NETWORK INTRUSION DETECTION SYSTEM IMPLEMENTATION & ANALYSIS USING SNORT

Prepared by,

Vandana Ashokan
Gourav Swaroop
Gopalkrishna
Maria Jose

CONTENTS

I. INTRODUCTION

1. IMPORTANCE OF NETWORK INTRUSION DETECTION SYSTEM (NIDS)

Network security has become a critical aspect of modern digital infrastructure as organizations face an ever-increasing number of sophisticated cyber threats. Attackers frequently employ techniques such as port scanning, brute force attempts, and SQL injections to compromise systems and gain unauthorized access. To counter these threats, security teams rely on Network Intrusion Detection Systems (NIDS), which continuously monitor network traffic to identify and alert on suspicious or malicious activity.

A NIDS functions by analyzing packets that traverse the network, comparing them against predefined signatures and behavioral patterns to detect anomalies. Unlike host-based systems, which monitor activities on individual machines, NIDS provides a broader perspective of the network environment, making it an essential component in detecting attacks at an early stage.

Snort, an open-source NIDS, is widely adopted for its flexibility, strong community support, and ability to detect diverse intrusions. It works in real time to capture traffic, generate alerts, and provide valuable insights for incident response. In this project, Snort was deployed in a virtual SOC lab to simulate real-world attacks and demonstrate the role of NIDS in enhancing proactive threat detection and network security.

2. OBJECTIVE

The main objective of this project is to design and implement a Network Intrusion Detection System (NIDS) using Snort in a virtual SOC environment. The project specifically aims to:

- Implement Snort as a Network Intrusion Detection System (NIDS) in a virtual SOC environment. Write and configure Snort rules for detecting attacks.
- Simulate real-world attacks such as port scans, brute force attempts and malware beaconing.

- Analyze generated alerts to evaluate Snort's effectiveness as a NIDS.

II. PREREQUISITES

Before implementing a Network Intrusion Detection System (NIDS) using Snort in a virtual SOC environment, certain prerequisites must be met. These involve ensuring adequate hardware, compatible software, and proper network configuration.

Hardware Requirements:

- Processor: 1.5 GHz Dual Core or higher
- RAM: Minimum 4 GB (8 GB recommended)
- Storage: 20 GB or more
- Network: Stable Internet Connection

Software Requirements

- Virtualization Platform: Oracle VirtualBox (for creating and managing VMs)
- Attacker Machine: Kali Linux (with penetration testing tools like Nmap, Hydra)
- Victim Machine: Linux distribution (with Snort NIDS installed and configured)
- NIDS Tool: Snort (open-source intrusion detection system)

III. TOOLS AND TECHNOLOGIES USED

1) **Virtualization Platform:** Oracle VirtualBox – for creating and managing virtual machines
2) **Attacker Machine:** Kali Linux VM with penetration testing tools
3) **Victim Machine:** Linux VM with Snort NIDS installed
4) **Snort IDS Engine:** For real-time packet analysis, logging, and intrusion detection
5) **Attack and Testing Tools:**
    a) **Nmap** → Reconnaissance and port scanning (SYN, FIN, Xmas)
    b) **Hydra** → Brute-force SSH/FTP login attempts
    c) **hping3** → Simulated malware beacon traffic
6) **Log Analysis:** Snort logs and alert files

IV. VIRTUAL SOC LAB SETUP

1. INSTALLING VIRTUALBOX AND CREATING VIRTUAL MACHINES

VirtualBox is used to create and manage virtual machines. Two VMs were created:

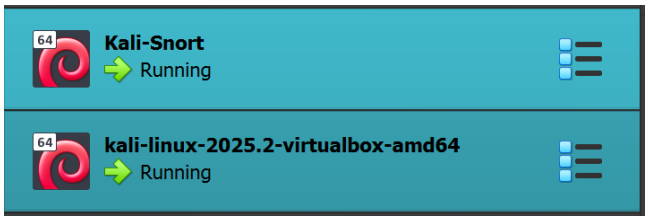
1) Kali Linux: Attacker machine.
2) Kali - Snort: Victim machine with Snort NIDS installed.

2. CONFIGURING NETWORK INTERFACES

The network was configured in Bridge Adapter mode to allow communication between the host, attacker, and victim machines. Static IP addresses were assigned for consistency.

3. INSTALLING KALI LINUX (ATTACKER) AND LINUX VM (VICTIM WITH SNORT)

- Installed Kali Linux as the attacker machine for running penetration tests.
- Installed another Linux distribution as the victim machine for Snort deployment.



V. INSTALLATION AND CONFIGURATION OF SNORT

1. INSTALLATION

The victim machine was updated, and Snort was installed:

sudo apt update && sudo apt upgrade -y

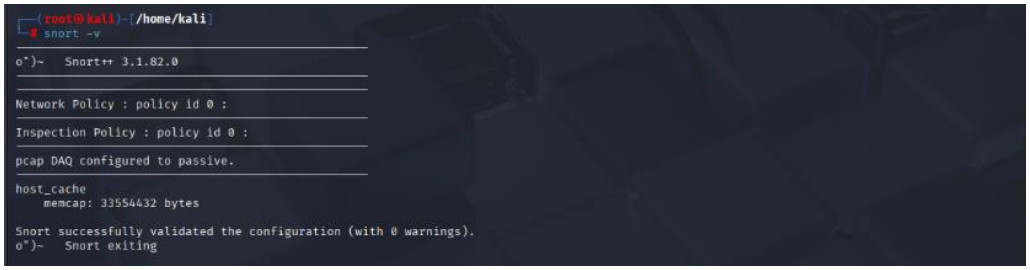
sudo apt install snort -y

```
┌──(root㉿kali)-[/home/kali]
└─# sudo apt update&& sudo apt upgrade -y
Hit:1 http://http.kali.org/kali kali-rolling InRelease
458 packages can be upgraded. Run 'apt list --upgradable' to see them.
The following packages were automatically installed and are no longer required:
  python3-packaging-whl  python3-pyinstaller-hooks-contrib  python3-wheel-whl
Use 'sudo apt autoremove' to remove them.

Upgrading:
  7zip                      libabsl20240722           libqt5sql5-sqlite         php8.4-cli
  adwaita-icon-theme        libapache2-mod-php8.4     libqt5sql5t64             php8.4-common
  apache2                   libapt-pkg7.0             libqt5test5t64            php8.4-mysql
  apache2-bin               libarchive13t64           libqt5widgets5t64         php8.4-opcache
  apache2-data              libavif16                 libqt5xml5t64             php8.4-readline
  apache2-utils             libblas3                  libqt6core6t64            plymouth
  apt                       libblkid1                 libqt6dbus6               plymouth-label
  apt-utils                 libblockdev-crypto3       libqt6gui6                preview-latex-style
  bash                      libblockdev-fs3           libqt6network6            procps
  bind9-dnsutils            libblockdev-loop3         libqt6opengl6             pyqt6-dev-tools
  bind9-host                libblockdev-mdraid3       libqt6openglwidgets6      python-matplotlib-data
  bind9-libs                libblockdev-nvme3         libqt6printsupport6       python3
  bluez                     libblockdev-part3         libqt6sql6                python3-bitstruct
  bluez-hcidump             libblockdev-swap3         libqt6sql6-sqlite         python3-bs4
  bluez-obexd               libblockdev-utils3        libqt6test6               python3-cffi
  bsdextrautils             libblockdev3              libqt6widgets6            python3-cffi-backend
  bsdutils                  libbluetooth3             libqt6xml6                python3-dev
  burpsuite                 libbson-1.0-0t64          librpm10                  python3-django
  busybox                   libc-bin                  librpmbuild10             python3-jq
  chromium                  libc-dev-bin              librpmio10                python3-ldb
  chromium-common           libc-l10n                 librpmsign10              python3-matplotlib
```

```
┌──(root㉿kali)-[/home/kali]
└─# sudo apt install snort -y
The following packages were automatically installed and are no longer required:
  python3-packaging-whl  python3-pyinstaller-hooks-contrib  python3-wheel-whl
Use 'sudo apt autoremove' to remove them.

Installing:
  snort

Installing dependencies:
  libdaq3  libestr0  libfastjson4  liblognorm5  oinkmaster  rsyslog  snort-common  snort-common-libraries  snort-rules-default

Suggested packages:
  rsyslog-doc      rsyslog-mongodb       rsyslog-hiredis     rsyslog-docker      | rsyslog-gnutls  snort-doc
  rsyslog-mysql    rsyslog-elasticsearch rsyslog-snmp        rsyslog-clickhouse  rsyslog-gssapi
  | rsyslog-pgsql  rsyslog-kafka         rsyslog-kubernetes  rsyslog-openssl     rsyslog-relp

Summary:
  Upgrading: 0, Installing: 10, Removing: 0, Not Upgrading: 0
  Download size: 3,680 kB
  Space needed: 15.8 MB / 12.8 GB available

Get:1 http://http.kali.org/kali kali-rolling/main amd64 snort-common-libraries amd64 3.1.82.0-0kali1+b1 [269 kB]
Get:5 http://kali.download/kali kali-rolling/main amd64 libfastjson4 amd64 1.2304.0-2 [28.9 kB]
Get:6 http://kali.download/kali kali-rolling/main amd64 liblognorm5 amd64 2.0.6-5 [66.5 kB]
Get:7 http://kali.download/kali kali-rolling/main amd64 rsyslog amd64 8.2504.0-1 [758 kB]
Get:4 http://mirror.primelink.net.id/kali kali-rolling/main amd64 libestr0 amd64 0.1.11-2 [9,048 B]
Get:3 http://mirror.sg.gs/kali kali-rolling/main amd64 snort-common all 3.1.82.0-0kali1 [117 kB]
Get:8 http://http.kali.org/kali kali-rolling/main amd64 libdaq3 amd64 3.0.12-0kali3+b1 [36.3 kB]
36% [3 snort-common 14.0 kB/117 kB 12%] [8 libdaq3 36.3 kB/36.3 kB 100%]
```

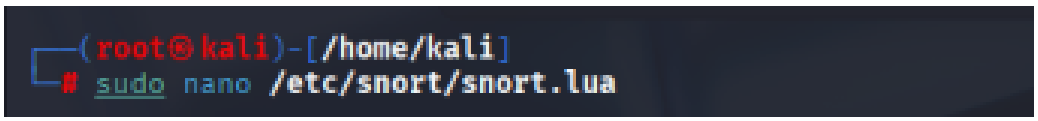
2. VERIFICATION

Snort -v



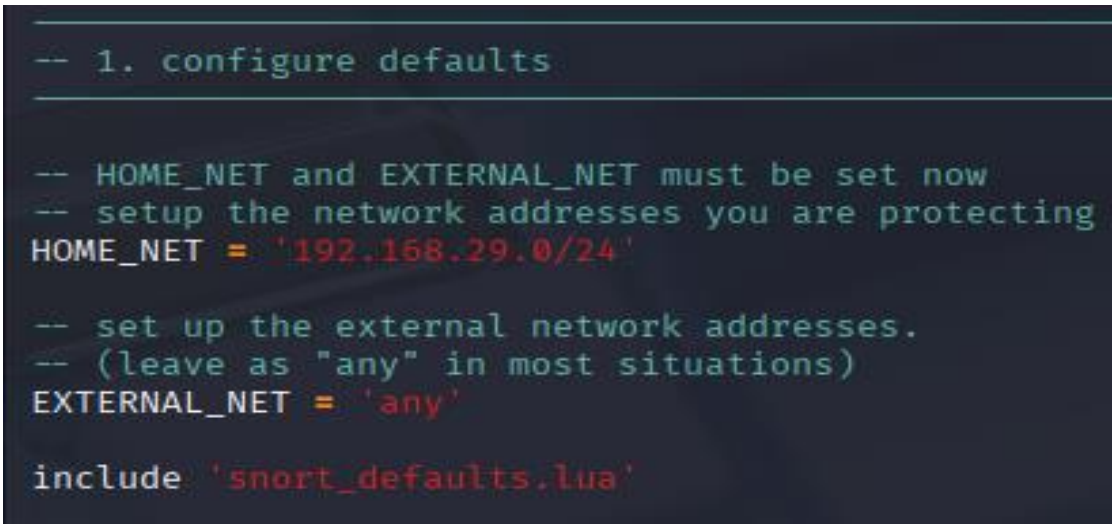
3. CONFIGURATION

Snort network variables were defined in **/etc/snort/snort.lua :**



HOME_NET = '192.168.29.0/24'
EXTERNAL_NET = 'any'

This ensured that Snort monitored the internal network and treated all other traffic as external.

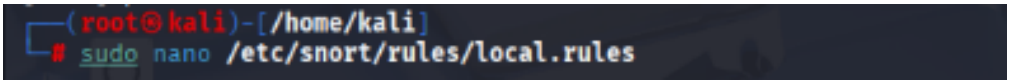


VI. NMAP SCAN DETECTION

Reconnaissance is the first stage of most attacks, where adversaries gather information about the target system. To simulate this, Nmap scans were launched from the attacker machine.

1. CUSTOM RULES IN **/etc/snort/rules/local.rules** :

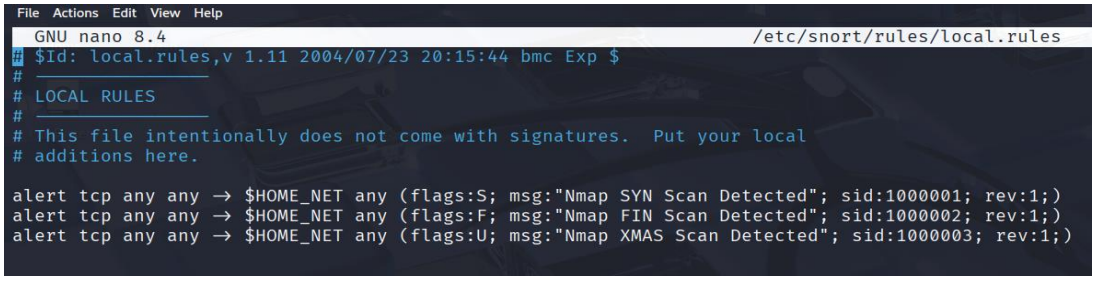
sudo nano /etc/snort/rules/local.rules



Alert tcp any any -> $HOME_NET any (msg: “NMAP SYN Scan Detected” ; flags: S; sid:1000001; rev:1;)

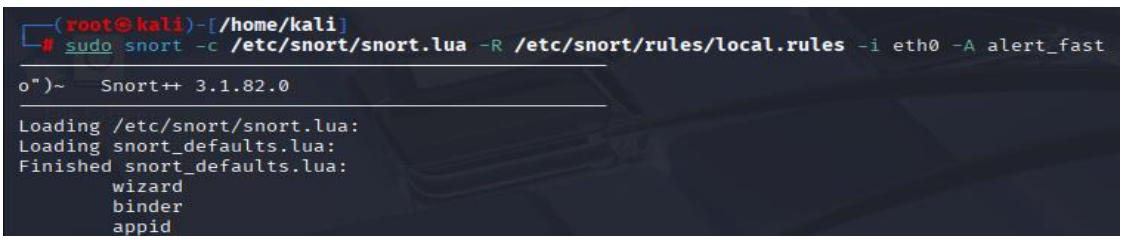
Alert tcp any any -> $HOME_NET any (msg: “NMAP FIN Scan Detected” ; flags: S; sid:1000002; rev:1;)

Alert tcp any any -> $HOME_NET any (msg: “NMAP XMAS Scan Detected” ; flags: S; sid:1000003; rev:1;)



2. RUNNING SNORT:

sudo snort -c /etc/snort/snort.lua -R /etc/snort/rules/local.rules -i eth0 -A alert_fast



3. ATTACKS:

a) SYN Scan

nmap -sS <victim_ip>

```
┌──(kali㉿kali)-[~]
└─$ nmap -sS 192.168.29.63
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-18 14:26 EDT
Nmap scan report for 192.168.29.63
Host is up (0.0013s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
MAC Address: 08:00:27:06:1A:ED (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.88 seconds
```

Observation:

```
Commencing packet processing
++ [0] eth0
08/18-23:56:47.145467 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.29.143 → 224.0.0.22
08/18-23:56:49.186848 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.29.143 → 224.0.0.22
08/18-23:56:54.768919 [**] [1:1000001:1] "Nmap SYN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60505 → 192.168.29.63:993
08/18-23:56:54.770192 [**] [1:1000001:1] "Nmap SYN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60505 → 192.168.29.63:199
08/18-23:56:54.770192 [**] [1:1000001:1] "Nmap SYN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60505 → 192.168.29.63:256
08/18-23:56:54.771279 [**] [1:1000001:1] "Nmap SYN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60505 → 192.168.29.63:22
08/18-23:56:54.771840 [**] [1:1000001:1] "Nmap SYN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60505 → 192.168.29.63:139
08/18-23:56:54.772968 [**] [1:1000001:1] "Nmap SYN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60505 → 192.168.29.63:1025
08/18-23:56:54.773457 [**] [1:1000001:1] "Nmap SYN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60505 → 192.168.29.63:80
08/18-23:56:54.774382 [**] [1:1000001:1] "Nmap SYN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60505 → 192.168.29.63:110
08/18-23:56:54.774889 [**] [1:1000001:1] "Nmap SYN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60505 → 192.168.29.63:3306
08/18-23:56:54.775410 [**] [1:1000001:1] "Nmap SYN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60505 → 192.168.29.63:5900
08/18-23:56:54.776434 [**] [1:1000001:1] "Nmap SYN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60505 → 192.168.29.63:995
08/18-23:56:54.777022 [**] [1:1000001:1] "Nmap SYN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60505 → 192.168.29.63:111
08/18-23:56:54.777734 [**] [1:1000001:1] "Nmap SYN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60505 → 192.168.29.63:8888
08/18-23:56:54.778880 [**] [1:1000001:1] "Nmap SYN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60505 → 192.168.29.63:53
```

Snort triggered alerts for TCP SYN packets (flags:S), confirming Nmap SYN scan detection.

b) FIN Scan:

nmap -sF <victim_ip>

```
┌──(kali㉿kali)-[~]
└─$ nmap -sF 192.168.29.63
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-18 14:28 EDT
Nmap scan report for 192.168.29.63
Host is up (0.0032s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE         SERVICE
21/tcp open|filtered ftp
22/tcp open|filtered ssh
MAC Address: 08:00:27:06:1A:ED (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.86 seconds
```

Observation:

```
08/18-23:58:16.816439 [**] [1:1000002:1] "Nmap FIN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:40068 → 192.168.29.63:1723
08/18-23:58:16.817329 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:40068 → 192.168.29.63:53
08/18-23:58:16.817329 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] {TCP} 192.168.29.173:40068 → 192.168.29.63:53
08/18-23:58:16.817329 [**] [1:1000002:1] "Nmap FIN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:40068 → 192.168.29.63:53
08/18-23:58:16.818275 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:40068 → 192.168.29.63:256
08/18-23:58:16.818275 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] {TCP} 192.168.29.173:40068 → 192.168.29.63:256
08/18-23:58:16.818275 [**] [1:1000002:1] "Nmap FIN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:40068 → 192.168.29.63:256
08/18-23:58:16.819232 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:40068 → 192.168.29.63:110
08/18-23:58:16.819232 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] {TCP} 192.168.29.173:40068 → 192.168.29.63:110
08/18-23:58:16.819232 [**] [1:1000002:1] "Nmap FIN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:40068 → 192.168.29.63:110
08/18-23:58:16.819812 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:40068 → 192.168.29.63:445
08/18-23:58:16.819812 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] {TCP} 192.168.29.173:40068 → 192.168.29.63:445
08/18-23:58:16.819812 [**] [1:1000002:1] "Nmap FIN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:40068 → 192.168.29.63:445
08/18-23:58:16.820404 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:40068 → 192.168.29.63:993
08/18-23:58:16.820404 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] {TCP} 192.168.29.173:40068 → 192.168.29.63:993
08/18-23:58:16.820404 [**] [1:1000002:1] "Nmap FIN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:40068 → 192.168.29.63:993
08/18-23:58:16.821001 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:40068 → 192.168.29.63:3389
08/18-23:58:16.821001 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] {TCP} 192.168.29.173:40068 → 192.168.29.63:3389
08/18-23:58:16.821001 [**] [1:1000002:1] "Nmap FIN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:40068 → 192.168.29.63:3389
08/18-23:58:16.821916 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:40068 → 192.168.29.63:21
08/18-23:58:16.821916 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] {TCP} 192.168.29.173:40068 → 192.168.29.63:21
08/18-23:58:16.821916 [**] [1:1000002:1] "Nmap FIN Scan Detected" [**] [Priority: 0] {TCP} 192.168.29.173:40068 → 192.168.29.63:21
08/18-23:58:16.822667 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:40068 → 192.168.29.63:995
08/18-23:58:16.822667 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] {TCP} 192.168.29.173:40068 → 192.168.29.63:995
```

Snort triggered alerts for TCP FIN packets (flags:F), confirming Nmap FIN scan detection.

c) XMAS Scan:

nmap -sX <victim_ip>

```
┌──(kali㉿kali)-[~]
└─$ nmap -sX 192.168.29.63
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-18 14:30 EDT
Nmap scan report for 192.168.29.63
Host is up (0.00095s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE         SERVICE
21/tcp open|filtered ftp
22/tcp open|filtered ssh
MAC Address: 08:00:27:06:1A:ED (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 2.03 seconds
```

Observation:

```
08/18-23:59:50.141743 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.29.143 → 224.0.0.22
08/18-23:59:59.117386 [**] [116:401:1] "(tcp) Nmap XMAS attack detected" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:587
08/18-23:59:59.117386 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:587
08/18-23:59:59.117386 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:587
08/18-23:59:59.118689 [**] [116:401:1] "(tcp) Nmap XMAS attack detected" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:8888
08/18-23:59:59.118689 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:8888
08/18-23:59:59.118689 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:8888
08/18-23:59:59.118689 [**] [116:401:1] "(tcp) Nmap XMAS attack detected" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:3389
08/18-23:59:59.118689 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:3389
08/18-23:59:59.118689 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:3389
08/18-23:59:59.120613 [**] [116:401:1] "(tcp) Nmap XMAS attack detected" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:5900
08/18-23:59:59.120613 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:5900
08/18-23:59:59.120613 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:5900
08/18-23:59:59.120613 [**] [116:401:1] "(tcp) Nmap XMAS attack detected" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:111
08/18-23:59:59.120613 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:111
08/18-23:59:59.120613 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:111
08/18-23:59:59.121878 [**] [116:401:1] "(tcp) Nmap XMAS attack detected" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:21
08/18-23:59:59.121878 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:21
08/18-23:59:59.121878 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:21
08/18-23:59:59.122392 [**] [116:401:1] "(tcp) Nmap XMAS attack detected" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:199
08/18-23:59:59.122392 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:199
08/18-23:59:59.122392 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:199
08/18-23:59:59.122393 [**] [116:401:1] "(tcp) Nmap XMAS attack detected" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:53
08/18-23:59:59.122393 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:53
08/18-23:59:59.122393 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:53
08/18-23:59:59.123879 [**] [116:401:1] "(tcp) Nmap XMAS attack detected" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:22
08/18-23:59:59.123879 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:22
08/18-23:59:59.123879 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] {TCP} 192.168.29.173:42459 → 192.168.29.63:22
```

Snort triggered alerts for TCP URG packets (flags:U), confirming Nmap Xmas scan detection.

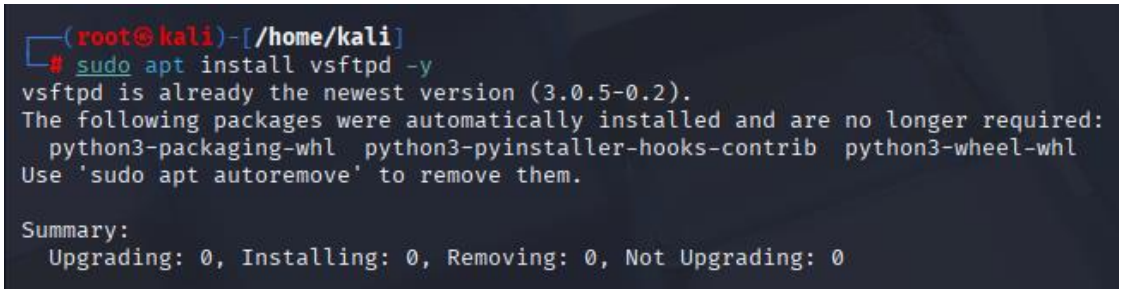
VI . BRUTE-FORCE ATTACK DETECTION

Brute force attcks are common methods attackers use to guess login credentials. To simulate this, Hydra was used against the SSH and FTP services of the victim.

1. INSTALLED SSH & FTP SERVERS ON VICTIM MACHINE:

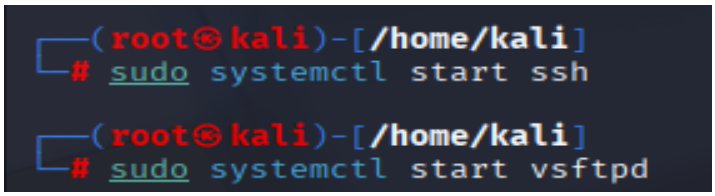
sudo apt install openssh-server -y

sudo apt install vsftpd -y



2. STARTING THE SERVICES:

sudo systemctl start ssh
sudo systemctl start vsftpd



3. ENABLING THE SERVICES:

```
┌──(root㉿kali)-[/home/kali]
└─# sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh

┌──(root㉿kali)-[/home/kali]
└─# sudo systemctl enable vsftpd
Synchronizing state of vsftpd.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable vsftpd
```

4. PASSWORD LIST ON ATTAACKER MACHINE:

echo -e “password123\nadmin\nroot\n123456\nqwerty” > pass.txt

A dummy password list was created with common weak passwords.

```
┌──(kali㉿kali)-[~]
└─$ echo -e "password123\nadmin\nroot\n123456\nqwerty" > pass.txt
```

5. ADDED DETECTION RULES IN LOCAL RULES:

sudo nano /etc/snort/rules/local.rules

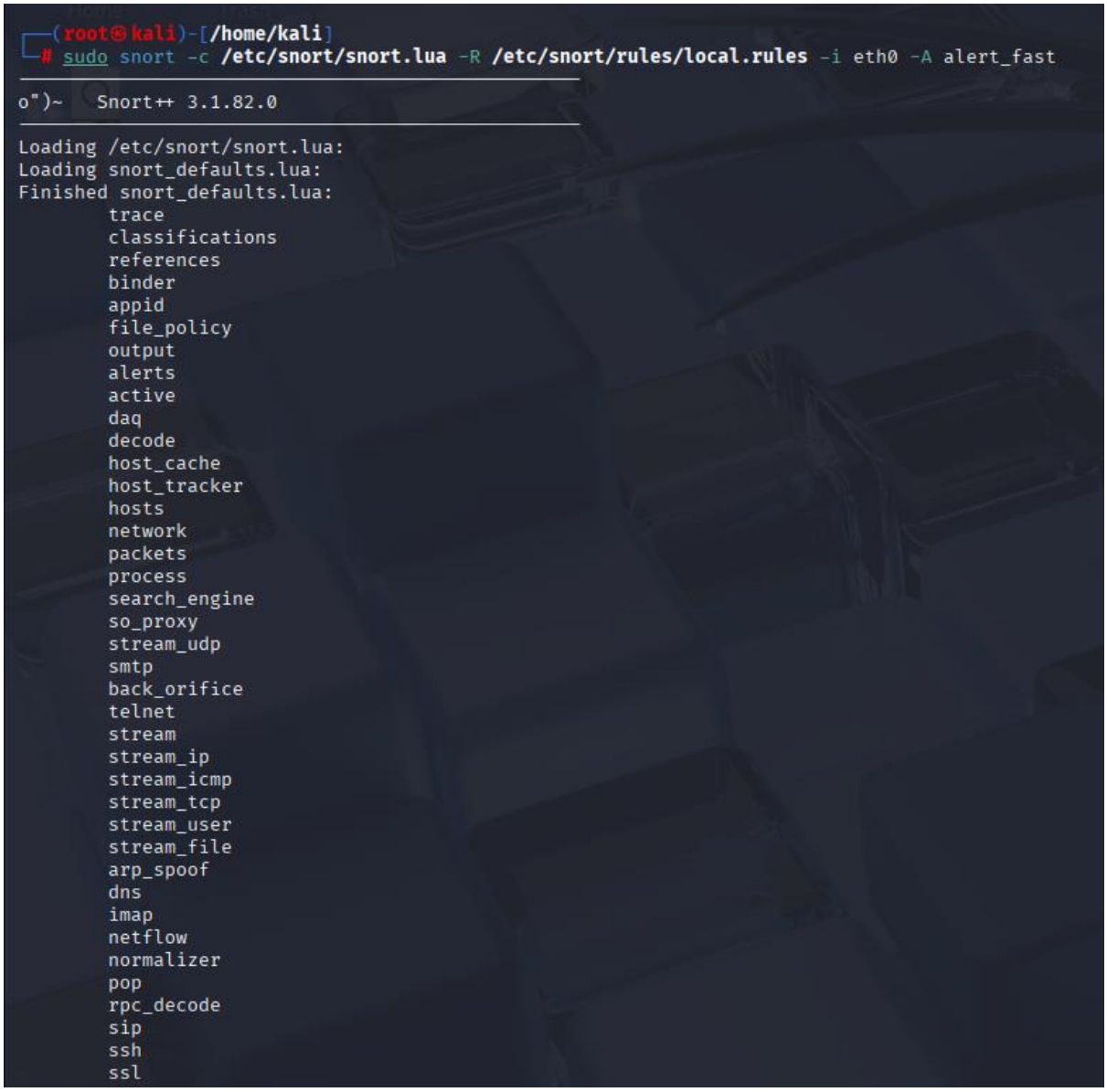
```
┌──(root㉿kali)-[/home/kali]
└─# sudo nano /etc/snort/rules/local.rules
```

alert tcp any any -> $HOME_NET 22 (msg:"SSH Brute-Force Attempt Detected"; flow:to_server,established; detection_filter:track by_src,count 5,seconds 60; sid:1000004; rev:1;)
alert tcp any any -> $HOME_NET 21 (msg:"FTP Brute-Force Attempt Detected"; flow:to_server,established; detection_filter:track by_src,count 5,seconds 60; sid:1000005; rev:1;)

```
root@kali: /home/kali
File  Actions  Edit  View  Help
 GNU nano 8.4                                    /etc/snort/rules/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 tcp any any → $HOME_NET any (flags: S; msg:"Nmap SYN Scan Detected"; sid:1000001; rev:1;)
alert tcp any any → $HOME_NET any (flags: F; msg:"Nmap FIN Scan Detected"; sid:1000002; rev:1;)
alert tcp any any → $HOME_NET any (flags: U; msg:"Nmap XMAS Scan Detected"; sid:1000003; rev:1;)
alert tcp any any → $HOME_NET 22 (msg:"SSH Brute-force Attempt Detected"; flow:to_server,established; detection_filter:track by_src,count 5,seconds 60;sid:1000004; rev:1;)
alert tcp any any → $HOME_NET 21 (msg:"FTP Brute-force Attempt Detected"; flow:to_server,established; detection_filter:track by_src,count 5,seconds 60;sid:1000005; rev:1;)
```

6. RAN SNORT IN THE VICTIM MACHINE:

sudo snort -c /etc/snort/snort.lua -R /etc/snort/rules/local.rules -i eth0 -A alert_fast



7. RAN HYDRA BRUTE-FORCE FROM ATTACKER MACHINE:

a) SSH

hydra -l testuser -P pass.txt ssh://<victim's ip>

```
┌──(kali㉿kali)-[~]
└─$ hydra -l testuser -P pass.txt ssh://192.168.29.63
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations,
or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-08-18 15:09:23
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 5 tasks per 1 server, overall 5 tasks, 5 login tries (l:1/p:5), ~1 try per task
[DATA] attacking ssh://192.168.29.63:22/
[ERROR] could not connect to ssh://192.168.29.63:22 - Socket error: Connection reset by peer
```

Observation

```
08/19-00:39:26.957125 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60440 → 192.168.29.63:22
08/19-00:39:26.957125 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60430 → 192.168.29.63:22
08/19-00:39:26.957125 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60440 → 192.168.29.63:22
08/19-00:39:26.957874 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60442 → 192.168.29.63:22
08/19-00:39:26.957874 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60442 → 192.168.29.63:22
08/19-00:39:26.957874 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60446 → 192.168.29.63:22
08/19-00:39:26.958440 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60446 → 192.168.29.63:22
08/19-00:39:26.969109 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60424 → 192.168.29.63:22
08/19-00:39:26.970101 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60424 → 192.168.29.63:22
08/19-00:39:26.984583 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60442 → 192.168.29.63:22
08/19-00:39:26.984957 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60442 → 192.168.29.63:22
08/19-00:39:26.990203 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60430 → 192.168.29.63:22
08/19-00:39:26.990725 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60430 → 192.168.29.63:22
08/19-00:39:27.002654 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60440 → 192.168.29.63:22
08/19-00:39:27.002654 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60440 → 192.168.29.63:22
08/19-00:39:27.007219 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60424 → 192.168.29.63:22
08/19-00:39:27.014971 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60442 → 192.168.29.63:22
08/19-00:39:27.015845 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60446 → 192.168.29.63:22
08/19-00:39:27.017087 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60446 → 192.168.29.63:22
08/19-00:39:27.022088 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60424 → 192.168.29.63:22
08/19-00:39:27.023864 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60430 → 192.168.29.63:22
08/19-00:39:27.030914 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60442 → 192.168.29.63:22
08/19-00:39:27.032528 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60430 → 192.168.29.63:22
08/19-00:39:27.033282 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60440 → 192.168.29.63:22
08/19-00:39:27.035552 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60446 → 192.168.29.63:22
08/19-00:39:27.039268 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60440 → 192.168.29.63:22
08/19-00:39:27.042490 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60446 → 192.168.29.63:22
08/19-00:39:27.068247 [**] [1:1000004:1] "SSH Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:60424 → 192.168.29.63:22
```

Snort triggered alerts for multiple failed SSH logins (port 22), confirming brute-force activity.

b) FTP:

hydra -l testuser -P pass.txt ftp://<victim's ip>

```
┌──(kali㉿kali)-[~]
└─$ hydra -l testuser -P pass.txt ftp://192.168.29.63
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations,
or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-08-18 15:12:23
[DATA] max 5 tasks per 1 server, overall 5 tasks, 5 login tries (l:1/p:5), ~1 try per task
[DATA] attacking ftp://192.168.29.63:21/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-08-18 15:12:28
```

Observation

```
08/19-00:42:19.277897 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59076 → 192.168.29.63:21
08/19-00:42:19.277897 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59104 → 192.168.29.63:21
08/19-00:42:19.283884 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59074 → 192.168.29.63:21
08/19-00:42:19.284225 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59094 → 192.168.29.63:21
08/19-00:42:19.285518 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59096 → 192.168.29.63:21
08/19-00:42:19.568472 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59094 → 192.168.29.63:21
08/19-00:42:19.568473 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59074 → 192.168.29.63:21
08/19-00:42:19.568473 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59096 → 192.168.29.63:21
08/19-00:42:19.571298 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59104 → 192.168.29.63:21
08/19-00:42:19.571298 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59094 → 192.168.29.63:21
08/19-00:42:19.571298 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59076 → 192.168.29.63:21
08/19-00:42:19.572546 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59074 → 192.168.29.63:21
08/19-00:42:19.573272 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59096 → 192.168.29.63:21
08/19-00:42:19.573272 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59104 → 192.168.29.63:21
08/19-00:42:19.573813 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59076 → 192.168.29.63:21
08/19-00:42:19.671636 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59094 → 192.168.29.63:21
08/19-00:42:19.673711 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59074 → 192.168.29.63:21
08/19-00:42:19.674434 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59104 → 192.168.29.63:21
08/19-00:42:19.674434 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59096 → 192.168.29.63:21
08/19-00:42:19.674434 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59076 → 192.168.29.63:21
08/19-00:42:22.586633 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59074 → 192.168.29.63:21
08/19-00:42:22.692324 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59074 → 192.168.29.63:21
08/19-00:42:22.693053 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59074 → 192.168.29.63:21
08/19-00:42:22.914652 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59096 → 192.168.29.63:21
08/19-00:42:22.915111 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59094 → 192.168.29.63:21
08/19-00:42:22.915112 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59076 → 192.168.29.63:21
08/19-00:42:23.018463 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59096 → 192.168.29.63:21
08/19-00:42:23.019349 [**] [1:1000005:1] "FTP Brute-force Attempt Detected" [**] [Priority: 0] {TCP} 192.168.29.173:59096 → 192.168.29.63:21
```

Snort triggered alerts for repeated failed FTP logins (port 21), confirming brute-force attempts.

VII. MALWARE BEACON TRAFFIC DETECTION

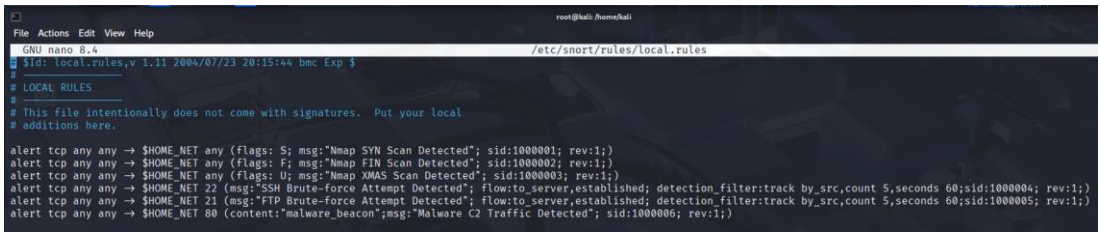
Malware often communicates with Command-and-Control (C2) servers usind beacon traffic. To simulate this, hping3 was used to send repetitive ICMP echo requests with a payload.

1. ADDED RULE IN **local.rules:**

   sudo nano /etc/snort/rules/local.rules

```
┌──(root㉿kali)-[/home/kali]
└─# sudo nano /etc/snort/rules/local.rules
```

   alert tcp any any -> $HOME_NET any (msg:"Malware C2 Traffic Detected"; content:"malware_beacon"; sid:1000006; rev:1;)



2. CREATED DUMMY BEACON PAYLOAD IN ATTACKER MACHINE:

   echo -n "malware_beacon" > beacon.txt

```
┌──(kali㉿kali)-[~]
└─$ echo "malware_beacon" > beacon.txt
```

3. RAN SNORT

   alert tcp any any -> $HOME_NET any (msg:"Malware C2 Traffic Detected"; content:"malware_beacon"; sid:1000006; rev:1;)

```
┌──(root㉿kali)-[/home/kali]
└─# sudo snort -c /etc/snort/snort.lua -R /etc/snort/rules/local.rules -i eth0 -A alert_fast
o")~   Snort++ 3.1.82.0
Loading /etc/snort/snort.lua:
Loading snort_defaults.lua:
Finished snort_defaults.lua:
        telnet
        iec104
        mms
        modbus
        s7commplus
        dce_smb
        dce_tcp
        dce_udp
        dce_http_proxy
        dce_http_server
        gtp_inspect
        output
        smtp
        ftp_client
        ftp_data
        http_inspect
        http2_inspect
```

4. SENT BEACON TRAFFIC:

hping3 -c 1 -d 20 --file beacon.txt -p 80 <victim's ip>

```
┌──(root㉿kali)-[/home/kali]
└─# hping3 -c 1 -d 20 --file beacon.txt -p 80 192.168.29.63
HPING 192.168.29.63 (eth0 192.168.29.63): NO FLAGS are set, 40 headers + 20 data bytes
[main] memlockall(): No such file or directory
Warning: can't disable memory paging!
len=46 ip=192.168.29.63 ttl=64 DF id=0 sport=80 flags=RA seq=0 win=0 rtt=11.5 ms

── 192.168.29.63 hping statistic ──
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 11.5/11.5/11.5 ms
```

5. OBSERVED REAL TIME ALERTS IN SNORT CONSOLE:

```
08/19-01:29:22.307836 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:2865 → 192.168.29.63:80
08/19-01:29:22.307836 [**] [1:1000006:1] "Malware C2 Traffic Detected" [**] [Priority: 0] {TCP} 192.168.29.173:2865 → 192.168.29.63:80
08/19-01:29:24.063676 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:2522 → 192.168.29.63:80
08/19-01:29:24.063676 [**] [1:1000006:1] "Malware C2 Traffic Detected" [**] [Priority: 0] {TCP} 192.168.29.173:2522 → 192.168.29.63:80
08/19-01:29:24.138517 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.29.143 → 224.0.0.22
08/19-01:29:24.291374 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.29.56 → 224.0.0.22
08/19-01:29:25.313869 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:1505 → 192.168.29.63:80
08/19-01:29:25.313869 [**] [1:1000006:1] "Malware C2 Traffic Detected" [**] [Priority: 0] {TCP} 192.168.29.173:1505 → 192.168.29.63:80
08/19-01:29:26.761195 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] {TCP} 192.168.29.173:2277 → 192.168.29.63:80
08/19-01:29:26.761195 [**] [1:1000006:1] "Malware C2 Traffic Detected" [**] [Priority: 0] {TCP} 192.168.29.173:2277 → 192.168.29.63:80
```

Snort triggered alerts for repetitive packets with "malware_beacon" over port 80, confirming simulated C2 beacon traffic.

VIII. RESULTS & ANALYSIS

Snort NIDS was tested against multiple simulated attack scenarios in the virtual SOC lab. The detection outcomes are;

SL NO.	ATTACK TYPE	CONDITION	OBSERVATION
1.	SYN Scan	flags:S	Alerts triggered for TCP SYN packets, confirming Nmap SYN scan detection.
2.	FIN Scan	flags:F	Alerts triggered for TCP FIN packets, confirming Nmap FIN scan detection.
3.	Xmas Scan	flags:U	Alerts triggered for TCP URG packets, confirming Nmap Xmas scan detection.
4.	SSH Brute force	port 22, detection filter for repeated logins	Alerts triggered for multiple failed SSH login attempts indicating brute force.
5.	FTP Brute force	port 21, detection filter for repeated logins	Alerts triggered for repeated failed FTP login attempts confirming brute force.
6.	Malware Beaconing	content:"malware_beacon", port 80	Alerts triggered for repetitive packets simulating malware beacon traffic.

Snort successfully identified all attack attempts, showing that the configured custom rules were effective in detecting different intrusion patterns.

IX. CONCLUSION

This project focused on building and testing a Network Intrusion Detection System (NIDS) using Snort in a virtual SOC lab environment. The setup included an attacker machine and a victim machine with Snort installed, allowing us to create a controlled space for simulating and studying real-world attack patterns.

Different attacks were carried out to test how Snort responds. These included Nmap scans (SYN, FIN, and Xmas), brute-force login attempts on SSH and FTP, and malware beaconing traffic. In each case, Snort generated alerts and logs, which proved its ability to recognize unusual or harmful network activity. The detection of these attacks showed how NIDS plays an important role in identifying intrusions before they cause damage.

Another key part of the project was the creation of custom Snort rules. By writing rules for each type of attack, it was possible to clearly see how signatures work and how they can be adjusted for different needs. This made the exercise very practical and closer to what analysts do in a Security Operations Centre (SOC).

Overall, this project gave a clear understanding of how Snort functions as a NIDS, how rules are written and tested, and how alerts can support analysis of suspicious activity. Even though the project was done in a lab setting, the tasks closely reflected real SOC monitoring and incident detection practices. This shows that Snort, even as an open-source tool, can be effectively used to strengthen network security and provide valuable insights for defenders.