

## Performing snort tool analysis

**Aim :-** To detect captured packets from an attacker (Kali) in Linux (Ubuntu) system.

1. We need two Linux VM install in our system with a host-only adapter.

Target Machine :- Ubuntu Linux

Attacking Machine :- Kali Linux

2. Boot up both Linux

3. In the target system we need to install Snort to detect the packets coming from the attacker machine.

Command :- `sudo apt-get install snort -y`

```
ubuntu@UbuntuVM:/etc/snort$ sudo apt-get install snort -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
snort is already the newest version (2.9.20-0+deb11u1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 86 not upgraded.
ubuntu@UbuntuVM:/etc/snort$
```

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

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

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

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

```
ubuntu@UbuntuVM:/etc/snort$ sudo snort -T -c /etc/snort/snort.conf
[sudo] password for ubuntu:
Running in Test mode

--== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
```

```

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 3.2 <Build 1>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: appid Version 1.1 <Build 5>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_S7COMMPLUS Version 1.0 <Build 1>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>

Total snort Fixed Memory Cost - MaxRss:105164
Snort successfully validated the configuration!
Snort exiting
ubuntu@UbuntuVM:/etc/snort/rules$

```

- Now we need to capture packets in our Ubuntu using snort. Type below command.

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

```

ubuntu@UbuntuVM:/etc/snort$ sudo snort -A console -c /etc/snort/snort.conf
[sudo] password for ubuntu:
Running in IDS mode

--== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP_PORTS' defined : [ 80:81 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4
343 4848 5250 6988 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:81
81 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555 ]
PortVar 'SHELLCODE_PORTS' defined : [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined : [ 1024:65535 ]
PortVar 'SSH_PORTS' defined : [ 22 ]
PortVar 'FTP_PORTS' defined : [ 21 2100 3535 ]
PortVar 'SIP_PORTS' defined : [ 5060:5061 5600 ]
PortVar 'FILE_DATA_PORTS' defined : [ 80:81 110 143 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 303
7 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 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002
55555 ]

```

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

```
(learnerprat@Pratik)-[~]
$ ping 192.168.226.129
PING 192.168.226.129 (192.168.226.129) 56(84) bytes of data.
64 bytes from 192.168.226.129: icmp_seq=1 ttl=64 time=0.678 ms
64 bytes from 192.168.226.129: icmp_seq=2 ttl=64 time=0.504 ms
64 bytes from 192.168.226.129: icmp_seq=3 ttl=64 time=0.533 ms
64 bytes from 192.168.226.129: icmp_seq=4 ttl=64 time=0.606 ms
64 bytes from 192.168.226.129: icmp_seq=5 ttl=64 time=0.552 ms
64 bytes from 192.168.226.129: icmp_seq=6 ttl=64 time=0.611 ms
^C
— 192.168.226.129 ping statistics —
6 packets transmitted, 6 received, 0% packet loss, time 5090ms
rtt min/avg/max/mdev = 0.504/0.580/0.678/0.057 ms

(learnerprat@Pratik)-[~]
$
```

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

```
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Commencing packet processing (pid=3415)
08/12-10:27:15.698989 [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.226.130 -> 192.168.226.129
08/12-10:27:15.699019 [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.226.129 -> 192.168.226.130
08/12-10:27:16.700646 [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.226.130 -> 192.168.226.129
08/12-10:27:16.700674 [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.226.129 -> 192.168.226.130
08/12-10:27:17.716844 [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.226.130 -> 192.168.226.129
08/12-10:27:17.716876 [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.226.129 -> 192.168.226.130
08/12-10:27:18.740987 [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.226.130 -> 192.168.226.129
08/12-10:27:18.741015 [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.226.129 -> 192.168.226.130
08/12-10:27:19.765101 [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.226.130 -> 192.168.226.129
08/12-10:27:19.765130 [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.226.129 -> 192.168.226.130
08/12-10:27:20.788782 [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.226.130 -> 192.168.226.129
08/12-10:27:20.788818 [**] [1:5889:1] kaushal msg [**] [Priority: 0] {ICMP} 192.168.226.129 -> 192.168.226.130
08/12-10:27:24.901131 [**] [1:5889:1] kaushal msg [**] [Priority: 0] {IPV6-ICMP} fe80::20c:29ff:fe46:8160 -> f
f02::2
```

9. Now we add custom rules in /etc/snort/rules/local.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;)
```

```
ubuntu@UbuntuVM: /etc/snort/rules
GNU nano 7.2 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 go to kali and start the ftp session using the below command.  
**ftp {ip of Ubuntu}**

```
(learnerprat@Pratik)-[~]  
$ ftp 192.168.226.129  
ftp: Can't connect to `192.168.226.129:21': Connection refused  
ftp: Can't connect to `192.168.226.129:ftp'  
ftp> █
```

11. we can see on ubuntu linux that (ftp\_attempted) message come up as we set rule

```
Network Scan] [Priority: 3] {UDP} 192.168.226.1:54457 -> 239.255.255.250:1900  
08/12-10:37:13.675520  [**] [1:5889:1] kaushal msg [**] [Priority: 0] {IPV6-ICMP} fe80::20c:29ff:feae:3c41 -> f  
f02::2  
08/12-10:37:52.493711  [**] [1:60001:1] ftp_attempted [**] [Priority: 0] {TCP} 192.168.226.130:52308 -> 192.168  
.226.129:21  
08/12-10:38:37.870348  [**] [1:60001:1] ftp_attempted [**] [Priority: 0] {TCP} 192.168.226.130:52868 -> 192.168  
.226.129:21
```

Result :-

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