



Veermata Jijabai Technological Institute, Mumbai 400019

Experiment No.: 05

Aim: Install SNORT, an open-source tool and implement IDS to detect attacks in the network.

Group : Kiran Patil - 211070904
Mayuresh Murudkar - 211070903
Pratiksha Sankhe – 201071049

Branch: Final Year B.Tech Computer Engineering

Batch: D

Theory:

The primary objective of this experiment is to establish an Intrusion Detection System (IDS) using SNORT, an open-source and widely-used tool for network security. The experiment involves the creation of a controlled network environment, simulating network attacks, and monitoring these attacks using SNORT. Here are additional details:

1. SNORT:

Purpose: SNORT is an open-source network intrusion detection and prevention system. It's designed to monitor network traffic and detect suspicious or malicious activity.

Features:

- **Packet inspection:** SNORT inspects network packets for patterns that match predefined rules.
- **Alerting:** When a rule is triggered, SNORT generates alerts and logs information about the detected activity.
- **Rule-based:** SNORT uses a rule-based language to specify what to look for and how to respond.
- **Customizable:** You can create custom rules to tailor SNORT to your network's specific needs.

Installation:

- SNORT can be installed on various Linux distributions. You can use package managers like apt for Debian-based systems or compile it from source.

2. Kali Linux:

Purpose:

- Kali Linux is a popular penetration testing and ethical hacking distribution. In this experiment, it will serve as the attacker machine.

Features:

- **Pre-installed hacking tools:** Kali Linux comes with a wide range of pre-installed security and hacking tools.
- **Customizable:** You can add or remove tools as needed for your security assessments.

Installation:

- Kali Linux can be downloaded and installed on a dedicated machine or run from a live USB.

3. Packet Crafting Tools:

Purpose:

- To simulate attacks, you may need packet crafting tools to generate malicious packets with specific characteristics.

Examples:

- **Scapy:** A powerful packet manipulation tool that allows you to create, send, and capture network packets.
- **Nmap:** A network scanning tool that can be used to simulate port scans and other network reconnaissance activities.
- **Hping:** A command-line tool for crafting and sending custom packets.

Installation:

- These tools can be installed on Kali Linux using package managers or by compiling from source.

4. Configuration Files:

- **SNORT Rules:**

SNORT uses rules to specify what to look for and how to respond to network activity. These rules are defined in .rules files and typically located in the SNORT configuration directory.

- **SNORT Configuration:**

The SNORT configuration file (usually snort.conf) specifies various settings, including which network interfaces to monitor and where to log alerts.

5. Log Files:

- **SNORT Logs:**

SNORT generates log files that contain information about detected network activity. Common log locations include /var/log/snort/ or a directory specified in the SNORT configuration.

6. Documentation and Screenshots:

- **Purpose:**

To document the experiment, you'll need to capture and annotate screenshots of key steps and results.

- **Tools:**

You can use built-in screenshot utilities on your operating system or third-party tools like Greenshot, Shutter, or Flameshot to capture and annotate screenshots.

7. Network Environment:

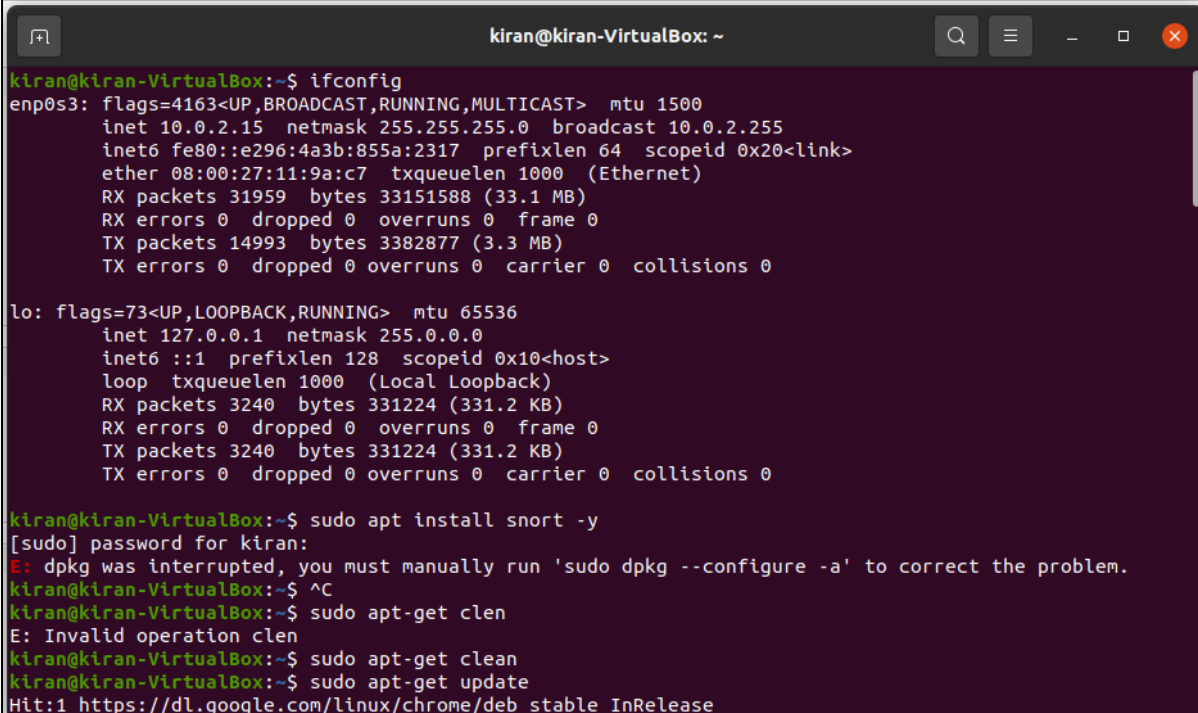
- **Isolation:**

It's crucial to conduct this experiment in an isolated and controlled network environment to prevent any unintended consequences on a live network.

Remember that conducting network attack simulations and using tools like SNORT should be done ethically and legally, with proper authorization and consent. Unauthorized or malicious activities are illegal and unethical. Always use these tools responsibly and for legitimate security testing or research purposes.

Implementation:

Ifconfig : to know your ip address



```
kiran@kiran-VirtualBox: ~  
kiran@kiran-VirtualBox:~$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255  
    inet6 fe80::e296:4a3b:855a:2317 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:11:9a:c7 txqueuelen 1000 (Ethernet)  
    RX packets 31959 bytes 33151588 (33.1 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 14993 bytes 3382877 (3.3 MB)  
    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 3240 bytes 331224 (331.2 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 3240 bytes 331224 (331.2 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
kiran@kiran-VirtualBox:~$ sudo apt install snort -y  
[sudo] password for kiran:  
E: dpkg was interrupted, you must manually run 'sudo dpkg --configure -a' to correct the problem.  
kiran@kiran-VirtualBox:~$ ^C  
kiran@kiran-VirtualBox:~$ sudo apt-get clen  
E: Invalid operation clen  
kiran@kiran-VirtualBox:~$ sudo apt-get clean  
kiran@kiran-VirtualBox:~$ sudo apt-get update  
Hit:1 https://dl.google.com/linux/chrome/deb stable InRelease
```

Install snort :

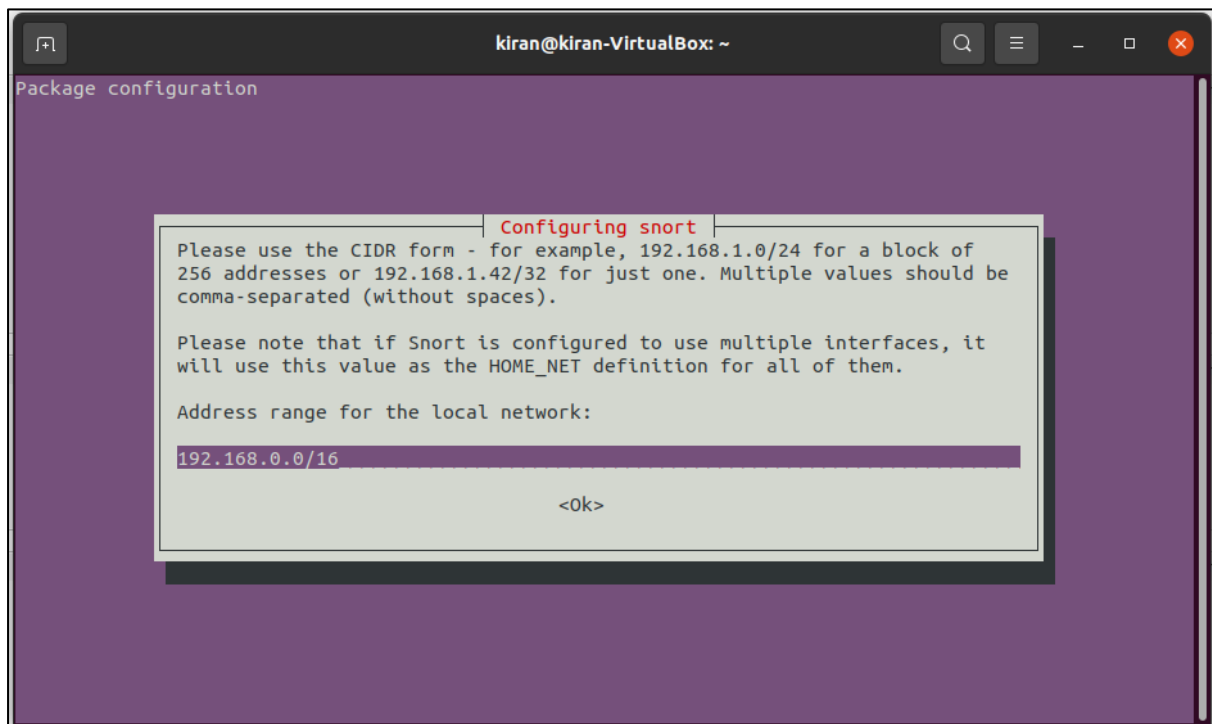
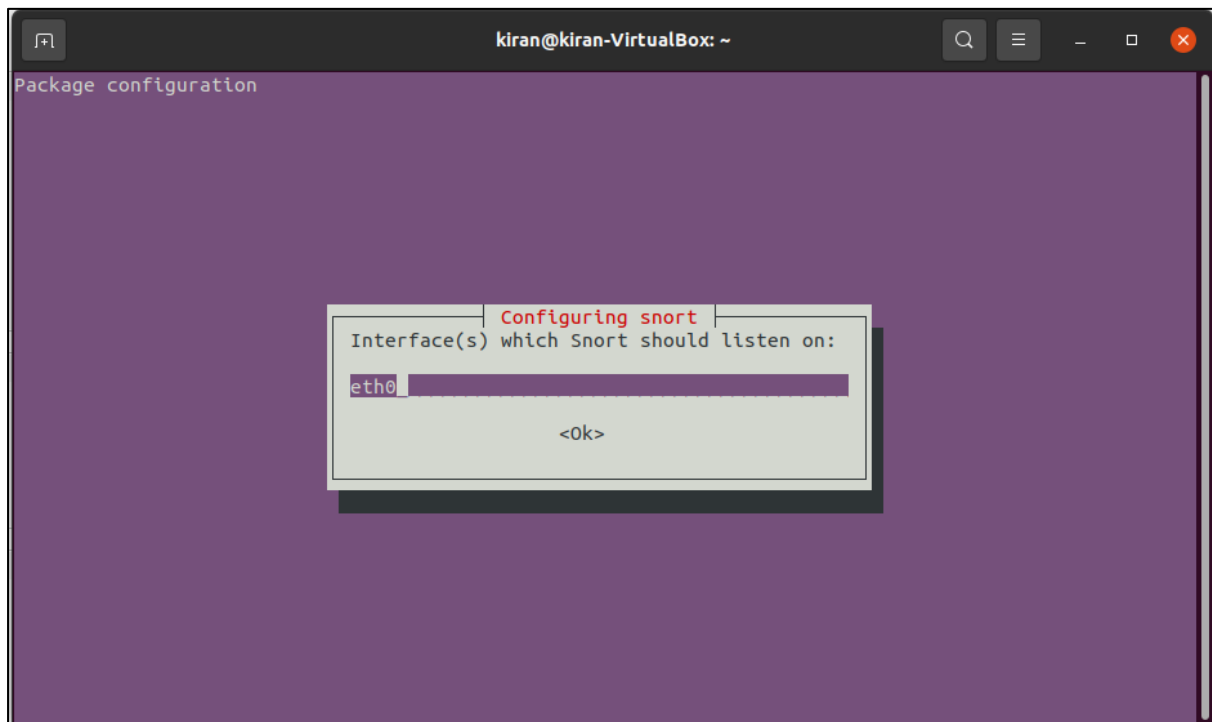
sudo apt install snort -y

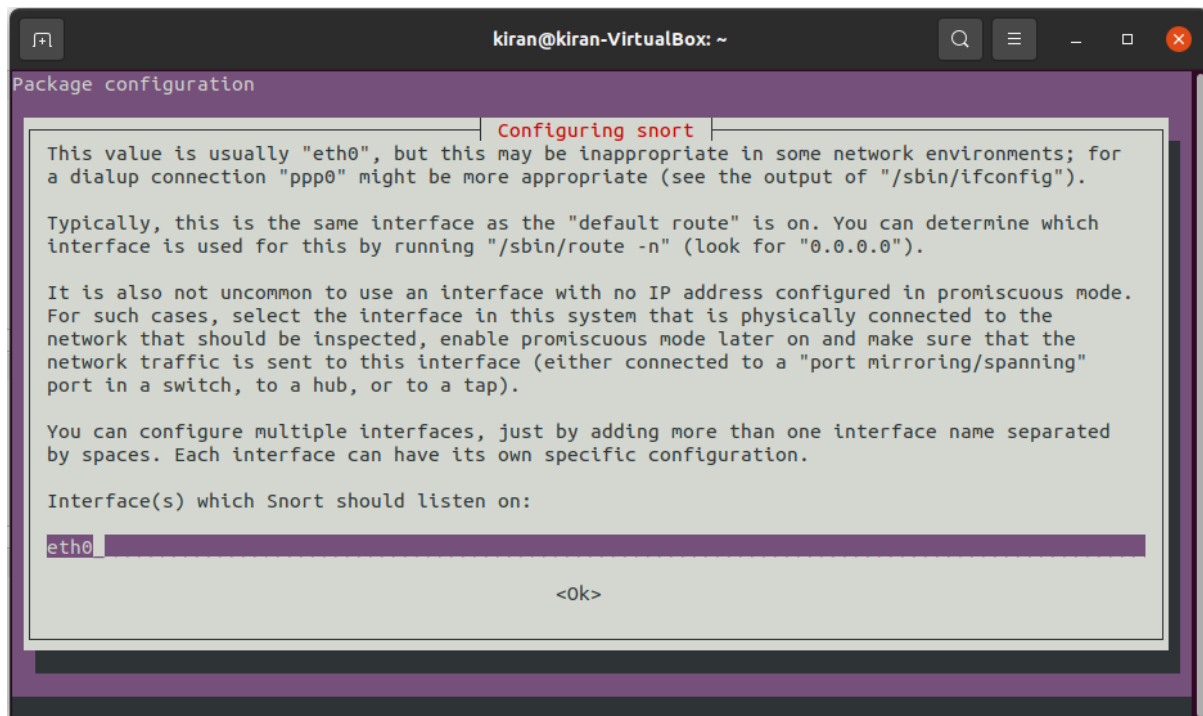
```
kiran@kiran-VirtualBox: ~  
kiran@kiran-VirtualBox:~$ sudo apt install snort -y  
[sudo] password for kiran:  
E: dpkg was interrupted, you must manually run 'sudo dpkg --configure -a' to correct the problem.  
kiran@kiran-VirtualBox:~$ ^C  
kiran@kiran-VirtualBox:~$ sudo apt-get clen  
E: Invalid operation clen  
kiran@kiran-VirtualBox:~$ sudo apt-get clean  
kiran@kiran-VirtualBox:~$ sudo apt-get update  
Hit:1 https://dl.google.com/linux/chrome/deb stable InRelease  
Hit:2 http://in.archive.ubuntu.com/ubuntu focal InRelease  
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease  
Get:4 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]  
Hit:5 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease  
Fetched 114 kB in 2s (64.6 kB/s)  
Reading package lists... Done  
kiran@kiran-VirtualBox:~$ sudo apt-get dist-upgrade  
E: dpkg was interrupted, you must manually run 'sudo dpkg --configure -a' to correct the problem.  
kiran@kiran-VirtualBox:~$ sudo dpkg --configure -a  
Setting up shared-mime-info (1.15-1) ...  
kiran@kiran-VirtualBox:~$ sudo apt install snort -y  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  lam-runtime liblam4  
Use 'sudo apt autoremove' to remove them.  
The following additional packages will be installed:  
  libdaq2 libdumbnet1 oinkmaster snort-common snort-common-libraries
```

Configure the snort

```
kiran@kiran-VirtualBox: ~  
Package configuration  
Configuring snort  
  
This value is usually "eth0", but this may be inappropriate in some  
network environments; for a dialup connection "ppp0" might be more  
appropriate (see the output of "/sbin/ifconfig").  
  
Typically, this is the same interface as the "default route" is on. You  
can determine which interface is used for this by running "/sbin/route  
-n" (look for "0.0.0.0").  
  
It is also not uncommon to use an interface with no IP address  
configured in promiscuous mode. For such cases, select the interface in  
this system that is physically connected to the network that should be  
inspected, enable promiscuous mode later on and make sure that the  
network traffic is sent to this interface (either connected to a "port  
mirroring/spanning" port in a switch, to a hub, or to a tap).  
  
<Ok>  
  
[.....]  
[.....]  
[.....]  
[.....]
```

Set the interface the default interface is set to eth0



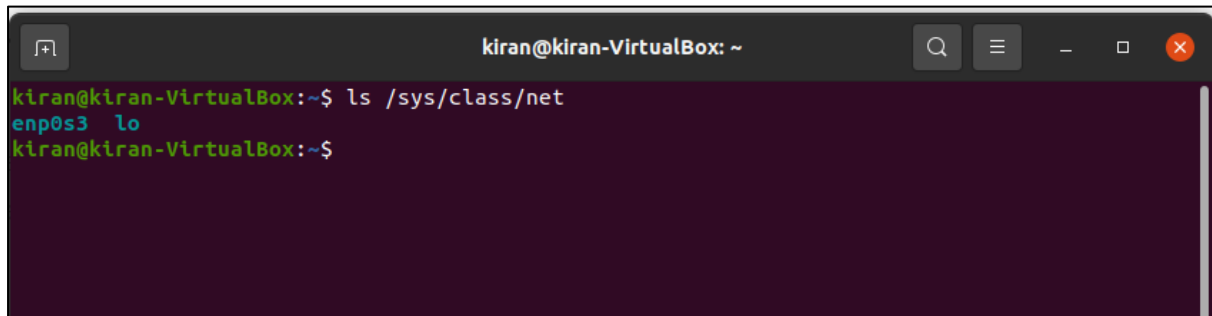


Here the eth0 interface is an invalid interface



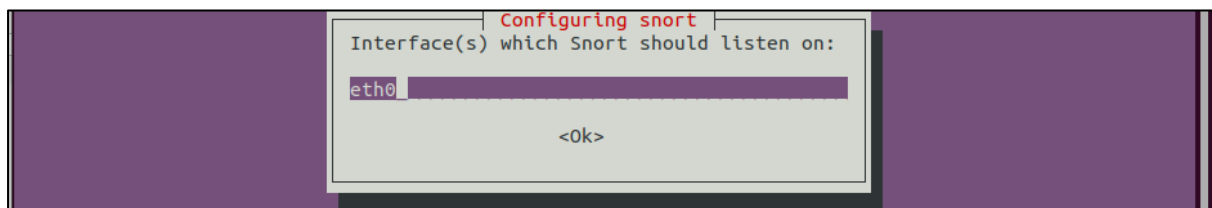
Get interface :

ls /sys/class/net



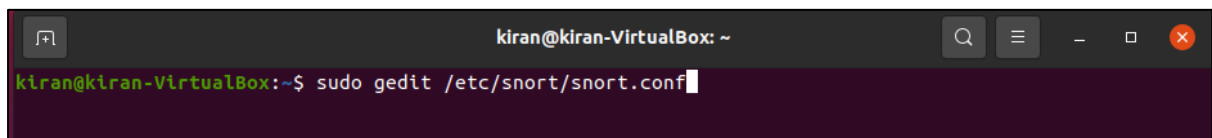
```
kiran@kiran-VirtualBox: ~  
kiran@kiran-VirtualBox:~$ ls /sys/class/net  
enp0s3  lo  
kiran@kiran-VirtualBox:~$
```

Then configure the snort with enp0s3 interface by changing it from eth0 to enp0s3



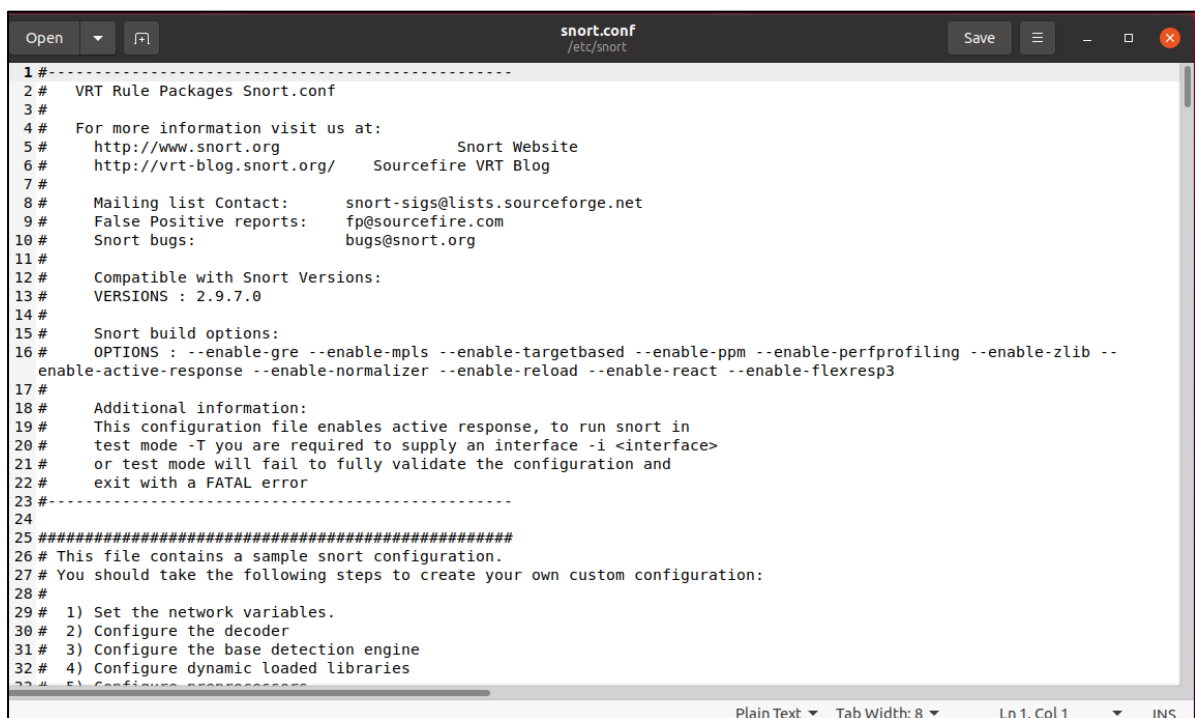
Now check the snort configuration file

Sudo gedit /etc/snort/snort.conf



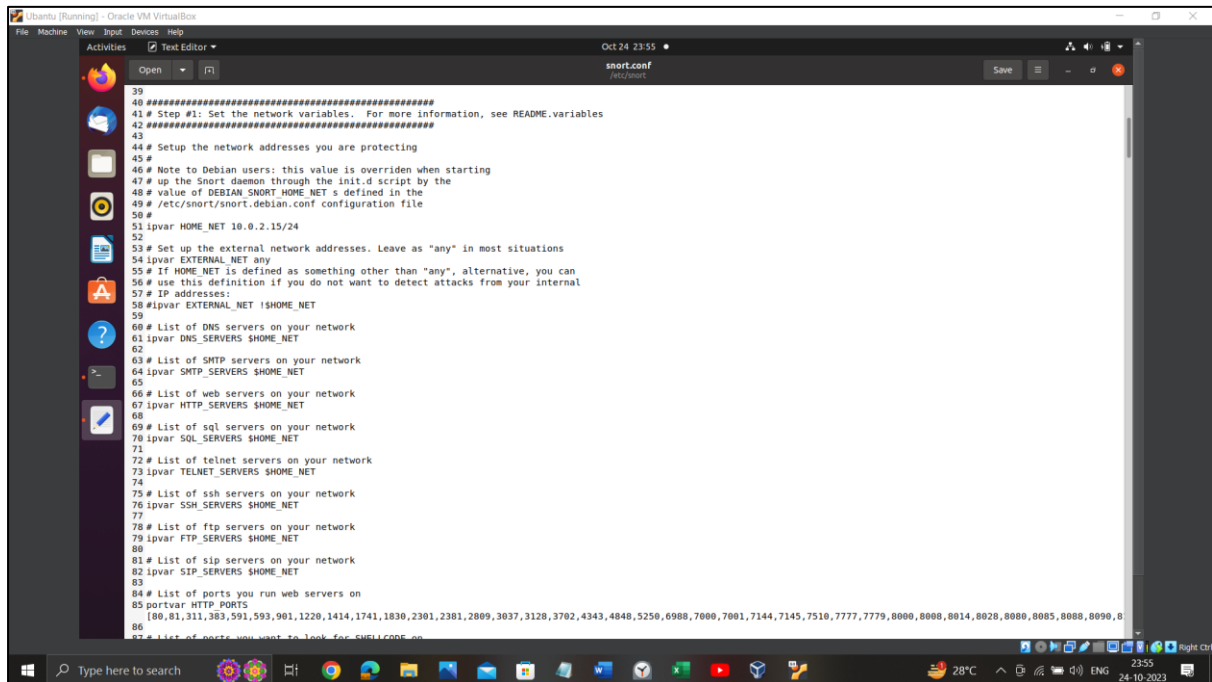
```
kiran@kiran-VirtualBox:~$ sudo gedit /etc/snort/snort.conf
```

Snort.conf file



```
1 #-----  
2 # VRT Rule Packages Snort.conf  
3 #  
4 # For more information visit us at:  
5 # http://www.snort.org Snort Website  
6 # http://vrt-blog.snort.org/ Sourcefire VRT Blog  
7 #  
8 # Mailing list Contact: snort-sigs@lists.sourceforge.net  
9 # False Positive reports: fp@sourcefire.com  
10 # Snort bugs: bugs@snort.org  
11 #  
12 # Compatible with Snort Versions:  
13 # VERSIONS : 2.9.7.0  
14 #  
15 # Snort build options:  
16 # OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-ppm --enable-perfprofiling --enable-zlib --  
enable-active-response --enable-normalizer --enable-reload --enable-react --enable-flexresp3  
17 #  
18 # Additional information:  
19 # This configuration file enables active response, to run snort in  
20 # test mode -T you are required to supply an interface -i <interface>  
21 # or test mode will fail to fully validate the configuration and  
22 # exit with a FATAL error  
23 #-----  
24  
25 #####  
26 # This file contains a sample snort configuration.  
27 # You should take the following steps to create your own custom configuration:  
28 #  
29 # 1) Set the network variables.  
30 # 2) Configure the decoder  
31 # 3) Configure the base detection engine  
32 # 4) Configure dynamic loaded libraries  
33 # 5) Configure the rule files
```


Change the ip address

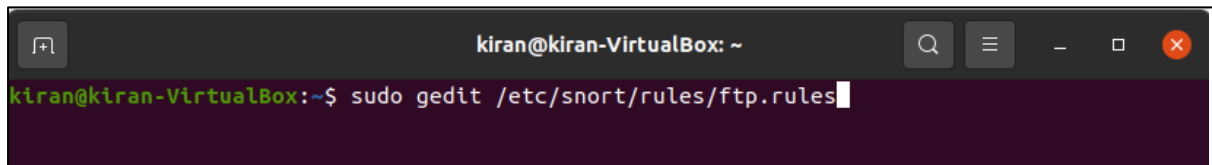


The screenshot shows a text editor window titled 'snort.conf' with a 'Save' button. The file content is a configuration for Snort, including comments and IP address definitions. The IP address 10.0.2.15/24 is defined for the HOME_NET variable.

```
39 #####
40 #####
41 # Step #1: Set the network variables. For more information, see README.variables
42 #####
43
44 # Setup the network addresses you are protecting
45 #
46 # Note to Debian users: this value is overridden when starting
47 # up the Snort daemon through the init.d script by the
48 # value of DEBIAN_SNORT_HOME_NET defined in the
49 # /etc/snort/snort.debian.conf configuration file
50 #
51 ipvar HOME_NET 10.0.2.15/24
52
53 # Set up the external network addresses. Leave as "any" in most situations
54 ipvar EXTERNAL_NET any
55 # If HOME_NET is defined as something other than "any", alternative, you can
56 # use this definition if you do not want to detect attacks from your internal
57 # IP addresses:
58 ipvar EXTERNAL_NET !$HOME_NET
59
60 # List of DNS servers on your network
61 ipvar DNS_SERVERS $HOME_NET
62
63 # List of SMTP servers on your network
64 ipvar SMTP_SERVERS $HOME_NET
65
66 # List of web servers on your network
67 ipvar HTTP_SERVERS $HOME_NET
68
69 # List of sql servers on your network
70 ipvar SQL_SERVERS $HOME_NET
71
72 # List of telnet servers on your network
73 ipvar TELNET_SERVERS $HOME_NET
74
75 # List of ssh servers on your network
76 ipvar SSH_SERVERS $HOME_NET
77
78 # List of ftp servers on your network
79 ipvar FTP_SERVERS $HOME_NET
80
81 # List of sip servers on your network
82 ipvar SIP_SERVERS $HOME_NET
83
84 # List of ports you run web servers on
85 portvar HTTP_PORTS
86 (80,81,311,383,591,593,901,1220,1414,1741,1830,2301,2381,2809,3037,3128,3702,4343,4848,5250,6988,7000,7001,7144,7145,7510,7777,7779,8000,8008,8014,8028,8080,8085,8088,8090,8
87
88 # List of ports you want to look for SQLiZOO on
```

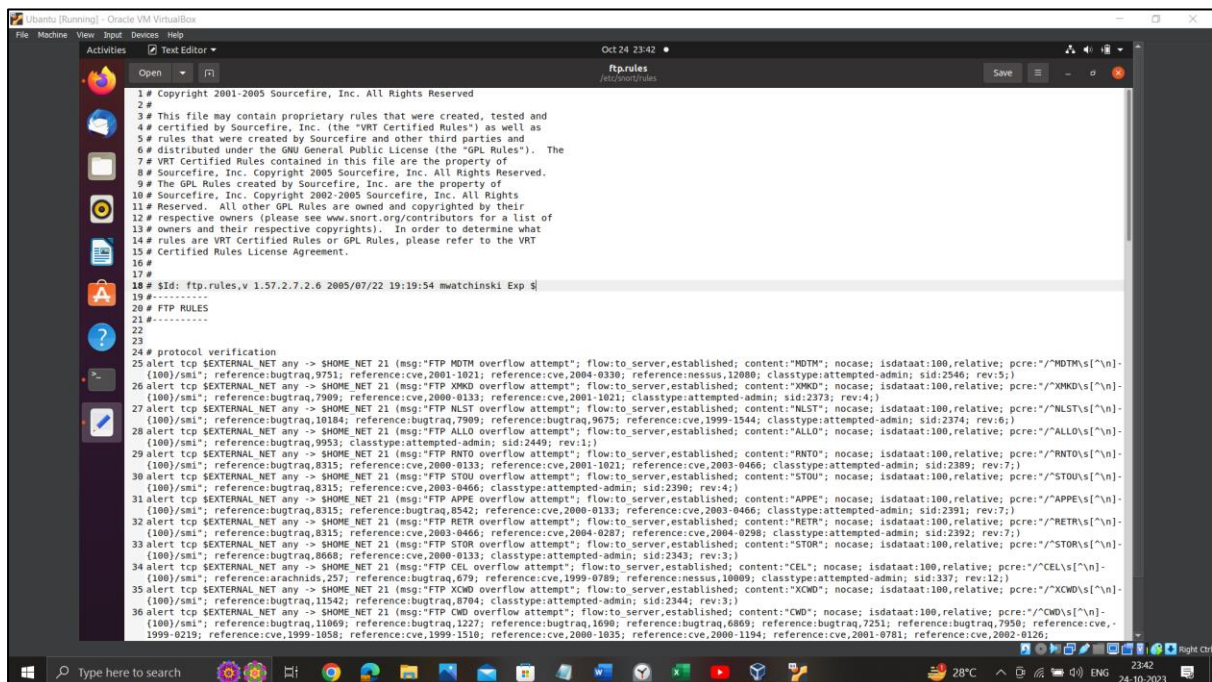
Rules files

Sudo gedit /etc/snort/rules/ftp.rules file



The screenshot shows a terminal window with the prompt 'kiran@kiran-VirtualBox: ~'. The command 'sudo gedit /etc/snort/rules/ftp.rules' has been entered and is highlighted.

```
kiran@kiran-VirtualBox:~$ sudo gedit /etc/snort/rules/ftp.rules
```



The screenshot shows a text editor window titled 'ftp.rules' with a 'Save' button. The file content is a list of Snort rules for FTP, including comments and rule definitions. The rules are numbered 1 through 36.

```
1 # Copyright 2001-2005 Sourcefire, Inc. All Rights Reserved
2 #
3 # This file may contain proprietary rules that were created, tested and
4 # certified by Sourcefire, Inc. (the "VRT Certified Rules") as well as
5 # rules that were created by Sourcefire and other third parties and
6 # distributed under the GNU General Public License (the "GPL Rules"). The
7 # VRT Certified Rules contained in this file are the property of
8 # Sourcefire, Inc. Copyright 2005 Sourcefire, Inc. All Rights Reserved.
9 # The GPL Rules created by Sourcefire, Inc. are the property of
10 # Sourcefire, Inc. Copyright 2002-2005 Sourcefire, Inc. All Rights
11 # Reserved. All other GPL Rules are owned and copyrighted by their
12 # respective owners (please see www.snort.org/contributors for a list of
13 # owners and their respective copyrights). In order to determine what
14 # rules are VRT Certified Rules or GPL Rules, please refer to the VRT
15 # Certified Rules License Agreement.
16 #
17 #
18 # $Id: ftp.rules,v 1.57.2.7.2.6 2005/07/22 19:19:54 mwatchinski Exp $
19 #-----
20 # FTP RULES
21 #-----
22
23
24 # protocol verification
25 alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP MDTM overflow attempt"; flow:to server,established; content:"MDTM"; nocase; isdataat:100,relative; pcre:"/MDTMs{\\n}-
(100)/smi"; reference:bugtraq,9751; reference:cve,2001-1021; reference:cve,2004-0330; reference:nessus,12080; classtype:attempted-admin; sid:2546; rev:5;)
26 alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP XMD overflow attempt"; flow:to server,established; content:"XMD"; nocase; isdataat:100,relative; pcre:"/XMDs{\\n}-
(100)/smi"; reference:bugtraq,7909; reference:cve,2000-0133; reference:cve,2001-1021; classtype:attempted-admin; sid:2373; rev:4;)
27 alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP NLST overflow attempt"; flow:to server,established; content:"NLST"; nocase; isdataat:100,relative; pcre:"/NLSTs{\\n}-
(100)/smi"; reference:bugtraq,10184; reference:bugtraq,7909; reference:bugtraq,9675; reference:cve,1999-1544; classtype:attempted-admin; sid:2374; rev:6;)
28 alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP ALLO overflow attempt"; flow:to server,established; content:"ALLO"; nocase; isdataat:100,relative; pcre:"/ALLOs{\\n}-
(100)/smi"; reference:bugtraq,9953; classtype:attempted-admin; sid:2449; rev:1;)
29 alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP RNTD overflow attempt"; flow:to server,established; content:"RNTD"; nocase; isdataat:100,relative; pcre:"/RNTDs{\\n}-
(100)/smi"; reference:bugtraq,8315; reference:cve,2000-0133; reference:cve,2001-1021; reference:cve,2003-0466; classtype:attempted-admin; sid:2389; rev:7;)
30 alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP STOU overflow attempt"; flow:to server,established; content:"STOU"; nocase; isdataat:100,relative; pcre:"/STOUs{\\n}-
(100)/smi"; reference:bugtraq,8315; reference:cve,2003-0466; classtype:attempted-admin; sid:2390; rev:4;)
31 alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP APPE overflow attempt"; flow:to server,established; content:"APPE"; nocase; isdataat:100,relative; pcre:"/APPEs{\\n}-
(100)/smi"; reference:bugtraq,8315; reference:cve,2003-0466; reference:cve,2004-0287; reference:cve,2004-0298; classtype:attempted-admin; sid:2392; rev:7;)
32 alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP RETR overflow attempt"; flow:to server,established; content:"RETR"; nocase; isdataat:100,relative; pcre:"/RETRs{\\n}-
(100)/smi"; reference:bugtraq,8315; reference:cve,2003-0466; reference:cve,2004-0287; reference:cve,2004-0298; classtype:attempted-admin; sid:2392; rev:7;)
33 alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP STOR overflow attempt"; flow:to server,established; content:"STOR"; nocase; isdataat:100,relative; pcre:"/STORs{\\n}-
(100)/smi"; reference:bugtraq,8668; reference:cve,2000-0133; classtype:attempted-admin; sid:2343; rev:3;)
34 alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP CEL overflow attempt"; flow:to server,established; content:"CEL"; nocase; isdataat:100,relative; pcre:"/CELS{\\n}-
(100)/smi"; reference:arachnids,257; reference:bugtraq,679; reference:cve,1999-0789; reference:nessus,10009; classtype:attempted-admin; sid:337; rev:12;)
35 alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP XCMD overflow attempt"; flow:to server,established; content:"XCMD"; nocase; isdataat:100,relative; pcre:"/XCMDs{\\n}-
(100)/smi"; reference:bugtraq,11542; reference:bugtraq,8704; classtype:attempted-admin; sid:2344; rev:3;)
36 alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP CMD overflow attempt"; flow:to server,established; content:"CMD"; nocase; isdataat:100,relative; pcre:"/CMDs{\\n}-
(100)/smi"; reference:bugtraq,11069; reference:bugtraq,1227; reference:bugtraq,6869; reference:bugtraq,7251; reference:bugtraq,7950; reference:cve,
1999-0219; reference:cve,1999-1058; reference:cve,1999-1510; reference:cve,2000-1035; reference:cve,2001-0781; reference:cve,2002-0126;
```

Check the snort is configured successfully and check for validation

Sudo snort -T -c /etc/snort/snort.conf -I enp0s3

```
kiran@kiran-VirtualBox: ~  
kiran@kiran-VirtualBox:~$ sudo snort -T -c /etc/snort/snort.conf -i enp0s3  
Running in Test mode  
  
--== Initializing Snort ==--  
Initializing Output Plugins!  
Initializing Preprocessors!  
Initializing Plug-ins!  
Parsing Rules file "/etc/snort/snort.conf"  
PortVar 'HTTP_PORTS' defined : [ 80:81 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 370  
2 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8  
180:8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 5555  
5 ]  
PortVar 'SHELLCODE_PORTS' defined : [ 0:79 81:65535 ]  
PortVar 'ORACLE_PORTS' defined : [ 1024:65535 ]  
PortVar 'SSH_PORTS' defined : [ 22 ]  
PortVar 'FTP_PORTS' defined : [ 21 2100 3535 ]  
PortVar 'SIP_PORTS' defined : [ 5060:5061 5600 ]  
PortVar 'FILE_DATA_PORTS' defined : [ 80:81 110 143 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809  
3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 809  
0 8118 8123 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 410  
80 50002 55555 ]  
PortVar 'GTP_PORTS' defined : [ 2123 2152 3386 ]  
Detection:  
  Search-Method = AC-Full-Q  
  Split Any/Any group = enabled  
  Search-Method-Optimizations = enabled
```

```
kiran@kiran-VirtualBox: ~  
'''' By Martin Roesch & The Snort Team: http://www.snort.org/contact#team  
Copyright (C) 2014 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: 8.39 2016-06-14  
Using ZLIB version: 1.2.11  
  
Rules Engine: SF_SNORT_DETECTION_ENGINE Version 2.4 <Build 1>  
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>  
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>  
Preprocessor Object: SF_GTP Version 1.1 <Build 1>  
Preprocessor Object: SF_POP Version 1.0 <Build 1>  
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>  
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>  
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>  
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>  
Preprocessor Object: SF_SIP Version 1.1 <Build 1>  
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>  
Preprocessor Object: SF_SSH Version 1.1 <Build 3>  
Preprocessor Object: SF_DNS Version 1.1 <Build 4>  
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>  
Preprocessor Object: SF_SDF Version 1.1 <Build 1>  
  
Snort successfully validated the configuration!  
Snort exiting  
kiran@kiran-VirtualBox:~$
```

Enable the snort and check for the traffic

```
kiran@kiran-VirtualBox: ~  
kiran@kiran-VirtualBox:~$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i enp0s3
```

Open the kali linux and then perform attacks to this machine

```
(kiran@Kali)-[~]  
$ nmap 192.168.1.18  
Starting Nmap 7.92 ( https://nmap.org ) at 2023-10-31 12:46 EDT
```

Network traffic and attacks reported

```
vjti@vjti-ThinkCentre-M70a-Gen-3: ~  
vjti@vjti-ThinkCentre-M70a-Gen-3:~$ sudo snort -A console -q -u snort -c /etc/snort/snort.conf -i enp2s0  
10/25-11:28:52.461388  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic]  
[Priority: 2] {IGMP} 0.0.0.0 -> 224.0.0.1  
10/25-11:30:57.903653  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic]  
[Priority: 2] {IGMP} 0.0.0.0 -> 224.0.0.1  
10/25-11:31:26.243065  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic]  
[Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67  
10/25-11:31:26.247331  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic]  
[Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67  
10/25-11:32:51.170695  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic]  
[Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67  
10/25-11:32:51.172468  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic]  
[Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67  
10/25-11:33:03.345206  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic]  
[Priority: 2] {IGMP} 0.0.0.0 -> 224.0.0.1  
10/25-11:33:55.120250  [**] [1:1415:9] SNMP Broadcast request [**] [Classification: Attempted Information Leak]  
[Priority: 2] {UDP} 172.18.37.245:41199 -> 255.255.255.255:161  
10/25-11:33:56.120667  [**] [1:1415:9] SNMP Broadcast request [**] [Classification: Attempted Information Leak]  
[Priority: 2] {UDP} 172.18.37.245:41199 -> 255.255.255.255:161  
10/25-11:34:03.591230  [**] [1:1415:9] SNMP Broadcast request [**] [Classification: Attempted Information Leak]  
[Priority: 2] {UDP} 172.18.37.245:60788 -> 255.255.255.255:161  
10/25-11:34:04.591811  [**] [1:1415:9] SNMP Broadcast request [**] [Classification: Attempted Information Leak]  
[Priority: 2] {UDP} 172.18.37.245:60788 -> 255.255.255.255:161
```

Conclusion:

In conclusion, this experiment involved the setup of SNORT as an Intrusion Detection System (IDS) in a controlled network environment. The experiment included configuring SNORT, creating custom rules, and simulating network attacks using a Kali Linux machine. SNORT effectively monitored the network traffic, detected malicious activity, and generated alerts and logs. Documentation and screenshots were used to capture the experiment's progress and findings. This experiment provided valuable insights into the capabilities of SNORT as an IDS for detecting network threats and enhancing network security.