ASSIGNMENT NO. 4

TITLE: Configure and demonstrate Snort tool for intrusion.

AIM: Configure and demonstrate use of vulnerability assessment tools such as Snort tool for intrusion or SSL Web security.

OBJECTIVE: Study any vulnerability assessment tool such as Snort tool and use its implementation features.

THEORY:

Introduction

Snort is a popular choice for running a network intrusion detection system or NIDS for short. It monitors the package data sent and received through a specific network interface.

NIDS can catch threats targeting your system vulnerabilities using signature-based detection and protocol analysis technologies. NIDS software, when installed and configured appropriately, can identify the latest attacks, malware infections, compromised systems, and network policy violations.

Platforms on which Snort runs

Snort runs on most UNIX and various windows.

- UNIX
- Applet, MAC, BEOS, JBM, AIX, BSD open etc.
- LINUX
- Mandrake LINUX, Red Hat, SUSE LINUX etc.
- WINDOWS
- Windows server 2003/XP/2000/NT

What can I do with Snort?

Snort has three primary uses:

- It can be used as a straight packet sniffer like tcpdump.
- A packet logger (useful for network traffic debugging, etc).
- As a full-blown network intrusion prevention system.

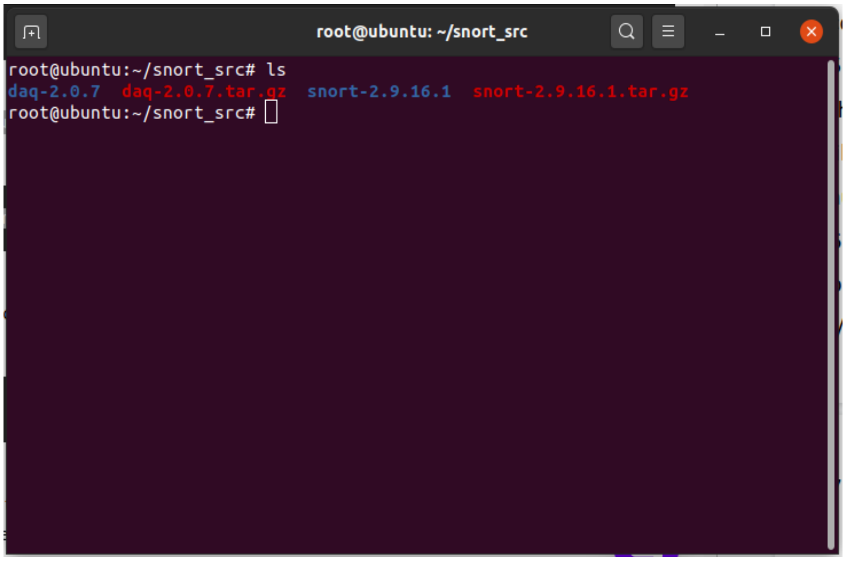
Installation

1. Install dependencies

```
sudo apt install -y gcc libpcre3-dev zlib1g-dev libluajit-5.1-dev \
libpcap-dev openssl libssl-dev libnghttp2-dev libdumbnet-dev \
bison flex libdnet autoconf libtool
```

2. create a temporary download folder in home directory

```
mkdir ~/snort_src && cd ~/snort_src
```



```
root@ubuntu:~/snort_src# ls
daq-2.0.7  daq-2.0.7.tar.gz  snort-2.9.16.1  snort-2.9.16.1.tar.gz
root@ubuntu:~/snort_src# 
```

3. Install Data Acquisition Library (DAQ) used to make the abstract calls to packet capture libraries. Download the latest DAQ using wget.

```
wget https://www.snort.org/downloads/snort/daq-2.0.7.tar.gz
```

4. Extract the code and go to the new directory

```
tar -xvzf daq-2.0.7.tar.gz
cd daq-2.0.7
```

5. The latest version requires an additional step to auto reconfigure DAQ before running the config. Use the command below which requires you need to have autoconf and libtool installed.

```
autoreconf -f -i
```

6. Afterwards, run the configuration script using its default values, then compile the program with make and finally install DAQ.

```
./configure && make && sudo make install
```

7. Now that DAQ is installed, change back to download folder
8. Next, download the Snort source code with wget.

```
Wget https://www.snort.org/downloads/snort/snort-2.9.16.1.tar.gz
```

9. Once the download is complete, extract the source and change into the new directory with these commands.

```
tar -xvzf snort-2.9.16.tar.gz
cd snort-2.9.16
```

10. Then configure the installation with sourcefire enabled, run make and make install

```
./configure --enable-sourcefire && make && sudo make install
```

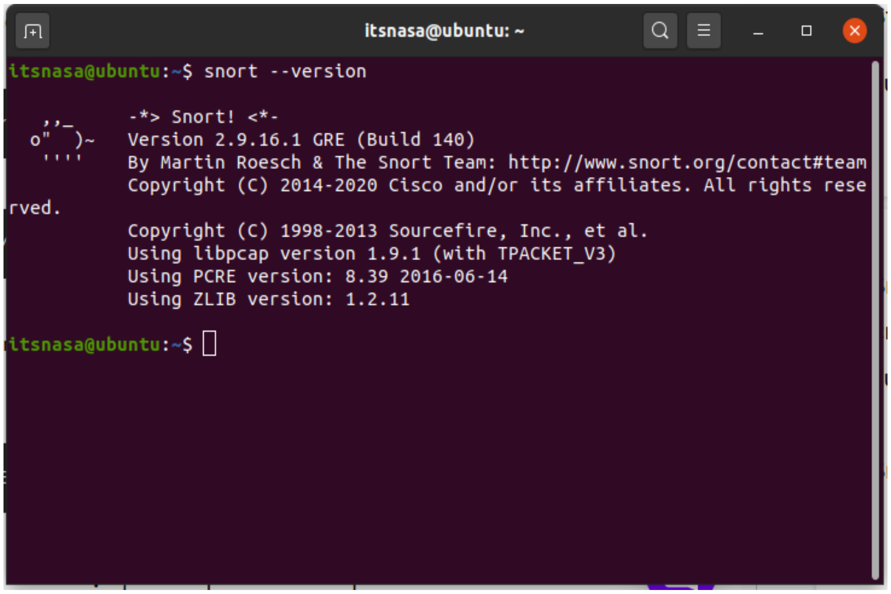
11. Start with updating the shared libraries using the command underneath.

```
sudo ldconfig
```

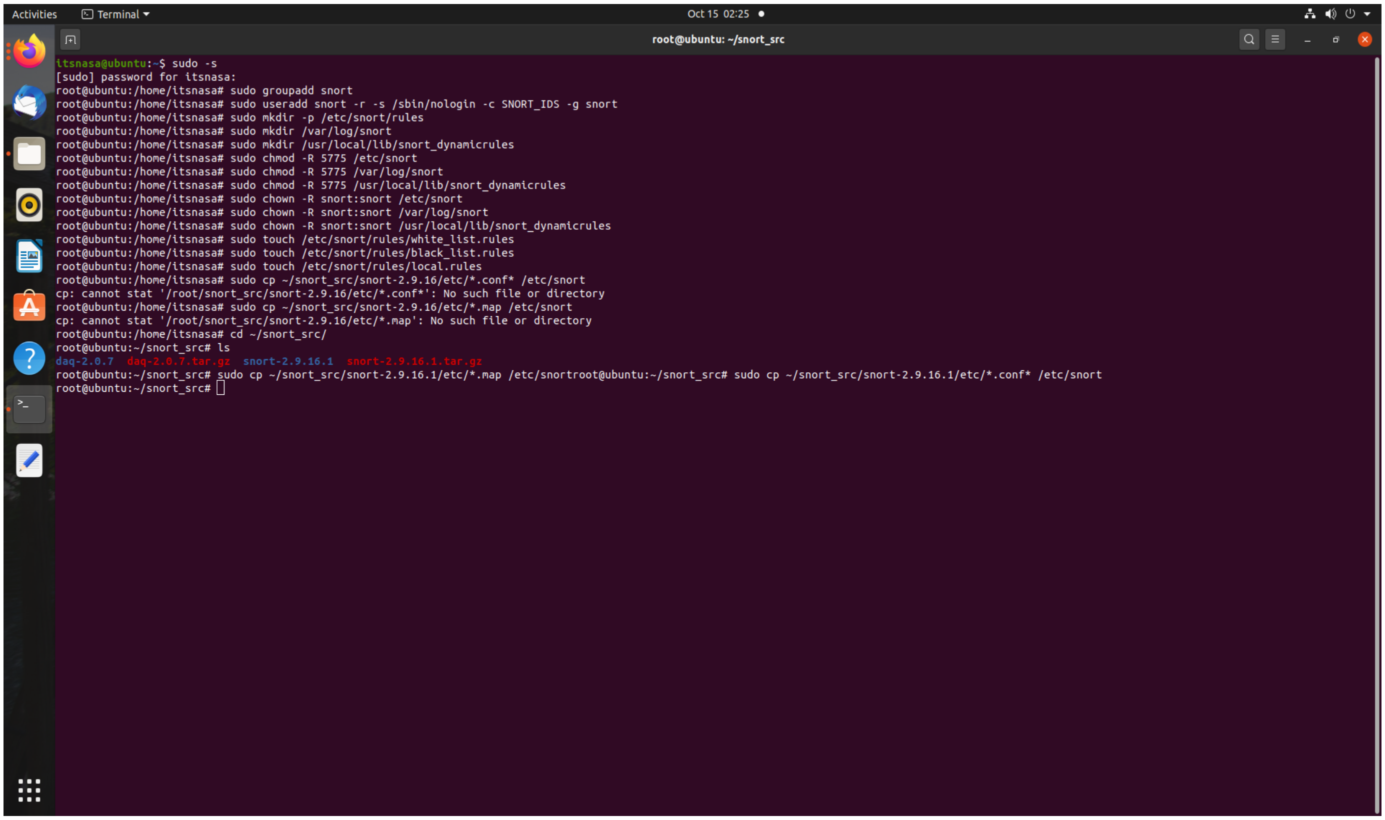
12. Snort on Ubuntu gets installed to /usr/local/bin/snort directory, it is good practice to create a symbolic link to /usr/sbin/snort.

```
sudo ln -s /usr/local/bin/snort /usr/sbin/snort
```

Snort is now up and running.



```
itsnasa@ubuntu:~$ snort --version

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.16.1 GRE (Build 140)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2020 Cisco and/or its affiliates. All rights rese
rved.
           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

itsnasa@ubuntu:~$ 
```



```
itsnasa@ubuntu:~$ sudo -s
[sudo] password for itsnasa:
root@ubuntu:/home/itsnasa# sudo groupadd snort
root@ubuntu:/home/itsnasa# sudo useradd snort -r -s /sbin/nologin -c SNORT_IDS -g snort
root@ubuntu:/home/itsnasa# sudo mkdir -p /etc/snort/rules
root@ubuntu:/home/itsnasa# sudo mkdir /var/log/snort
root@ubuntu:/home/itsnasa# sudo mkdir /usr/local/lib/snort_dynamicrules
root@ubuntu:/home/itsnasa# sudo chmod -R 5775 /etc/snort
root@ubuntu:/home/itsnasa# sudo chmod -R 5775 /var/log/snort
root@ubuntu:/home/itsnasa# sudo chmod -R 5775 /usr/local/lib/snort_dynamicrules
root@ubuntu:/home/itsnasa# sudo chown -R snort:snort /etc/snort
root@ubuntu:/home/itsnasa# sudo chown -R snort:snort /var/log/snort
root@ubuntu:/home/itsnasa# sudo chown -R snort:snort /usr/local/lib/snort_dynamicrules
root@ubuntu:/home/itsnasa# sudo touch /etc/snort/rules/white_list.rules
root@ubuntu:/home/itsnasa# sudo touch /etc/snort/rules/black_list.rules
root@ubuntu:/home/itsnasa# sudo touch /etc/snort/rules/local.rules
root@ubuntu:/home/itsnasa# sudo cp ~/snort_src/snort-2.9.16/etc/*.conf* /etc/snort
cp: cannot stat '/root/snort_src/snort-2.9.16/etc/*.conf*': No such file or directory
root@ubuntu:/home/itsnasa# sudo cp ~/snort_src/snort-2.9.16/etc/*.map /etc/snort
cp: cannot stat '/root/snort_src/snort-2.9.16/etc/*.map': No such file or directory
root@ubuntu:/home/itsnasa# cd ~/snort_src/
root@ubuntu:~/snort_src# ls
daq-2.0.7  daq-2.0.7.tar.gz  snort-2.9.16.1  snort-2.9.16.1.tar.gz
root@ubuntu:~/snort_src# sudo cp ~/snort_src/snort-2.9.16.1/etc/*.map /etc/snortroot@ubuntu:~/snort_src# sudo cp ~/snort_src/snort-2.9.16.1/etc/*.conf* /etc/snort
root@ubuntu:~/snort_src# 
```

Setting rules for Snort

1. Grab the community rules using wget with the command below.

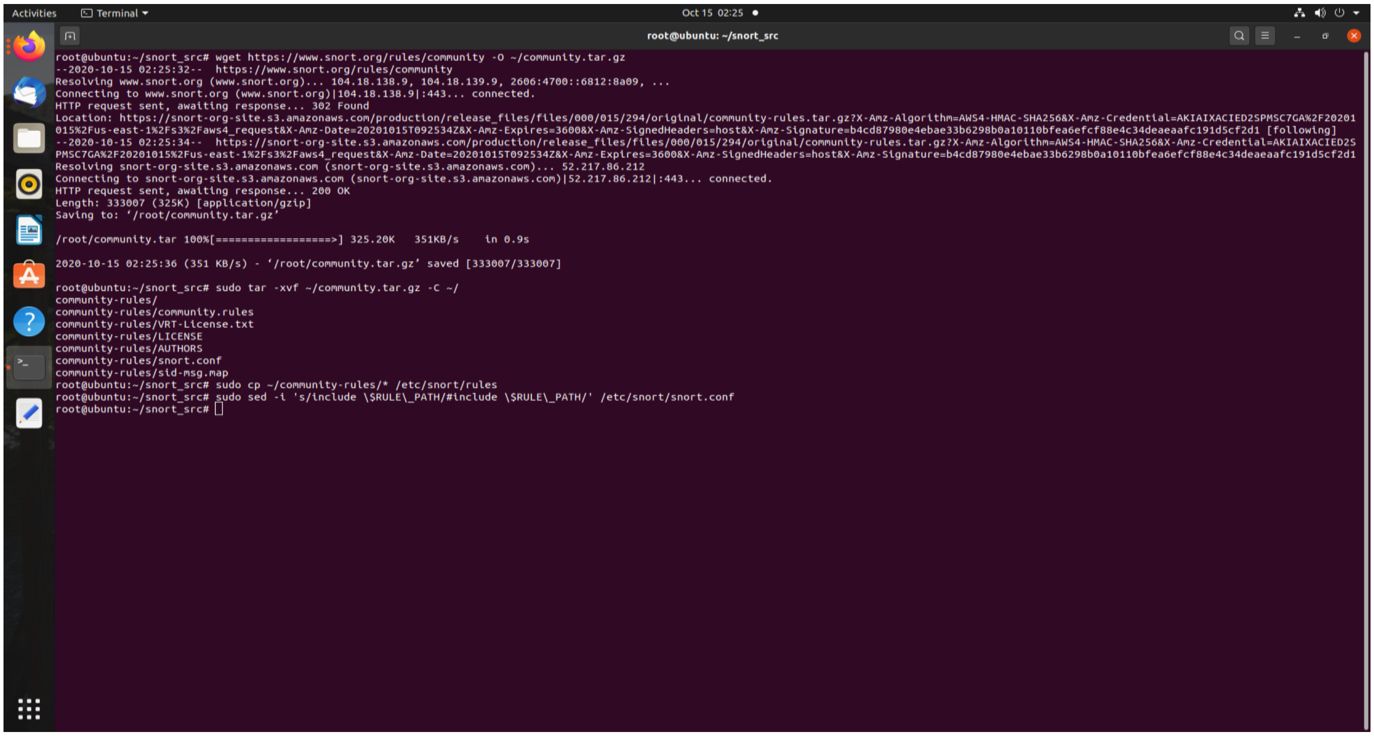
```
wget https://www.snort.org/rules/community -O ~/community.tar.gz
```

2. Extract the rules and copy them to your configuration folder.

```
sudo tar -xvf ~/community.tar.gz -C ~/
sudo cp ~/community-rules/* /etc/snort/rules
```

3. By default, Snort on Ubuntu expects to find a number of different rule files which are not included in the community rules. You can easily comment out the unnecessary lines using the sed command underneath.

```
sudo sed -
i 's/include \$RULE\_PATH/#include \$RULE\_PATH/' /etc/snort/snort.conf
```

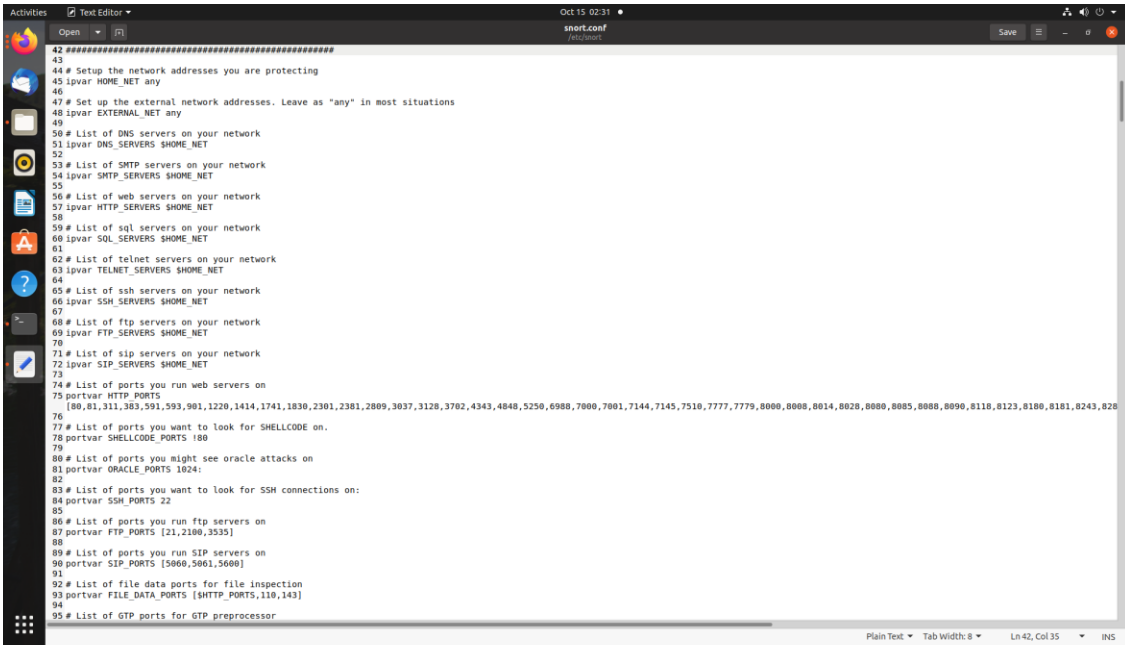


```
root@ubuntu:~/snort_src# wget https://www.snort.org/rules/community -O ~/community.tar.gz
--2020-10-15 02:25:32--  https://www.snort.org/rules/community
Resolving www.snort.org (www.snort.org)... 104.18.138.9, 104.18.139.9, 2606:4700::6812:8a09, ...
Connecting to www.snort.org (www.snort.org)|104.18.138.9|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://snort-org-site.s3.amazonaws.com/production/release_files/files/000/015/294/original/community-rules.tar.gz?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIXACIED2SPMSC7GA%2F20201
015%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20201015T092534Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=b4cd87980e4ebae33b6298b0a10110bfea6efcf88e4c34deaeaafc191d5cf2d1 [following]
--2020-10-15 02:25:34--  https://snort-org-site.s3.amazonaws.com/production/release_files/files/000/015/294/original/community-rules.tar.gz?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIXACIED2S
PMSC7GA%2F20201015%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20201015T092534Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=b4cd87980e4ebae33b6298b0a10110bfea6efcf88e4c34deaeaafc191d5cf2d1
Resolving snort-org-site.s3.amazonaws.com (snort-org-site.s3.amazonaws.com)... 52.217.86.212
Connecting to snort-org-site.s3.amazonaws.com (snort-org-site.s3.amazonaws.com)|52.217.86.212|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 333007 (325K) [application/gzip]
Saving to: ‘/root/community.tar.gz’

/root/community.tar 100%[=================>] 325.20K   351KB/s    in 0.9s

2020-10-15 02:25:36 (351 KB/s) - ‘/root/community.tar.gz’ saved [333007/333007]

root@ubuntu:~/snort_src# sudo tar -xvf ~/community.tar.gz -C ~/
community-rules/
community-rules/community.rules
community-rules/VRT-License.txt
community-rules/LICENSE
community-rules/AUTHORS
community-rules/snort.conf
community-rules/sid-msg.map
root@ubuntu:~/snort_src# sudo cp ~/community-rules/* /etc/snort/rules
root@ubuntu:~/snort_src# sudo sed -i 's/include \$RULE\_PATH/#include \$RULE\_PATH/' /etc/snort/snort.conf
root@ubuntu:~/snort_src# 
```

With the configuration and rule files in place, edit the snort.conf to modify a few parameters.

Open the configuration file in your favourite text editor, for example using Gedit with the command below

sudo gedit /etc/snort/snort.conf

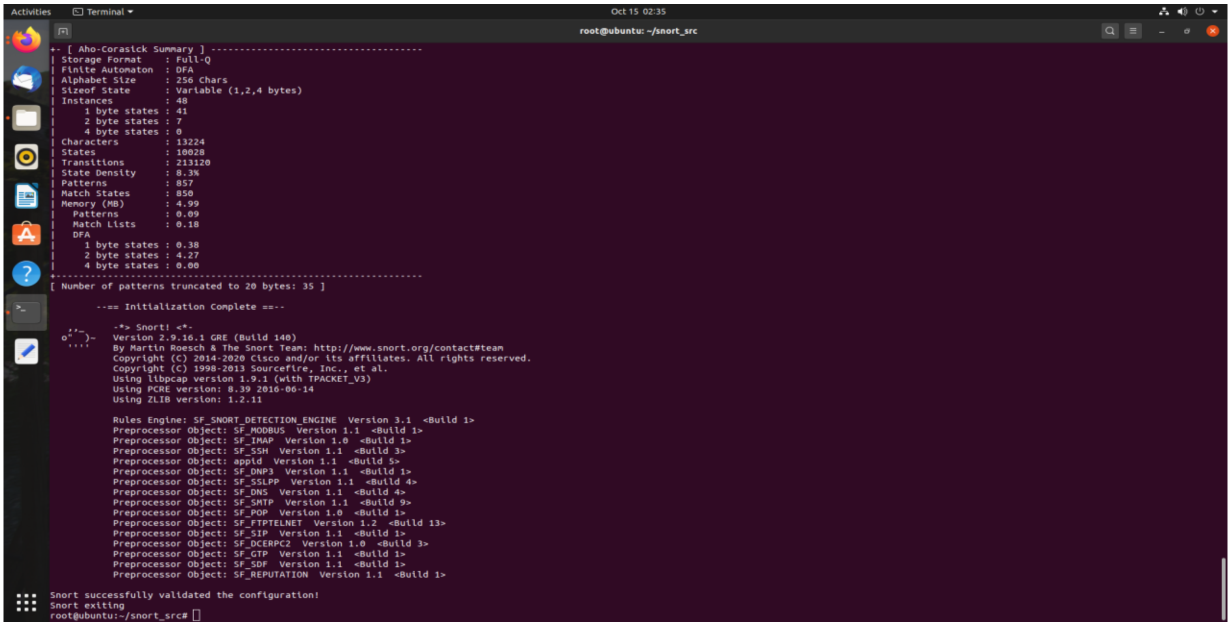


```
###################################################

# Setup the network addresses you are protecting
ipvar HOME_NET any

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

# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
ipvar SMTP_SERVERS $HOME_NET

# List of web servers on your network
ipvar HTTP_SERVERS $HOME_NET

# List of sql servers on your network
ipvar SQL_SERVERS $HOME_NET

# List of telnet servers on your network
ipvar TELNET_SERVERS $HOME_NET

# List of ssh servers on your network
ipvar SSH_SERVERS $HOME_NET

# List of ftp servers on your network
ipvar FTP_SERVERS $HOME_NET

# List of sip servers on your network
ipvar SIP_SERVERS $HOME_NET

# List of ports you run web servers on
portvar HTTP_PORTS
[80,81,311,383,591,593,901,1220,1414,1741,1830,2301,2381,2809,3037,3128,3702,4343,4848,5250,6988,7000,7001,7144,7145,7510,7777,7779,8000,8008,8014,8028,8080,8085,8088,8090,8118,8123,8180,8181,8243,828

# List of ports you want to look for SHELLCODE on.
portvar SHELLCODE_PORTS !80

# List of ports you might see oracle attacks on
portvar ORACLE_PORTS 1024:

# List of ports you want to look for SSH connections on:
portvar SSH_PORTS 22

# List of ports you run ftp servers on
portvar FTP_PORTS [21,2100,3535]

# List of ports you run SIP servers on
portvar SIP_PORTS [5060,5061,5600]

# List of file data ports for file inspection
portvar FILE_DATA_PORTS [$HTTP_PORTS,110,143]

# List of GTP ports for GTP preprocessor
```

Edit your path files.

Your Snort should now be ready to run. Test the configuration using the parameter -T to enable test mode.

```
sudo snort -T -c /etc/snort/snort.conf
```



```
+- [ Aho-Corasick Summary ] -------------------------------------
| Storage Format    : Full-Q
| Finite Automaton  : DFA
| Alphabet Size     : 256 Chars
| Sizeof State      : Variable (1,2,4 bytes)
| Instances         : 48
|     1 byte states : 41
|     2 byte states : 7
|     4 byte states : 0
| Characters        : 13224
| States            : 10028
| Transitions       : 213120
| State Density     : 8.3%
| Patterns          : 857
| Match States      : 850
| Memory (MB)       : 4.99
|   Patterns        : 0.09
|   Match Lists     : 0.18
|   DFA
|     1 byte states : 0.38
|     2 byte states : 4.27
|     4 byte states : 0.00
+----------------------------------------------------------------
[ Number of patterns truncated to 20 bytes: 35 ]

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.16.1 GRE (Build 140)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2020 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 3.1  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: appid  Version 1.1  <Build 5>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_POP  Version 1.0  <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_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>

Snort successfully validated the configuration!
Snort exiting
root@ubuntu:~/snort_src# 
```

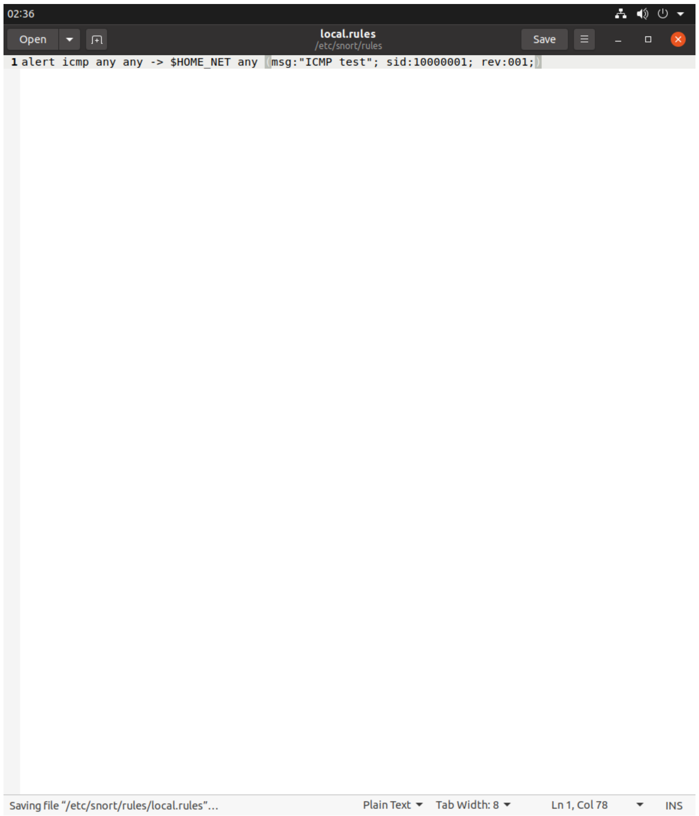
Testing the configuration

To test if Snort is logging alerts as intended, add a custom detection rule alert on incoming ICMP connections to the local.rules file. Open your local rules in a text editor.

```
sudo nano /etc/snort/rules/local.rules
```

Then add the following line to the file.

```
alert icmp any any -> $HOME_NET any (msg:"ICMP test"; sid:10000001; rev:001;)
```



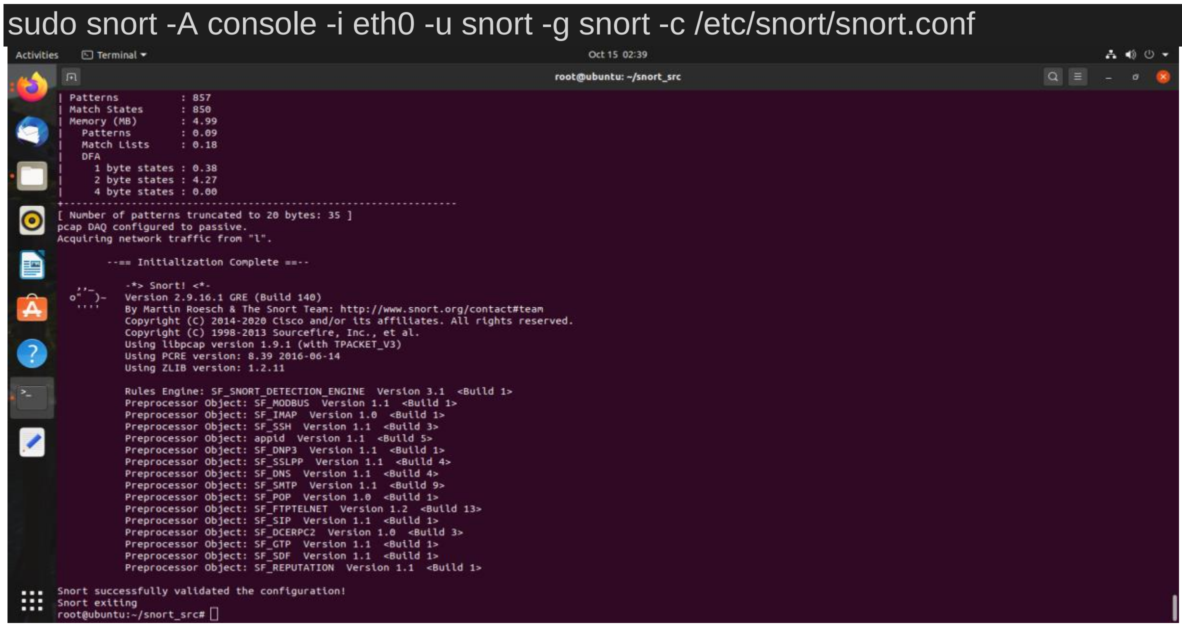
```
alert icmp any any -> $HOME_NET any (msg:"ICMP test"; sid:10000001; rev:001;)
```

Saving file "/etc/snort/rules/local.rules"…

Save the local.rules and exit the editor.

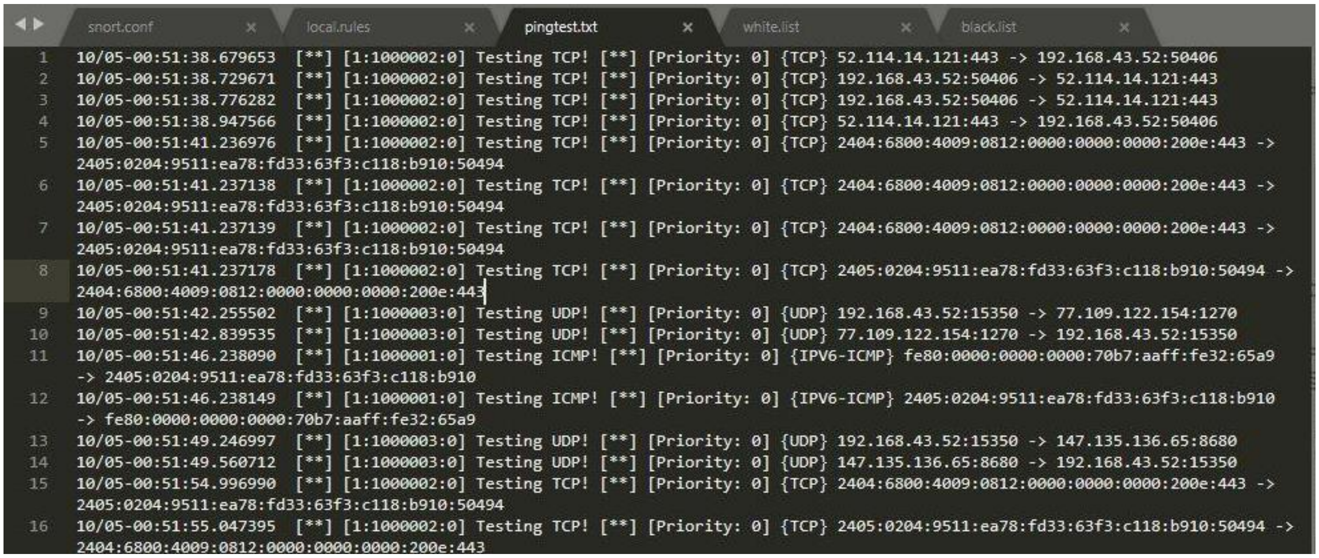
Start Snort with -A console options to print the alerts to stdout. You will need to select the correct network interface with the public IP address of your server, for example, eth0.

```
sudo snort -A console -i eth0 -u snort -g snort -c /etc/snort/snort.conf
```



```
| Patterns         : 857
| Match States     : 850
| Memory (MB)      : 4.99
|   Patterns       : 0.09
|   Match Lists    : 0.18
|   DFA
|     1 byte states : 0.38
|     2 byte states : 4.27
|     4 byte states : 0.00
+----------------------------------------------------------------
[ Number of patterns truncated to 20 bytes: 35 ]
pcap DAQ configured to passive.
Acquiring network traffic from "l".

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.16.1 GRE (Build 140)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2020 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 3.1  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: appid  Version 1.1  <Build 5>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_POP  Version 1.0  <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_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>

Snort successfully validated the configuration!
Snort exiting
root@ubuntu:~/snort_src# 
```

Check the file after some time.



```
10/05-00:51:38.679653  [**] [1:1000002:0] Testing TCP! [**] [Priority: 0] {TCP} 52.114.14.121:443 -> 192.168.43.52:50406
10/05-00:51:38.729671  [**] [1:1000002:0] Testing TCP! [**] [Priority: 0] {TCP} 192.168.43.52:50406 -> 52.114.14.121:443
10/05-00:51:38.776282  [**] [1:1000002:0] Testing TCP! [**] [Priority: 0] {TCP} 192.168.43.52:50406 -> 52.114.14.121:443
10/05-00:51:38.947566  [**] [1:1000002:0] Testing TCP! [**] [Priority: 0] {TCP} 52.114.14.121:443 -> 192.168.43.52:50406
10/05-00:51:41.236976  [**] [1:1000002:0] Testing TCP! [**] [Priority: 0] {TCP} 2404:6800:4009:0812:0000:0000:0000:200e:443 -> 2405:0204:9511:ea78:fd33:63f3:c118:b910:50494
10/05-00:51:41.237138  [**] [1:1000002:0] Testing TCP! [**] [Priority: 0] {TCP} 2404:6800:4009:0812:0000:0000:0000:200e:443 -> 2405:0204:9511:ea78:fd33:63f3:c118:b910:50494
10/05-00:51:41.237139  [**] [1:1000002:0] Testing TCP! [**] [Priority: 0] {TCP} 2404:6800:4009:0812:0000:0000:0000:200e:443 -> 2405:0204:9511:ea78:fd33:63f3:c118:b910:50494
10/05-00:51:41.237178  [**] [1:1000002:0] Testing TCP! [**] [Priority: 0] {TCP} 2405:0204:9511:ea78:fd33:63f3:c118:b910:50494 -> 2404:6800:4009:0812:0000:0000:0000:200e:443
10/05-00:51:42.255502  [**] [1:1000003:0] Testing UDP! [**] [Priority: 0] {UDP} 192.168.43.52:15350 -> 77.109.122.154:1270
10/05-00:51:42.839535  [**] [1:1000003:0] Testing UDP! [**] [Priority: 0] {UDP} 77.109.122.154:1270 -> 192.168.43.52:15350
10/05-00:51:46.238090  [**] [1:1000001:0] Testing ICMP! [**] [Priority: 0] {IPV6-ICMP} fe80:0000:0000:0000:70b7:aaff:fe32:65a9 -> 2405:0204:9511:ea78:fd33:63f3:c118:b910
10/05-00:51:46.238149  [**] [1:1000001:0] Testing ICMP! [**] [Priority: 0] {IPV6-ICMP} 2405:0204:9511:ea78:fd33:63f3:c118:b910 -> fe80:0000:0000:0000:70b7:aaff:fe32:65a9
10/05-00:51:49.246997  [**] [1:1000003:0] Testing UDP! [**] [Priority: 0] {UDP} 192.168.43.52:15350 -> 147.135.136.65:8680
10/05-00:51:49.560712  [**] [1:1000003:0] Testing UDP! [**] [Priority: 0] {UDP} 147.135.136.65:8680 -> 192.168.43.52:15350
10/05-00:51:54.996990  [**] [1:1000002:0] Testing TCP! [**] [Priority: 0] {TCP} 2404:6800:4009:0812:0000:0000:0000:200e:443 -> 2405:0204:9511:ea78:fd33:63f3:c118:b910:50494
10/05-00:51:55.047395  [**] [1:1000002:0] Testing TCP! [**] [Priority: 0] {TCP} 2405:0204:9511:ea78:fd33:63f3:c118:b910:50494 -> 2404:6800:4009:0812:0000:0000:0000:200e:443
```

It is successfully working.

CONCLUSION:

Thus, installation and implementation of snort is completed in this assignment.