hping and finally we performed a rule that would generate alerts when we perform Nmap remote scans in the network.

Brute force (ssh)

```
[kali㉿kali-g4-s5]:~[~]
└─$ hydra -l /home/kali/Desktop/user.txt -P /home/kali/Desktop/pass.txt 172.20.21.1 ssh
Hydra v9.1 (c) 2019 by van Hauser/TMC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 13:29:30
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 400 login tries (!:20:p:20), -25 tries per task
[DATA] attacking ssh://172.20.21.1:22
[22]ssh host: 172.20.21.1 login: vagrant password: vagrant
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-11 13:30:12

[kali㉿kali-g4-s5]:~[~]
```

Figure 20: Performing a brute force attack using Hydra to attack Metasploitable 2 machine

04/11-13:30:39.274978 172.20.21.1:22 -> 172.20.21.10:45170
TCP TTL:64 TOS:0x0 ID:59024 Iplen:20 DgmLen:120 DF
@ Seq: 0x85510ECE5 Ack: 0x60a06539 Win: 0x1F5 TcpLen: 32
TCF Options (3) => NOP NOP TS: 1825231248 566284916
=====

04/11-13:30:39.274978 172.20.21.1:22 -> 172.20.21.10:45170
TCP TTL:64 TOS:0x0 ID:4787 Iplen:20 DgmLen:136 DF
@ Seq: 0x60a0664E5 Ack: 0x85510F09 Win: 0xF861 TcpLen: 32
TCF Options (3) => NOP NOP TS: 566284918 1825231248
=====

04/11-13:30:39.275045 172.20.21.1:22 -> 172.20.21.1:22
TCP TTL:64 TOS:0x0 ID:59083 Iplen:20 DgmLen:136 DF
@ Seq: 0x85510F09 Ack: 0x60a06539 Win: 0x1F5 TcpLen: 32
TCF Options (3) => NOP NOP TS: 1825231266 566284918
=====

04/11-13:30:39.276483 172.20.21.1:22 -> 172.20.21.10:45170
TCP TTL:128 TOS:0x0 ID:4788 Iplen:20 DgmLen:136 DF
@ Seq: 0x60a066539 Ack: 0x85510F05 Win: 0xF80D TcpLen: 32
TCF Options (3) => NOP NOP TS: 566284918 1825231266
=====

04/11-13:30:39.276904 172.20.21.1:22 -> 172.20.21.1:22
TCP TTL:64 TOS:0x0 ID:59026 Iplen:20 DgmLen:52 DF
@ Seq: 0x85510F05 Ack: 0x60a0650D Win: 0x1F5 TcpLen: 32
TCF Options (3) => NOP NOP TS: 1825231269 566284918
=====

04/11-13:30:39.278194 172.20.21.1:22 -> 172.20.21.10:45170
TCP TTL:128 TOS:0x0 ID:4789 Iplen:20 DgmLen:52 DF
@ Seq: 0x60a0650D Ack: 0x85510F5E Win: 0xF80D TcpLen: 32
TCF Options (3) => NOP NOP TS: 566284919 1825231269
=====

04/11-13:30:39.288357 172.20.21.1:22 -> 172.20.21.10:45170
TCP TTL:128 TOS:0x0 ID:4790 Iplen:20 DgmLen:52 DF
@ Seq: 0x60a06650C Ack: 0x85510F5E Win: 0xF80D TcpLen: 32
TCF Options (3) => NOP NOP TS: 566284920 1825231269
=====

04/11-13:30:39.288415 172.20.21.1:22 -> 172.20.21.1:22
TCP TTL:64 TOS:0x8 ID: 0 Iplen:20 DgmLen:52 DF
@ Seq: 0x85510F5E Ack: 0x60a0650C Win: 0x1F5 TcpLen: 32
TCF Options (3) => NOP NOP TS: 1825231279 566284920
=====

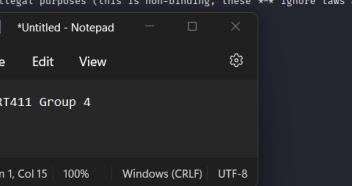


Figure 21:Snort Collecting the logs of the attack

```

GNU nano 2.9.8          bruteforce.log

[**] [1:10000002:1] Possible SSH brute forcing! [**]
[Priority: 0]
04/11-01:47:55.439050 172.20.21.10:42940 -> 172.20.21.1:22
TCP TTL:64 TOS:0x0 ID:64279 Iplen:28 DgmLen:68 DF
*****S* Seq: 0x5941D1B4 Ack: 0x0 Win: 0xF0F0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 1783067382 0 NOP WS: ? 

[**] [1:10000002:1] Possible SSH brute forcing! [**]
[Priority: 0]
04/11-01:47:55.439051 172.20.21.10:42942 -> 172.20.21.1:22
TCP TTL:64 TOS:0x0 ID:64279 Iplen:28 DgmLen:68 DF
*****S* Seq: 0x5F67F35F Ack: 0x0 Win: 0xF0F0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 1783067382 0 NOP WS: ? 

[**] [1:10000002:1] Possible SSH brute forcing! [**]
[Priority: 0]
04/11-01:47:55.439182 172.20.21.10:42944 -> 172.20.21.1:22
TCP TTL:64 TOS:0x0 ID:33379 Iplen:28 DgmLen:68 DF
*****S* Seq: 0x7A116D9A Ack: 0x0 Win: 0xF0F0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 1783067383 0 NOP WS: ? 

[**] [1:10000002:1] Possible SSH brute forcing! [**]
[Priority: 0]
04/11-01:47:55.439183 172.20.21.10:42946 -> 172.20.21.1:22
TCP TTL:64 TOS:0x0 ID:54384 Iplen:28 DgmLen:68 DF
*****S* Seq: 0x5067D65A Ack: 0x0 Win: 0xF0F0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 1783067383 0 NOP WS: ? 

[**] [1:10000002:1] Possible SSH brute forcing! [**]
[Priority: 0]
04/11-01:47:55.439407 172.20.21.10:42950 -> 172.20.21.1:22
TCP TTL:64 TOS:0x0 ID:36821 Iplen:28 DgmLen:68 DF
*****S* Seq: 0xA82D9E50 Ack: 0x0 Win: 0xF0F0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 1783067383 0 NOP WS: ? 

[**] [1:10000002:1] Possible SSH brute forcing! [**]

```

[Read 2646 lines]

Figure 22: Alerts showing the SID and the generated alert log

Brute force (FTP)

```

File Actions Edit View Help

--(kali㉿kali-g4-s5)-[~]
$ hydra -l /home/kali/Desktop/user.txt -P /home/kali/Desktop/pass.txt 172.20.21.1 ftp
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, the
ore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/hydra) starting at 2022-04-11 16:32:33
[DATA] max 16 tasks per 1 server, overall 16 tasks, 400 login tries (l:10/p:20), ~25 tries per task
[DATA] attacking ftp://172.20.21.1:21/
[21][ftp] host: 172.20.21.1 login: vagrant password: vagrant
[21][ftp] host: 172.20.21.1 login: Vagrant password: vagrant
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/hydra) finished at 2022-04-11 16:32:41

--(kali㉿kali-g4-s5)-[~]
$ ping 172.20.34.5
PING 172.20.34.5 (172.20.34.5) 56(84) bytes of data.
64 bytes from 172.20.34.5: icmp_seq=1 ttl=128 time=0.352 ms
64 bytes from 172.20.34.5: icmp_seq=2 ttl=128 time=0.340 ms
...
64 bytes from 172.20.34.5: icmp_seq=10 ttl=128 time=0.352 ms
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.340/0.346/0.352/0.005 ms

```

Figure 23: Created the FTP Brute force session on the Metasploitable 2 machine

```
GNU nano 2.9.8          FTPbruteforce.log

[**] [1:1000003:1] Possible FTP brute force! [**]
[Priority: 0]
04/11-12:55:39.466055 172.20.21.10:43436 -> 172.20.21.1:21
TCP TTL:64 TOS:0x0 ID:58271 Iplen:28 DgmLen:68 DF
*****S* Seq: 0xB876F9B8 Ack: 0x0 Win: 0xF0F0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 1823131455 0 NOP WS: ? 

[**] [1:1000003:1] Possible FTP brute force! [**]
[Priority: 0]
04/11-12:55:39.466061 172.20.21.10:43438 -> 172.20.21.1:21
TCP TTL:64 TOS:0x0 ID:16231 Iplen:28 DgmLen:68 DF
*****S* Seq: 0xE44C3A80 Ack: 0x0 Win: 0xF0F0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 1823131455 0 NOP WS: ? 

[**] [1:1000003:1] Possible FTP brute force! [**]
[Priority: 0]
04/11-12:55:39.466075 172.20.21.10:43442 -> 172.20.21.1:21
TCP TTL:64 TOS:0x0 ID:43789 Iplen:28 DgmLen:68 DF
*****S* Seq: 0xE7E01A78 Ack: 0x0 Win: 0xF0F0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 1823131455 0 NOP WS: ? 

[**] [1:1000003:1] Possible FTP brute force! [**]
[Priority: 0]
04/11-12:55:39.466077 172.20.21.10:43440 -> 172.20.21.1:21
TCP TTL:64 TOS:0x0 ID:68470 Iplen:28 DgmLen:68 DF
*****S* Seq: 0x0074E1AC Ack: 0x0 Win: 0xF0F0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 1823131455 0 NOP WS: ? 

[**] [1:1000003:1] Possible FTP brute force! [**]
[Priority: 0]
04/11-12:55:39.466083 172.20.21.10:43432 -> 172.20.21.1:21
TCP TTL:64 TOS:0x0 ID:244 Iplen:28 DgmLen:68 DF
*****S* Seq: 0x15C28C62 Ack: 0x0 Win: 0xF0F0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 1823131454 0 NOP WS: ? 

[**] [1:1000003:1] Possible FTP brute force! [**]
[Priority: 0]
04/11-12:55:39.466084 172.20.21.10:43434 -> 172.20.21.1:21
TCP TTL:64 TOS:0x0 ID:41021 Iplen:28 DgmLen:68 DF
*****S* Seq: 0x2D02DA78 Ack: 0x0 Win: 0xF0F0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 1823131454 0 NOP WS: ? 

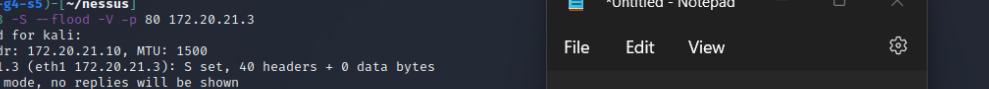
[**] [1:1000003:1] Possible FTP brute force! [**]
```

Figure 24: Generated Alerts from the FTP Brute force session

DDoS Session

```
(kali㉿kali-g4-s5) [~/nessus]
└─$ sudo hping3 -S --flood -V -p 80 172.20.21.3
[sudo] password for kali:
using eth1, addr: 172.20.21.10, MTU: 1500
HPING 172.20.21.3 (eth1 172.20.21.3): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 172.20.21.3 hping statistic ---
186630 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

(kali㉿kali-g4-s5) [~/nessus]
└─$
```



The screenshot shows a terminal window on the left and a Notepad window on the right. The terminal window displays the command hping3 -S --flood -V -p 80 172.20.21.3 being run, resulting in a flood of SYN packets to port 80 of the target host. The Notepad window contains the text "Group 4 SRT411".

Figure 25: Performing an hping DDoS attack on Machine 4

```
GNU nano 2.9.8                                /var/log/snort/ddos.log

[**] [1:1000001:0] DDoS encounter [**]
[Priority: 0]
04/10-22:15:27.381822 172.20.21.10:2879 -> 172.20.21.3:80
TCP TTL:64 TOS:0x0 ID:63528 IplLen:20 DgmLen:48
*****$* Seq: 0x1BCF10B9 Ack: 0x5B2840A1 Win: 0x200 TcpLen: 20

[**] [1:1000001:0] DDoS encounter [**]
[Priority: 0]
04/10-22:15:27.381824 172.20.21.10:2880 -> 172.20.21.3:80
TCP TTL:64 TOS:0x0 ID:35586 IplLen:20 DgmLen:48
*****$* Seq: 0x7B1FCFFC Ack: 0x2FDE6898 Win: 0x200 TcpLen: 20

[**] [1:1000001:0] DDoS encounter [**]
[Priority: 0]
04/10-22:15:27.381840 172.20.21.10:2881 -> 172.20.21.3:80
TCP TTL:64 TOS:0x0 ID:34692 IplLen:20 DgmLen:48
*****$* Seq: 0x30444424 Ack: 0x1F4F2D8 Win: 0x200 TcpLen: 20

[**] [1:1000001:0] DDoS encounter [**]
[Priority: 0]
04/10-22:15:27.381868 172.20.21.10:2882 -> 172.20.21.3:80
TCP TTL:64 TOS:0x0 ID:38654 IplLen:20 DgmLen:48
*****$* Seq: 0x2259C6AF Ack: 0x6B7F72C5 Win: 0x200 TcpLen: 20

[**] [1:1000001:0] DDoS encounter [**]
[Priority: 0]
04/10-22:15:27.381870 172.20.21.10:2883 -> 172.20.21.3:80
TCP TTL:64 TOS:0x0 ID:31463 IplLen:20 DgmLen:48
*****$* Seq: 0x51A543AB Ack: 0x4C77DF0E Win: 0x200 TcpLen: 20

[**] [1:1000001:0] DDoS encounter [**]
[Priority: 0]
04/10-22:15:27.381876 172.20.21.10:2884 -> 172.20.21.3:80
TCP TTL:64 TOS:0x0 ID:60492 IplLen:20 DgmLen:48
*****$* Seq: 0x54FA3568 Ack: 0x2C5322F5 Win: 0x200 TcpLen: 20

[**] [1:1000001:0] DDoS encounter [**]
[Priority: 0]
04/10-22:15:27.381877 172.20.21.10:2885 -> 172.20.21.3:80
TCP TTL:64 TOS:0x0 ID:56847 IplLen:20 DgmLen:48
*****$* Seq: 0x213A3136 Ack: 0x14DF3BB5 Win: 0x200 TcpLen: 20

[**] [1:1000001:0] DDoS encounter [**]
```

Figure 26: The Generated Logs of the DDoS attack

Any Nmap Scan

Figure 27: Performing an Nmap Scan on the Network

The screenshot shows a terminal window titled "nmap nano 2.9.8" displaying multiple lines of Nmap scan logs. The logs detail various TCP connections and their properties, such as source and destination IP addresses, port numbers, and sequence numbers. Below the terminal is a "Notepad" application window titled "*Untitled - Notepad". The Notepad window contains a single line of text: "Group 4 SRT411". At the bottom of the Notepad window, there is a toolbar with various icons and labels corresponding to standard Windows-style text operations like Cut, Copy, Paste, Undo, and Redo.

```

GNU nano 2.9.8                               nmapscan.log

[**] [1:1000004:1] Nmap Scan is happening [**]
[Priority: 0]
04/11-13:15:15.604875 172.20.21.10:50438 -> 172.20.21.2:995
TCP TTL:38 TOS:0x0 ID:52113 Iplen:20 DgmLen:44
*****S* Seq: 0x6029E66E Ack: 0x0 Win: 0x400 TcpLen: 24
TCP Options (1) => MSS: 1468

[**] [1:1000004:1] Nmap Scan is happening [**]
[Priority: 0]
04/11-13:15:15.604876 172.20.21.10:50438 -> 172.20.21.2:8888
TCP TTL:55 TOS:0x0 ID:2154 Iplen:20 DgmLen:44
*****S* Seq: 0x6029E66E Ack: 0x0 Win: 0x400 TcpLen: 24
TCP Options (1) => MSS: 1468

[**] [1:1000004:1] Nmap Scan is happening [**]
[Priority: 0]
04/11-13:15:15.604879 172.20.21.10:50438 -> 172.20.21.2:23
TCP TTL:37 TOS:0x0 ID:39180 Iplen:20 DgmLen:44
*****S* Seq: 0x6029E66E Ack: 0x0 Win: 0x400 TcpLen: 24
TCP Options (1) => MSS: 1468

[**] [1:1000004:1] Nmap Scan is happening [**]
[Priority: 0]
04/11-13:15:15.604879 172.20.21.10:50438 -> 172.20.21.2:256
TCP TTL:39 TOS:0x0 ID:30149 Iplen:20 DgmLen:44
*****S* Seq: 0x6029E66E Ack: 0x0 Win: 0x400 TcpLen: 24
TCP Options (1) => MSS: 1468

[**] [1:1000004:1] Nmap Scan is happening [**]
[Priority: 0]
04/11-13:15:15.604880 172.20.21.10:50438 -> 172.20.21.2:993
TCP TTL:53 TOS:0x0 ID:41752 Iplen:20 DgmLen:44
*****S* Seq: 0x6029E66E Ack: 0x0 Win: 0x400 TcpLen: 24
TCP Options (1) => MSS: 1468

[**] [1:1000004:1] Nmap Scan is happening [**]
[Priority: 0]
04/11-13:15:15.604880 172.20.21.10:50438 -> 172.20.21.2:135
TCP TTL:41 TOS:0x0 ID:48304 Iplen:20 DgmLen:44
*****S* Seq: 0x6029E66E Ack: 0x0 Win: 0x400 TcpLen: 24
TCP Options (1) => MSS: 1468

[**] [1:1000004:1] Nmap Scan is happening [**]

```

Figure 28: Performed the log collection for the Nmap scans

The Security Goal on this Log collection

The main Security Goal in this Log collection is to identify the importance of an intrusion detection system within a network. This is important mainly because we would not be able to identify any attacks happening within the network and that is the main functionality of snort. Snort detects these suspicious activity present within the Environment through the use of rules that we generated to detect these activities. By collecting the logs of the alerts, we are able to discover the extent of the suspicious activities present.

Vulnerabilities and Threats

Most of the Vulnerabilities and threats found within the Home Lab environment was the improper use of software and having many programs that ahs either exceeded its end of life, which makes them a prime target for attackers to exploit and reach machines. Other Threats where the lack of strong password generation, within the home environment, we were able to discover that most of the machines have week authentication segments which result in the exploitation of them through the use of attacks such as Brute-Force attacks.

How the collected Data would help me reach the goal

The collected Data will have a significant amount for us to reach the target, mainly because we were able to generate CSV files that will be used as logs to generate them in a Visualization platform, mainly for the purpose of understanding the main visualization of the logs collected. So, we can look into the activity and then track down any suspicious behaviours within the collected logs.

The Collected Data and Data Cleaning Techniques

The Collected data will be generated into CSV files, and they will be sent to a new location so that they can be easily Collected to visualize through the use of ELK. In terms of Data Cleaning techniques, we will look into columns with unusual information that might be unreadable, we would not include them into the log collection.

Collected CSV files

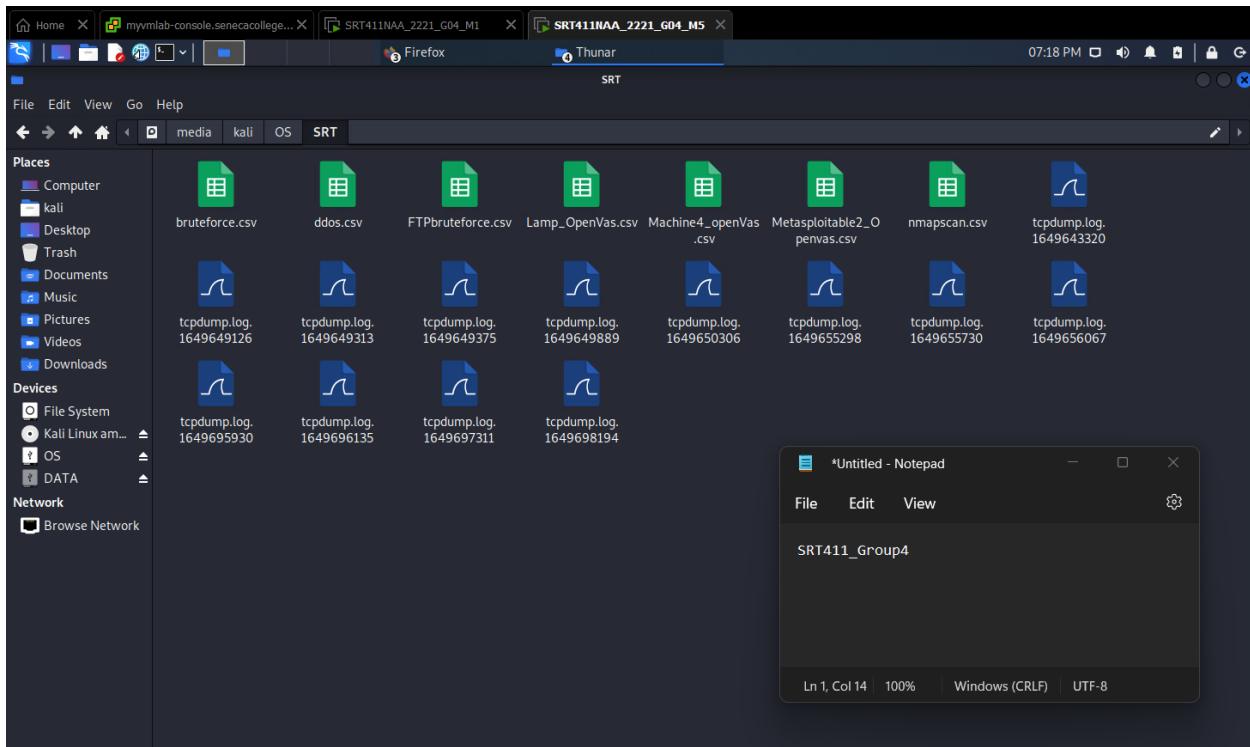


Figure 29: All the generated Alerts, OpenVAS CVEs, and the Snort Log files are generated

DDoS Attack CSV

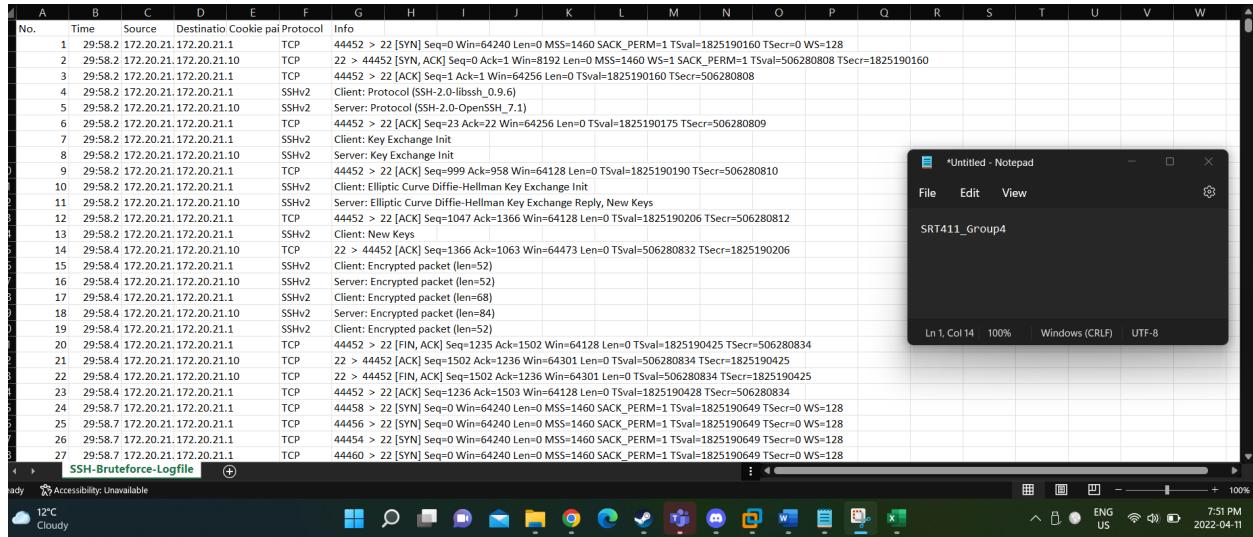
Figure 30: DDoS CSV

FTP Brute force

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
No.	Time	Source	Destinatio	Cookie	pai	Protocol	Info															
1	52:15:5 172.20.21.172.20.21.1	TCP	43390 > 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSeqval=1822927481 TSeqrc=0 WS=128																			
2	52:15:5 172.20.21.172.20.21.1	TCP	43392 > 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSeqval=1822927481 TSeqrc=0 WS=128																			
3	52:15:5 172.20.21.172.20.21.1	TCP	43394 > 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSeqval=1822927482 TSeqrc=0 WS=128																			
4	52:15:5 172.20.21.172.20.21.1	TCP	43400 > 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSeqval=1822927482 TSeqrc=0 WS=128																			
5	52:15:5 172.20.21.172.20.21.1	TCP	43402 > 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSeqval=1822927482 TSeqrc=0 WS=128																			
6	52:15:5 172.20.21.172.20.21.1	TCP	43408 > 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSeqval=1822927482 TSeqrc=0 WS=128																			
7	52:15:5 172.20.21.172.20.21.1	TCP	43410 > 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSeqval=1822927482 TSeqrc=0 WS=128																			
8	52:15:5 172.20.21.172.20.21.1	TCP	43390 > 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSeqval=1822927482 TSeqrc=506054540																			
9	52:15:5 172.20.21.172.20.21.1	TCP	43392 > 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSeqval=1822927482 TSeqrc=506054540																			
10	52:15:5 172.20.21.172.20.21.1	TCP	43394 > 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSeqval=1822927482 TSeqrc=506054540																			
11	52:15:5 172.20.21.172.20.21.1	TCP	43416 > 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSeqval=1822927482 TSeqrc=0 WS=128																			
12	52:15:5 172.20.21.172.20.21.1	TCP	43396 > 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSeqval=1822927482 TSeqrc=0 WS=128																			
13	52:15:5 172.20.21.172.20.21.1	TCP	43418 > 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSeqval=1822927482 TSeqrc=0 WS=128																			
14	52:15:5 172.20.21.172.20.21.1	TCP	43400 > 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSeqval=1822927482 TSeqrc=506054540																			
15	52:15:5 172.20.21.172.20.21.1	TCP	43398 > 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSeqval=1822927482 TSeqrc=0 WS=128																			
16	52:15:5 172.20.21.172.20.21.1	TCP	43412 > 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSeqval=1822927482 TSeqrc=506054540																			
17	52:15:5 172.20.21.172.20.21.1	TCP	43396 > 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSeqval=1822927482 TSeqrc=506054540																			
18	52:15:5 172.20.21.172.20.21.1	TCP	43398 > 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSeqval=1822927482 TSeqrc=506054540																			
19	52:15:5 172.20.21.172.20.21.1	TCP	43402 > 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSeqval=1822927482 TSeqrc=506054540																			
20	52:15:5 172.20.21.172.20.21.1	TCP	43404 > 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSeqval=1822927482 TSeqrc=0 WS=128																			
21	52:15:5 172.20.21.172.20.21.1	TCP	43408 > 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSeqval=1822927482 TSeqrc=506054540																			
22	52:15:5 172.20.21.172.20.21.1	TCP	43406 > 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSeqval=1822927482 TSeqrc=0 WS=128																			
23	52:15:5 172.20.21.172.20.21.1	TCP	43420 > 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSeqval=1822927482 TSeqrc=0 WS=128																			
24	52:15:5 172.20.21.172.20.21.1	TCP	43422 > 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSeqval=1822927482 TSeqrc=0 WS=128																			
25	52:15:5 172.20.21.172.20.21.1	TCP	43404 > 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSeqval=1822927482 TSeqrc=506054540																			
26	52:15:5 172.20.21.172.20.21.1	TCP	43410 > 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSeqval=1822927482 TSeqrc=506054540																			
27	52:15:5 172.20.21.172.20.21.1	TCP	43406 > 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSeqval=1822927482 TSeqrc=506054540																			

Figure 31: FTP Brute-Force Attack

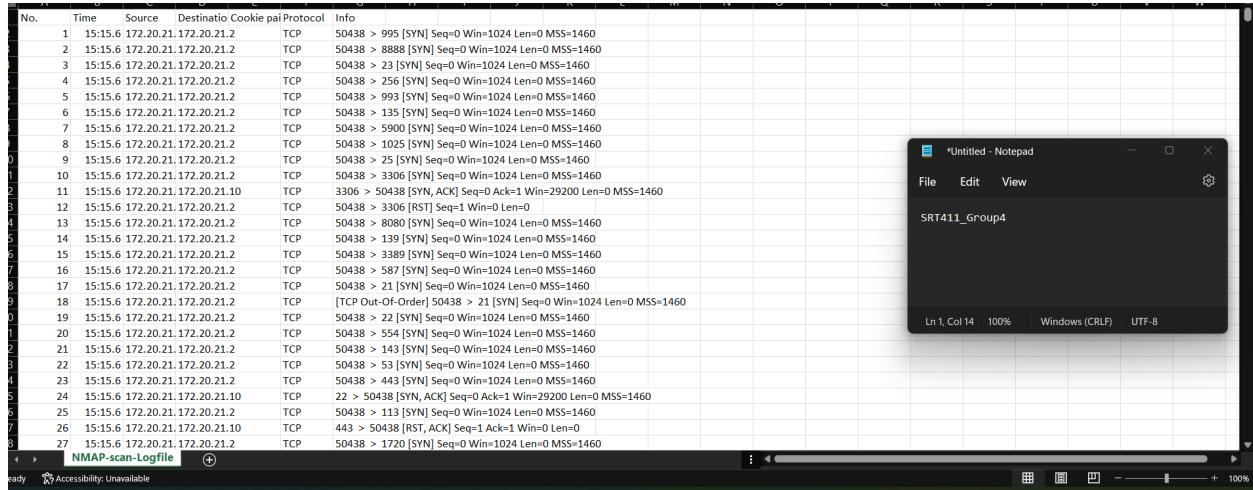
SSH Brute force Attack



No.	Time	Source	Destinatio	Cookie	pai	Protocol	Info
1	29-58.2 172.20.21.172.20.21.1	TCP	44452 > 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 Tsvl=1825190160 TSecr=0 WS=128				
2	29-58.2 172.20.21.172.20.21.10	TCP	22 > 44452 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=1 SACK_PERM=1 Tsvl=506280808 TSecr=1825190160				
3	29-58.2 172.20.21.172.20.21.1	TCP	44452 > 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 Tsvl=1825190160 TSecr=506280808				
4	29-58.2 172.20.21.172.20.21.1	SSHv2	Client: Protocol (SSH-2.0-libssh_0.9.6)				
5	29-58.2 172.20.21.172.20.21.10	SSHv2	Server: Protocol (SSH-2.0-OpenSSH_7.1)				
6	29-58.2 172.20.21.172.20.21.1	TCP	44452 > 22 [ACK] Seq=23 Ack=22 Win=64256 Len=0 Tsvl=1825190175 TSecr=506280809				
7	29-58.2 172.20.21.172.20.21.1	SSHv2	Client: Key Exchange Init				
8	29-58.2 172.20.21.172.20.21.1	SSHv2	Server: Key Exchange Init				
9	29-58.2 172.20.21.172.20.21.1	TCP	44452 > 22 [ACK] Seq=999 Ack=958 Win=64128 Len=0 Tsvl=1825190190 TSecr=506280810				
10	29-58.2 172.20.21.172.20.21.1	SSHv2	Client: Elliptic Curve Diffie-Hellman Key Exchange Init				
11	29-58.2 172.20.21.172.20.21.10	SSHv2	Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys				
12	29-58.2 172.20.21.172.20.21.1	TCP	44452 > 22 [ACK] Seq=1047 Ack=1366 Win=64128 Len=0 Tsvl=1825190206 TSecr=506280812				
13	29-58.2 172.20.21.172.20.21.1	SSHv2	Client: New Keys				
14	29-58.4 172.20.21.172.20.21.10	TCP	22 > 44452 [ACK] Seq=1366 Ack=1063 Win=64473 Len=0 Tsvl=506280823 TSecr=1825190206				
15	29-58.4 172.20.21.172.20.21.1	SSHv2	Client: Encrypted packet (len=52)				
16	29-58.4 172.20.21.172.20.21.10	SSHv2	Server: Encrypted packet (len=52)				
17	29-58.4 172.20.21.172.20.21.1	SSHv2	Client: Encrypted packet (len=68)				
18	29-58.4 172.20.21.172.20.21.10	SSHv2	Server: Encrypted packet (len=84)				
19	29-58.4 172.20.21.172.20.21.1	SSHv2	Client: Encrypted packet (len=52)				
20	29-58.4 172.20.21.172.20.21.1	TCP	44452 > 22 [FIN, ACK] Seq=1235 Ack=1502 Win=64128 Len=0 Tsvl=1825190425 TSecr=506280834				
21	29-58.4 172.20.21.172.20.21.10	TCP	22 > 44452 [ACK] Seq=1502 Ack=1236 Win=64301 Len=0 Tsvl=506280834 TSecr=1825190425				
22	29-58.4 172.20.21.172.20.21.10	TCP	22 > 44452 [FIN, ACK] Seq=1502 Ack=1236 Win=64301 Len=0 Tsvl=506280834 TSecr=1825190425				
23	29-58.4 172.20.21.172.20.21.1	TCP	44452 > 22 [ACK] Seq=1236 Ack=1503 Win=64128 Len=0 Tsvl=1825190428 TSecr=506280834				
24	29-58.7 172.20.21.172.20.21.1	TCP	44458 > 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 Tsvl=1825190649 TSecr=0 WS=128				
25	29-58.7 172.20.21.172.20.21.1	TCP	44456 > 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 Tsvl=1825190649 TSecr=0 WS=128				
26	29-58.7 172.20.21.172.20.21.1	TCP	44454 > 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 Tsvl=1825190649 TSecr=0 WS=128				
27	29-58.7 172.20.21.172.20.21.1	TCP	44460 > 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 Tsvl=1825190649 TSecr=0 WS=128				

Figure 32: SSH-Brute Force attack

NMAP scan



No.	Time	Source	Destinatio	Cookie	pai	Protocol	Info
1	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
2	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 8888 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
3	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 23 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
4	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 256 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
5	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 993 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
6	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 133 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
7	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
8	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 1025 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
9	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
10	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 3306 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
11	15:15.6 172.20.21.172.20.21.10	TCP	3306 > 50438 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460				
12	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 3306 [RST] Seq=1 Win=0 Len=0				
13	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 8080 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
14	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 138 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
15	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 3389 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
16	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 587 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
17	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
18	15:15.6 172.20.21.172.20.21.2	TCP	[TCP Out-Of-Order] 50438 > 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
19	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
20	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 354 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
21	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 143 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
22	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 53 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
23	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
24	15:15.6 172.20.21.172.20.21.10	TCP	22 > 50438 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460				
25	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				
26	15:15.6 172.20.21.172.20.21.10	TCP	443 > 50438 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0				
27	15:15.6 172.20.21.172.20.21.2	TCP	50438 > 1720 [SYN] Seq=0 Win=1024 Len=0 MSS=1460				

Figure 33: NMAP Scan

References

<https://upcloud.com/community/tutorials/installing-snort-on-centos/>

<https://www.rapid7.com/blog/post/2016/12/09/understanding-and-configuring-snort-rules/#:~:text=Usually%2C%20Snort%20rules%20were%20written,large%20and%20difficult%20to%20understand.>

https://www.researchgate.net/publication/338660054_DETECTING_DDoS_ATTACK_USING_Snort

<https://www.clearos.com/clearfoundation/social/community/solved-snort-rule-for-ftp-brute-force>