



EE6042/ET4028 Host & Network Security

Intrusion Detection Systems Assignment

Ashutosh Yadav

18249094

Department of Electronics & Computer Engineering

University of Limerick

Table of Contents

| | |
|---|----|
| Table of Figures | 2 |
| Introduction..... | 3 |
| Attack 1: Ping Scan, TCP Scan & XMAS Scan | 4 |
| Attack 2: Brute Force FTP Attack | 7 |
| Attack 3: Denial of Service Attack | 10 |
| Additional Information | 13 |
| References..... | 14 |

Table of Figures

| | |
|--|----|
| Figure 1 NMAP scan rules..... | 5 |
| Figure 2 NMAP scan detection..... | 5 |
| Figure 3 Ncrack use | 7 |
| Figure 4 Snippet of alert_BRUTEFORCE | 8 |
| Figure 5 hping3 use..... | 10 |
| Figure 6 Snippet from alert_DDOS | 11 |

Introduction

The assignment was performed using the following virtual machines through Oracle VirtualBox.

- Ubuntu 20.04 LTS Victim Machine (IDS & Victim)
- Ubuntu 20.04 LTS Attacker Machine (Virtual Machine)

The IP addresses of the two virtual machines are:

- *192.168.0.10 (Attacker, Dynamic IP)*
- *192.168.0.24 / 192.168.0.21 (Victim & IDS Dynamic IP)*

The snort IDS command used:

```
# sudo snort -A full -u snort -g snort -c /etc/snort/snort.conf -I enp0s3
```

Description of all the components/options:

-A full : Full alert mode. This is the default alert mode. **-A console** was used during early stages such as the ping/tcp and xmas scan.

-u : Changes the default user ID or UID under which snort runs after start-up.

-g : Changes the default group ID or GID under which snort runs after start-up.

-c : Prints the character data found in the packet payload. In this case it is printed to **snort.conf**.

-i : Specifies the interface snort should listen on. Gathered using **ip addr** command which is **enp0s3**.

Attack 1: Ping Scan, TCP Scan & XMAS Scan

When using this **TCP scan**, Nmap sends TCP and UDP packets to a particular port, and then analyses its response.

Ping Scan is used to identify any active hosts on the network. It identifies all IP addresses that are currently online without sending any packets to the hosts themselves. This is particularly useful when scanning a large network to discover machines in what could be a sea of IP addresses, both active and inactive.

Xmas Scan sends a packet with FIN, URG and PSH flags set. This was considered a very stealthy scan as if the port is open, no response is needed. But if its closed, the victim would respond with an ACK packet.

NNAMP Commands used for attack:

sudo nmap -sX 192.168.0.24 (Xmas scan)

sudo nmap -sT 192.168.0.24 (TCP Scan)

sudo nmap -sP 192.168.0.24 --disable-arp-ping (Ping Scan)

Rules:

```
566 #NMAP RULES FOR PING SCANS
567
568 #NMAP Ping scan
569
570 alert icmp any any -> 192.168.0.24 any (msg: "NMAP Ping Scan"; dsize:0;sid:10000004; rev: 1;)
571
572 #To detect TCP Scan
573
574 alert tcp any any -> 192.168.0.24 any (msg: "NMAP TCP Scan"; classtype:network-scan; sid:10000005; rev:2; )
575
576 #XMAS Scan
577
578 alert tcp any any -> 192.168.0.24 any (msg:"Nmap XMAS Scan"; flags:FPU; classtype:network-scan; sid:10000006; rev:1; )
579
580 #####
```

Figure 1 NMAP scan rules

Detection:

```
ash@ash-VirtualBox:~$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i enp0s3
04/12-22:29:21.691934 1:10000005:2 NMAP TCP Scan 1:10000005:2 [Priority: 0] {TCP} 93.184.220.70:443 -> 192.168.0.24:49164
04/12-22:29:51.706309 1:10000005:2 NMAP TCP Scan 1:10000005:2 [Priority: 0] {TCP} 93.184.220.70:443 -> 192.168.0.24:49164
04/12-22:29:54.834027 1:10000004:1 NMAP Ping Scan 1:10000004:1 [Priority: 0] {ICMP} 192.168.0.10 -> 192.168.0.24
04/12-22:29:54.834243 1:10000005:2 NMAP TCP Scan 1:10000005:2 [Priority: 0] {TCP} 192.168.0.10:3153 -> 192.168.0.24:443
04/12-22:29:55.334738 1:10000005:2 NMAP TCP Scan 1:10000005:2 [Priority: 0] {TCP} 192.168.0.10:3153 -> 192.168.0.24:443
04/12-22:29:55.835159 1:10000005:2 NMAP TCP Scan 1:10000005:2 [Priority: 0] {TCP} 192.168.0.10:3153 -> 192.168.0.24:443
04/12-22:29:56.336197 1:10000005:2 NMAP TCP Scan 1:10000005:2 [Priority: 0] {TCP} 192.168.0.10:3153 -> 192.168.0.24:443
04/12-22:29:56.836496 1:10000005:2 NMAP TCP Scan 1:10000005:2 [Priority: 0] {TCP} 192.168.0.10:3153 -> 192.168.0.24:443
04/12-22:30:07.922403 1:10000004:1 NMAP Ping Scan 1:10000004:1 [Priority: 0] {ICMP} 192.168.0.10 -> 192.168.0.24
04/12-22:30:07.922526 1:10000005:2 NMAP TCP Scan 1:10000005:2 [Priority: 0] {TCP} 192.168.0.10:3156 -> 192.168.0.24:443
04/12-22:30:08.423032 1:10000005:2 NMAP TCP Scan 1:10000005:2 [Priority: 0] {TCP} 192.168.0.10:3156 -> 192.168.0.24:443
04/12-22:30:08.924318 1:10000005:2 NMAP TCP Scan 1:10000005:2 [Priority: 0] {TCP} 192.168.0.10:3156 -> 192.168.0.24:443
04/12-22:30:09.424774 1:10000005:2 NMAP TCP Scan 1:10000005:2 [Priority: 0] {TCP} 192.168.0.10:3156 -> 192.168.0.24:443
04/12-22:30:09.925162 1:10000005:2 NMAP TCP Scan 1:10000005:2 [Priority: 0] {TCP} 192.168.0.10:3156 -> 192.168.0.24:443
04/12-22:30:21.749823 1:10000005:2 NMAP TCP Scan 1:10000005:2 [Priority: 0] {TCP} 93.184.220.70:443 -> 192.168.0.24:49164
```

Figure 2 NMAP scan detection

Rule and file location:

Rule(s) location: **snort.conf**

Line: 564 - 572

Log file and alert location:

Alert file: **alert_Ports**

Log file: **snort.log.1649856623**

Description of all the components/options:

alert icmp any any -> 192.168.0.24 any (msg: "NMAP Ping Scan"; dsize:0;sid:10000004; rev:1;)

alert tcp any any -> 192.168.0.24 any (msg: "NMAP TCP Scan"; classtype:network-scan; sid:10000005; rev:2;)

alert tcp any any -> 192.168.0.24 any (msg:"Nmap XMAS Scan"; flags:FPU; classtype:network-scan; sid:10000006; rev:1;)

- **Alert** – Generates alert when condition is met
- **TCP** – Traffic type
- **ICMP** – Traffic type (used for ping scan here)
- **Any** – Source IP address as the scan can come from anywhere
- **Any** – Source port, same as above, can be any port
- **192.168.0.24** – Destination IP address
- **Any** – Snort will look at any port range here for the above IP address
- **Msg** – Message contents inserted in the alert file
- **Flags** – Snort will look for FPU flag being set (Xmas scan only)
- **Classtype** – Used to categorise rule as detecting particular class of attack. Network scan chosen here for all 3 scans.
- **sid** - sid:1000004, sid:10000005, sid:10000006 – Unique identification for the snort rules.
- **Rev** – Revision number of rule.

Attack 2: Brute Force FTP Attack

A brute force attack is a hacking method that would use a trial and error approach in an attempt to crack credentials for login, passwords as well encryption keys. The attacker would work through a dictionary of common and possible usernames/passwords and attempts to use all of them until they find the correct logins.

Text files for user/pass were created and used which are enclosed with the submission and can be found in the “*users*” folder.

I purposefully included the correct username and password to demonstrate the use of ncrack and successful credential discovery.

This attack will be carried out using Ncrack on FTP.

NCRACK Commands used for attack:

```
# sudo ncrack -u users.txt -P pass.txt ftp://192.168.0.24
```

See the attack below:

```
ash@ash-VirtualBox:~/Desktop/users$ ncrack -U users.txt -P pass.txt ftp://192.168.0.24

Starting Ncrack 0.7 ( http://ncrack.org ) at 2022-04-13 15:53 IST

Discovered credentials for ftp on 192.168.0.24 21/tcp:
192.168.0.24 21/tcp ftp: 'ash' 'pass1234'

Ncrack done: 1 service scanned in 37.37 seconds.

Ncrack finished.
```

Figure 3 Ncrack use

Rule and file location:

Rule(s) location: **snort.conf**

Line: 578

Log file and alert location:

Alert file: **alert_BRUTEFORCE**

Log file: **snort.log.1649861582**

Detection:

Snippet from alert_BRUTEFORCE file

```
[**] [1:100000003:1] Brute Force Attack, Intruder attempting FTP Access [**]
[Classification: Unsuccessful User Privilege Gain] [Priority: 1]
04/13-15:53:11.393138 192.168.0.25:36386 -> 192.168.0.24:21
TCP TTL:64 TOS:0x0 ID:13115 IpLen:20 DgmLen:63 DF
***AP*** Seq: 0xC2D3ED92 Ack: 0x63306D19 Win: 0x1F6 Tcplen: 32
TCP Options (3) => NOP NOP TS: 4282013748 3310536503

[**] [1:100000003:1] Brute Force Attack, Intruder attempting FTP Access [**]
[Classification: Unsuccessful User Privilege Gain] [Priority: 1]
04/13-15:53:14.749429 192.168.0.25:36390 -> 192.168.0.24:21
TCP TTL:64 TOS:0x0 ID:46110 IpLen:20 DgmLen:63 DF
***AP*** Seq: 0xD3D974C9 Ack: 0xC1C441A4 Win: 0x1F6 Tcplen: 32
TCP Options (3) => NOP NOP TS: 4282017104 3310539860

[**] [1:100000003:1] Brute Force Attack, Intruder attempting FTP Access [**]
[Classification: Unsuccessful User Privilege Gain] [Priority: 1]
04/13-15:53:14.749597 192.168.0.25:36394 -> 192.168.0.24:21
TCP TTL:64 TOS:0x0 ID:6711 IpLen:20 DgmLen:65 DF
***AP*** Seq: 0xF6306646 Ack: 0x9ACB060C Win: 0x1F6 Tcplen: 32
TCP Options (3) => NOP NOP TS: 4282017104 3310539860

[**] [1:100000003:1] Brute Force Attack, Intruder attempting FTP Access [**]
[Classification: Unsuccessful User Privilege Gain] [Priority: 1]
04/13-15:53:14.750560 192.168.0.25:36406 -> 192.168.0.24:21
TCP TTL:64 TOS:0x0 ID:54348 IpLen:20 DgmLen:63 DF
***AP*** Seq: 0x478C30AA Ack: 0x63B050AD Win: 0x1F6 Tcplen: 32
TCP Options (3) => NOP NOP TS: 4282017105 3310539860

[**] [1:100000003:1] Brute Force Attack, Intruder attempting FTP Access [**]
[Classification: Unsuccessful User Privilege Gain] [Priority: 1]
04/13-15:53:14.751122 192.168.0.25:36402 -> 192.168.0.24:21
TCP TTL:64 TOS:0x0 ID:58246 IpLen:20 DgmLen:63 DF
***AP*** Seq: 0xCAD9AFC2 Ack: 0xBAF98B29 Win: 0x1F6 Tcplen: 32
TCP Options (3) => NOP NOP TS: 4282017106 3310539861
```

Figure 4 Snippet of alert_BRUTEFORCE

Description of all the components/options:

alert tcp any any -> 192.168.0.24 any (msg:"Brute Force Attack, Intruder attempting FTP Access"; pcre:"/USER/PASS/i"; flow:to_server; classtype:unsuccessful-user; threshold: type threshold, track by_src, count 3, seconds 60; sid:100000007; rev:1;)

- **Alert** – Generates alert when condition is met
- **TCP** – Traffic type
- **Any** – Source IP address as the scan can come from anywhere
- **Any** – Source port, same as above, can be any port
- **192.168.0.24** – Destination IP address
- **Any** – Snort will look at any port range here for the above IP address
- **Msg** – Message contents inserted in the alert file
- **pcre** – Snort IDS will look for anything between the 2 “/” and “i” is used to ensure it is not case sensitive.
- **Classtype** – Used to categorise rule as detecting particular class of attack. Using *unsuccessful-user* for this classtype.
- **sid** - sid:10000007 – Unique identification for the snort rules.
- **Rev** – Revision number of rule.
- **flow** – to_server Trigger on client request from A to B.
- **Threshold** – Sends an alert whenever the event occurs. Track by src maintains for any unique IP. Count is set to 3 and seconds is 60 for the time period.

Attack 3: Denial of Service Attack

A denial of service attack or commonly known as “DoS” is an attack that is meant to shut down a network or machine(s) which results in the loss of access to the user. This is accomplished by flooding the target machine with information or “traffic” to trigger a crash. This attack is commonly used on high profile organizations or targets such as web servers, banking infrastructure, government and state companies as an example. This doesn’t result in theft or data or loss of information but can cause significant downtime and resources to resolve.

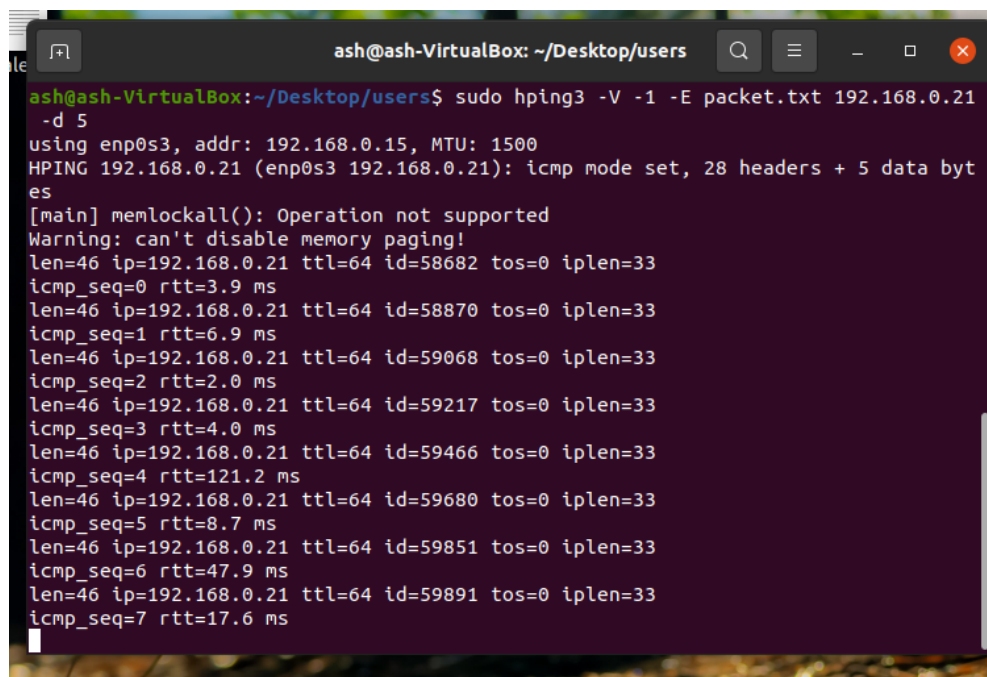
The tool used here is hping3 which is a free packet generator and analyser for the TCP/IP protocol.

The command used to execute the attack can be seen below and content of “packet.txt” was “Ash”.

Hping3 Commands used for attack:

sudo hping3 -V -I -E packet.txt 192.168.0.21 -d 5

See the attack below:



```
ash@ash-VirtualBox: ~/Desktop/users
ash@ash-VirtualBox:~/Desktop/users$ sudo hping3 -V -I -E packet.txt 192.168.0.21 -d 5
using enp0s3, addr: 192.168.0.15, MTU: 1500
HPING 192.168.0.21 (enp0s3 192.168.0.21): icmp mode set, 28 headers + 5 data bytes
[main] memlockall(): Operation not supported
Warning: can't disable memory paging!
len=46 ip=192.168.0.21 ttl=64 id=58682 tos=0 iplen=33
icmp_seq=0 rtt=3.9 ms
len=46 ip=192.168.0.21 ttl=64 id=58870 tos=0 iplen=33
icmp_seq=1 rtt=6.9 ms
len=46 ip=192.168.0.21 ttl=64 id=59068 tos=0 iplen=33
icmp_seq=2 rtt=2.0 ms
len=46 ip=192.168.0.21 ttl=64 id=59217 tos=0 iplen=33
icmp_seq=3 rtt=4.0 ms
len=46 ip=192.168.0.21 ttl=64 id=59466 tos=0 iplen=33
icmp_seq=4 rtt=121.2 ms
len=46 ip=192.168.0.21 ttl=64 id=59680 tos=0 iplen=33
icmp_seq=5 rtt=8.7 ms
len=46 ip=192.168.0.21 ttl=64 id=59851 tos=0 iplen=33
icmp_seq=6 rtt=47.9 ms
len=46 ip=192.168.0.21 ttl=64 id=59891 tos=0 iplen=33
icmp_seq=7 rtt=17.6 ms
```

Figure 5 hping3 use

Rule and file location:

Rule(s) location: **snort.conf**

Line: 583

Log file and alert location:

Alert file: **alert_DDOS**

Log file: **snort.log.1650115880**

Detection:

Snippet from alert_DDOS file

```
[**] [1:100000002:1] DDOS attack! We have a problem [**]
[Classification: Detection of a Denial of Service Attack] [Priority: 2]
04/16-14:31:25.865065 192.168.0.15 -> 192.168.0.21
ICMP TTL:64 TOS:0x0 ID:56430 IpLen:20 DgmLen:33
Type:8 Code:0 ID:59914 Seq:256 ECHO

[**] [1:100000002:1] DDOS attack! We have a problem [**]
[Classification: Detection of a Denial of Service Attack] [Priority: 2]
04/16-14:31:26.867971 192.168.0.15 -> 192.168.0.21
ICMP TTL:64 TOS:0x0 ID:24766 IpLen:20 DgmLen:33
Type:8 Code:0 ID:59914 Seq:512 ECHO

[**] [1:100000002:1] DDOS attack! We have a problem [**]
[Classification: Detection of a Denial of Service Attack] [Priority: 2]
04/16-14:31:27.868839 192.168.0.15 -> 192.168.0.21
ICMP TTL:64 TOS:0x0 ID:57247 IpLen:20 DgmLen:33
Type:8 Code:0 ID:59914 Seq:768 ECHO

[**] [1:100000002:1] DDOS attack! We have a problem [**]
[Classification: Detection of a Denial of Service Attack] [Priority: 2]
04/16-14:31:28.869106 192.168.0.15 -> 192.168.0.21
ICMP TTL:64 TOS:0x0 ID:49529 IpLen:20 DgmLen:33
Type:8 Code:0 ID:59914 Seq:1024 ECHO

[**] [1:100000002:1] DDOS attack! We have a problem [**]
[Classification: Detection of a Denial of Service Attack] [Priority: 2]
04/16-14:31:29.870221 192.168.0.15 -> 192.168.0.21
ICMP TTL:64 TOS:0x0 ID:40283 IpLen:20 DgmLen:33
Type:8 Code:0 ID:59914 Seq:1280 ECHO
```

Figure 6 Snippet from alert_DDOS

Description of all the components/options:

*alert icmp any any -> 192.168.0.21 any (msg:"DDOS attack! We have a problem";
content:"Ash"; itype:8; classtype:denial-of-service; sid:100000007; rev:1;)*

- **Alert** – Generates alert when condition is met
- **ICMP** – Traffic type
- **Any** – Source IP address as the scan can come from anywhere
- **Any** – Source port, same as above, can be any port
- **192.168.0.21** – Destination IP address
- **Any** – Snort will look at any port range here for the above IP address
- **Msg** – Message contents inserted in the alert file
- **Content** – Searches for the content in the attacker packet file which for this attack was “Ash” and was present in the packet.txt file used in the attack.
- **itype** – This is a rule that test the value of the ICMP type field. The value is a numeric value and the values can be set out of range to detect invalid ICMP type values that are often used in DoS attacks. Here it is set to an echo request (8). Usual format: itype: “ICMP_type_number” [10].
- **Classtype** - Used to categorise rule as detecting particular class of attack. Using *denial-of-service* for this classtype.
- **sid** - sid:100000008 – Unique identification for the snort rules.
- **Rev** – Revision number of rule.

Additional Information

All 3 attacks are in separate folders containing the following:

- 1x alert file
- 1x snort.log file

The snort.conf file is included in the main directory of the submission folder.

A “users” folder is included containing the user,pass and packet text files used during the attacks.

The required options can be seen below:

content option used: Denial of service attack; Attack 3

pcrc option used: Brute Force FTP Attack; Attack 2

flow option used: Brute Force FTP Attack; Attack 2

itype option used: Denial of service attack; Attack 3

References

- [1] Cloudsavvyit.com, 2022. [Online]. Available: <https://www.cloudsavvyit.com/6424/how-to-use-the-snort-intrusion-detection-system-on-linux/>. [Accessed: 10- Apr- 2022].
- [2] R.Chandel, "Comprehensive Guide on Snort (Part 1) - Hacking Articles", Hacking Articles, 2022. [Online]. Available: <https://www.hackingarticles.in/comprehensive-guide-on-snort-part-1/>. [Accessed: 11- Apr- 2022].
- [3] R. Chandel, "How to Detect NMAP Scan Using Snort - Hacking Articles", Hacking Articles, 2022. [Online]. Available: <https://www.hackingarticles.in/detect-nmap-scan-using-snort/>. [Accessed: 12- Apr- 2022].
- [4] R. Chandel, "Comprehensive Guide on Ncrack - A Brute Forcing Tool - Hacking Articles", Hacking Articles, 2022. [Online]. Available: <https://www.hackingarticles.in/comprehensive-guide-on-ncrack-a-brute-forcing-tool/>. [Accessed: 13- Apr- 2022].
- [5] "Ncrack Reference Guide (Man Page) | Ncrack Reference Guide (Man Page)", Nmap.org, 2022. [Online]. Available: <https://nmap.org/ncrack/man.html>. [Accessed: 13- Apr- 2022].
- [6] "Chapter 15. Nmap Reference Guide | Nmap Network Scanning", Nmap.org, 2022. [Online]. Available: <https://nmap.org/book/man.html>. [Accessed: 13- Apr- 2022].
- [7] "Writing Snort Rules", Paginas.fe.up.pt, 2022. [Online]. Available: https://paginas.fe.up.pt/~mgi98020/pgr/writing_snort_rules.htm. [Accessed: 14- Apr- 2022].
- [8] "3.3 Command-Line Options", Books.gigatux.nl, 2022. [Online]. Available: <http://books.gigatux.nl/mirror/snortids/0596006616/snortids-CHP-3-SECT-3.html>. [Accessed: 14- Apr- 2022].
- [9] Gogoi, Manas & Mishra, Sourav. (2018). DETECTING DDoS ATTACK USING Snort. [Online]. Available: https://www.researchgate.net/publication/338660054_DETECTING_DDoS_ATTACK_USING_Snort [Accessed: 14- Apr- 2022].

[10] "Rule Options | Working with Snort Rules | InformIT", Informit.com, 2022. [Online]. Available: <https://www.informit.com/articles/article.aspx?p=101171&seqNum=6>. [Accessed: 16- Apr- 2022].