

Cyber Security and Snort IDS/IPS Implementation Report

Internship Title:

Cyber Security and Snort IDS/IPS installation, configuration, detect and validate the rules and alerts

Prepared By: VIJAY S R

Date : 28/11/2025

1. Introduction

This report provides foundational knowledge of cybersecurity and a comprehensive, step-by-step guide to installing, configuring, and operationalizing the Snort Intrusion Detection and Prevention System (IDS/IPS) on an Ubuntu machine. It also includes detailed instructions for custom rules creation and validation of triggered alerts.

2. Cyber Security Basics

Cyber security refers to the practice of protecting systems, networks, data, and applications from digital threats. These threats include malware, unauthorized access, data breaches, phishing attacks, Denial of Service (DoS), and more.

2.1 Key Cyber Security Concepts

- Confidentiality: Ensuring only authorized users have access to data.
- Integrity: Ensuring data is accurate and unchanged.
- Availability: Ensuring systems and data are accessible when required.
- Threat: Any circumstance that can exploit vulnerabilities.
- Vulnerability: A weakness in system security.
- Exploit: A method used by attackers to leverage vulnerabilities.

2.2 Key Security Domains

- Network Security
- Endpoint Security
- Application Security
- Identity and Access Management (IAM)
- Security Operations

3. Snort IDS/IPS Overview

Snort is an open-source, signature-based network intrusion detection and prevention system. It performs real-time packet inspection and traffic analysis.

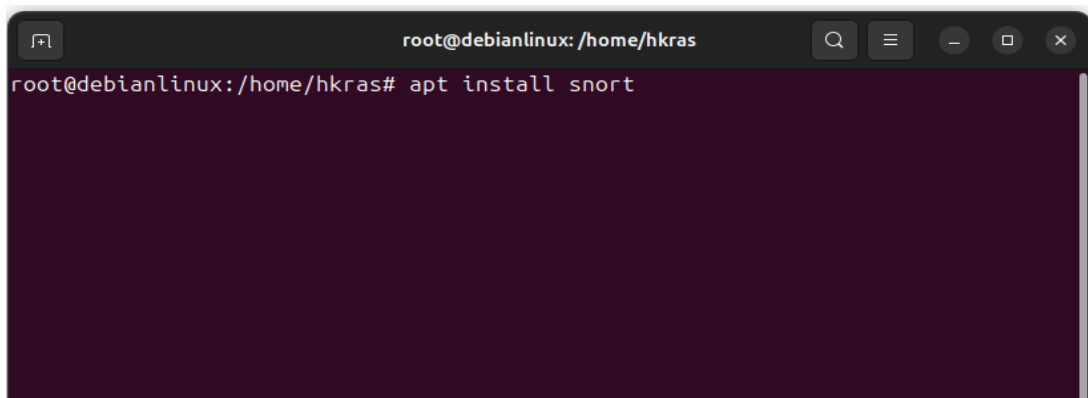
3.1 Snort Modes

- Sniffer Mode
- Packet Logger Mode
- Network IDS Mode
- IPS Mode

4. Snort Installation on Ubuntu

Installation steps for Snort 2.9.20 . Open the terminal type

`sudo apt install snort`



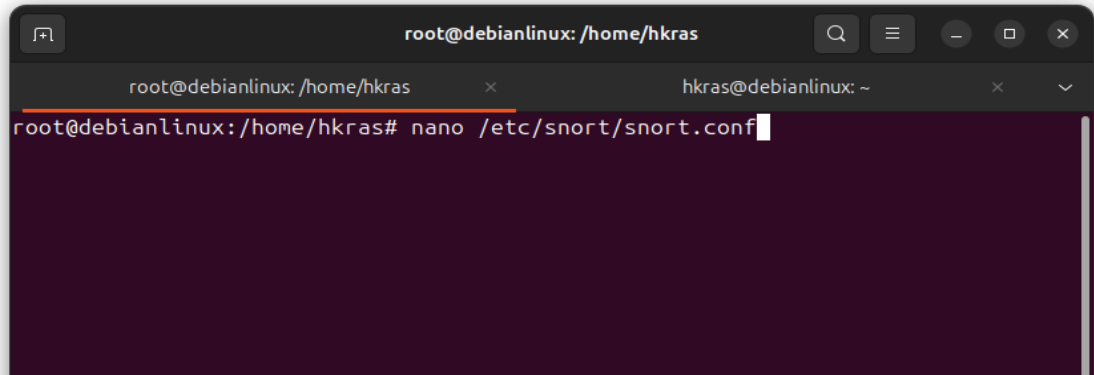
Then check the version:- `sudo snort -V` or `snort --version`

5. Snort Configuration

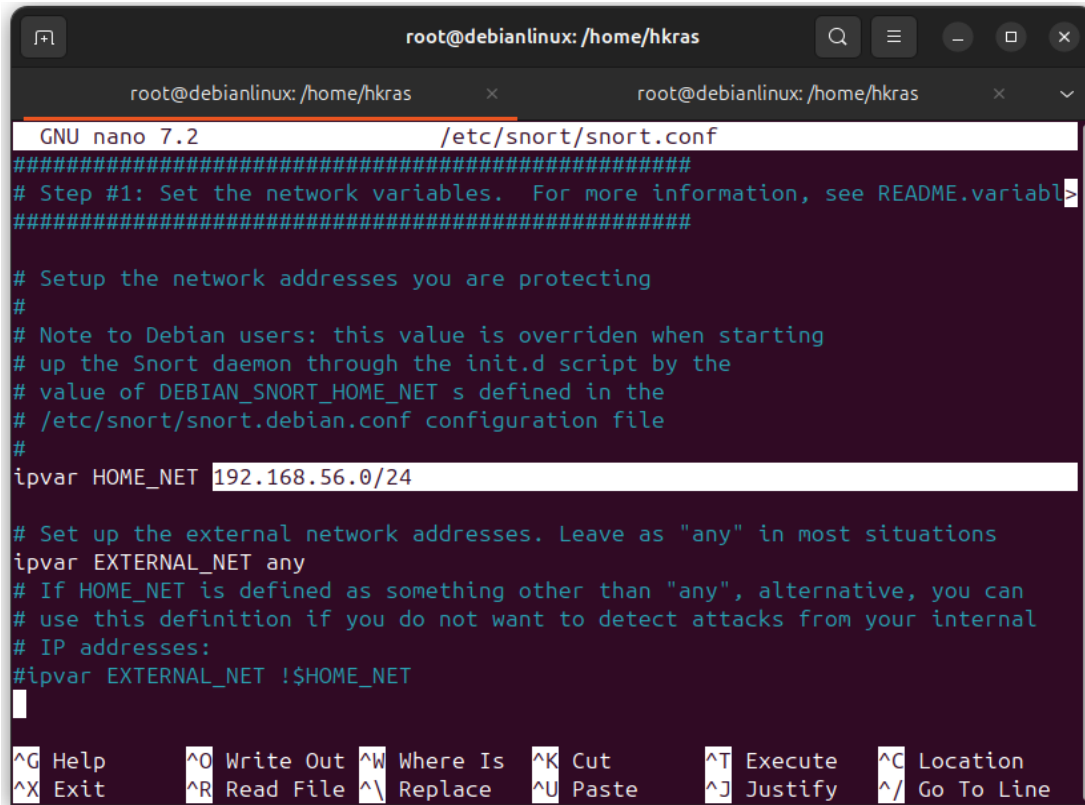
Configuration of directories, rule paths, snort.conf tuning, and enabling local rules.

You should do some changes in snort.conf file.

```
sudo nano /etc/snort/snort.conf
```



We should configure our entire network with snort. So you make change your conf file like the below image



```
GNU nano 7.2 /etc/snort/snort.conf
#####
# Step #1: Set the network variables.  For more information, see README.variables
#####

# Setup the network addresses you are protecting
#
# Note to Debian users: this value is overridden 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 192.168.56.0/24

# 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

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

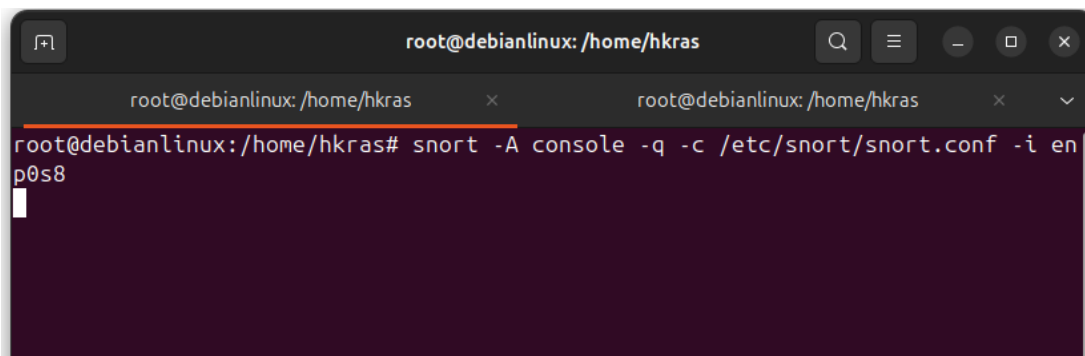
Add your entire network ip range in ipvar HOME_NET . See the above image for your reference and your ip range may vary depends on your internet service provider. So check your ip address and range using ip -a command on ubuntu.

6. Running Snort in IDS and IPS Modes

Commands for:

- IDS mode

`sudo snort -A console -q -c /etc/snort/snort.conf -i enp0s8`



```
root@debianlinux: /home/hkras
root@debianlinux:/home/hkras# snort -A console -q -c /etc/snort/snort.conf -i enp0s8
```

Interface may change depends on your vms like eth0, eth1, eth2 this

- enable and run Inline IPS mode using NFQUEUE and iptables redirection using following commands

Enable NFQUEUE:

```
sudo apt install -y libnetfilter-queue-dev
```

Run snort inline:

```
sudo snort -Q -daq nfq -c /etc/snort/snort.conf -i enp0s8
```

Add iptables rules to divert packets:

```
sudo iptables -I INPUT -j NFQUEUE --queue-num 0
```

```
sudo iptables -I OUTPUT -j NFQUEUE --queue-num 0
```

7. Snort Rules Creation

7.1 Rule Structure

Action protocol source_ip source_port -> dest_ip dest_port (options)

A screenshot of a code editor window titled 'local.rules' with a subtitle '/etc/snort/rules'. The editor shows a list of Snort rules. The first rule is for SSH login attempts, followed by ICMP ping requests, and then six rules for SQL injection attempts with various payloads. Each rule line is color-coded: 'alert' is red, 'tcp' and 'icmp' are blue, and other keywords are black. The rules are as follows:

```
# SSH login Rules
alert tcp any any -> $HOME_NET 22 (msg:"SSH Login Attempt Detected"; sid:1000002; rev:1;)

# ICMP Ping Req Rules
alert icmp any any -> $HOME_NET any (msg:"ping request Detected"; sid:1000001; rev:1;)

# SQL Injection Rule 1
alert tcp any any -> $HOME_NET 80 (msg:"SQL Injection Attempt"; content:"' or '1'='1"; nocase; sid:1000003; rev:1;)

# SQL Injection Rule 2
alert tcp any any -> $HOME_NET 80 (msg:"SQL Injection Attempt"; content:" ' order by *"; nocase; sid:1000004; rev:1;)

# SQL Injection Rule 3
alert tcp any any -> $HOME_NET 80 (msg:"SQL Injection Attempt"; content:"union"; nocase; sid:1000005; rev:1;)

# SQL Injection Rule 4
alert tcp any any -> $HOME_NET 80 (msg:"SQL Injection Attempt"; content:"select"; nocase; sid:1000006; rev:1;)

# SQL Injection Rule 5
alert tcp any any -> $HOME_NET 80 (msg:"SQL Injection Attempt"; content:"' and '1'='1"; nocase; sid:1000007; rev:1;)

# SQL Injection Rule 6
alert tcp any any -> $HOME_NET 80 (msg:"SQL Injection Attempt"; content:"version()"; nocase; sid:1000008; rev:1;)
```

```
Nov 27 01:55
Open ▾
community.rules
local.rules
/etc/snort/rules
local.rules x

# SQL Injection Rule 4
alert tcp any any -> $HOME_NET 80 (msg:"SQL Injection Attempt"; content:"select"; nocase; sid:1000006; rev:1;)

# SQL Injection Rule 5
alert tcp any any -> $HOME_NET 80 (msg:"SQL Injection Attempt"; content:"'" and '1'='1"; nocase; sid:1000007; rev:1;)

# SQL Injection Rule 6
alert tcp any any -> $HOME_NET 80 (msg:"SQL Injection Attempt"; content:"version()"; nocase; sid:1000008; rev:1;)

# SQL Injection Rule 7
alert tcp any any -> $HOME_NET 80 (msg:"SQL Injection Attempt"; content:"sleep"; nocase; sid:1000009; rev:1;)

# FTP Anonymous login Rule
alert tcp any any -> $HOME_NET 21 (msg:"FTP Anonymous Login Attempt"; content:"USER anonymous"; sid:1000010; rev:1;)
```

7.2 Example Rules

- ICMP Ping Detection Rule

```
alert icmp any any -> $HOME_NET any (msg:"icmp ping detected"; sid:1000001; rev:1;)
```

- FTP Anonymous Login Detection

```
alert tcp any any -> $HOME_NET 21 (msg:"FTP Anonymous Login Attempt"; content:"USER anonymous"; sid:1000002; rev:1;)
```

- SSH Bruteforce attempt Rule

```
alert tcp any any -> $HOME_NET 22 (msg:"SSH Bruteforce Attempt"; threshold:type both, track by_src, count 5, seconds 60; sid:1000003; rev:1;)
```

- SQL Injection Detection Rule

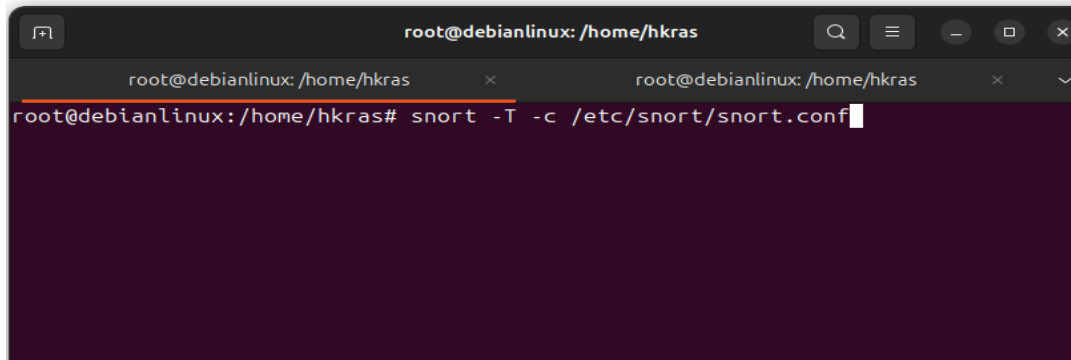
```
alert tcp any any -> $HOME_NET 80 (msg:"SQL Injection Attempt"; content:"select"; nocase; sid:1000004; rev:1;)
```

```
Alert tcp any any -> $HOME_NET 80 (msg:"SQL Injection Attempt"; content:"union"; nocase; sid:1000005; rev:1;)
```

8. Validating Snort Rules (testing Alerts)

8.1 validate Configuration

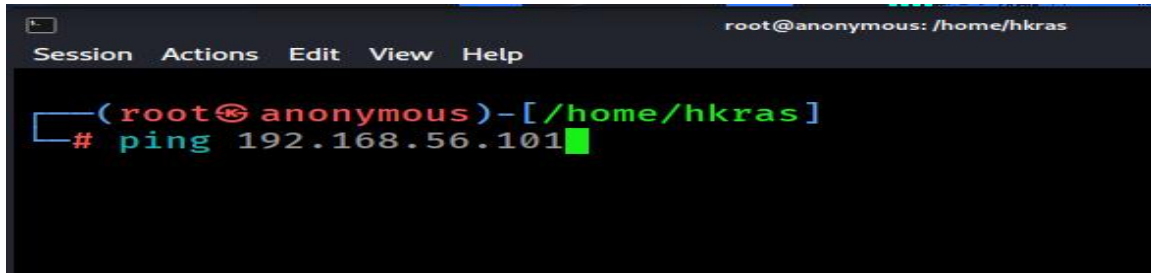
`sudo snort -T -c /etc/snort/snort.conf`



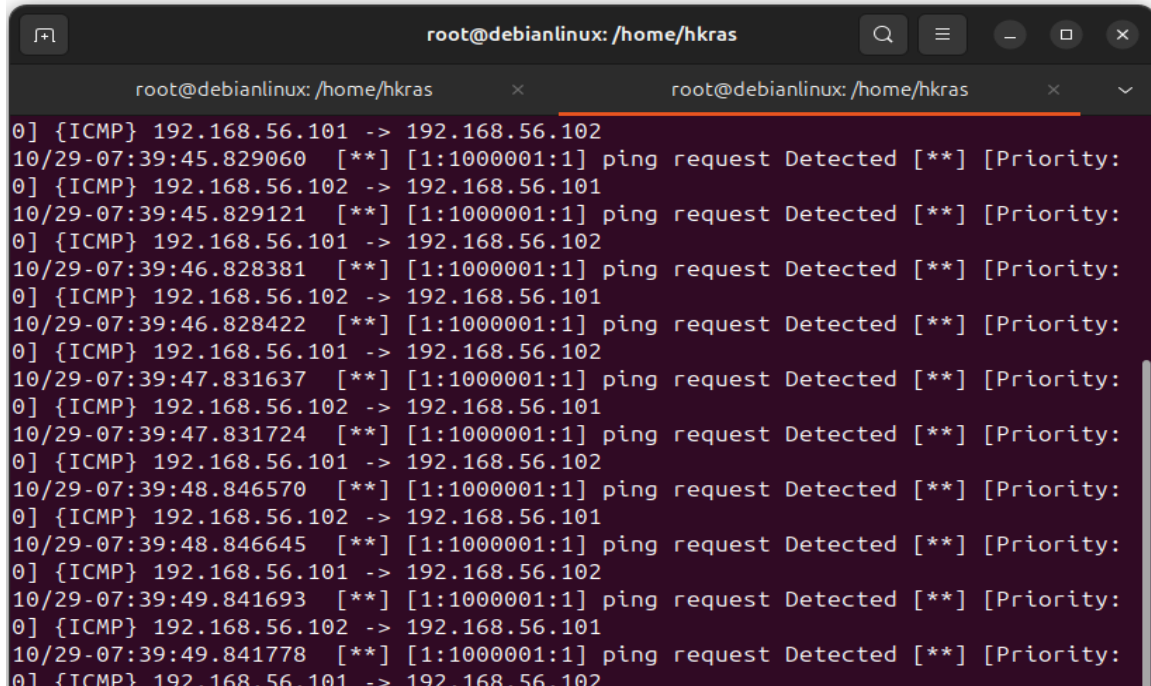
```
root@debianlinux: /home/hkras
root@debianlinux: /home/hkras# snort -T -c /etc/snort/snort.conf
```

8.2 Testing ICMP Rule

Ping 192.168.x.x (type the snort installed machine ip address)



```
root@anonymous: /home/hkras
Session Actions Edit View Help
# ping 192.168.56.101
```

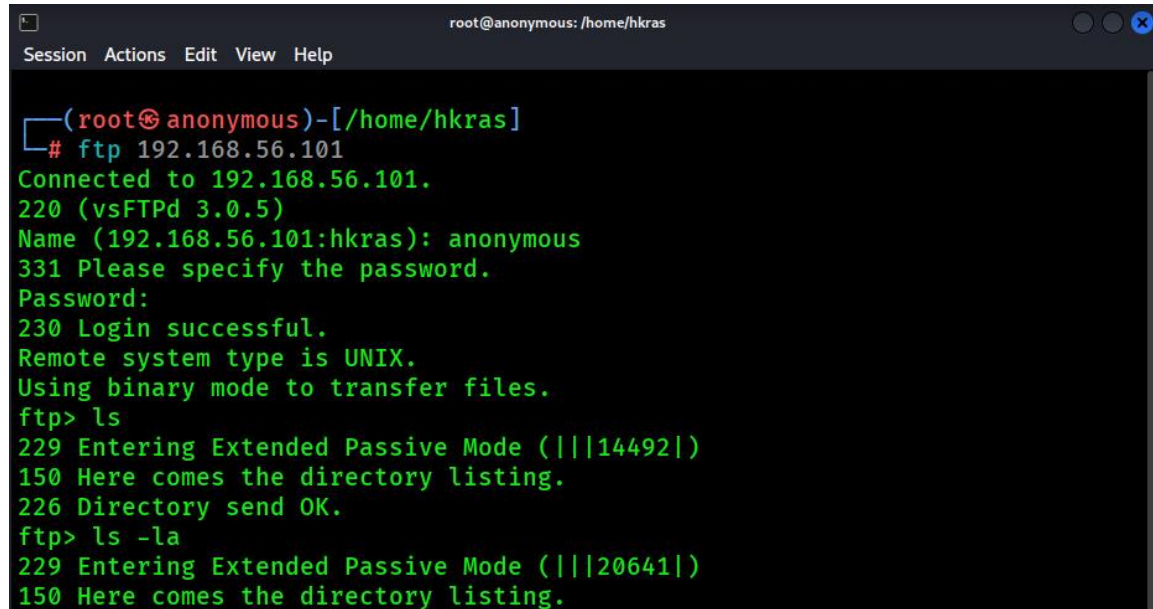


```
root@debianlinux: /home/hkras
root@debianlinux: /home/hkras# snort -T -c /etc/snort/snort.conf
0] {ICMP} 192.168.56.101 -> 192.168.56.102
10/29-07:39:45.829060  [**] [1:1000001:1] ping request Detected [**] [Priority:
0] {ICMP} 192.168.56.102 -> 192.168.56.101
10/29-07:39:45.829121  [**] [1:1000001:1] ping request Detected [**] [Priority:
0] {ICMP} 192.168.56.101 -> 192.168.56.102
10/29-07:39:46.828381  [**] [1:1000001:1] ping request Detected [**] [Priority:
0] {ICMP} 192.168.56.102 -> 192.168.56.101
10/29-07:39:46.828422  [**] [1:1000001:1] ping request Detected [**] [Priority:
0] {ICMP} 192.168.56.101 -> 192.168.56.102
10/29-07:39:47.831637  [**] [1:1000001:1] ping request Detected [**] [Priority:
0] {ICMP} 192.168.56.102 -> 192.168.56.101
10/29-07:39:47.831724  [**] [1:1000001:1] ping request Detected [**] [Priority:
0] {ICMP} 192.168.56.101 -> 192.168.56.102
10/29-07:39:48.846570  [**] [1:1000001:1] ping request Detected [**] [Priority:
0] {ICMP} 192.168.56.102 -> 192.168.56.101
10/29-07:39:48.846645  [**] [1:1000001:1] ping request Detected [**] [Priority:
0] {ICMP} 192.168.56.101 -> 192.168.56.102
10/29-07:39:49.841693  [**] [1:1000001:1] ping request Detected [**] [Priority:
0] {ICMP} 192.168.56.102 -> 192.168.56.101
10/29-07:39:49.841778  [**] [1:1000001:1] ping request Detected [**] [Priority:
0] {ICMP} 192.168.56.101 -> 192.168.56.102
```

Ping request detected and the alert triggered

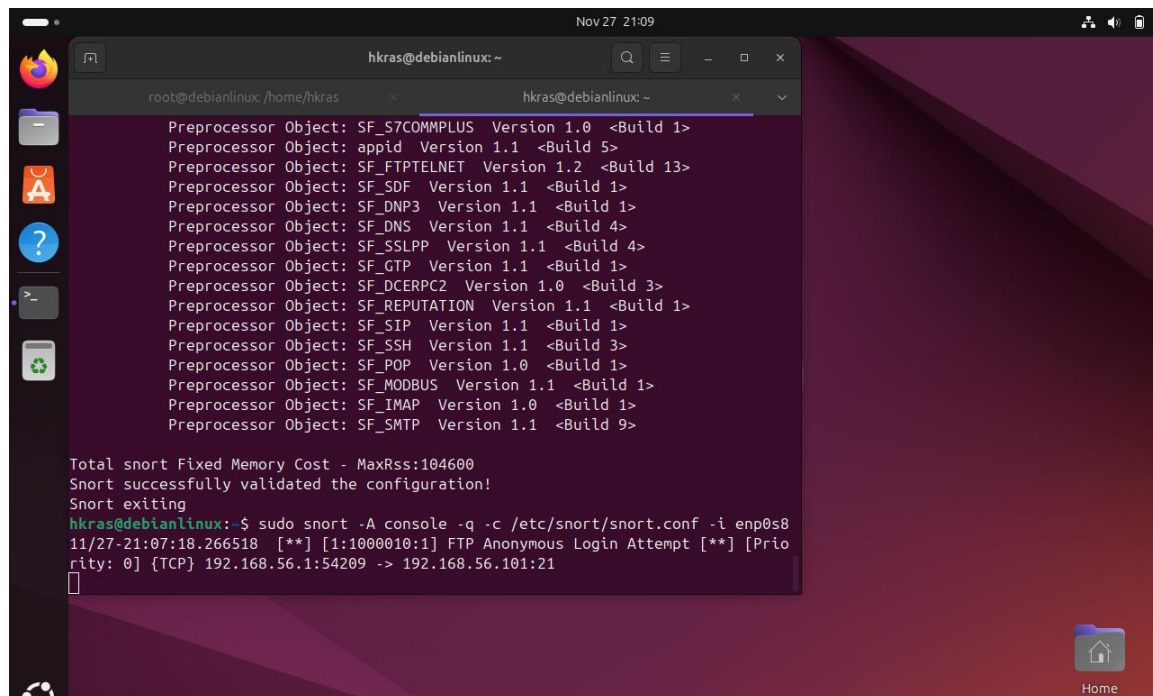
8.3 Testing FTP Anonymous Rule

ftp 192.168.x.x (snort installed machine ip)



```
root@anonymous: /home/hkras
Session Actions Edit View Help

(root@anonymous)-[/home/hkras]
# ftp 192.168.56.101
Connected to 192.168.56.101.
220 (vsFTPd 3.0.5)
Name (192.168.56.101:hkras): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||14492|)
150 Here comes the directory listing.
226 Directory send OK.
ftp> ls -la
229 Entering Extended Passive Mode (|||20641|)
150 Here comes the directory listing.
```



```
hkras@debianlinux: ~
root@debianlinux: /home/hkras
Preprocessor Object: SF_S7COMPLUS 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_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>

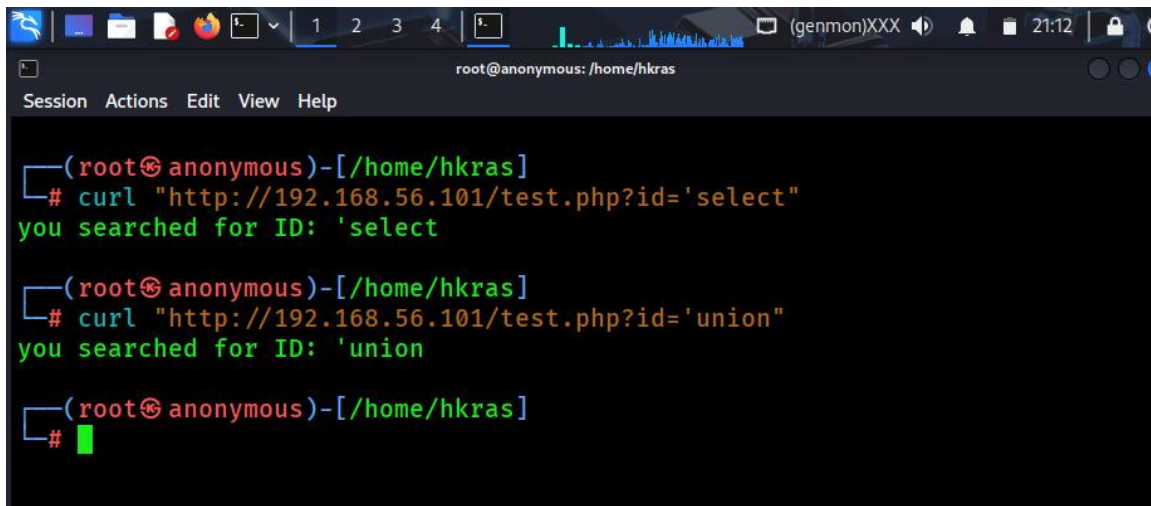
Total snort Fixed Memory Cost - MaxRss:104600
Snort successfully validated the configuration!
Snort exiting
hkras@debianlinux:~$ sudo snort -A console -q -c /etc/snort/snort.conf -i enp0s8
11/27-21:07:18.266518  [**] [1:1000010:1] FTP Anonymous Login Attempt [**] [Priority: 0] {TCP} 192.168.56.1:54209 -> 192.168.56.101:21
```

FTP anonymous login attempt detected and the alert triggered.

8.4 Testing SQL Injection Rule

Curl "http://192.168.56.101/test.php?id='union'"

Curl "http://192.168.56.101/test.php?id='select'"

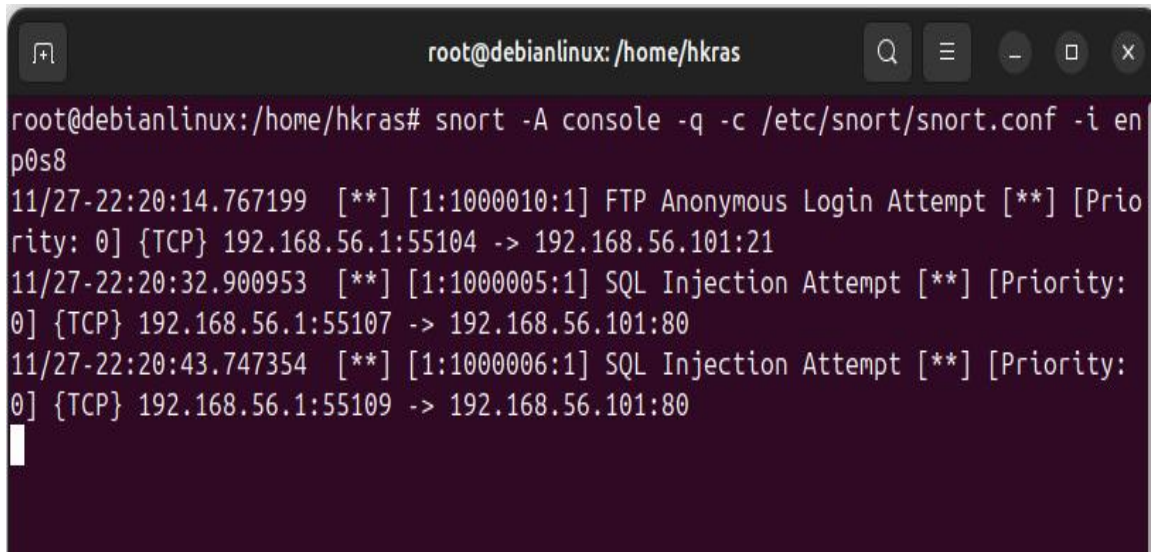


```
root@anonymous: /home/hkras
Session Actions Edit View Help

(root@anonymous)-[/home/hkras]
# curl "http://192.168.56.101/test.php?id='select'"
you searched for ID: 'select'

(root@anonymous)-[/home/hkras]
# curl "http://192.168.56.101/test.php?id='union'"
you searched for ID: 'union'

(root@anonymous)-[/home/hkras]
#
```



```
root@debianlinux: /home/hkras
root@debianlinux:/home/hkras# snort -A console -q -c /etc/snort/snort.conf -i en
p0s8
11/27-22:20:14.767199  [**] [1:1000010:1] FTP Anonymous Login Attempt [**] [Prio
rity: 0] {TCP} 192.168.56.1:55104 -> 192.168.56.101:21
11/27-22:20:32.900953  [**] [1:1000005:1] SQL Injection Attempt [**] [Priority:
0] {TCP} 192.168.56.1:55107 -> 192.168.56.101:80
11/27-22:20:43.747354  [**] [1:1000006:1] SQL Injection Attempt [**] [Priority:
0] {TCP} 192.168.56.1:55109 -> 192.168.56.101:80
```

SQL Injection attempt detected and alert triggered in Snort IDS Mode.

9. Viewing Alerts

Snort logs alerts to:

```
/var/log/snort/alert
```

To view:

```
sudo tail -f /var/log/snort/alert
```

10. Conclusion

This report covered the cybersecurity fundamentals and provided a complete guide to deploying Snort IDS/IPS on Ubuntu, including installation, configuration, rule authoring, and alert validation. By following these instructions, you can use snort as an effective network monitoring and intrusion detection tool on your environment.