Project 2: Network intrusion detection and prevention- Snort and Syslog

Submitted by: Shantanu Bhusari (1211213728)

1. Summary

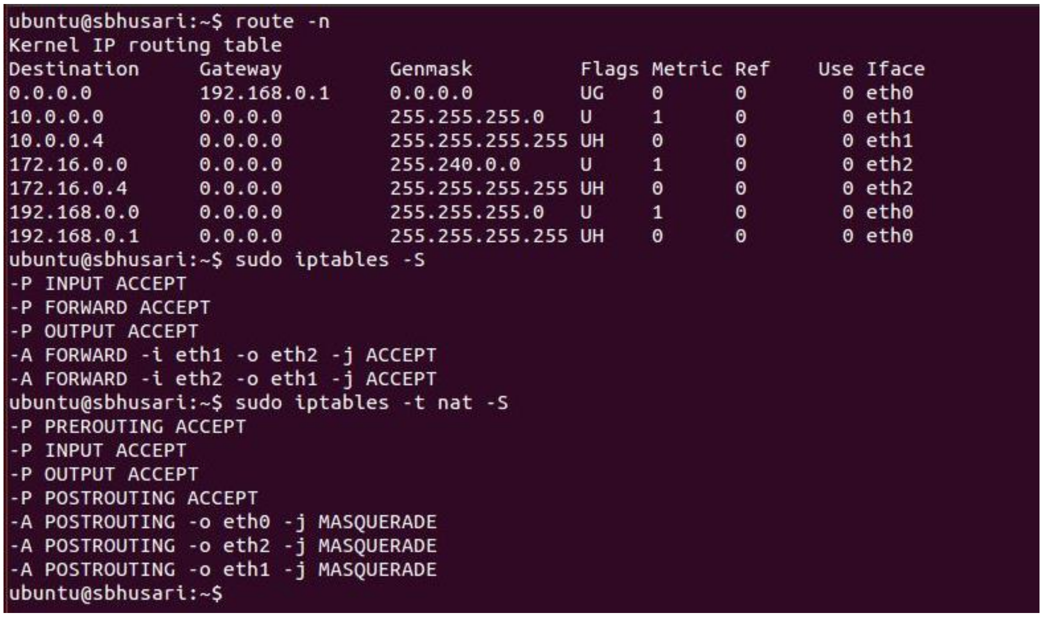
Snort is an open source network Intrusion detection and prevention system that has ability to perform real-life network traffic analysis, packet logging and packet handling capability. In this project, I will demonstrate the usage of snort to perform packet logging, alert generation, and packet handling to secure a server from network intrusions. Given network configuration has three Linux system (Client, server and gateway). The gateway host is responsible for analyzing and handling traffic generated for server host, which is running an apache server. The snort will be running on the gateway host. Using required rules with snort, intrusion detection and prevention for server will be performed. Further, using rsyslog, those alerts will be stored on the server.

2. List of software packages used

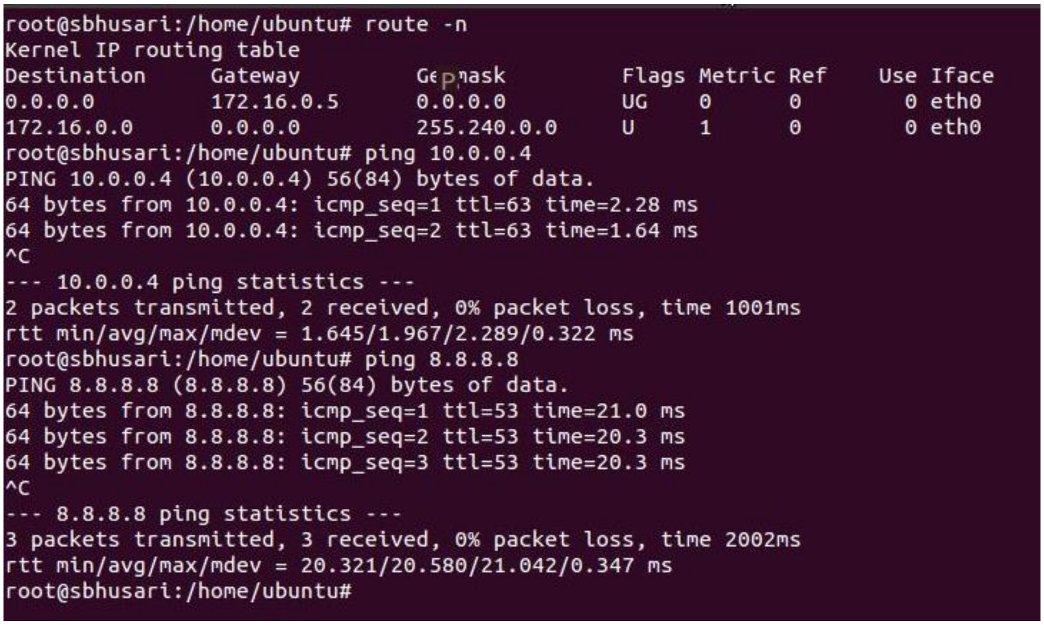
- Linux Ubuntu 14.04
- Apache2
- Snort 2.9.11
- rsyslog

3. Description

In the given setup, the gateway host acts as a gateway which allows communication between server and client, also between internal network and internet. Initially, the routes on server and client are incorrect, so correct gateway addresses were provided in respective routing tables so that server and client can communicate with each other. Further, no NAT setup is done on the gateway host blocking client and server to communicate with the external network.



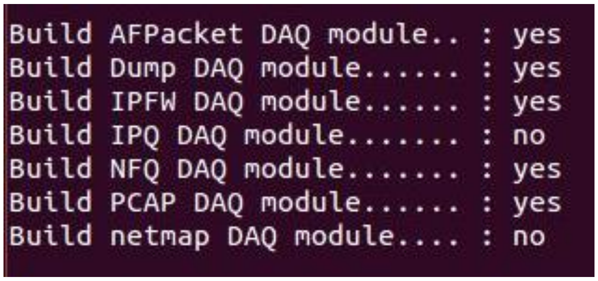
```
ubuntu@sbhusari:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.0.1     0.0.0.0         UG    0      0        0 eth0
10.0.0.0        0.0.0.0         255.255.255.0   U     1      0        0 eth1
10.0.0.4        0.0.0.0         255.255.255.255 UH    0      0        0 eth1
172.16.0.0      0.0.0.0         255.240.0.0     U     1      0        0 eth2
172.16.0.4      0.0.0.0         255.255.255.255 UH    0      0        0 eth2
192.168.0.0     0.0.0.0         255.255.255.0   U     1      0        0 eth0
192.168.0.1     0.0.0.0         255.255.255.255 UH    0      0        0 eth0
ubuntu@sbhusari:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A FORWARD -i eth1 -o eth2 -j ACCEPT
-A FORWARD -i eth2 -o eth1 -j ACCEPT
ubuntu@sbhusari:~$ sudo iptables -t nat -S
-P PREROUTING ACCEPT
-P INPUT ACCEPT
-P OUTPUT ACCEPT
-P POSTROUTING ACCEPT
-A POSTROUTING -o eth0 -j MASQUERADE
-A POSTROUTING -o eth2 -j MASQUERADE
-A POSTROUTING -o eth1 -j MASQUERADE
ubuntu@sbhusari:~$
```



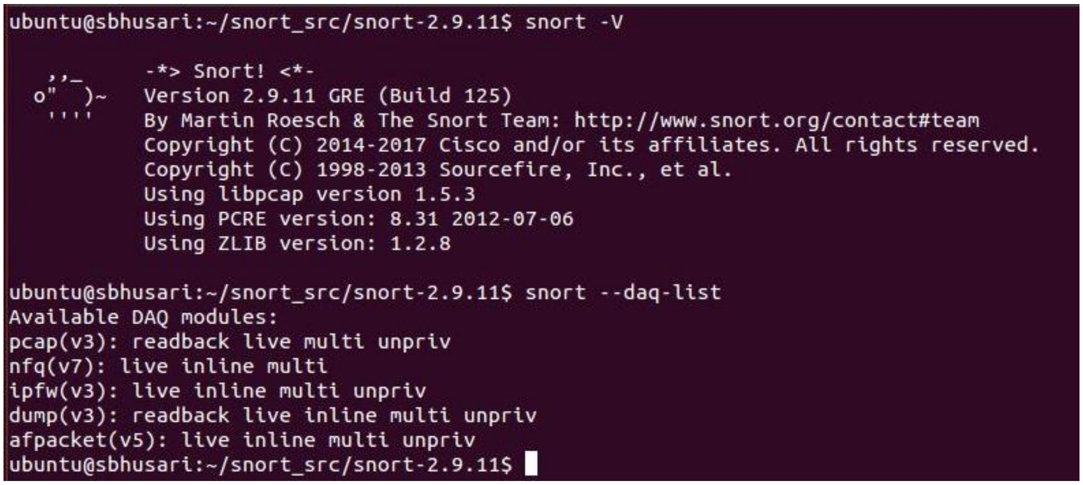
```
root@sbhusari:/home/ubuntu# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.16.0.5      0.0.0.0         UG    0      0        0 eth0
172.16.0.0      0.0.0.0         255.240.0.0     U     1      0        0 eth0
root@sbhusari:/home/ubuntu# ping 10.0.0.4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=63 time=2.28 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=63 time=1.64 ms
^C
--- 10.0.0.4 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.645/1.967/2.289/0.322 ms
root@sbhusari:/home/ubuntu# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=53 time=21.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=53 time=20.3 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=53 time=20.3 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 20.321/20.580/21.042/0.347 ms
root@sbhusari:/home/ubuntu#
```

Successful ping activity on the client

After fixing firewall and routes, Snort is installed on the gateway. Required modules are installed to perform IPS with NFQ on Snort.

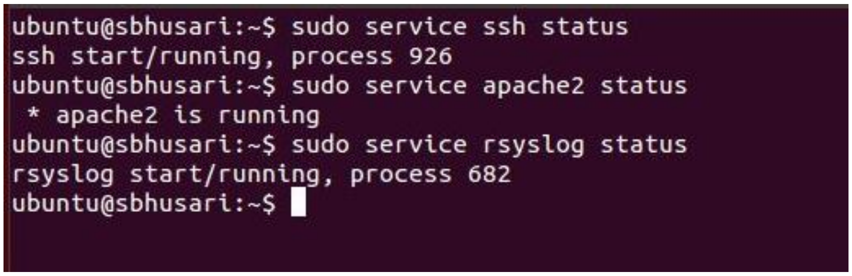


```
Build AFPacket DAQ module.. : yes
Build Dump DAQ module...... : yes
Build IPFW DAQ module...... : yes
Build IPQ DAQ module....... : no
Build NFQ DAQ module....... : yes
Build PCAP DAQ module...... : yes
Build netmap DAQ module.... : no
```



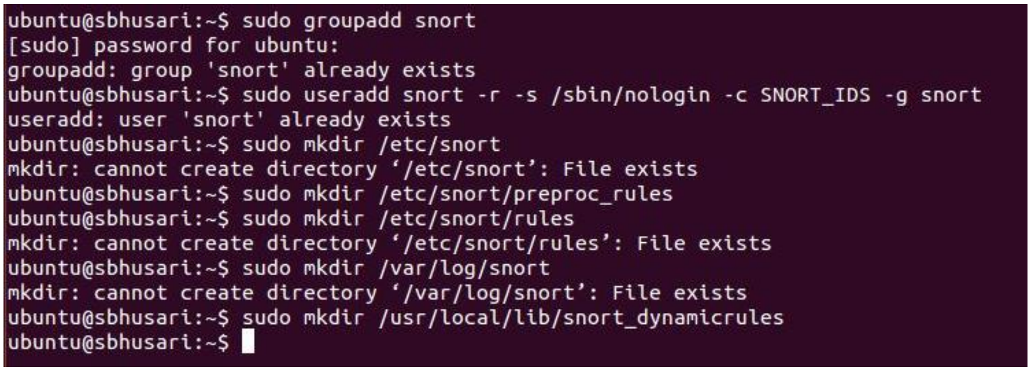
```
ubuntu@sbhusari:~/snort_src/snort-2.9.11$ snort -V

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.11 GRE (Build 125)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2017 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.5.3
           Using PCRE version: 8.31 2012-07-06
           Using ZLIB version: 1.2.8

ubuntu@sbhusari:~/snort_src/snort-2.9.11$ snort --daq-list
Available DAQ modules:
pcap(v3): readback live multi unpriv
nfq(v7): live inline multi
ipfw(v3): live inline multi unpriv
dump(v3): readback live inline multi unpriv
afpacket(v5): live inline multi unpriv
ubuntu@sbhusari:~/snort_src/snort-2.9.11$ 
```

Now, apache2 is installed and configured on the server. Rsyslo service is deployed with Linux OS and is generally active on the system.

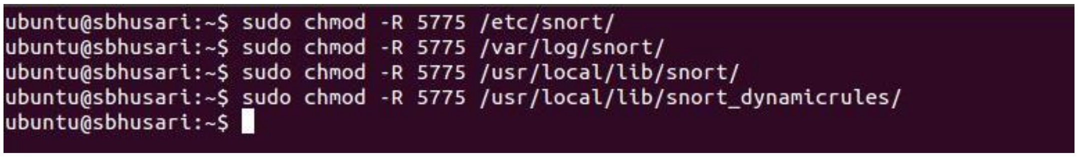


```
ubuntu@sbhusari:~$ sudo service ssh status
ssh start/running, process 926
ubuntu@sbhusari:~$ sudo service apache2 status
 * apache2 is running
ubuntu@sbhusari:~$ sudo service rsyslog status
rsyslog start/running, process 682
ubuntu@sbhusari:~$ 
```

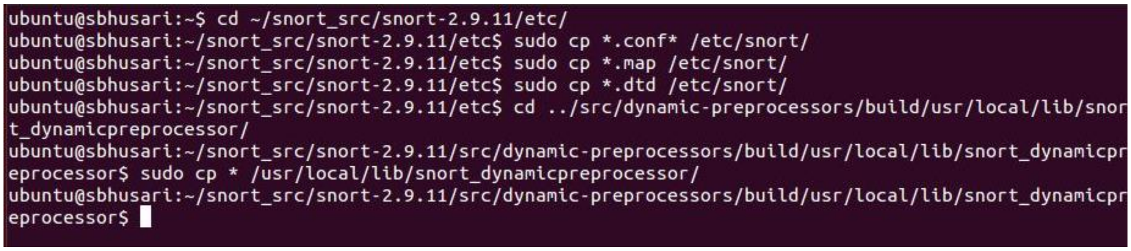
After installation, snort is configured as per given instructions.



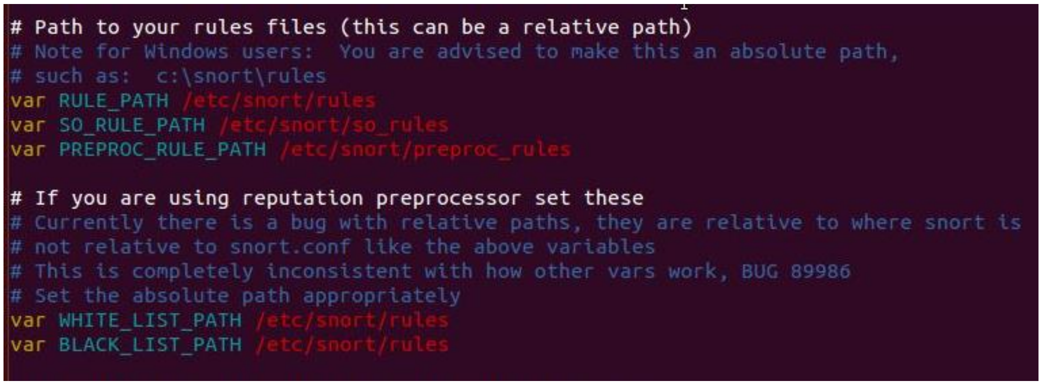
```
ubuntu@sbhusari:~$ sudo groupadd snort
[sudo] password for ubuntu:
groupadd: group 'snort' already exists
ubuntu@sbhusari:~$ sudo useradd snort -r -s /sbin/nologin -c SNORT_IDS -g snort
useradd: user 'snort' already exists
ubuntu@sbhusari:~$ sudo mkdir /etc/snort
mkdir: cannot create directory ‘/etc/snort’: File exists
ubuntu@sbhusari:~$ sudo mkdir /etc/snort/preproc_rules
ubuntu@sbhusari:~$ sudo mkdir /etc/snort/rules
mkdir: cannot create directory ‘/etc/snort/rules’: File exists
ubuntu@sbhusari:~$ sudo mkdir /var/log/snort
mkdir: cannot create directory ‘/var/log/snort’: File exists
ubuntu@sbhusari:~$ sudo mkdir /usr/local/lib/snort_dynamicrules
ubuntu@sbhusari:~$ █
```



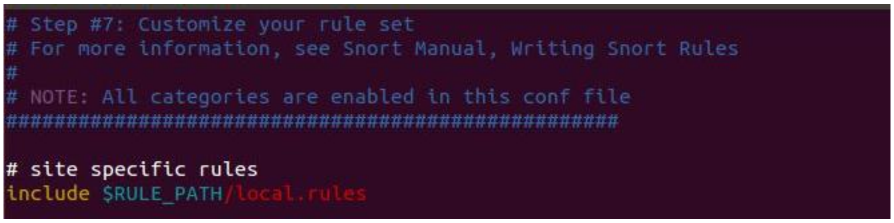
```
ubuntu@sbhusari:~$ sudo chmod -R 5775 /etc/snort/
ubuntu@sbhusari:~$ sudo chmod -R 5775 /var/log/snort/
ubuntu@sbhusari:~$ sudo chmod -R 5775 /usr/local/lib/snort/
ubuntu@sbhusari:~$ sudo chmod -R 5775 /usr/local/lib/snort_dynamicrules/
ubuntu@sbhusari:~$ █
```



```
ubuntu@sbhusari:~$ cd ~/snort_src/snort-2.9.11/etc/
ubuntu@sbhusari:~/snort_src/snort-2.9.11/etc$ sudo cp *.conf* /etc/snort/
ubuntu@sbhusari:~/snort_src/snort-2.9.11/etc$ sudo cp *.map /etc/snort/
ubuntu@sbhusari:~/snort_src/snort-2.9.11/etc$ sudo cp *.dtd /etc/snort/
ubuntu@sbhusari:~/snort_src/snort-2.9.11/etc$ cd ../src/dynamic-preprocessors/build/usr/local/lib/snor
t_dynamicpreprocessor/
ubuntu@sbhusari:~/snort_src/snort-2.9.11/src/dynamic-preprocessors/build/usr/local/lib/snort_dynamicpr
eprocessor$ sudo cp * /usr/local/lib/snort_dynamicpreprocessor/
ubuntu@sbhusari:~/snort_src/snort-2.9.11/src/dynamic-preprocessors/build/usr/local/lib/snort_dynamicpr
eprocessor$ █
```

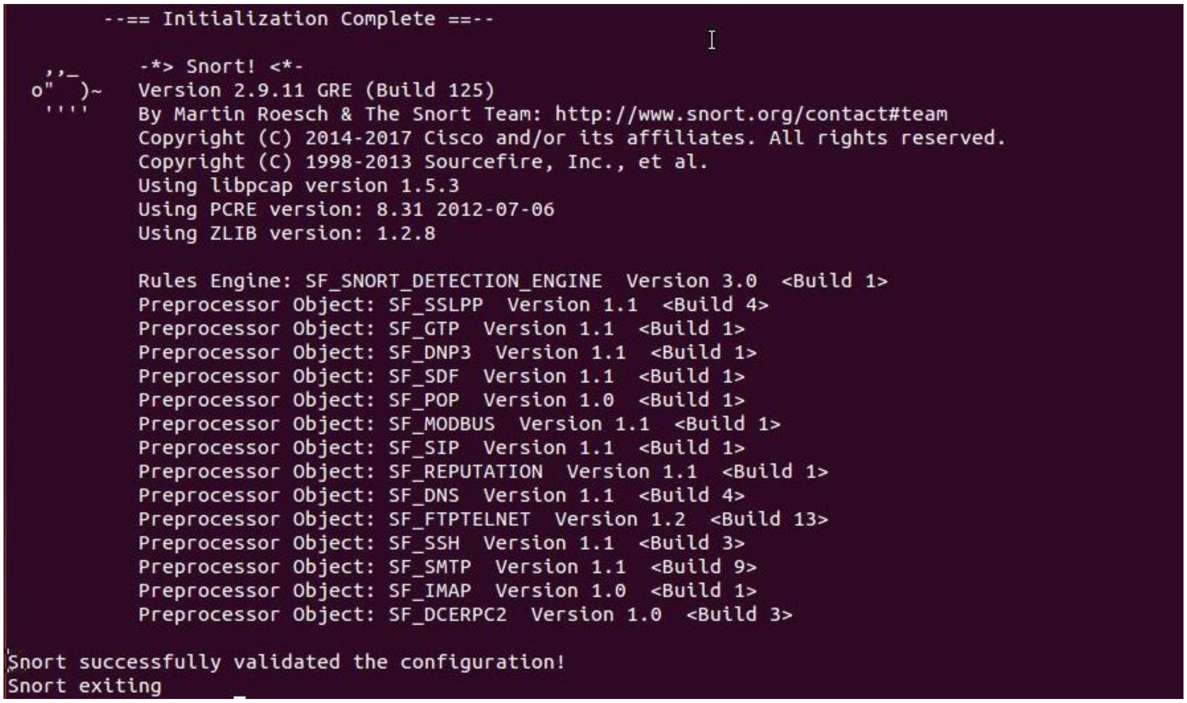


```
# Path to your rules files (this can be a relative path)
# Note for Windows users:  You are advised to make this an absolute path,
# such as:  c:\snort\rules
var RULE_PATH /etc/snort/rules
var SO_RULE_PATH /etc/snort/so_rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules

# If you are using reputation preprocessor set these
# Currently there is a bug with relative paths, they are relative to where snort is
# not relative to snort.conf like the above variables
# This is completely inconsistent with how other vars work, BUG 89986
# Set the absolute path appropriately
var WHITE_LIST_PATH /etc/snort/rules
var BLACK_LIST_PATH /etc/snort/rules
```



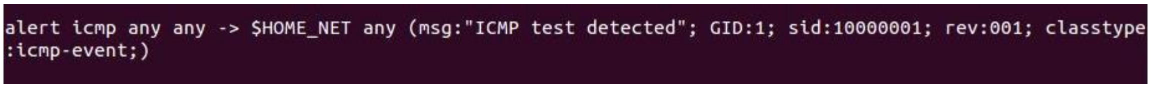
```
# Step #7: Customize your rule set
# For more information, see Snort Manual, Writing Snort Rules
#
# NOTE: All categories are enabled in this conf file
###################################################

# site specific rules
include $RULE_PATH/local.rules
```

Snort configuration is validated after these steps.



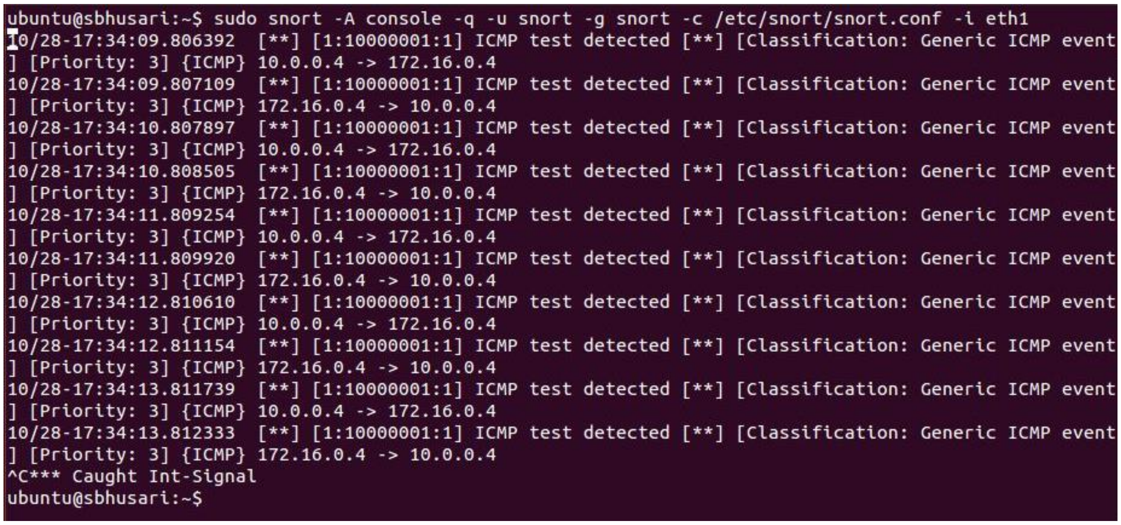
```
        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.11 GRE (Build 125)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2017 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.5.3
           Using PCRE version: 8.31 2012-07-06
           Using ZLIB version: 1.2.8

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.0  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           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_DCERPC2  Version 1.0  <Build 3>

Snort successfully validated the configuration!
Snort exiting
```

Now, for testing purpose, a simple rule is created in local.rules to log any ping (ICMP traffic) inside the network.

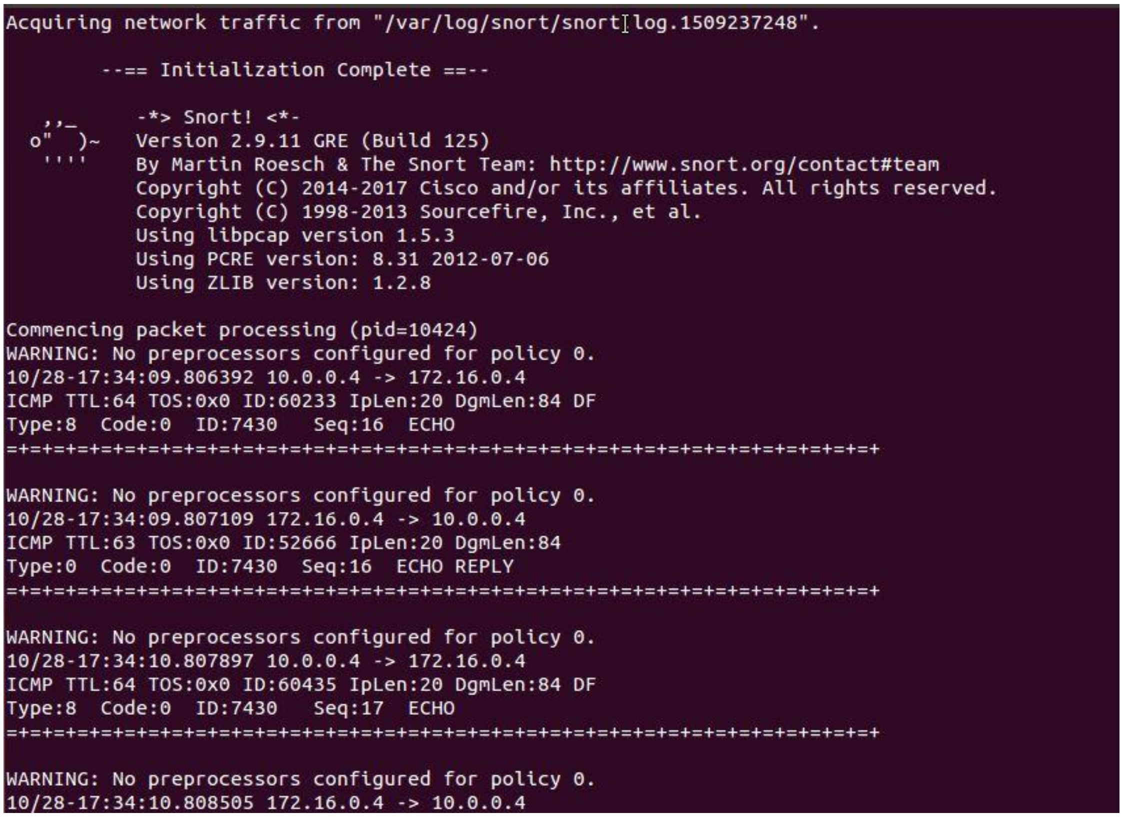


```
alert icmp any any -> $HOME_NET any (msg:"ICMP test detected"; GID:1; sid:10000001; rev:001; classtype
:icmp-event;)
```

After running the snort, we can view the ICMP traffic inside the network. Also, the alerts are logged in a file by Snort.



```
ubuntu@sbhusari:~$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth1
10/28-17:34:09.806392  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP event
] [Priority: 3] {ICMP} 10.0.0.4 -> 172.16.0.4
10/28-17:34:09.807109  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP event
] [Priority: 3] {ICMP} 172.16.0.4 -> 10.0.0.4
10/28-17:34:10.807897  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP event
] [Priority: 3] {ICMP} 10.0.0.4 -> 172.16.0.4
10/28-17:34:10.808505  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP event
] [Priority: 3] {ICMP} 172.16.0.4 -> 10.0.0.4
10/28-17:34:11.809254  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP event
] [Priority: 3] {ICMP} 10.0.0.4 -> 172.16.0.4
10/28-17:34:11.809920  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP event
] [Priority: 3] {ICMP} 172.16.0.4 -> 10.0.0.4
10/28-17:34:12.810610  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP event
] [Priority: 3] {ICMP} 10.0.0.4 -> 172.16.0.4
10/28-17:34:12.811154  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP event
] [Priority: 3] {ICMP} 172.16.0.4 -> 10.0.0.4
10/28-17:34:13.811739  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP event
] [Priority: 3] {ICMP} 10.0.0.4 -> 172.16.0.4
10/28-17:34:13.812333  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP event
] [Priority: 3] {ICMP} 172.16.0.4 -> 10.0.0.4
^C*** Caught Int-Signal
ubuntu@sbhusari:~$
```

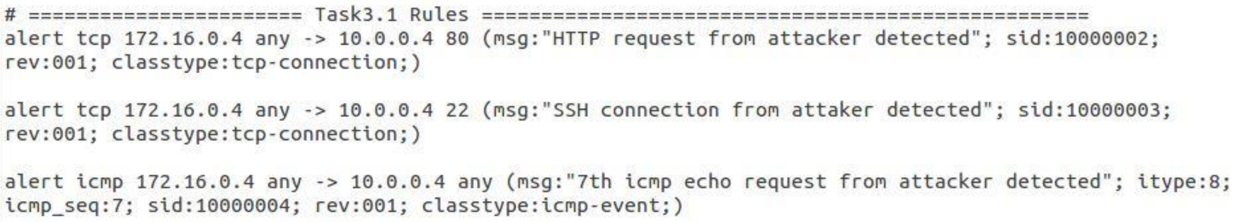


```
Acquiring network traffic from "/var/log/snort/snort.log.1509237248".

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.11 GRE (Build 125)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2017 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.5.3
           Using PCRE version: 8.31 2012-07-06
           Using ZLIB version: 1.2.8

Commencing packet processing (pid=10424)
WARNING: No preprocessors configured for policy 0.
10/28-17:34:09.806392 10.0.0.4 -> 172.16.0.4
ICMP TTL:64 TOS:0x0 ID:60233 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:7430   Seq:16  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/28-17:34:09.807109 172.16.0.4 -> 10.0.0.4
ICMP TTL:63 TOS:0x0 ID:52666 IpLen:20 DgmLen:84
Type:0  Code:0  ID:7430  Seq:16  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/28-17:34:10.807897 10.0.0.4 -> 172.16.0.4
ICMP TTL:64 TOS:0x0 ID:60435 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:7430   Seq:17  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/28-17:34:10.808505 172.16.0.4 -> 10.0.0.4
```

After the validation and testing of Snort, the rules for Intrusion detection and prevention are created and implemented to fulfill the project requirements in following steps:

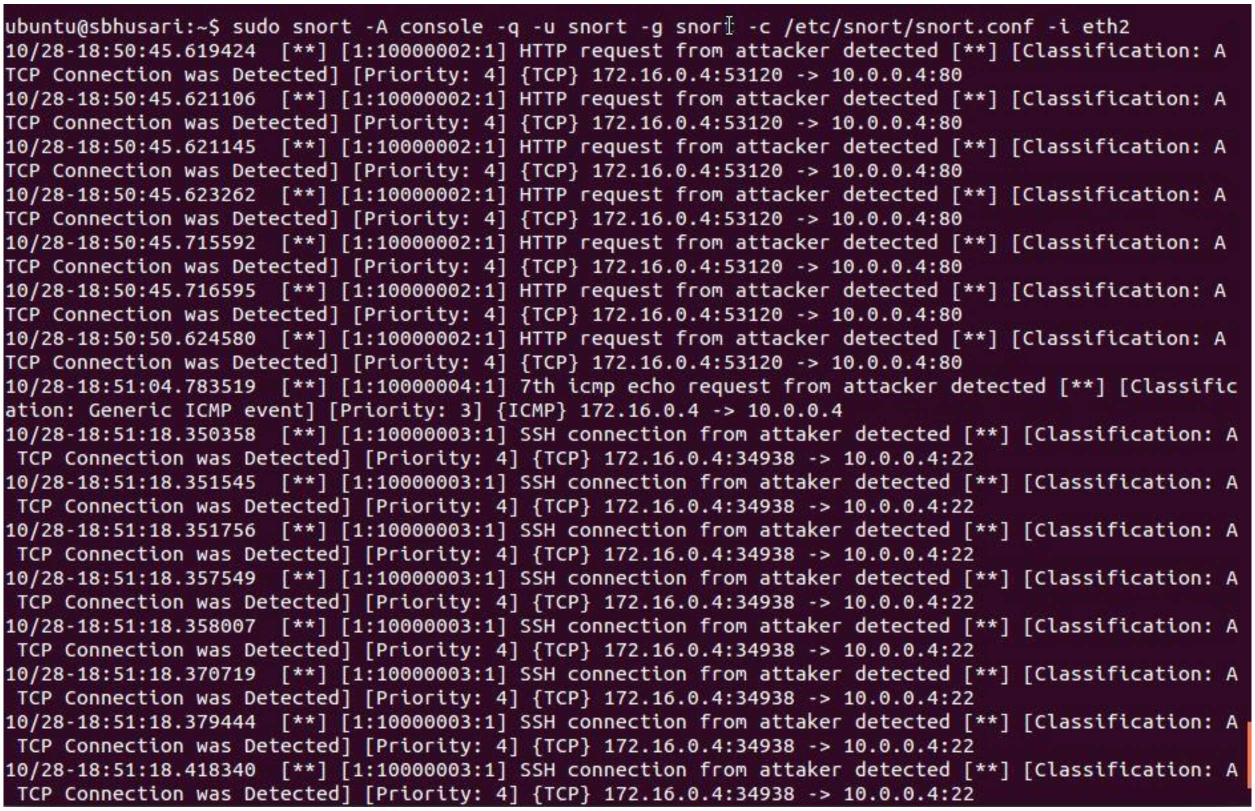
1) Simple rules are created to log traffic from client (attacker) to server. In this step, Snort is sunning as IDS. Following traffic is logged by Snort:
    - Any HTP connection request from client to server
    - Any SSH connection request from client to server

- ICMP echo request packet with sequence number 7

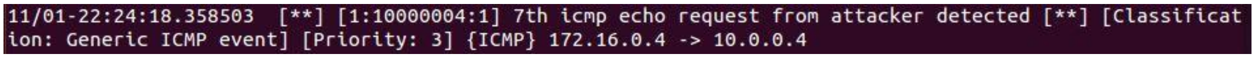


```
# ======================= Task3.1 Rules ==================================================
alert tcp 172.16.0.4 any -> 10.0.0.4 80 (msg:"HTTP request from attacker detected"; sid:10000002;
rev:001; classtype:tcp-connection;)

alert tcp 172.16.0.4 any -> 10.0.0.4 22 (msg:"SSH connection from attaker detected"; sid:10000003;
rev:001; classtype:tcp-connection;)

alert icmp 172.16.0.4 any -> 10.0.0.4 any (msg:"7th icmp echo request from attacker detected"; itype:8;
icmp_seq:7; sid:10000004; rev:001; classtype:icmp-event;)
```

After running Snort, logs can be displayed on the console.

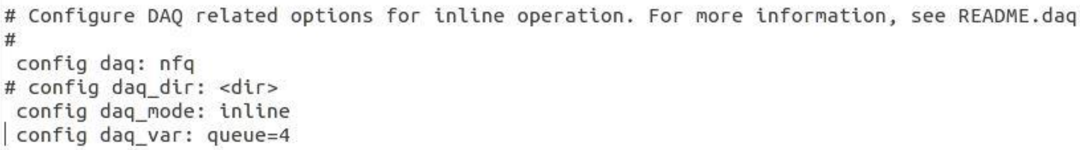


```
ubuntu@sbhusari:~$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth2
10/28-18:50:45.619424  [**] [1:10000002:1] HTTP request from attacker detected [**] [Classification: A
TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:53120 -> 10.0.0.4:80
10/28-18:50:45.621106  [**] [1:10000002:1] HTTP request from attacker detected [**] [Classification: A
TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:53120 -> 10.0.0.4:80
10/28-18:50:45.621145  [**] [1:10000002:1] HTTP request from attacker detected [**] [Classification: A
TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:53120 -> 10.0.0.4:80
10/28-18:50:45.623262  [**] [1:10000002:1] HTTP request from attacker detected [**] [Classification: A
TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:53120 -> 10.0.0.4:80
10/28-18:50:45.715592  [**] [1:10000002:1] HTTP request from attacker detected [**] [Classification: A
TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:53120 -> 10.0.0.4:80
10/28-18:50:45.716595  [**] [1:10000002:1] HTTP request from attacker detected [**] [Classification: A
TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:53120 -> 10.0.0.4:80
10/28-18:50:50.624580  [**] [1:10000002:1] HTTP request from attacker detected [**] [Classification: A
TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:53120 -> 10.0.0.4:80
10/28-18:51:04.783519  [**] [1:10000004:1] 7th icmp echo request from attacker detected [**] [Classific
ation: Generic ICMP event] [Priority: 3] {ICMP} 172.16.0.4 -> 10.0.0.4
10/28-18:51:18.350358  [**] [1:10000003:1] SSH connection from attaker detected [**] [Classification: A
 TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:34938 -> 10.0.0.4:22
10/28-18:51:18.351545  [**] [1:10000003:1] SSH connection from attaker detected [**] [Classification: A
 TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:34938 -> 10.0.0.4:22
10/28-18:51:18.351756  [**] [1:10000003:1] SSH connection from attaker detected [**] [Classification: A
 TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:34938 -> 10.0.0.4:22
10/28-18:51:18.357549  [**] [1:10000003:1] SSH connection from attaker detected [**] [Classification: A
 TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:34938 -> 10.0.0.4:22
10/28-18:51:18.358007  [**] [1:10000003:1] SSH connection from attaker detected [**] [Classification: A
 TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:34938 -> 10.0.0.4:22
10/28-18:51:18.370719  [**] [1:10000003:1] SSH connection from attaker detected [**] [Classification: A
 TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:34938 -> 10.0.0.4:22
10/28-18:51:18.379444  [**] [1:10000003:1] SSH connection from attaker detected [**] [Classification: A
 TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:34938 -> 10.0.0.4:22
10/28-18:51:18.418340  [**] [1:10000003:1] SSH connection from attaker detected [**] [Classification: A
 TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:34938 -> 10.0.0.4:22
```



```
11/01-22:24:18.358503  [**] [1:10000004:1] 7th icmp echo request from attacker detected [**] [Classificat
ion: Generic ICMP event] [Priority: 3] {ICMP} 172.16.0.4 -> 10.0.0.4
```

2) Now Snort in Intrusion prevention system (IPS) mode with DAQ type NFQ is demonstrated in this step. To make snort work in inline mode, following changes are made in snort.conf file.

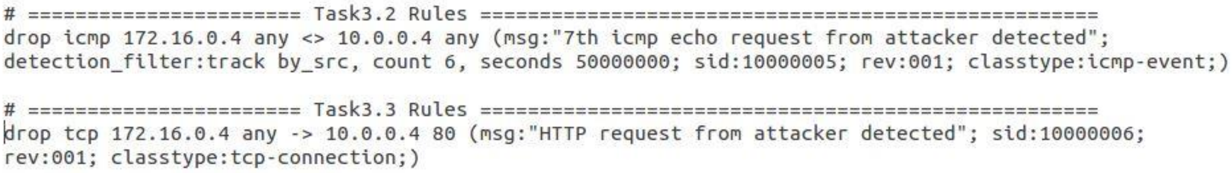


```
# Configure DAQ related options for inline operation. For more information, see README.daq
#
 config daq: nfq
# config daq_dir: <dir>
 config daq_mode: inline
 config daq_var: queue=4
```

Also iptables changes are made to allow Snort in IPS mode. Now following rules are created in local.rules file:

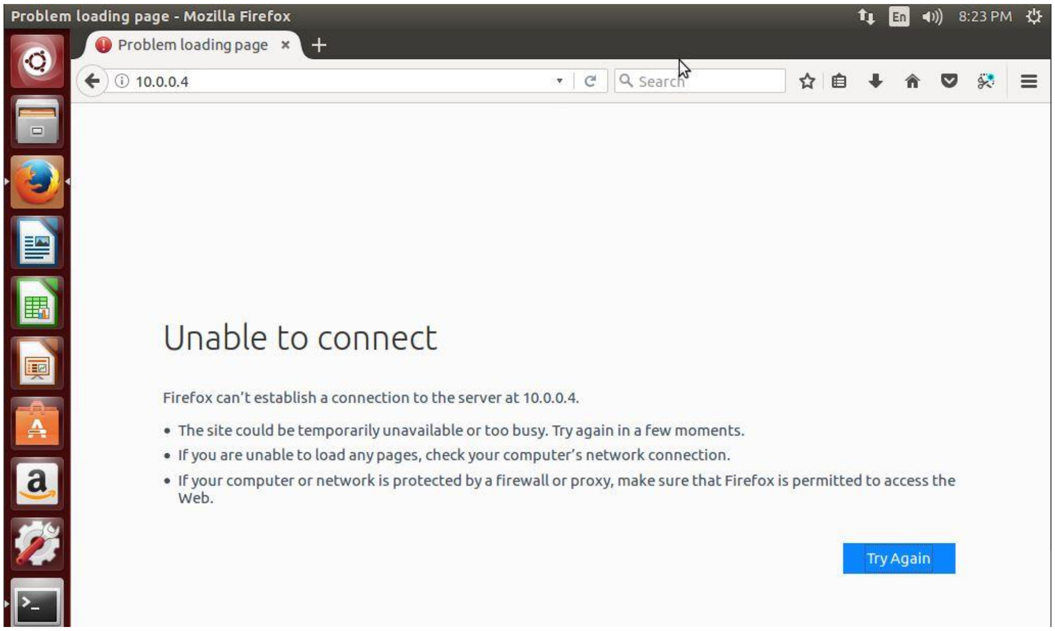
- Block ping traffic from attacker to server after 7th ICMP echo request packet is detected

- Block HTTP request from attacker to server



```
# ====================== Task3.2 Rules ====================================================
drop icmp 172.16.0.4 any <> 10.0.0.4 any (msg:"7th icmp echo request from attacker detected";
detection_filter:track by_src, count 6, seconds 50000000; sid:10000005; rev:001; classtype:icmp-event;)

# ====================== Task3.3 Rules ====================================================
drop tcp 172.16.0.4 any -> 10.0.0.4 80 (msg:"HTTP request from attacker detected"; sid:10000006;
rev:001; classtype:tcp-connection;)
```

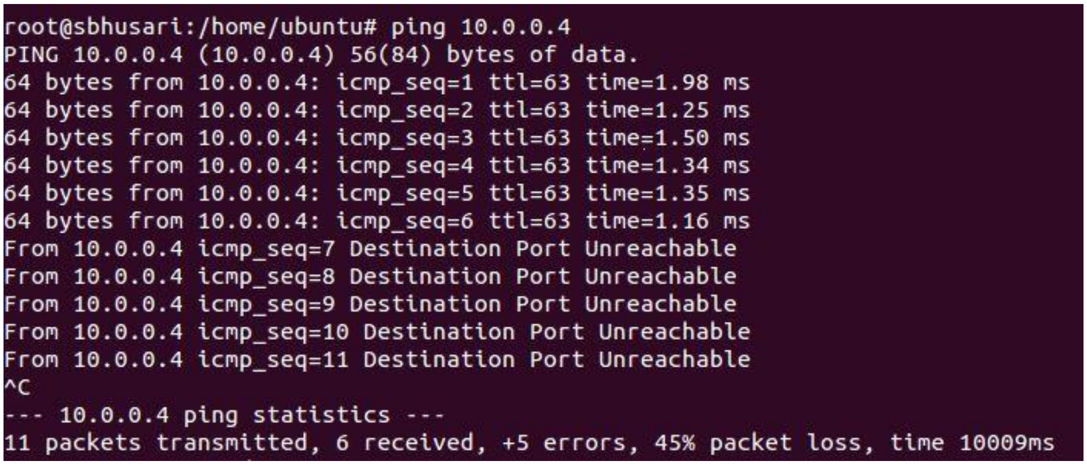
After running snort using given rules, we get desired results.



Unable to connect

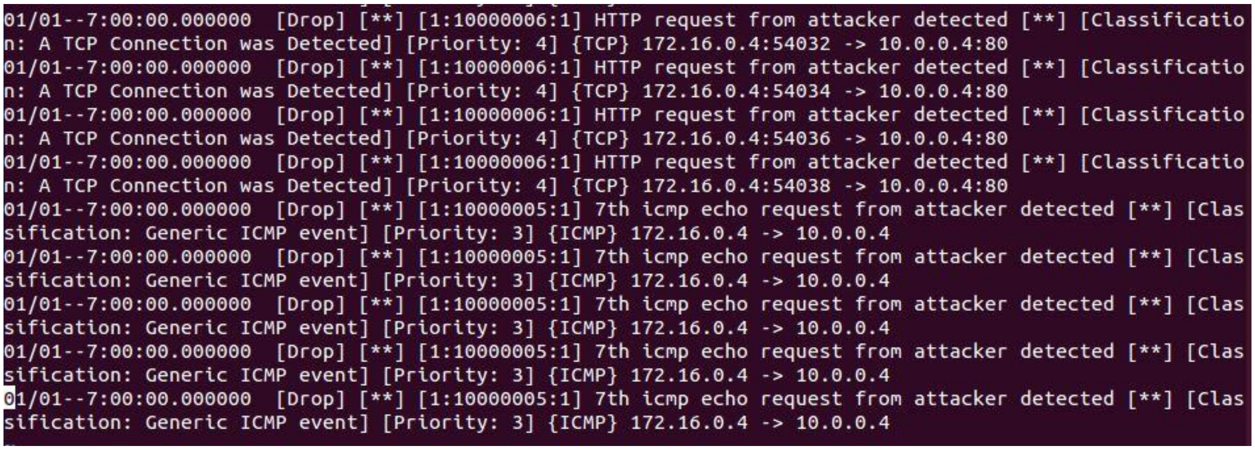
Firefox can't establish a connection to the server at 10.0.0.4.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.



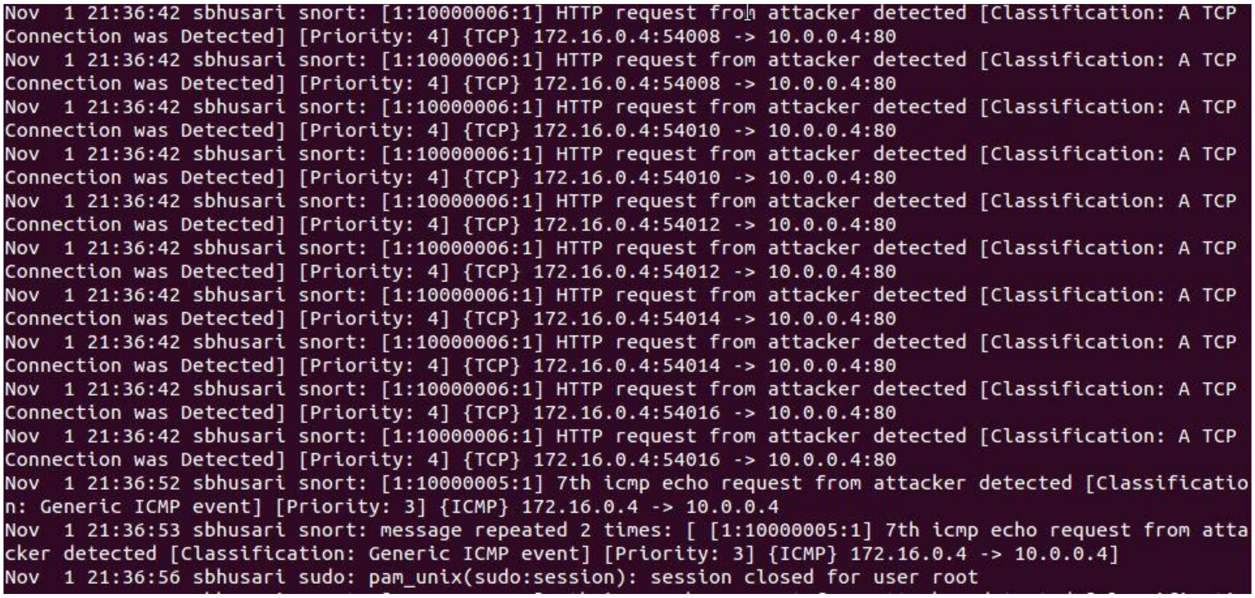
```
root@sbhusari:/home/ubuntu# ping 10.0.0.4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=63 time=1.98 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=63 time=1.25 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=63 time=1.50 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=63 time=1.34 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=63 time=1.35 ms
64 bytes from 10.0.0.4: icmp_seq=6 ttl=63 time=1.16 ms
From 10.0.0.4 icmp_seq=7 Destination Port Unreachable
From 10.0.0.4 icmp_seq=8 Destination Port Unreachable
From 10.0.0.4 icmp_seq=9 Destination Port Unreachable
From 10.0.0.4 icmp_seq=10 Destination Port Unreachable
From 10.0.0.4 icmp_seq=11 Destination Port Unreachable
^C
--- 10.0.0.4 ping statistics ---
11 packets transmitted, 6 received, +5 errors, 45% packet loss, time 10009ms
```

The snort also logs all the dropped packets.



```
01/01--7:00:00.000000  [Drop] [**] [1:10000006:1] HTTP request from attacker detected [**] [Classificatio
n: A TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:54032 -> 10.0.0.4:80
01/01--7:00:00.000000  [Drop] [**] [1:10000006:1] HTTP request from attacker detected [**] [Classificatio
n: A TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:54034 -> 10.0.0.4:80
01/01--7:00:00.000000  [Drop] [**] [1:10000006:1] HTTP request from attacker detected [**] [Classificatio
n: A TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:54036 -> 10.0.0.4:80
01/01--7:00:00.000000  [Drop] [**] [1:10000006:1] HTTP request from attacker detected [**] [Classificatio
n: A TCP Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:54038 -> 10.0.0.4:80
01/01--7:00:00.000000  [Drop] [**] [1:10000005:1] 7th icmp echo request from attacker detected [**] [Clas
sification: Generic ICMP event] [Priority: 3] {ICMP} 172.16.0.4 -> 10.0.0.4
01/01--7:00:00.000000  [Drop] [**] [1:10000005:1] 7th icmp echo request from attacker detected [**] [Clas
sification: Generic ICMP event] [Priority: 3] {ICMP} 172.16.0.4 -> 10.0.0.4
01/01--7:00:00.000000  [Drop] [**] [1:10000005:1] 7th icmp echo request from attacker detected [**] [Clas
sification: Generic ICMP event] [Priority: 3] {ICMP} 172.16.0.4 -> 10.0.0.4
01/01--7:00:00.000000  [Drop] [**] [1:10000005:1] 7th icmp echo request from attacker detected [**] [Clas
sification: Generic ICMP event] [Priority: 3] {ICMP} 172.16.0.4 -> 10.0.0.4
01/01--7:00:00.000000  [Drop] [**] [1:10000005:1] 7th icmp echo request from attacker detected [**] [Clas
sification: Generic ICMP event] [Priority: 3] {ICMP} 172.16.0.4 -> 10.0.0.4
```

3) In this step, we will send snort alerts to syslog server. In the given network, server host in the syslog server. To store snort alerts in a single file, fast alert option is enabled by adding following line in snort.conf (output alert_fast: /var/log/snort/fastalert). The fastalert file will store all the generated logs. After making appropriate changes in configuration file of rsyslog on both gateway and server host, logs can be obtained on the server.



```
Nov  1 21:36:42 sbhusari snort: [1:10000006:1] HTTP request from attacker detected [Classification: A TCP
Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:54008 -> 10.0.0.4:80
Nov  1 21:36:42 sbhusari snort: [1:10000006:1] HTTP request from attacker detected [Classification: A TCP
Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:54008 -> 10.0.0.4:80
Nov  1 21:36:42 sbhusari snort: [1:10000006:1] HTTP request from attacker detected [Classification: A TCP
Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:54010 -> 10.0.0.4:80
Nov  1 21:36:42 sbhusari snort: [1:10000006:1] HTTP request from attacker detected [Classification: A TCP
Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:54010 -> 10.0.0.4:80
Nov  1 21:36:42 sbhusari snort: [1:10000006:1] HTTP request from attacker detected [Classification: A TCP
Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:54012 -> 10.0.0.4:80
Nov  1 21:36:42 sbhusari snort: [1:10000006:1] HTTP request from attacker detected [Classification: A TCP
Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:54012 -> 10.0.0.4:80
Nov  1 21:36:42 sbhusari snort: [1:10000006:1] HTTP request from attacker detected [Classification: A TCP
Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:54014 -> 10.0.0.4:80
Nov  1 21:36:42 sbhusari snort: [1:10000006:1] HTTP request from attacker detected [Classification: A TCP
Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:54014 -> 10.0.0.4:80
Nov  1 21:36:42 sbhusari snort: [1:10000006:1] HTTP request from attacker detected [Classification: A TCP
Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:54016 -> 10.0.0.4:80
Nov  1 21:36:42 sbhusari snort: [1:10000006:1] HTTP request from attacker detected [Classification: A TCP
Connection was Detected] [Priority: 4] {TCP} 172.16.0.4:54016 -> 10.0.0.4:80
Nov  1 21:36:52 sbhusari snort: [1:10000005:1] 7th icmp echo request from attacker detected [Classificatio
n: Generic ICMP event] [Priority: 3] {ICMP} 172.16.0.4 -> 10.0.0.4
Nov  1 21:36:53 sbhusari snort: message repeated 2 times: [ [1:10000005:1] 7th icmp echo request from atta
cker detected [Classification: Generic ICMP event] [Priority: 3] {ICMP} 172.16.0.4 -> 10.0.0.4]
Nov  1 21:36:56 sbhusari sudo: pam_unix(sudo:session): session closed for user root
```

4. Conclusion

Given setup of Snort in IDS and IPS mode is successfully performed. It can be observed that Snort works as a powerful Intrusion detection system as well as Intrusion prevention system with appropriate configurations. Also, rsyslog facility helps to store alerts and logs generated in the network in a centralized server, to protect and properly utilize logs for system administration.

5. Appendix

- Link for the demonstration of project on youtube - https://youtu.be/Xv9E4QJS9g8