Project Report: Basic Network Packet Sniffer

1. Introduction

This project aims to create a Python-based network packet sniffer using the Scapy library. The sniffer c

2. Abstract

The network packet sniffer captures packets such as HTTP, FTP, or other protocols, extracts key inform

3. Tools Used

- Python 3
- Scapy library
- Kali Linux OS
- Text editor (VS Code, nano)

4. Steps Involved in Building the Project

- Install Python and Scapy on Kali Linux.
- Write a Python script (packet_sniffer.py) using Scapy to capture live packets.
- Implement filters to capture specific protocols like HTTP or FTP.
- Extract key packet fields: IP addresses, ports, payloads, etc.
- Log captured packet details into a text file with timestamps.
- Run the script and monitor network traffic in real-time.

5. Screenshots/Outputs

Example log output from the sniffer script:

Timestamp: 2025-09-08 12:00:01

Source IP: 192.168.1.2

Destination IP: 93.184.216.34

Protocol: HTTP

Payload: GET /index.html

---

6. Conclusion

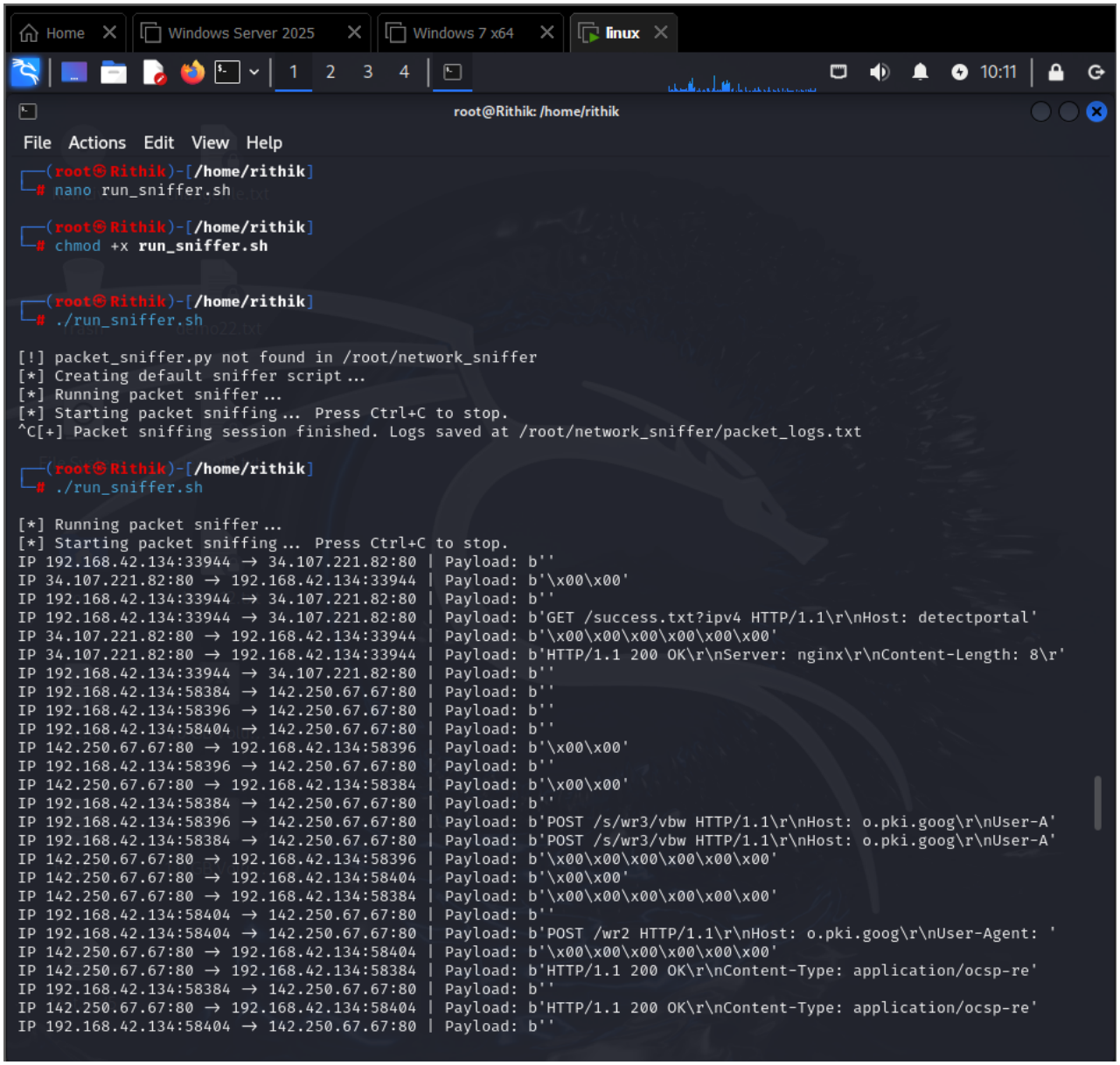
The project demonstrates how to capture and analyze network packets using Python and Scapy. It high

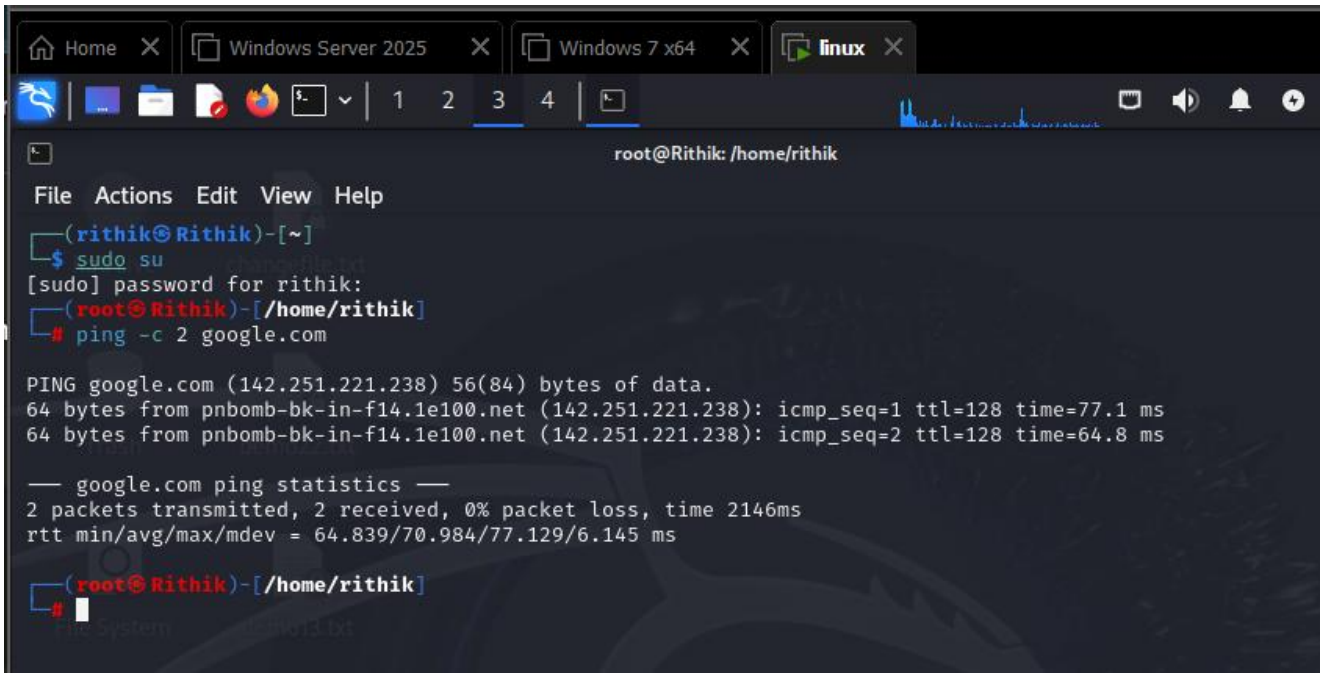
7. Challenges Faced

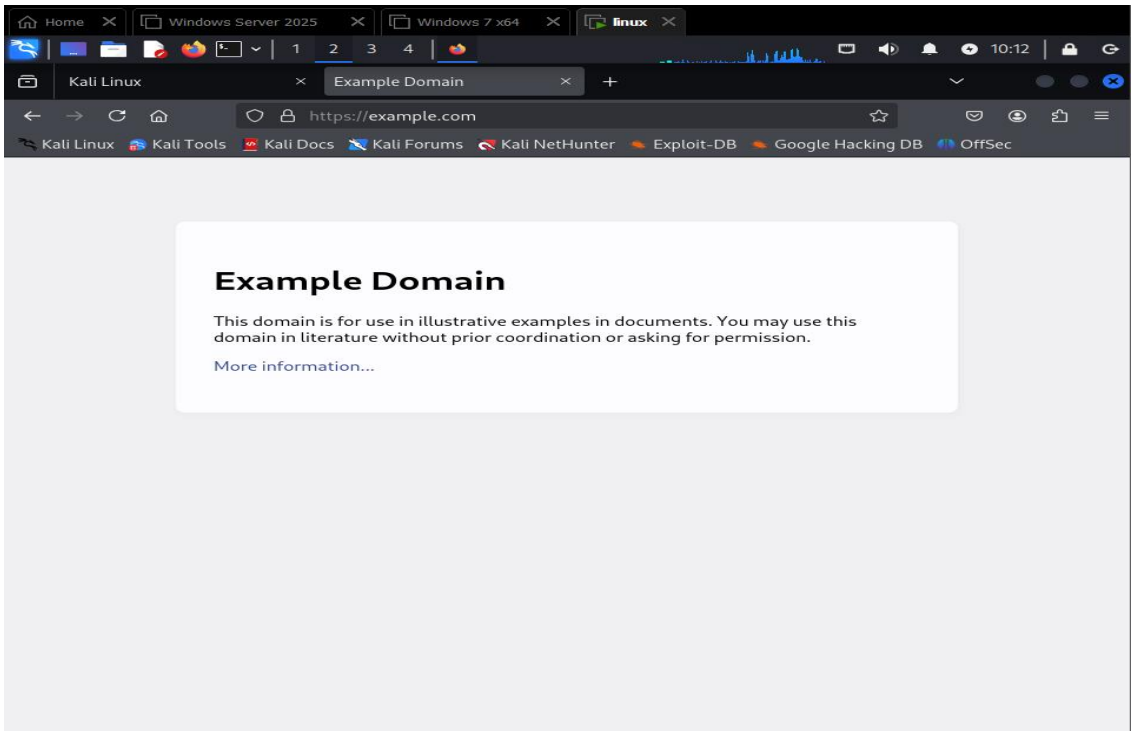
- Ensuring proper permissions to capture network packets (requires root/admin access).
- Filtering and parsing packets efficiently.
- Logging packets with readable timestamps and formatting.

8. References

- Scapy Documentation: https://scapy.readthedocs.io/en/latest/
- Python Official Documentation: https://docs.python.org/3/
- Kali Linux Tools: https://www.kali.org/tools/




Home
Windows Server 2025
Windows 7 x64
linux
1 2 3 4
root@Rithik: /home/rithik
File Actions Edit View Help
┌──(rithik㉿Rithik)-[~]
└─$ sudo su
[sudo] password for rithik:
┌──(root㉿Rithik)-[/home/rithik]
└─# ping -c 2 google.com
PING google.com (142.251.221.238) 56(84) bytes of data.
64 bytes from pnbomb-bk-in-f14.1e100.net (142.251.221.238): icmp_seq=1 ttl=128 time=77.1 ms
64 bytes from pnbomb-bk-in-f14.1e100.net (142.251.221.238): icmp_seq=2 ttl=128 time=64.8 ms
— google.com ping statistics —
2 packets transmitted, 2 received, 0% packet loss, time 2146ms
rtt min/avg/max/mdev = 64.839/70.984/77.129/6.145 ms
┌──(root㉿Rithik)-[/home/rithik]
└─#


Home
Windows Server 2025
Windows 7 x64
linux
1 2 3 4
10:12
Kali Linux
Example Domain
https://example.com
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Example Domain
This domain is for use in illustrative examples in documents. You may use this domain in literature without prior coordination or asking for permission.
More information...

Home | Windows Server 2025 | Windows 7 x64 | linux

1 2 3 4

root@Rithik: /home/rithik

File Actions Edit View Help

GNU nano 8.2 run_sniffer.sh

```
    cat > "$SNIF_SCRIPT" <<'EOF'
#!/usr/bin/env python3
from scapy.all import sniff
from scapy.layers.inet import IP, TCP
from datetime import datetime

LOG_FILE = "packet_logs.txt"

def log_packet(packet_info):
    timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    with open(LOG_FILE, "a") as f:
        f.write(f"[{timestamp}] {packet_info}\n")

def process_packet(packet):
    if IP in packet:
        src_ip = packet[IP].src
        dst_ip = packet[IP].dst
        if TCP in packet:
            src_port = packet[TCP].sport
            dst_port = packet[TCP].dport
            payload = bytes(packet[TCP].payload)
            if dst_port in [80, 21] or src_port in [80, 21]:
                info = f"IP {src_ip}:{src_port} → {dst_ip}:{dst_port} | Payload: {payload[:50]}"
                print(info)
                log_packet(info)

print("[*] Starting packet sniffing... Press Ctrl+C to stop.")
sniff(filter="tcp", prn=process_packet, store=False)
EOF
    chmod +x "$SNIF_SCRIPT"
fi

# ────────────────────────
# Run the sniffer
# ────────────────────────
echo "[*] Running packet sniffer..."
sudo python3 "$SNIF_SCRIPT"

# ────────────────────────
# Completion message
# ────────────────────────
echo "[+] Packet sniffing session finished. Logs saved at $LOG_FILE"
```

^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location

Home | Windows Server 2025 | Windows 7 x64 | linux

1 2 3 4

root@Rithik: /home/rithik

File Actions Edit View Help

```
  GNU nano 8.2                                          run_sniffer.sh
#!/bin/bash
# ═══════════════════════════════════════════════════════════════════
# Packet Sniffer Project Wrapper
# Author: Your Name
# Description: Sets up and runs the Python packet sniffer
# ═══════════════════════════════════════════════════════════════════

# ──────────────────────────────
# Project Config
# ──────────────────────────────
PROJECT_DIR="$HOME/network_sniffer"
SNIF_SCRIPT="$PROJECT_DIR/packet_sniffer.py"
LOG_FILE="$PROJECT_DIR/packet_logs.txt"

# ──────────────────────────────
# Ensure project directory exists
# ──────────────────────────────
mkdir -p "$PROJECT_DIR"
cd "$PROJECT_DIR" || exit 1

# ──────────────────────────────
# Check if sniffer script exists
# ──────────────────────────────
if [ ! -f "$SNIF_SCRIPT" ]; then
    echo "[!] packet_sniffer.py not found in $PROJECT_DIR"
    echo "[*] Creating default sniffer script..."
    cat > "$SNIF_SCRIPT" <<'EOF'
#!/usr/bin/env python3
from scapy.all import sniff
from scapy.layers.inet import IP, TCP
from datetime import datetime

LOG_FILE = "packet_logs.txt"

def log_packet(packet_info):
    timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    with open(LOG_FILE, "a") as f:
        f.write(f"[{timestamp}] {packet_info}\n")

def process_packet(packet):
    if IP in packet:
        src_ip = packet[IP].src
        dst_ip = packet[IP].dst
```