s

Experiment 12

Study of Network security : Set up Snort and study the logs.

Roll No.	19
Name	Manav Jawrani
Class	D15-A
Subject	Security Lab
LO Mapped	LO6: Demonstrate the network security system using open source tools.

Aim: Study of Network security : Set up Snort and study the logs.

Introduction:

What is Snort?

Snort is an open source network intrusion detection system created by Sourcefire founder and former CTO Martin Roesch. Cisco now develops and maintains Snort.

Snort is referred to as a packet sniffer that monitors network traffic, scrutinizing each packet closely to detect a dangerous payload or suspicious anomalies. Long a leader among enterprise intrusion prevention and detection tools, users can compile Snort on most Linux operating systems (OSes) or Unix. A version is also available for Windows.

How does Snort work?

Snort is based on library packet capture (libpcap). Libpcap is a tool that is widely used in Transmission Control Protocol/Internet Protocol address traffic sniffers, content searching and analyzers for packet logging, real-time traffic analysis, protocol analysis and content matching. Users can configure Snort as a sniffer, packet logger -- like TCPdump or Wireshark -- or network intrusion prevention methods.

Intrusion detection systems (IDS) vs. intrusion prevention systems (IPS)

Intrusion prevention system mode

As an open source network intrusion prevention system, Snort will monitor network traffic and compare it against a user-defined Snort rule set -- the file would be labeled snort.conf. This is Snort's most important function.

Snort applies rules to monitored traffic and issues alerts when it detects certain kinds of questionable activity on the network.

It can identify cybersecurity attack methods, including OS fingerprinting, denial of service, buffer overflow, common gateway interface attacks, stealth port scans and Server Message Block probes.

When Snort detects suspicious behavior, it acts as a firewall and sends a real-time alert to Syslog, to a separate alerts file or through a pop-up window.

Snort Modes:

Snort runs in three different modes:

1. Sniffer mode

2. Packet logger mode
3. Intrusion detection mode.

1. Sniffer mode:

The controller allows you to configure an access point to act as a network "sniffer," capturing and forwarding all packets on a specific channel to a remote machine running packet analyzer software. These packets include data such as time stamps, signal strength, packet sizes, and so on. Sniffers enable you to monitor and record network activity while also detecting problems. Snort will scan and identify network packets if a subscriber configures it to operate as a sniffer. Snort can also save those packets to disc.

2. Packet logger mode:

When run in this mode, Snort collects every packet it sees and puts it in hierarchical mode in the log directory. In other words, a new directory is created for each address collected, and data related to that address is stored in that directory. Snort puts the packets into ASCII files whose filenames are generated from the protocol and port number. This organization makes it easy to see who has connected to your network and what ports and protocols they are using: just use ls -R (or dir /s on Windows) to list the protocol directory. Make sure to specify your home network variable (either in your configuration file or by using the -h switch) to specify logging only for your home network.

This hierarchical organization is most useful when you are dealing with a limited number of hosts or when you want to see the IP addresses of collected hosts at a glance. However, the log directory can become very overloaded over time due to the ever-increasing number of directories and files. If you're logging all traffic on a very active network, it's even possible that you're missing the Inodes (a Unix data structure that limits the total number of files on a system) run out .-files) much

before running out of disk space. If someone did a full scan of your network and mapped all 65,536 TCP ports as well as 65,536 UDP ports, you would suddenly have 131,000 or more files, possibly all in a single directory. This file explosion can test the limits of most computers and easily escalate into a full-blown denial-of-service attack.

If you're not careful you can cause a real headache.

3. Intrusion Detection mode:

Only malicious packets will be logged by SNORT in NIDS mode. It accomplishes this by relying on the predefined characteristics of malicious packets defined in its rules. The action that SNORT takes is also defined in the network administrator's rules. Snort is an adequate network sniffer, but it is an excellent tool for detecting intruders. Snort provides near real-time intrusion detection when used as an NIDS.

Installing snort on Ubuntu Machine:

Use this website to install snort :

https://youtu.be/U6xMp-MIEfA

After successful installation you will see such a output:

```
manav@manav-virtual-machine:~$ sudo snort -v
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "ens33".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    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

Commencing packet processing (pid=15948)
09/24-04:33:19.302998 192.168.189.128:68 -> 192.168.189.254:67
UDP TTL:64 TOS:0xC0 ID:789 IpLen:20 DgmLen:327 DF
Len: 299
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
09/24-04:33:19.304648 192.168.189.254:67 -> 192.168.189.128:68
UDP TTL:16 TOS:0x10 ID:0 IpLen:20 DgmLen:328
Len: 300
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Snort Rules:

Snort is a signature-based IDS, and it defines rules to detect the intrusions. All rules of Snort are stored under /etc/snort/rules directory. The screenshot below shows the files that contain rules of Snort.

```
manav@manav-virtual-machine:~$ sudo ls /etc/snort/rules
[sudo] password for manav:
attack-responses.rules   community-game.rules          community-smtp.rules          deleted.rules      LICENSE           pop2.rules       sql.rules            web-misc.rules
AUTHORS                  community-icmp.rules          community-sql-injection.rules dns.rules          local.rules       pop3.rules       telnet.rules         web-php.rules
backdoor.rules           community-imap.rules          community-virus.rules         dos.rules          misc.rules        porn.rules       tftp.rules           white_list.rules
bad-traffic.rules        community-inappropriate.rules community-web-attacks.rules   experimental.rules multimedia.rules rpc.rules        virus.rules          x11.rules
black_list.rules         community-mail-client.rules   community-web-cgi.rules       exploit.rules      mysql.rules       rservices.rules  VRT-License.txt
chat.rules               community-misc.rules          community-web-client.rules    finger.rules       netbios.rules     scan.rules       web-attacks.rules
community-bot.rules      community-nntp.rules          community-web-dos.rules       ftp.rules          nntp.rules        shellcode.rules  web-cgi.rules
community-deleted.rules  community-oracle.rules        community-web-iis.rules       icmp-info.rules    oracle.rules      sid-msg.map      web-client.rules
community-dos.rules      community-policy.rules        community-web-misc.rules      icmp.rules         other-ids.rules   smtp.rules       web-coldfusion.rules
community-exploit.rules  community.rules               community-web-php.rules       imap.rules         p2p.rules         snmp.rules       web-frontpage.rules
community-ftp.rules      community-sip.rules           ddos.rules                    info.rules         policy.rules      snort.conf       web-iis.rules
manav@manav-virtual-machine:~$
```

****Note - To use snort in different modes you should run all the commands in root user.****

Different Modes of Snort:

1. Snort in Sniffer Mode:

We can use the following commands to run snort as Sniffer

A. Enable sniffer mode for Snort using the -v flag:

snort -v -c /etc/snort/snort.conf

```
WARNING: /etc/snort/rules/community-web-php.rules(474) GID 1 SID 100000934 in rule duplicates previous rule. Ignoring old rule.

4151 Snort rules read
    3477 detection rules
    0 decoder rules
    0 preprocessor rules
3477 Option Chains linked into 271 Chain Headers
0 Dynamic rules
+++++++++++++++++++++++++++++++++++++++++++++++++++

+-------------------[Rule Port Counts]-------------------------------------
|             tcp     udp    icmp      ip
|     src     151      18       0       0
|     dst    3306     126       0       0
|     any     383      48     146      22
|      nc      27       8      95      20
|     s+d      12       5       0       0
+--------------------------------------------------------------------------

+-----------------------[detection-filter-config]------------------------------
| memory-cap : 1048576 bytes
+-----------------------[detection-filter-rules]------------------------------
| none
-----------------------------------------------------------------------------

+-----------------------[rate-filter-config]-----------------------------------
| memory-cap : 1048576 bytes
+-----------------------[rate-filter-rules]------------------------------------
| none
-----------------------------------------------------------------------------

+-----------------------[event-filter-config]----------------------------------
| memory-cap : 1048576 bytes
+-----------------------[event-filter-global]----------------------------------
| none
+-----------------------[event-filter-local]-----------------------------------
| gen-id=1      sig-id=2495       type=Both      tracking=dst count=20  seconds=60
| gen-id=1      sig-id=3152       type=Threshold tracking=src count=5   seconds=2
| gen-id=1      sig-id=2924       type=Threshold tracking=dst count=10  seconds=60
| gen-id=1      sig-id=3273       type=Threshold tracking=src count=5   seconds=2
| gen-id=1      sig-id=1991       type=Limit     tracking=src count=1   seconds=60
| gen-id=1      sig-id=2275       type=Threshold tracking=dst count=5   seconds=60
| gen-id=1      sig-id=2494       type=Both      tracking=dst count=20  seconds=60
| gen-id=1      sig-id=2523       type=Both      tracking=dst count=10  seconds=10
```

B. Upon startup, Snort displays the mode, the logging directory, and the interface on which it is currently listening. When initialization completes, Snort begins dumping packets to the screen. This output is fairly basic: it displays only the IP and TCP/UDP/ICMP headers and little else. To break out of sniffer mode, use Ctrl-C. Snort exits by generating a summary of packets captured, including the protocols, packet fragmentation statistics, and stream reassembly stats. To view application data, use the -d flag. This option provides even more detailed output:

snort -vd -c /etc/snort/snort.conf

```
   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    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_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           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_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
Commencing packet processing (pid=3139)
09/26-23:58:30.416646 192.168.189.128:48318 -> 192.168.189.2:53
UDP TTL:64 TOS:0x0 ID:5814 IpLen:20 DgmLen:86 DF
Len: 58
F0 43 01 00 00 01 00 00 00 00 00 01 12 63 6F 6E  .C...........con
6E 65 63 74 69 76 69 74 79 2D 63 68 65 63 6B 06  nectivity-check.
75 62 75 6E 74 75 03 63 6F 6D 00 00 1C 00 01 00  ubuntu.com......
00 29 02 00 00 00 00 00 00 00                    .).........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

09/26-23:58:30.428947 192.168.189.2:53 -> 192.168.189.128:48318
UDP TTL:128 TOS:0x0 ID:21027 IpLen:20 DgmLen:147
Len: 119
F0 43 81 80 00 01 00 00 00 01 00 01 12 63 6F 6E  .C...........con
6E 65 63 74 69 76 69 74 79 2D 63 68 65 63 6B 06  nectivity-check.
75 62 75 6E 74 75 03 63 6F 6D 00 00 1C 00 01 C0  ubuntu.com......
1F 00 06 00 01 00 00 00 05 00 31 03 6E 73 31 09  ..........1.ns1.
63 61 6E 6F 6E 69 63 61 6C C0 26 0A 68 6F 73 74  canonical.&.host
6D 61 73 74 65 72 C0 3F 78 49 16 8C 00 00 2A 30  master.?xI....*0
00 00 0E 10 00 09 3A 80 00 00 0E 10 00 00 29 10  ......:.......).
00 00 00 00 05 00 00                             .......
```

```
    Post parameters extracted:            0
    HTTP response Headers extracted:      0
    HTTP Response Cookies extracted:      0
    Unicode:                              0
    Double unicode:                       0
    Non-ASCII representable:              0
    Directory traversals:                 0
    Extra slashes ("//"):                 0
    Self-referencing paths ("./"):        0
    HTTP Response Gzip packets extracted: 0
    Gzip Compressed Data Processed:       n/a
    Gzip Decompressed Data Processed:     n/a
    Total packets processed:              1
===============================================================================
SMTP Preprocessor Statistics
  Total sessions                                   : 0
  Max concurrent sessions                          : 0
===============================================================================
dcerpc2 Preprocessor Statistics
  Total sessions: 0
===============================================================================
SSL Preprocessor:
   SSL packets decoded: 7
          Client Hello: 0
          Server Hello: 4
           Certificate: 0
           Server Done: 0
   Client Key Exchange: 0
   Server Key Exchange: 0
         Change Cipher: 4
              Finished: 0
    Client Application: 0
    Server Application: 5
                 Alert: 0
  Unrecognized records: 2
  Completed handshakes: 0
        Bad handshakes: 0
      Sessions ignored: 1
    Detection disabled: 1
===============================================================================
SIP Preprocessor Statistics
  Total sessions: 0
===============================================================================
Snort exiting
root@manav-virtual-machine:~#
```

C. To view more details and see results similar to tcpdump—including data link layer headers, as opposed to just the application layer data—use the -e flag as well. Using both the -d and -e options displays almost all the data within the packet:

snort -vde -c /etc/snort/snort.conf

```
*** Caught Int-Signal
09/27-00:06:34.307626 00:50:56:C0:00:08 -> FF:FF:FF:FF:FF:FF type:0x800 len:0xF3
192.168.189.1:138 -> 192.168.189.255:138 UDP TTL:128 TOS:0x0 ID:38223 IpLen:20 DgmLen:229
Len: 201
11 02 C7 81 C0 A8 BD 01 00 8A 00 BB 00 00 20 45  ............... E
45 45 46 46 44 45 4C 46 45 45 50 46 41 43 4E 46  EEFFDELFEEPFACNF
45 46 45 45 4B 45 44 45 47 46 45 44 49 43 41 00  EFEEKEDEGFEDICA.
20 46 48 45 50 46 43 45 4C 45 48 46 43 45 50 46   FHEPFCELEHFCEPF
46 46 41 43 41 43 41 43 41 43 41 43 41 43 41 42  FFACACACACACACAB
4F 00 FF 53 4D 42 25 00 00 00 00 00 00 00 00 00  O..SMB%.........
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 11 00 00 21 00 00 00 00 00 00 00 00 00 E8  .....!..........
03 00 00 00 00 00 00 00 00 21 00 56 00 03 00 01  .........!.V....
00 00 00 02 00 32 00 5C 4D 41 49 4C 53 4C 4F 54  .....2.\MAILSLOT
5C 42 52 4F 57 53 45 00 0F 00 80 FC 0A 00 44 45  \BROWSE.......DE
53 4B 54 4F 50 2D 54 54 4A 43 46 54 38 00 0A 00  SKTOP-TTJCFT8...
03 10 05 00 0F 01 55 AA 00                       ......U..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

===============================================================================
Run time for packet processing was 253.138316 seconds
Snort processed 68 packets.
Snort ran for 0 days 0 hours 4 minutes 13 seconds
   Pkts/min:           17
   Pkts/sec:            0
===============================================================================
Memory usage summary:
  Total non-mmapped bytes (arena):       46260224
  Bytes in mapped regions (hblkhd):      13574144
  Total allocated space (uordblks):      40670208
  Total free space (fordblks):           5590016
  Topmost releasable block (keepcost):   127712
===============================================================================
Packet I/O Totals:
   Received:           68
   Analyzed:           68 (100.000%)
    Dropped:            0 (  0.000%)
   Filtered:            0 (  0.000%)
Outstanding:            0 (  0.000%)
   Injected:            0
===============================================================================
Breakdown by protocol (includes rebuilt packets):
        Eth:           69 (100.000%)
       VLAN:            0 (  0.000%)
        IP4:           51 ( 73.913%)
```

```
              UDP Discards: 0
                     Events: 0
           Internal Events: 0
           TCP Port Filter
                  Filtered: 0
                 Inspected: 0
                   Tracked: 4
           UDP Port Filter
                  Filtered: 0
                 Inspected: 0
                   Tracked: 14
==================================================================================
HTTP Inspect - encodings (Note: stream-reassembled packets included):
    POST methods:                          0
    GET methods:                           0
    HTTP Request Headers extracted:        0
    HTTP Request Cookies extracted:        0
    Post parameters extracted:             0
    HTTP response Headers extracted:       0
    HTTP Response Cookies extracted:       0
    Unicode:                               0
    Double unicode:                        0
    Non-ASCII representable:               0
    Directory traversals:                  0
    Extra slashes ("//"):                  0
    Self-referencing paths ("./"):         0
    HTTP Response Gzip packets extracted: 0
    Gzip Compressed Data Processed:        n/a
    Gzip Decompressed Data Processed:      n/a
    Total packets processed:               1
==================================================================================
SMTP Preprocessor Statistics
  Total sessions                                   : 0
  Max concurrent sessions                          : 0
==================================================================================
dcerpc2 Preprocessor Statistics
  Total sessions: 0
==================================================================================
==================================================================================
SIP Preprocessor Statistics
  Total sessions: 0
==================================================================================
Snort exiting
root@manav-virtual-machine:~#
root@manav-virtual-machine:~#
```

2. Snort in Packet Logger Mode:

Command to check all the log details:

u2spewfoo /var/log/snort/snort.log

```
root@manav-virtual-machine:~# u2spewfoo /var/log/snort/snort.log

(Event)
        sensor id: 0    event id: 1     event second: 1664217188        event microsecond: 42424
        sig id: 1917    gen id: 1       revision: 6      classification: 23
        priority: 3     ip source: 192.168.189.1        ip destination: 239.255.255.250
        src port: 56401 dest port: 1900 protocol: 17    impact_flag: 0  blocked: 0
        mpls label: 0   vland id: 0     policy id: 0

Packet
        sensor id: 0    event id: 1     event second: 1664217188
        packet second: 1664217188       packet microsecond: 42424
        linktype: 1     packet_length: 217
[    0] 01 00 5E 7F FF FA 00 50 56 C0 00 08 08 00 45 00  ..^....PV.....E.
[   16] 00 CB C6 8E 00 00 01 11 84 EF C0 A8 BD 01 EF FF  ................
[   32] FF FA DC 51 07 6C 00 B7 32 82 4D 2D 53 45 41 52  ...Q.l..2.M-SEAR
[   48] 43 48 20 2A 20 48 54 54 50 2F 31 2E 31 0D 0A 48  CH * HTTP/1.1..H
[   64] 4F 53 54 3A 20 32 33 39 2E 32 35 35 2E 32 35 35  OST: 239.255.255
[   80] 2E 32 35 30 3A 31 39 30 30 0D 0A 4D 41 4E 3A 20  .250:1900..MAN:
[   96] 22 73 73 64 70 3A 64 69 73 63 6F 76 65 72 22 0D  "ssdp:discover".
[  112] 0A 4D 58 3A 20 31 0D 0A 53 54 3A 20 75 72 6E 3A  .MX: 1..ST: urn:
[  128] 64 69 61 6C 2D 6D 75 6C 74 69 73 63 72 65 65 6E  dial-multiscreen
[  144] 2D 6F 72 67 3A 73 65 72 76 69 63 65 3A 64 69 61  -org:service:dia
[  160] 6C 3A 31 0D 0A 55 53 45 52 2D 41 47 45 4E 54 3A  l:1..USER-AGENT:
[  176] 20 4D 69 63 72 6F 73 6F 66 74 20 45 64 67 65 2F   Microsoft Edge/
[  192] 31 30 35 2E 30 2E 31 33 34 33 2E 35 30 20 57 69  105.0.1343.50 Wi
[  208] 6E 64 6F 77 73 0D 0A 0D 0A                       ndows....

(Event)
        sensor id: 0    event id: 2     event second: 1664217188        event microsecond: 121225
        sig id: 1917    gen id: 1       revision: 6      classification: 23
        priority: 3     ip source: 192.168.189.1        ip destination: 239.255.255.250
        src port: 56405 dest port: 1900 protocol: 17    impact_flag: 0  blocked: 0
        mpls label: 0   vland id: 0     policy id: 0

Packet
        sensor id: 0    event id: 2     event second: 1664217188
        packet second: 1664217188       packet microsecond: 121225
        linktype: 1     packet_length: 217
[    0] 01 00 5E 7F FF FA 00 50 56 C0 00 08 08 00 45 00  ..^....PV.....E.
[   16] 00 CB C6 8F 00 00 01 11 84 EE C0 A8 BD 01 EF FF  ................
[   32] FF FA DC 55 07 6C 00 B7 42 A0 4D 2D 53 45 41 52  ...U.l..B.M-SEAR
[   48] 43 48 20 2A 20 48 54 54 50 2F 31 2E 31 0D 0A 48  CH * HTTP/1.1..H
[   64] 4F 53 54 3A 20 32 33 39 2E 32 35 35 2E 32 35 35  OST: 239.255.255
[   80] 2E 32 35 30 3A 31 39 30 30 0D 0A 4D 41 4E 3A 20  .250:1900..MAN:
```

```
[   16] 00 00 00 24 00 01 FE 80 00 00 00 00 00 00 58 5B  ...$..........X[
[   32] D5 5C E3 B9 A4 76 FF 02 00 00 00 00 00 00 00 00  .\...v..........
[   48] 00 00 00 00 00 16 3A 00 05 02 00 00 01 00 8F 00  ......:.........
[   64] BB 1E 00 00 00 01 03 00 00 00 FF 02 00 00 00 00  ................
[   80] 00 00 00 00 00 00 00 01 00 03                    ..........

(IPv6 Event)
        sensor id: 0    event id: 20    event second: 1664217333        event microsecond: 807029
        sig id: 1000001 gen id: 1       revision: 1      classification: 0
        priority: 0     ip source: fe80::585b:d55c:e3b9:a476    ip destination: ff02::16
        src port: 0     dest port: 0    protocol: 58    impact_flag: 0  blocked: 0
        mpls label: 0   vland id: 0     policy id: 0

Packet
        sensor id: 0    event id: 20    event second: 1664217333
        packet second: 1664217333       packet microsecond: 807029
        linktype: 1     packet_length: 90
[    0] 33 33 00 00 00 16 00 50 56 C0 00 08 86 DD 60 00  33.....PV.....`.
[   16] 00 00 00 24 00 01 FE 80 00 00 00 00 00 00 58 5B  ...$..........X[
[   32] D5 5C E3 B9 A4 76 FF 02 00 00 00 00 00 00 00 00  .\...v..........
[   48] 00 00 00 00 00 16 3A 00 05 02 00 00 01 00 8F 00  ......:.........
[   64] BA 1E 00 00 00 01 04 00 00 00 FF 02 00 00 00 00  ................
[   80] 00 00 00 00 00 00 00 01 00 03                    ..........

(IPv6 Event)
        sensor id: 0    event id: 21    event second: 1664217334        event microsecond: 235647
        sig id: 1000001 gen id: 1       revision: 1      classification: 0
        priority: 0     ip source: fe80::585b:d55c:e3b9:a476    ip destination: ff02::16
        src port: 0     dest port: 0    protocol: 58    impact_flag: 0  blocked: 0
        mpls label: 0   vland id: 0     policy id: 0

Packet
        sensor id: 0    event id: 21    event second: 1664217334
        packet second: 1664217334       packet microsecond: 235647
        linktype: 1     packet_length: 90
[    0] 33 33 00 00 00 16 00 50 56 C0 00 08 86 DD 60 00  33.....PV.....`.
[   16] 00 00 00 24 00 01 FE 80 00 00 00 00 00 00 58 5B  ...$..........X[
[   32] D5 5C E3 B9 A4 76 FF 02 00 00 00 00 00 00 00 00  .\...v..........
[   48] 00 00 00 00 00 16 3A 00 05 02 00 00 01 00 8F 00  ......:.........
[   64] BA 1E 00 00 00 01 04 00 00 00 FF 02 00 00 00 00  ................
[   80] 00 00 00 00 00 00 00 01 00 03                    ..........
get_record: (2) Failed to allocate memory.
free(): double free detected in tcache 2
Aborted (core dumped)
root@manav-virtual-machine:~#
```

3. Snort in Intrusion Detection mode:
sudo gedit /etc/snort/snort.conf

When the snort.conf file opens, scroll down until you find the ipvar HOME_NET setting. You'll want to change the IP address to be your actual class C subnet. Currently, it should be 192.168.X.0/24. You'll simply change the IP address part to match your Ubuntu Server VM IP, making sure to leave the ".0/24″ on the end.

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

# Setup the network addresses you are protecting
#
# 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 192.168.189.0/24

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
# If HOME_NET is defined as something other than "any", alternative, you can
# use this definition if you do not want to detect attacks from your internal
# IP addresses:
#ipvar EXTERNAL_NET !$HOME_NET

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

Select Save from the bar on top and close the file. At this point, Snort is ready to run. Except, it doesn't have any rules loaded. To verify, run the following command:

sudo snort -T -i eth0 -c /etc/snort/snort.conf

```
| State Density     : 10.6%
| Patterns          : 5055
| Match States      : 3855
| Memory (MB)       : 17.00
|   Patterns        : 0.51
|   Match Lists     : 1.02
|   DFA
|     1 byte states : 1.02
|     2 byte states : 14.05
|     4 byte states : 0.00
+-------------------------------------------------------------------
[ Number of patterns truncated to 20 bytes: 1039 ]
pcap DAQ configured to passive.
Acquiring network traffic from "eth0".

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    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_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           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_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>

Snort successfully validated the configuration!
Snort exiting
root@manav-virtual-machine:~#
```

Writing and Adding a Snort Rule: -

1. For ICMP Packet found.

Next, we are going to add a simple snort rule. You should add your own rules at /etc/snort/rules/local.rules. Add the following line into the local.rules file

alert icmp any any -> any any (msg:"ICMP Packet found"; sid:1000001; rev:1;)

Basically, this rule defines that an alert will be logged if an ICMP packet is found. The ICMP packet could be from any IP address and the rule ID is 1000001. Make sure to pick a SID greater than 1000000 for your own rules. The screenshot below shows the contents of the local.rules file after adding the rule.

```
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# additions here.
alert icmp any any -> any any (msg:"ICMP Packet found"; sid:1000001; rev:1;)
```

Now start pinging your Ubuntu Server with the following command (use your Ubuntu Server IP instead of *.x.x*):

ping 192.168.x.x

```
C:\Users\MANAV>ping 192.168.189.128

Pinging 192.168.189.128 with 32 bytes of data:
Reply from 192.168.189.128: bytes=32 time<1ms TTL=64
Reply from 192.168.189.128: bytes=32 time=1ms TTL=64
Reply from 192.168.189.128: bytes=32 time=1ms TTL=64
Reply from 192.168.189.128: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.189.128:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\MANAV>_
```

Now return to your Ubuntu Server running Snort IDS. You should see alerts generated for every ICMP Echo request and Echo reply message, with the message text we specified in the msg option:

```
root@manav-virtual-machine: ~
root@manav-virtual-machine:~# sudo snort -A console -q -c /etc/snort/snort.conf
09/27-00:48:47.912054  [**] [1:382:7] ICMP PING Windows [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.189.1 -> 192.168.189.128
09/27-00:48:47.912054  [**] [1:1000001:1] ICMP Packet found [**] [Priority: 0] {ICMP} 192.168.189.1 -> 192.168.189.128
09/27-00:48:47.912054  [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.189.1 -> 192.168.189.128
09/27-00:48:47.912086  [**] [1:1000001:1] ICMP Packet found [**] [Priority: 0] {ICMP} 192.168.189.128 -> 192.168.189.1
09/27-00:48:47.912086  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.189.128 -> 192.168.189.1
09/27-00:48:48.930112  [**] [1:382:7] ICMP PING Windows [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.189.1 -> 192.168.189.128
09/27-00:48:48.930112  [**] [1:1000001:1] ICMP Packet found [**] [Priority: 0] {ICMP} 192.168.189.1 -> 192.168.189.128
09/27-00:48:48.930112  [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.189.1 -> 192.168.189.128
09/27-00:48:48.930159  [**] [1:1000001:1] ICMP Packet found [**] [Priority: 0] {ICMP} 192.168.189.128 -> 192.168.189.1
09/27-00:48:48.930159  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.189.128 -> 192.168.189.1
09/27-00:48:49.940058  [**] [1:382:7] ICMP PING Windows [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.189.1 -> 192.168.189.128
09/27-00:48:49.940058  [**] [1:1000001:1] ICMP Packet found [**] [Priority: 0] {ICMP} 192.168.189.1 -> 192.168.189.128
09/27-00:48:49.940058  [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.189.1 -> 192.168.189.128
09/27-00:48:49.940102  [**] [1:1000001:1] ICMP Packet found [**] [Priority: 0] {ICMP} 192.168.189.128 -> 192.168.189.1
09/27-00:48:49.940102  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.189.128 -> 192.168.189.1
09/27-00:48:50.951960  [**] [1:382:7] ICMP PING Windows [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.189.1 -> 192.168.189.128
09/27-00:48:50.951960  [**] [1:1000001:1] ICMP Packet found [**] [Priority: 0] {ICMP} 192.168.189.1 -> 192.168.189.128
09/27-00:48:50.951960  [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.189.1 -> 192.168.189.128
09/27-00:48:50.952003  [**] [1:1000001:1] ICMP Packet found [**] [Priority: 0] {ICMP} 192.168.189.128 -> 192.168.189.1
09/27-00:48:50.952003  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.189.128 -> 192.168.189.1
```

Here in output you can see the message which we have written in our local rules **"ICMP Packet found"**

2. For FTP connection:
Rule:

alert tcp any any -> any any (msg:”FTP connection attempt”; sid:1000002; rev:1;)

```
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# additions here.
alert icmp any any -> any any (msg:"ICMP Packet found"; sid:1000001; rev:1;)
alert tcp any any -> any any (msg:”FTP connection attempt”; sid:1000002; rev:1;)
```

Here we changed the protocol to TCP, and changed the alert message text. Save and close the file. Now let’s run Snort in IDS mode again, but this time, we are going to add one more option, as follows:

Now from your host type:

ftp 192.162.X.X

```
C:\Users\MANAV>ftp 192.168.189.128
> ftp: connect :Connection refused
ftp>
```

Then

sudo snort -A console -q -c /etc/snort/snort.conf

```
09/27-01:12:35.200023  [**] [1:1000002:1] "FTP connection attempt" [**] [Priority: 0] {TCP} 192.168.189.1:56869 -> 192.168.189.128:21
09/27-01:12:35.200067  [**] [1:1000002:1] "FTP connection attempt" [**] [Priority: 0] {TCP} 192.168.189.128:21 -> 192.168.189.1:56869
09/27-01:12:35.705266  [**] [1:1000002:1] "FTP connection attempt" [**] [Priority: 0] {TCP} 192.168.189.1:56869 -> 192.168.189.128:21
09/27-01:12:35.705305  [**] [1:1000002:1] "FTP connection attempt" [**] [Priority: 0] {TCP} 192.168.189.128:21 -> 192.168.189.1:56869
09/27-01:12:36.219182  [**] [1:1000002:1] "FTP connection attempt" [**] [Priority: 0] {TCP} 192.168.189.1:56869 -> 192.168.189.128:21
09/27-01:12:36.219232  [**] [1:1000002:1] "FTP connection attempt" [**] [Priority: 0] {TCP} 192.168.189.128:21 -> 192.168.189.1:56869
09/27-01:12:36.732509  [**] [1:1000002:1] "FTP connection attempt" [**] [Priority: 0] {TCP} 192.168.189.1:56869 -> 192.168.189.128:21
09/27-01:12:36.732571  [**] [1:1000002:1] "FTP connection attempt" [**] [Priority: 0] {TCP} 192.168.189.128:21 -> 192.168.189.1:56869
09/27-01:12:37.246974  [**] [1:1000002:1] "FTP connection attempt" [**] [Priority: 0] {TCP} 192.168.189.1:56869 -> 192.168.189.128:21
09/27-01:12:37.247016  [**] [1:1000002:1] "FTP connection attempt" [**] [Priority: 0] {TCP} 192.168.189.128:21 -> 192.168.189.1:56869
```

Here in output you can see the message which we have written in our local rules **"FTP connection attempt"**

<u>Conclusion</u>:

Thus, we have successfully set up snort, studied the logs and used snort in different modes.