Exp No:6 Perform Snort tool analysis with screenshot for each step.

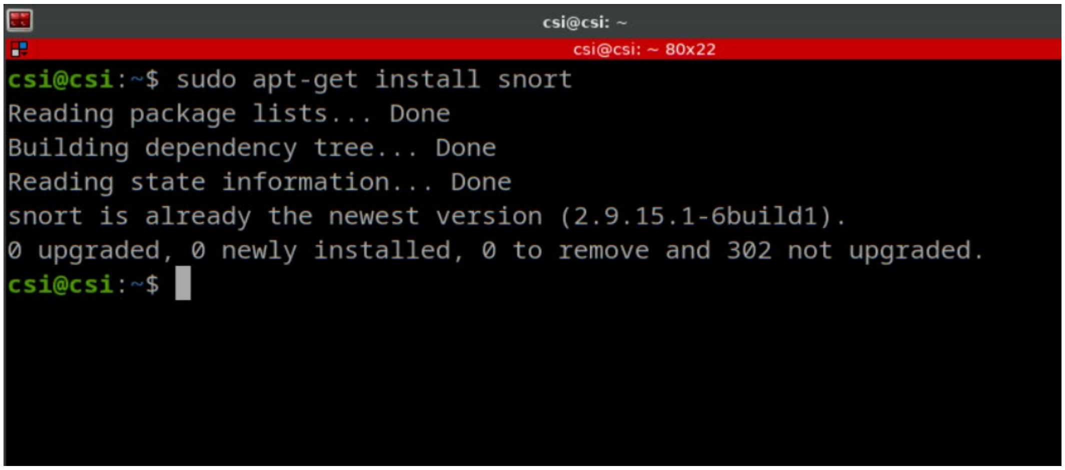
Date: 19 September 2022

Aim: to detect captured packets from attacker in Linux system.

Algorithm:

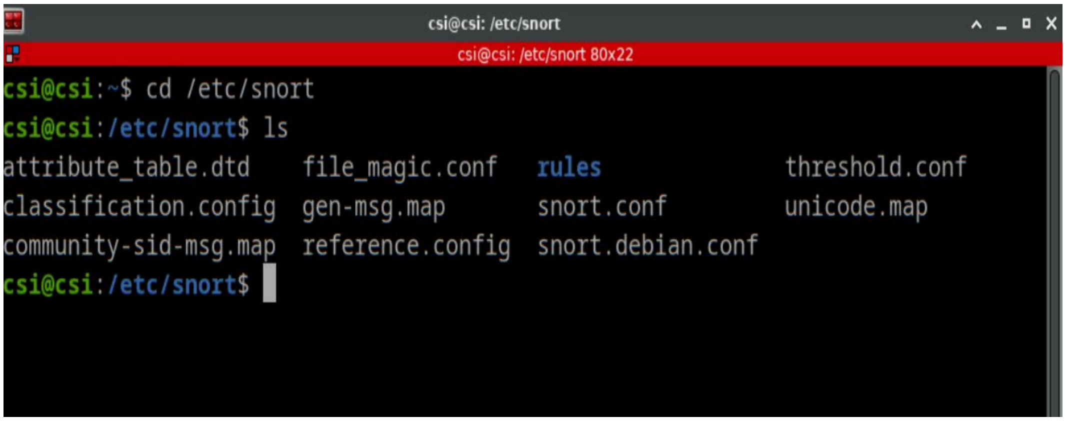
1. We need two Linux VM install in our system with host-only adapter.
2. Boot up both Linux.
3. In target system(CSI Linux) we need to install snort using below command.

   sudo apt-get install snort -y



```
csi@csi:~$ sudo apt-get install snort
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
snort is already the newest version (2.9.15.1-6build1).
0 upgraded, 0 newly installed, 0 to remove and 302 not upgraded.
csi@csi:~$
```

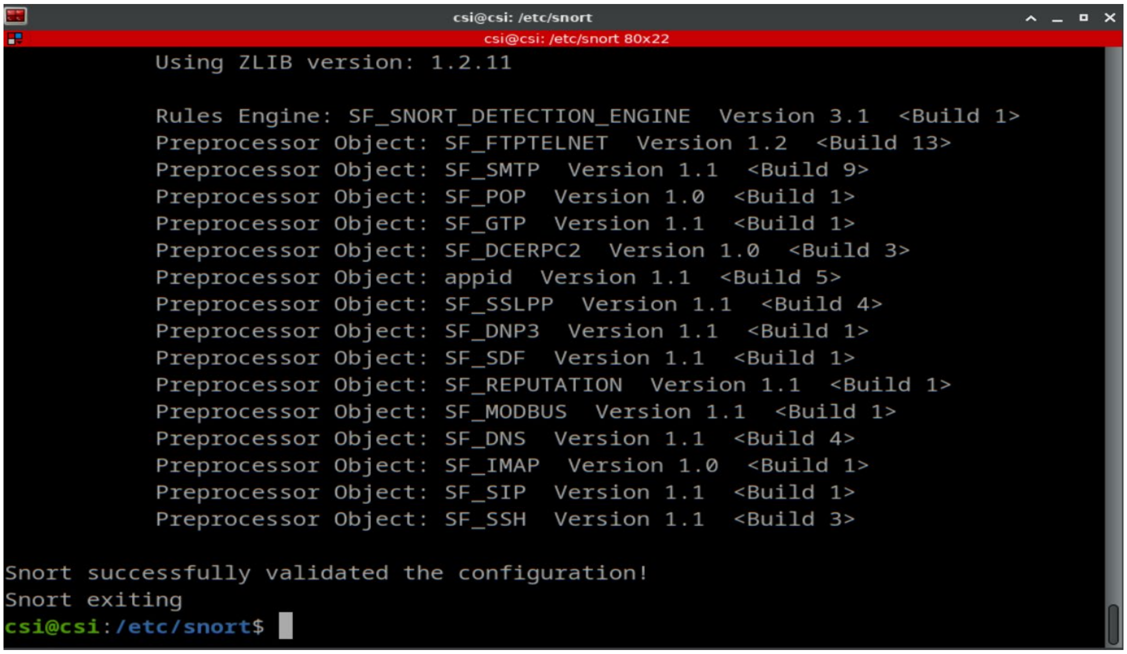
4. Now navigate to the /etc/snort directory and analyze all files.



```
csi@csi:~$ cd /etc/snort
csi@csi:/etc/snort$ ls
attribute_table.dtd    file_magic.conf   rules               threshold.conf
classification.config  gen-msg.map       snort.conf          unicode.map
community-sid-msg.map  reference.config  snort.debian.conf
csi@csi:/etc/snort$
```

5. Using below command test the snort.conf file its fine or not

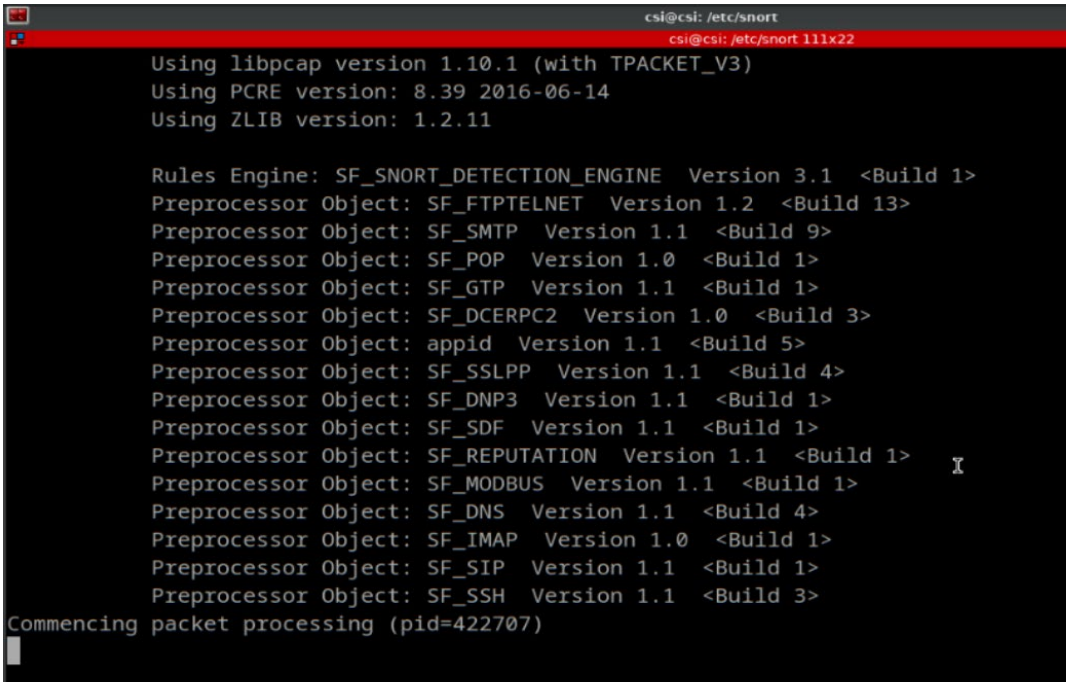
sudo snort -T -c /etc/snort/snort.conf



```
          Using ZLIB version: 1.2.11

          Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.1  <Build 1>
          Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
          Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
          Preprocessor Object: SF_POP  Version 1.0  <Build 1>
          Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
          Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
          Preprocessor Object: appid  Version 1.1  <Build 5>
          Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
          Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
          Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
          Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
          Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
          Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
          Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
          Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
          Preprocessor Object: SF_SSH  Version 1.1  <Build 3>

Snort successfully validated the configuration!
Snort exiting
csi@csi:/etc/snort$
```

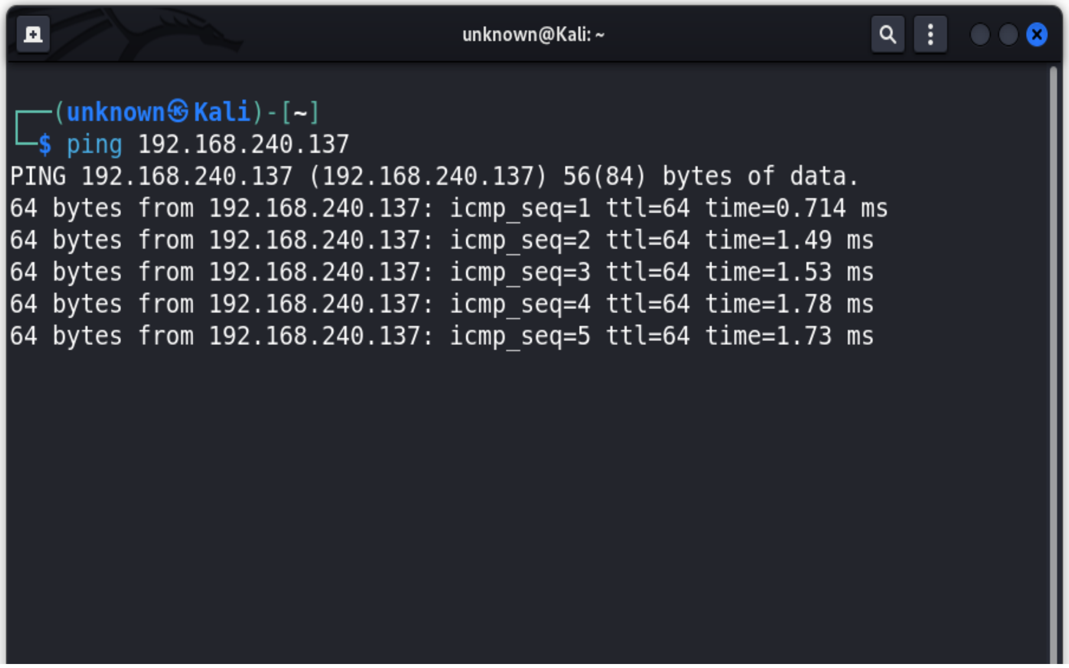
6. Now we need to capture packets in our CSI using snort. Type below command.

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



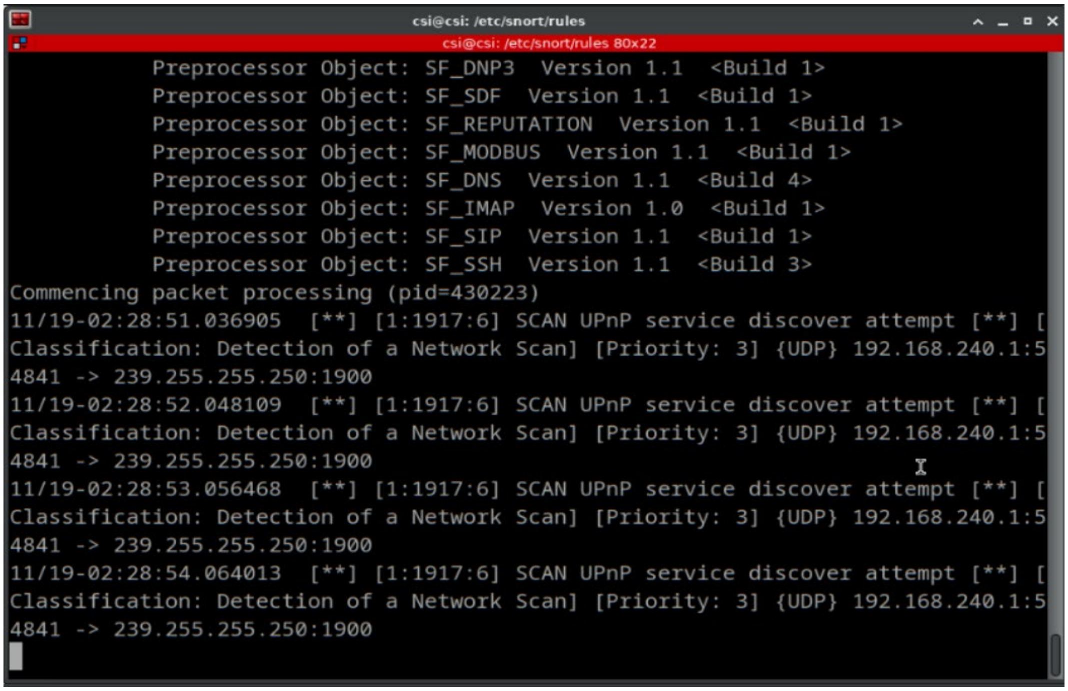
```
          Using libpcap version 1.10.1 (with TPACKET_V3)
          Using PCRE version: 8.39 2016-06-14
          Using ZLIB version: 1.2.11

          Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.1  <Build 1>
          Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
          Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
          Preprocessor Object: SF_POP  Version 1.0  <Build 1>
          Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
          Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
          Preprocessor Object: appid  Version 1.1  <Build 5>
          Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
          Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
          Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
          Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
          Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
          Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
          Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
          Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
          Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
Commencing packet processing (pid=422707)
```

7. Now go to attacker linux(kali Linux) open terminal and ping ip(CSI linux).



```
┌──(unknown㉿Kali)-[~]
└─$ ping 192.168.240.137
PING 192.168.240.137 (192.168.240.137) 56(84) bytes of data.
64 bytes from 192.168.240.137: icmp_seq=1 ttl=64 time=0.714 ms
64 bytes from 192.168.240.137: icmp_seq=2 ttl=64 time=1.49 ms
64 bytes from 192.168.240.137: icmp_seq=3 ttl=64 time=1.53 ms
64 bytes from 192.168.240.137: icmp_seq=4 ttl=64 time=1.78 ms
64 bytes from 192.168.240.137: icmp_seq=5 ttl=64 time=1.73 ms
```

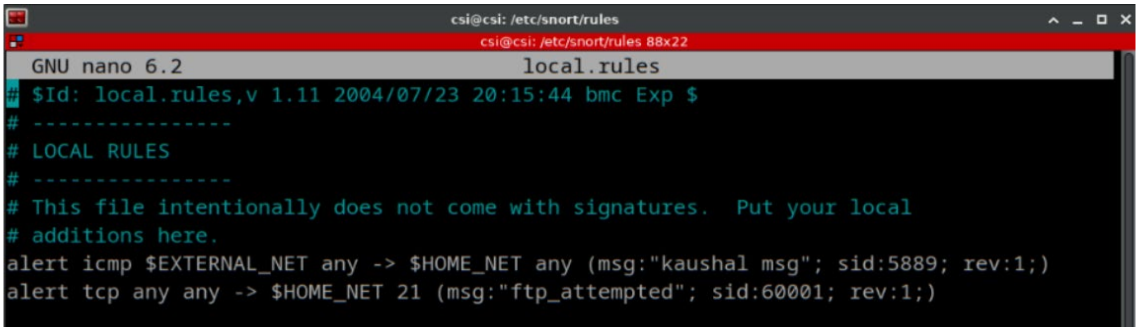
8. In CSI linux it shows detected packets from kali linux.



```
          Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
          Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
          Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
          Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
          Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
          Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
          Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
          Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
Commencing packet processing (pid=430223)
11/19-02:28:51.036905  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [
Classification: Detection of a Network Scan] [Priority: 3] {UDP} 192.168.240.1:5
4841 -> 239.255.255.250:1900
11/19-02:28:52.048109  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [
Classification: Detection of a Network Scan] [Priority: 3] {UDP} 192.168.240.1:5
4841 -> 239.255.255.250:1900
11/19-02:28:53.056468  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [
Classification: Detection of a Network Scan] [Priority: 3] {UDP} 192.168.240.1:5
4841 -> 239.255.255.250:1900
11/19-02:28:54.064013  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [
Classification: Detection of a Network Scan] [Priority: 3] {UDP} 192.168.240.1:5
4841 -> 239.255.255.250:1900
```

9. Now we add custom rules in /etc/snort/rules/locul.rules file in CSI linux shown in below screenshot and save it.

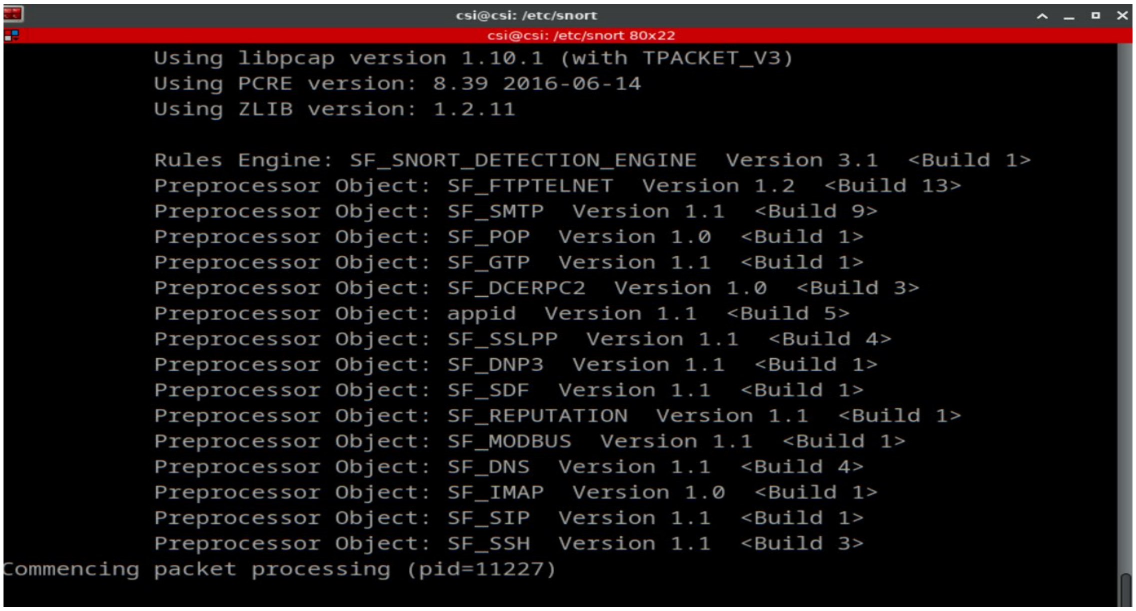
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"kaushal msg"; sid:5889; rev:1;)
alert tcp any any -> $HOME_NET 21 (msg:"ftp_attempted"; sid:60001; rev:1;)



```
  GNU nano 6.2                         local.rules
# $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 $EXTERNAL_NET any -> $HOME_NET any (msg:"kaushal msg"; sid:5889; rev:1;)
alert tcp any any -> $HOME_NET 21 (msg:"ftp_attempted"; sid:60001; rev:1;)
```

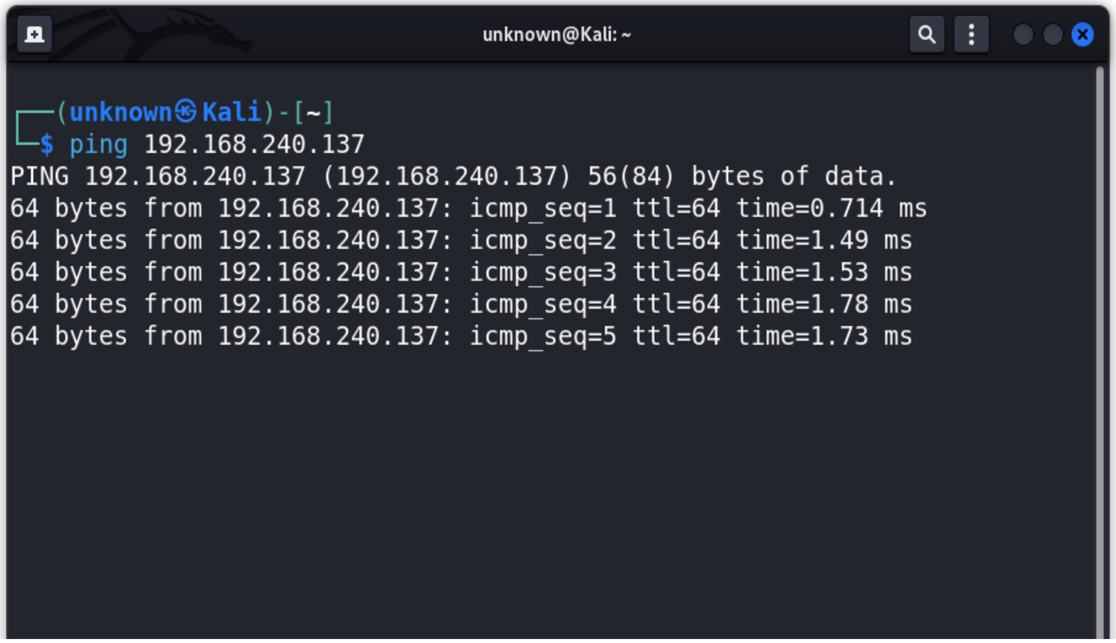
10. Now again test the snort.conf file and start capturing using below commands

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



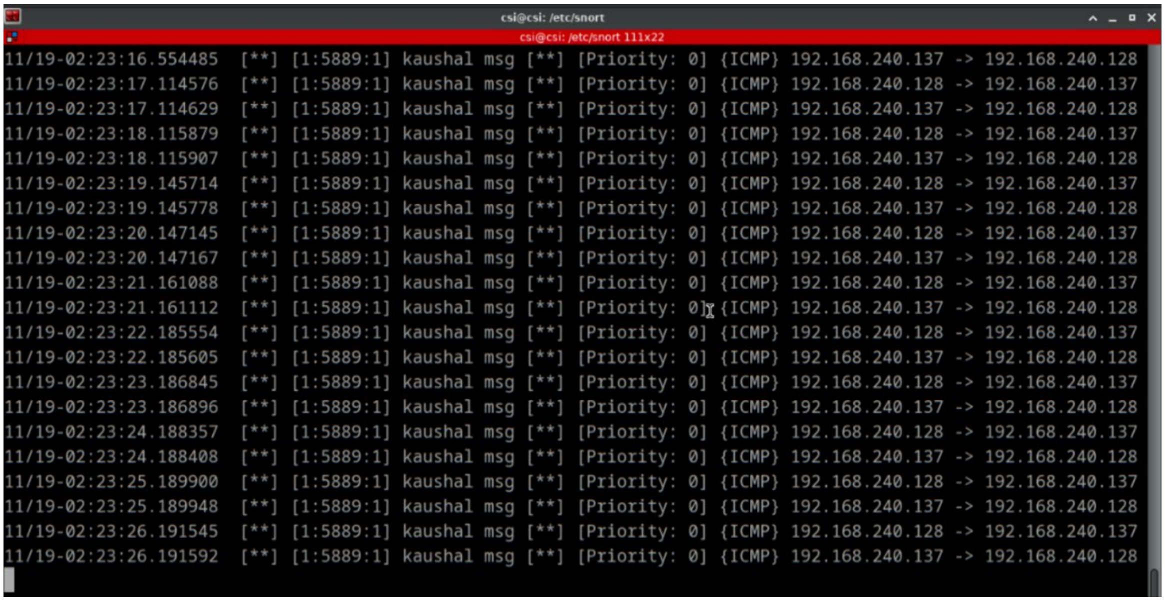
```
           Using libpcap version 1.10.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.1  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: appid  Version 1.1  <Build 5>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
Commencing packet processing (pid=11227)
```

11. Now go to kali machine and again ping ip(CSI linux)



```
┌──(unknown㉿Kali)-[~]
└─$ ping 192.168.240.137
PING 192.168.240.137 (192.168.240.137) 56(84) bytes of data.
64 bytes from 192.168.240.137: icmp_seq=1 ttl=64 time=0.714 ms
64 bytes from 192.168.240.137: icmp_seq=2 ttl=64 time=1.49 ms
64 bytes from 192.168.240.137: icmp_seq=3 ttl=64 time=1.53 ms
64 bytes from 192.168.240.137: icmp_seq=4 ttl=64 time=1.78 ms
64 bytes from 192.168.240.137: icmp_seq=5 ttl=64 time=1.73 ms
```

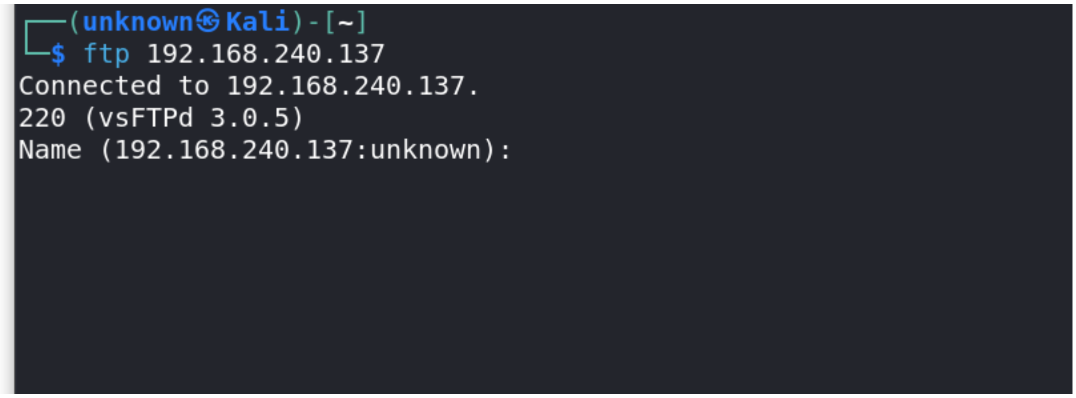
12. Now see the result in CSI linux (kaushal msg) message come up as we set in rules.



```
11/19-02:23:16.554485  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.137 -> 192.168.240.128
11/19-02:23:17.114576  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.128 -> 192.168.240.137
11/19-02:23:17.114629  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.137 -> 192.168.240.128
11/19-02:23:18.115879  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.128 -> 192.168.240.137
11/19-02:23:18.115907  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.137 -> 192.168.240.128
11/19-02:23:19.145714  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.128 -> 192.168.240.137
11/19-02:23:19.145778  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.137 -> 192.168.240.128
11/19-02:23:20.147145  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.128 -> 192.168.240.137
11/19-02:23:20.147167  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.137 -> 192.168.240.128
11/19-02:23:21.161088  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.128 -> 192.168.240.137
11/19-02:23:21.161112  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.137 -> 192.168.240.128
11/19-02:23:22.185554  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.128 -> 192.168.240.137
11/19-02:23:22.185605  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.137 -> 192.168.240.128
11/19-02:23:23.186845  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.128 -> 192.168.240.137
11/19-02:23:23.186896  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.137 -> 192.168.240.128
11/19-02:23:24.188357  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.128 -> 192.168.240.137
11/19-02:23:24.188408  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.137 -> 192.168.240.128
11/19-02:23:25.189900  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.128 -> 192.168.240.137
11/19-02:23:25.189948  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.137 -> 192.168.240.128
11/19-02:23:26.191545  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.128 -> 192.168.240.137
11/19-02:23:26.191592  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.240.137 -> 192.168.240.128
```

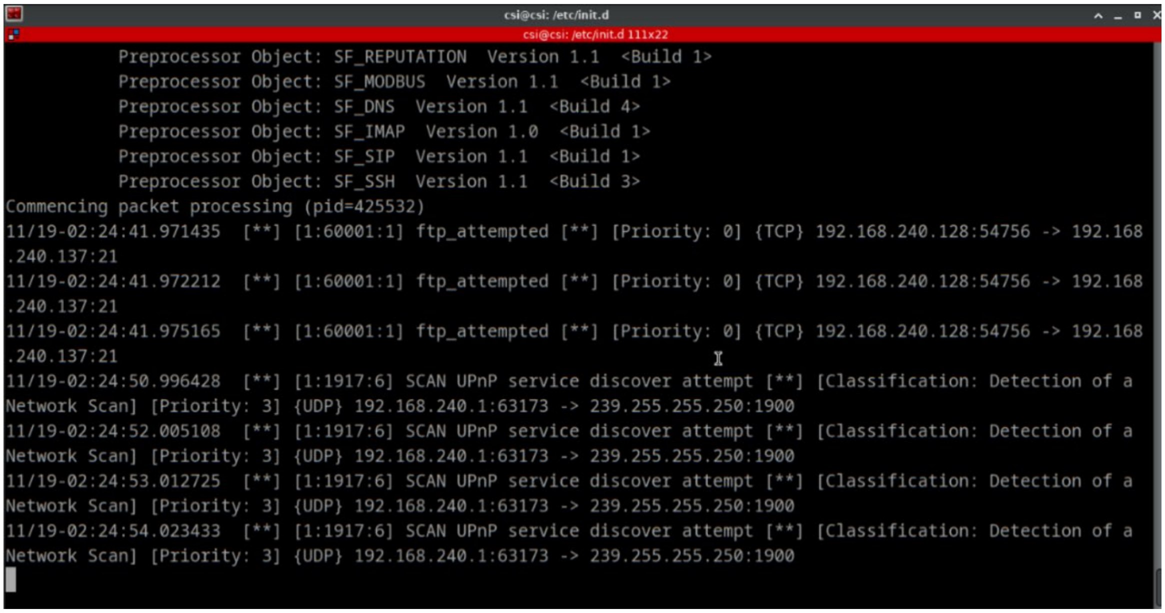
13. Now go to kali and start ftp session using below command.

ftp ip(CSI linux)



```
┌──(unknown㉿Kali)-[~]
└─$ ftp 192.168.240.137
Connected to 192.168.240.137.
220 (vsFTPd 3.0.5)
Name (192.168.240.137:unknown):
```

14. we can see on CSI linux that (ftp_attempted) message come up as we set rule



```
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
Commencing packet processing (pid=425532)
11/19-02:24:41.971435  [**] [1:60001:1] ftp_attempted [**] [Priority: 0] {TCP} 192.168.240.128:54756 -> 192.168
.240.137:21
11/19-02:24:41.972212  [**] [1:60001:1] ftp_attempted [**] [Priority: 0] {TCP} 192.168.240.128:54756 -> 192.168
.240.137:21
11/19-02:24:41.975165  [**] [1:60001:1] ftp_attempted [**] [Priority: 0] {TCP} 192.168.240.128:54756 -> 192.168
.240.137:21
11/19-02:24:50.996428  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a
Network Scan] [Priority: 3] {UDP} 192.168.240.1:63173 -> 239.255.255.250:1900
11/19-02:24:52.005108  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a
Network Scan] [Priority: 3] {UDP} 192.168.240.1:63173 -> 239.255.255.250:1900
11/19-02:24:53.012725  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a
Network Scan] [Priority: 3] {UDP} 192.168.240.1:63173 -> 239.255.255.250:1900
11/19-02:24:54.023433  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [Classification: Detection of a
Network Scan] [Priority: 3] {UDP} 192.168.240.1:63173 -> 239.255.255.250:1900
```

Result:

- Using snort tool we can detect and prevent incidents are coming in our machine.
- Also we can identify from where incidents are come from and Who is attacking our system or which process harm our system