

The World Islamic Sciences and Education University

جامعة العلوم الاسلامية العالمية

Faculty of Information Technology

كلية تكنولوجيا المعلومات



NETWORK SECURITY PROJECT

Title

***Design and Implementation of an Intrusion Prevention System (IPS)
using Snort 3***

Students

Abdaullah Mohammad Mustafa Abughallous (3220603043)

Supervisor

Dr.Firas Alzobi

SEMESTER I

2025/2026

Contents

INTRODUCTION.....	1
PROJECT PROBLEM.....	1
PROJECT OBJECTIVES.....	1
PROJECT SCOPE AND BOUNDARIES	1
THEORETICAL BACKGROUND.....	2
SYSTEM ARCHITECTURE.....	2
TOOLS AND TECHNOLOGIES USED	3
METHODOLOGY	3
IMPLEMENTATION	3
TESTING AND EVALUATION	3
RESULTS AND DISCUSSION	4
CONCLUSION	4
FUTURE WORK.....	4
REFERENCES.....	4
APPENDICES	5
1. IPS RULES SCREENSHOT :	5
2. IPS TESTING SCREEN SHOT :.....	6
<input type="checkbox"/> PING	6
<input type="checkbox"/> NMAP(SSH).....	7
<input type="checkbox"/> NMAP(SCAN).....	7
3. IDS (RULES).....	8
4. IDS TESTING SCREENSHOT :	8
<input type="checkbox"/> PING	8
<input type="checkbox"/> NMAP (SCAN).....	9
<input type="checkbox"/> NMAP (SSH).....	9

INTRODUCTION

In the contemporary digital landscape, network security has evolved from simple perimeter defense to complex real-time threat mitigation. Intrusion Prevention Systems (IPS) represent a critical evolution of security technology, providing the ability to not only detect but also actively block malicious traffic. This project explores the deployment of Snort 3, the latest iteration of the world's leading open-source IPS, to protect network integrity through deep packet inspection and automated response.

PROJECT PROBLEM

Traditional security measures often rely on passive detection (IDS) which generates alerts after a breach has already initiated. This delay allows attackers to complete reconnaissance or execute payloads. Furthermore, standard firewalls struggle to identify application-layer threats. A significant technical challenge addressed in this project is "Kernel Leakage"—where packets bypass security software due to OS-level routing—necessitating a specialized transparent bridge configuration to ensure 100% traffic inspection.

PROJECT OBJECTIVES

1. First Objective: To design and implement a Transparent Network Bridge that forces all inbound and outbound traffic to pass through the Snort 3 engine for inspection the make IDS to show the different .
2. Second Objective: To achieve active threat mitigation (Prevention) by configuring Snort 3 in Inline Mode using the DAQ AFPacket module, ensuring malicious packets are dropped before reaching the target.

PROJECT SCOPE AND BOUNDARIES

Included: Implementation of a 3-node network (Attacker, IPS Bridge, Victim), configuration of Snort 3 Inline mode, mitigation of ICMP floods, and prevention of Nmap-based reconnaissance scans.

Not Included: Protection against encrypted HTTPS traffic (SSL/TLS decryption), hardware-based acceleration (FPGA/ASIC), or cloud-native WAF deployments.

THEORETICAL BACKGROUND

This project is built upon the concepts of Signature-based Detection and Inline Processing. Unlike IDS, which uses a "tap" or "span" port to copy traffic, an IPS sits directly in the communication path. Snort 3 utilizes a multi-threaded architecture and the DAQ (Data Acquisition) library to interface with network hardware, allowing it to intercept, analyze, and either "Pass" or "Drop" packets based on pre-defined security rules.

SYSTEM ARCHITECTURE

The architecture follows a Transparent Bridge model. The IPS node acts as a virtual switch with two interfaces (eth0 and eth1) but no visible IP address to the attacker, making it a "Stealth IPS."

Attacker Node: Kali Linux (192.168.56.102).

IPS Node: Snort 3 Intermediary (Transparent Bridge).

Victim Node: Kali Linux (192.168.56.110).



TOOLS AND TECHNOLOGIES USED

Operating Systems: Kali Linux (Rolling Edition).

Core Software: Snort 3.1+ (Snort++).

Libraries: LibDAQ 3.x (AFPacket module).

Scanning Tools: Nmap, ping.

Network Utilities: Iproute2, Sysctl (for kernel tuning).

METHODOLOGY

Environment Preparation: Setting up two network interfaces and flushing existing IP configurations to prevent kernel-level routing.

Bridge Configuration: Unbinding eth0 and eth1 from the IP stack and enabling them in promiscuous mode.

IPS Configuration: Modifying snort.lua to recognize the DAQ AFPacket variables.

Rule Development: Converting detection rules (alert) into prevention rules (drop).

Execution: Launching Snort with the -Q flag for inline operation.

IMPLEMENTATION

The practical application involved executing Snort with specific DAQ variables to handle packet replacement:

- Ensuring no traffic leaks through the Kernel
`sudo sysctl -w net.ipv4.ip_forward=0`
- Running the IPS Engine
`sudo snort -Q --daq afpacket -i eth0:eth1 -c /etc/snort/snort.lua -A alert_fast --daq-var replace=1`

Manual verification included checking Snort logs for [drop] tags which indicate successful packet interception.

TESTING AND EVALUATION

Actual Result	Expected Result	Tool Used	Test Scenario
%100Packet Loss	All packets dropped	hping3 --flood	ICMP Flood
Status: Filtered	Port status: Filtered	nmap -sS	Stealth Scan

Blocked after threshold	Connection Reset	Hydra / Nmap	SSH Brute Force
-------------------------	------------------	--------------	------------------------

RESULTS AND DISCUSSION

The evaluation proved that the IPS was 100% effective in blocking ICMP-based DoS attacks. Most notably, the Nmap scan results changed from Open (before IPS) to Filtered (after IPS). This "Filtered" state confirms that Snort dropped the SYN packets silently, preventing the attacker from even knowing if the service existed, thus successfully neutralizing the reconnaissance phase.

CONCLUSION

The project successfully demonstrated that Snort 3, when correctly configured in Inline Mode, provides robust protection against common network threats. By resolving the Kernel Leakage issue and utilizing the AFPacket DAQ, we achieved a stable IPS environment with negligible latency (~2ms), suitable for production-level security.

FUTURE WORK

Proposed enhancements include:

- Integrating the ELK Stack (Elasticsearch, Logstash, Kibana) for real-time visual threat analysis.
- Implementing Machine Learning pre-processors to detect anomalous traffic patterns that do not match existing signatures.

REFERENCES

Cisco Systems. (2025). Snort 3 User Guide: Advanced Intrusion Prevention.

<https://docs.snort.org/>

Kaur, P., & Gupta, A. (2024). Deep Packet Inspection and Mitigation Techniques in Snort 3 IPS. IJCSDF, 13(1), 45-58.

Talos Intelligence Group. (2024). Snort 3 Rule Writing Best Practices. Cisco Talos Whitepaper.

Verma, S. (2023). Performance Analysis of Snort 3 in Multi-Gigabit Deployments. Journal of Network Security.

DAQ Contributors. (2025). DAQ v3.x Specifications.

<https://github.com/snort3/libdaq>

Sharma, R. (2022). Transitioning to Snort 3: Architectural Benefits. ICCS 2022 Proceedings

APPENDICES

1. IPS RULES SCREENSHOT :

```
# 1. Ping Block (إذا زاد عن 10 نوا Ping خطر ال)
drop icmp any any → any any (msg:"*** ICMP FLOOD DETECTED ***"; detection_filter: track by_src, count 5, seconds 30)

# 2. Telnet Block (خطر محاولات الدخول المتكررة - بروتوكول غير آمن)
drop tcp any any → any 23 (msg:"*** TELNET BRUTE FORCE DETECTED ***"; detection_filter: track by_src, count 3, seconds 30)

# 3. SSH Scan Block (للمنفذ 22 أكثر من 5 مرات في الدقيقة Scan خطر من يقوم بعمل)
drop tcp any any → any 22 (flags:S; msg:"*** SSH SCAN DETECTED ***"; detection_filter: track by_src, count 5, seconds 30)

# 4. Nmap Port Scan Block (خطر فحص المنافذ السريع)
drop tcp any any → any any (flags:S,12; msg:"*** NMAP SCAN DETECTED ***"; detection_filter: track by_src, count 2, seconds 30)

# 5. Nmap OS Detection Block (خطر محاولة معرفة نوع نظام التشغيل)
drop tcp any any → any any (flags:FPU; msg:"*** OS FINGERPRINTING DETECTED ***"; detection_filter: track by_src, count 2, seconds 30)

# 6. Suspicious HTTP (بسيط DoS يحمي من - GET خطر إغراق الموقع بطلبات)
drop tcp any any → any 80 (content:"GET"; msg:"*** HTTP FLOOD DETECTED ***"; detection_filter: track by_src, count 2, seconds 30)
```

TRACKING

2. IPS TESTING SCREEN SHOT :

- **PING**

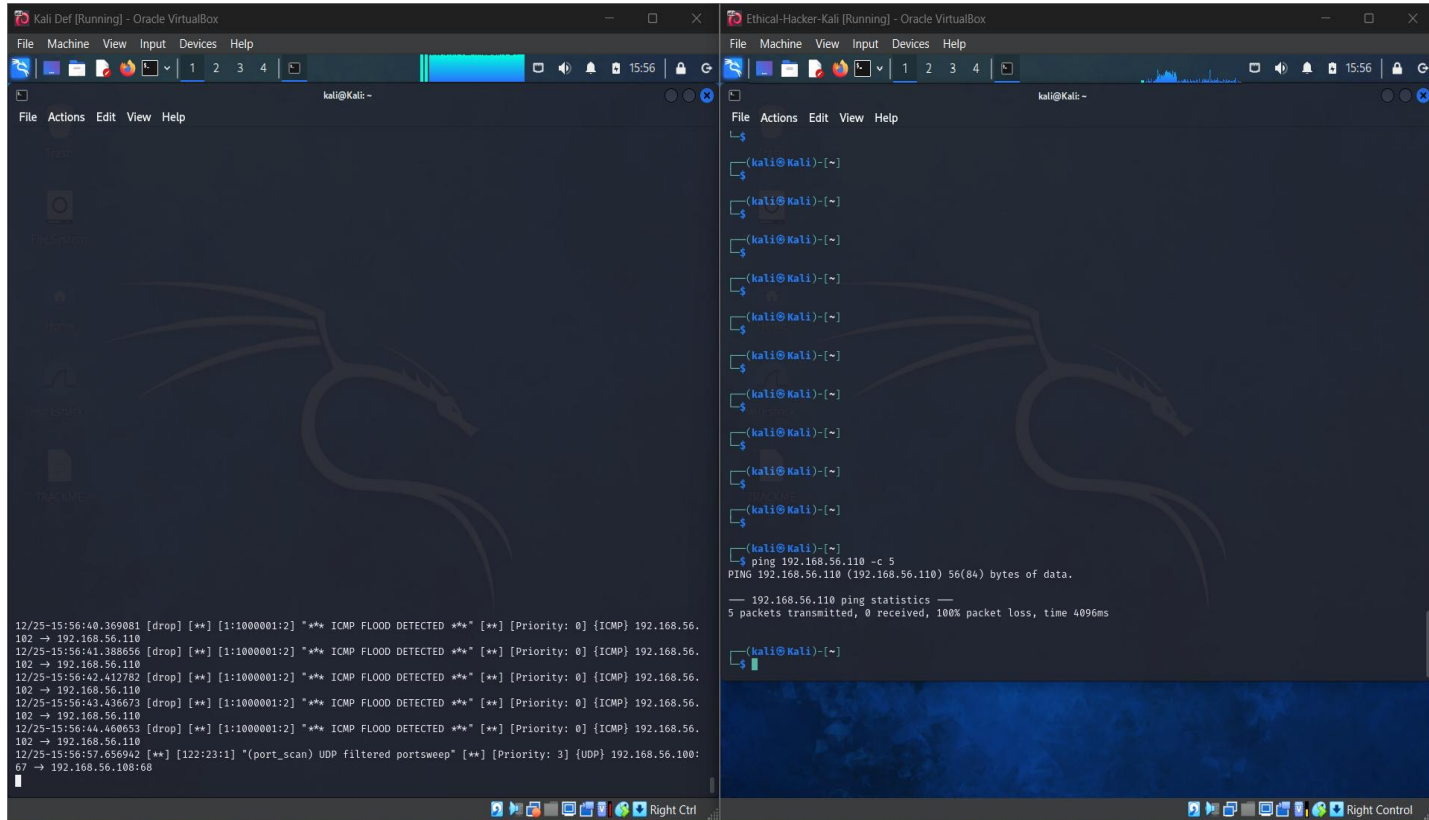


Figure 1 PING

- **NMAP(SSH)**

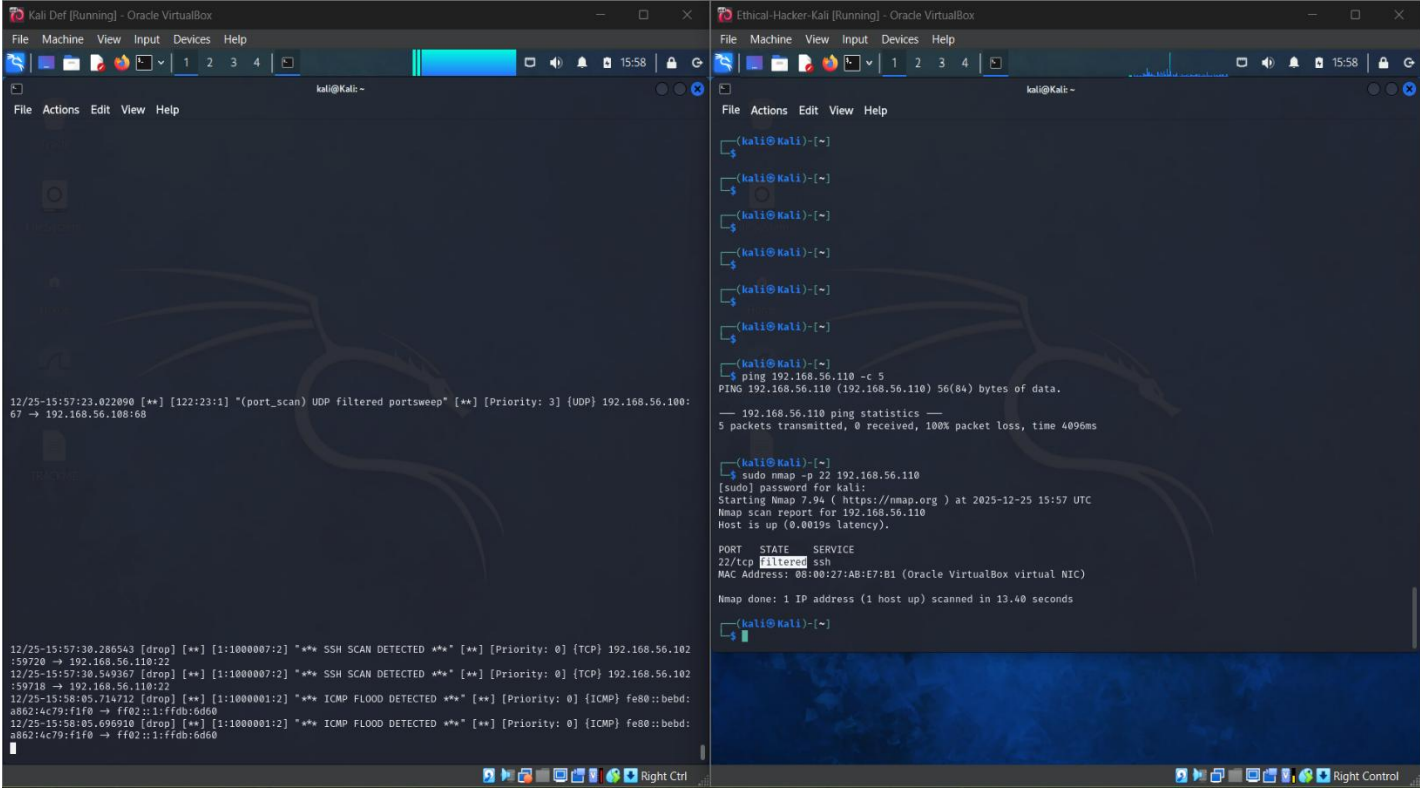
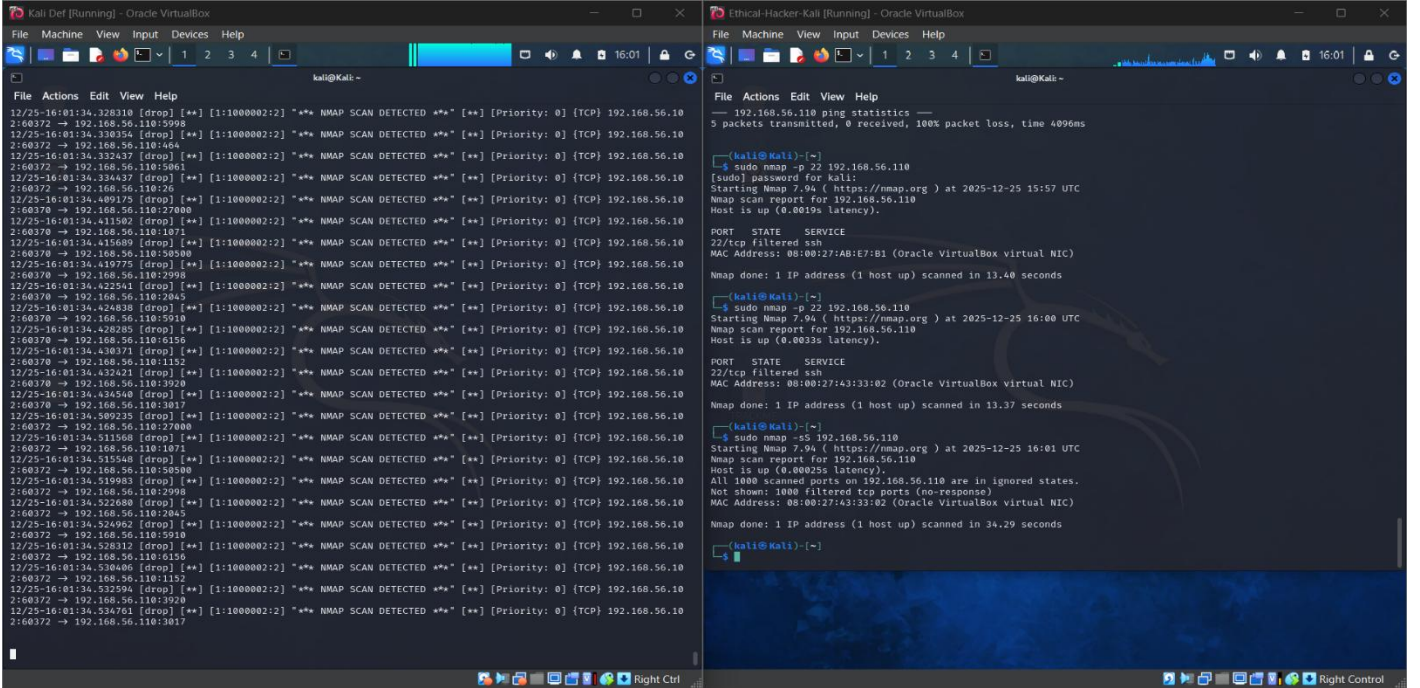


Figure 2 NMAP SSH

- **NMAP(SCAN)**



3. IDS (RULES)

Kali Def (Purwing) - Oracle VM VirtualBox

File Machine View Input Devices Help

1 2 3 4

kali@kali: ~/Desktop

File Actions Edit View Help

GNU nano 2.8.5 /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.
#
# ping rule
alert icmp any any -> any any (msg:"ICMP detected from attacker"; sid:1000001; rev:1;)
#2:telnet
alert tcp any any -> any 23 (msg:"Telnet connection attempt detected"; sid:1000005; rev:1;)
#3:nmap(ssh scan)
alert tcp any any -> any 22 (msg:"SSH scan detected"; sid:1000002; rev:1;)
#4:nmap(port scan)
alert tcp any any -> any any (flags:S; msg:"TCP SYN Scan detected"; sid:1000002; rev:1;)
#5:nmap(os detection)
alert tcp any any -> any any (flags:FPU; msg:"Nmap OS Detection attempt"; sid:1000003; rev:1;)
#http get request
alert tcp any any -> any 80 (msg:"HTTP GET request detected"; content:"GET"; sid:1000006; rev:1;)
```

Read 19 lines

Undo Redo Set Mark To Bracket Previous Back Prev Word Home
Undo Redo Copy Where Was Next Forward Next Word End

Help Write Out Where Is Cut Execute Location
Exit Read File Replace Paste Justify Go To Line

4. IDS TESTING SCREENSHOT :

- **PING**

[illegible]

• NMAP (SCAN)

```

Ethical-Hacker-Kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help

kali@kali:~$ nmap -sS 192.168.56.110
You requested a scan type which requires root privileges.
QUITTING!

kali@kali:~$ sudo nmap -sS 192.168.56.110
Starting Nmap 7.94 ( https://nmap.org ) at 2025-12-17 09:31 UTC
Nmap scan report for 192.168.56.110
Host is up (0.00093s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
Nmap done: 1 IP address (1 host up) scanned in 13.27 seconds

kali@kali:~$

```

```

Kali Def [Running] - Oracle VirtualBox
File Machine View Input Devices Help

12/17-09:29:08.667501 [**] [112:1:1] "(arp_spoof) unicast ARP request" [**] [Priority: 3] [ARP] →
12/17-09:29:08.752473 [**] [1:1000001:1] "ICMP detected from attacker" [**] [Priority: 0] [ICMP] 192.168.56.102 →
192.168.56.110
12/17-09:29:04.752512 [**] [1:1000001:1] "ICMP detected from attacker" [**] [Priority: 0] [ICMP] 192.168.56.110 →
192.168.56.102
12/17-09:29:05.778013 [**] [1:1000001:1] "ICMP detected from attacker" [**] [Priority: 0] [ICMP] 192.168.56.102 →
192.168.56.110
12/17-09:29:05.778061 [**] [1:1000001:1] "ICMP detected from attacker" [**] [Priority: 0] [ICMP] 192.168.56.110 →
192.168.56.102
12/17-09:29:06.778497 [**] [1:1000001:1] "ICMP detected from attacker" [**] [Priority: 0] [ICMP] 192.168.56.102 →
192.168.56.110
12/17-09:29:06.778544 [**] [1:1000001:1] "ICMP detected from attacker" [**] [Priority: 0] [ICMP] 192.168.56.110 →
192.168.56.102
12/17-09:29:30.756716 [**] [1:1000001:1] "ICMP detected from attacker" [**] [Priority: 0] [ICMP] fe80::a00:27ff:fe5
c:52ce → ff02::2
12/17-09:31:42.258295 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:48325 → 1
92.168.56.110:5900
12/17-09:31:42.258296 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:48325 → 1
92.168.56.110:8000
12/17-09:31:42.258296 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:48325 → 1
92.168.56.110:22
12/17-09:31:42.258297 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:48325 → 1
92.168.56.110:199
12/17-09:31:42.258297 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:48325 → 1
92.168.56.110:256
12/17-09:31:42.258297 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:48325 → 1
92.168.56.110:3389
12/17-09:31:42.258297 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:48325 → 1
92.168.56.110:587
12/17-09:31:42.258591 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:48325 → 1
92.168.56.110:1005
12/17-09:31:42.258591 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:48325 → 1
92.168.56.110:135
12/17-09:31:42.260775 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:48325 → 1
92.168.56.110:1725
12/17-09:31:42.260775 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:48325 → 1
92.168.56.110:995
12/17-09:31:42.260775 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:48325 → 1
92.168.56.110:111
12/17-09:31:42.260776 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:48325 → 1
92.168.56.110:25
12/17-09:31:42.260776 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:48325 → 1

```

• NMAP (SSH)

```

Ethical-Hacker-Kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help

kali@kali:~$ telnet 192.168.56.110
Trying 192.168.56.110...
telnet: Unable to connect to remote host: Connection refused

kali@kali:~$ sudo telnet 192.168.56.110
Trying 192.168.56.110...
telnet: Unable to connect to remote host: Connection refused

kali@kali:~$ nmap -p 22,23,80,443 192.168.56.110
Starting Nmap 7.94 ( https://nmap.org ) at 2025-12-17 09:38 UTC
Nmap scan report for 192.168.56.110
Host is up (0.0013s latency).
PORT      STATE SERVICE
22/tcp    open  ssh
23/tcp    closed telnet
80/tcp    closed http
443/tcp   closed https
Nmap done: 1 IP address (1 host up) scanned in 13.05 seconds

kali@kali:~$

```

```

Kali Def [Running] - Oracle VirtualBox
File Machine View Input Devices Help

92.168.56.110:1
12/17-09:34:22.543592 [**] [116:401:1] "(tcp) Nmap XMAS attack detected" [**] [Priority: 3] [TCP] 192.168.56.102:54
071 → 192.168.56.110:1
12/17-09:34:22.543592 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] [TCP] 192.168.56.102:
54071 → 192.168.56.110:1
12/17-09:34:22.543592 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] [TCP]
192.168.56.102:54071 → 192.168.56.110:1
12/17-09:34:22.548662 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] [TCP] 192.168.56.102:
54866 → 192.168.56.110:22
12/17-09:34:22.596583 [**] [116:401:1] "(tcp) Nmap XMAS attack detected" [**] [Priority: 3] [TCP] 192.168.56.102:54
067 → 192.168.56.110:22
12/17-09:34:22.596583 [**] [116:420:1] "(tcp) TCP SYN with FIN" [**] [Priority: 3] [TCP] 192.168.56.102:54067 → 19
2.168.56.110:22
12/17-09:34:22.596583 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] [TCP]
192.168.56.102:54067 → 192.168.56.110:22
12/17-09:34:22.672422 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] [TCP] 192.168.56.102:
54866 → 192.168.56.110:22
12/17-09:34:22.696637 [**] [116:420:1] "(tcp) TCP SYN with FIN" [**] [Priority: 3] [TCP] 192.168.56.102:54067 → 19
2.168.56.110:22
12/17-09:34:22.696637 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] [TCP]
192.168.56.102:54067 → 192.168.56.110:22
12/17-09:34:22.772008 [**] [116:423:1] "(tcp) TCP has no SYN, ACK, or RST" [**] [Priority: 3] [TCP] 192.168.56.102:
54866 → 192.168.56.110:22
12/17-09:34:22.772008 [**] [116:401:1] "(tcp) Nmap XMAS attack detected" [**] [Priority: 3] [TCP] 192.168.56.102:54
067 → 192.168.56.110:22
12/17-09:34:22.777771 [**] [116:420:1] "(tcp) TCP SYN with FIN" [**] [Priority: 3] [TCP] 192.168.56.102:54067 → 19
2.168.56.110:22
12/17-09:34:22.777771 [**] [116:422:1] "(tcp) TCP PDU missing ack for established session" [**] [Priority: 3] [TCP]
192.168.56.102:54067 → 192.168.56.110:22
12/17-09:34:26.683587 [**] [112:1:1] "(arp_spoof) unicast ARP request" [**] [Priority: 3] [ARP] →
12/17-09:35:32.232120 [**] [1:1000001:1] "ICMP detected from attacker" [**] [Priority: 0] [ICMP] fe80::a00:27ff:fe5
c:52ce → ff02::2
12/17-09:36:11.506786 [**] [1:1000005:1] "Telnet connection attempt detected" [**] [Priority: 0] [TCP] 192.168.56.1
02:36654 → 192.168.56.110:23
12/17-09:36:11.506786 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:36654 → 1
92.168.56.110:23
12/17-09:36:16.507890 [**] [112:1:1] "(arp_spoof) unicast ARP request" [**] [Priority: 3] [ARP] →
12/17-09:36:16.626231 [**] [112:1:1] "(arp_spoof) unicast ARP request" [**] [Priority: 3] [ARP] →
12/17-09:36:18.289713 [**] [1:1000005:1] "Telnet connection attempt detected" [**] [Priority: 0] [TCP] 192.168.56.1
02:36650 → 192.168.56.110:23
12/17-09:36:18.289713 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:36650 → 1
92.168.56.110:23
12/17-09:38:31.923792 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:51854 → 1
92.168.56.110:80
12/17-09:38:31.923792 [**] [1:1000002:1] "TCP SYN Scan detected" [**] [Priority: 0] [TCP] 192.168.56.102:54844 → 1
92.168.56.110:443
12/17-09:38:37.051690 [**] [112:1:1] "(arp_spoof) unicast ARP request" [**] [Priority: 3] [ARP] →
12/17-09:38:37.109761 [**] [112:1:1] "(arp_spoof) unicast ARP request" [**] [Priority: 3] [ARP] →

```