



CLASS ACTIVITY

Developing a Sniffer

CS4061

Ethical Hacking Concepts & Practices

Submitted by: Muhammad Azan Afzal

Roll number: 22I-1741

Date: April 27, 2025



Table of Contents

• Introduction	3
• Steps	3
1. Objectives.....	3
2. System Design	3
2.1 Technologies Used	3
2.2 Architecture Overview	3
3. Code Overview	4
4. Screenshots and Testing Results.....	4
4.1 Interface Selection Prompt	5
4.2 Packet Capture in Progress	6
4.3 Captured Packets Log File	7
5. Ethical Considerations.....	8
6. Future Improvements	8
• CONCLUSION	8
• References	9
End of Report	9



National University of Computer and Emerging Sciences

Islamabad Campus

• Introduction

Network traffic analysis and packet sniffing are crucial components of ethical hacking and network security. Packet sniffers allow security professionals to monitor and analyze network communication to detect vulnerabilities, diagnose network issues, and ensure system integrity. In this quiz, the objective was to develop a custom packet sniffing tool in Python capable of capturing, parsing, and optionally filtering network traffic from both wired (Ethernet) and wireless (Wi-Fi) interfaces. This practical exercise provided hands-on experience in packet analysis, the OSI model, and network troubleshooting methodologies within a secure and ethical environment.

• Steps

1. Objectives

- To design and implement a custom packet sniffing tool using Python.
- To capture all network traffic, including broadcast, unicast, and multicast packets, from both wired (Ethernet) and wireless (Wi-Fi) interfaces.
- To parse captured packets and analyze critical fields at different OSI layers.
- To log captured packet information for future analysis.
- To demonstrate optional filtering capabilities based on protocols like HTTP, DNS, ARP, TCP, etc.
- To ensure the tool operates ethically within a controlled test environment.

2. System Design

2.1 Technologies Used

- **Programming Language:** Python 3
- **Libraries:** Scapy (for packet capturing and parsing)
- **Operating System:** Kali Linux (root access utilized for sniffing)
- **Network Modes:**
 - **Monitor Mode** (for Wi-Fi interfaces)
 - **Promiscuous Mode** (for Ethernet interfaces)

2.2 Architecture Overview

- **User Interface:** Command-line based.
- **Interface Selection:** User provides the network interface (e.g., wlan0mon, eth0).
- **Packet Capturing:**
 - Sniff packets using Scapy's sniff() method.



National University of Computer and Emerging Sciences Islamabad Campus

- Capture all types of packets (broadcast, multicast, unicast).
- **Packet Parsing:**
 - Identify and extract fields from Ethernet, IP, TCP, UDP, and ARP layers.
 - Display source MAC/IP, destination MAC/IP, and protocol type.
- **Logging:**
 - Packet summary is written to captured_packets_log.txt file.
- **Filtering Support:**
 - Users can apply custom BPF filters to capture specific traffic types.
- **Graceful Exit:**
 - Handles KeyboardInterrupt (CTRL+C) and other runtime errors.

3. Code Overview

The code comprises the following main components:

- **Root Check:**
 - The tool checks if it is run with root privileges, essential for sniffing operations.
- **Main Function:**
 - Prompts the user for network interface.
 - Asks if the user wants to apply a BPF filter.
 - Initiates packet sniffing.
- **Packet Callback Function:**
 - For each captured packet:
 - Check and extract information from Ethernet, IP, TCP, UDP, and ARP layers.
 - Print relevant fields (e.g., source MAC/IP, destination MAC/IP, protocol type).
 - Append a summary of the packet to a text file.
- **Packet Sniffing:**
 - sniff(iface=interface, prn=packet_callback, filter=bpf_filter, store=0)
- **Error Handling:**
 - Displays meaningful errors if the interface is incorrect or if there are permission issues.

4. Screenshots and Testing Results

Testing was conducted in a controlled virtual lab setup using Kali Linux to generate network traffic.



National University of Computer and Emerging Sciences Islamabad Campus

4.1 Interface Selection Prompt

The screenshot shows a Kali Linux virtual machine running on VMware Workstation. The main window displays a terminal with a Python script named `packet-sniffer.py` being executed. The script uses Scapy to sniff network traffic on a specified interface. The user has entered `eth0` as the network interface. The script prompts for a BPF filter, and the user has entered `udp`. The script then starts capturing packets on `eth0`. A secondary window titled `captured_packets_log.txt` is also visible, showing the output of the script.

```
1 from scapy.all import sniff, Ether, IP, TCP, UDP, ARP
2 import sys
3 import os
4
5 def packet_callback(packet):
6     print("\n===== New Packet Captured =====")
7
8     if packet.haslayer(Ether):
9         eth_layer = packet[Ether]
10        print(f"Ethernet Frame: {eth_layer}")
11
12    if packet.haslayer(IP):
13        ip_layer = packet[IP]
14        print(f"IP Packet: {ip_layer.src}")
15
16    if packet.haslayer(TCP):
17        print("Protocol: TCP")
18    elif packet.haslayer(UDP):
19        print("Protocol: UDP")
20    elif packet.haslayer(ARP):
21        print("Protocol: ARP")
22    else:
23        print("Protocol: Other")
24
25    # Save to log file
26    with open("captured_packets_log.txt", "a") as log_file:
27        log_file.write(packet.summary() + "\n")
28
29 def main():
30     print("\n=====")
31     print("Simple Python Packet Sniffer")
32     print("=====")
33
34     interface = input("Enter the network interface to sniff on (e.g., eth0, wlan0mon): ")
35
36     # Optional: filter by protocol
37     apply_filter = input("Do you want to filter packets? (y/n): ")
38     if apply_filter.lower() == 'y':
39         bpf_filter = input("Enter BPF filter (e.g., tcp, udp, arp, port 80): ")
40         sniff(interface, filter=bpf_filter, prn=packet_callback)
41     else:
42         sniff(interface, prn=packet_callback)
43
44 if __name__ == '__main__':
45     main()
```

Terminal Output:

```
kali@kali: ~/Desktop
$ sudo python packet-sniffer.py
[sudo] password for kali:
Simple Python Packet Sniffer
Enter the network interface to sniff on (e.g., eth0, wlan0mon): eth0
Do you want to filter packets? (y/n): y
Enter BPF filter (e.g., tcp, udp, arp, port 80): udp
[*] Starting packet capture on eth0... Press CTRL+C to stop.
```

The screenshot shows a Kali Linux virtual machine running on VMware Workstation. The terminal window displays the following commands and output:

```
[sudo] password for kali:
Simple Python Packet Sniffer

Enter the network interface to sniff on (e.g., eth0, wlanmon): eth0

Do you want to filter packets? (y/n): y
Enter UDP filter (e.g., tcp, udp, arp, port 80): udp

[*] Starting packet capture on eth0... Press CTRL+C to stop.

===== New Packet Captured =====
Ethernet Frame: 00:0c:29:59:40:40 -> 00:50:56:ee:34:aa
IP Packet: 192.168.8.128 -> 192.168.8.2
Protocol: UDP

===== New Packet Captured =====
Ethernet Frame: 00:0c:29:59:40:40 -> 00:50:56:ee:34:aa
IP Packet: 192.168.8.128 -> 192.168.8.2
Protocol: UDP

===== New Packet Captured =====
Ethernet Frame: 00:50:56:ee:34:aa -> 00:0c:29:59:40:40
IP Packet: 192.168.8.2 -> 192.168.8.128
Protocol: UDP

===== New Packet Captured =====
Ethernet Frame: 00:50:56:ee:34:aa -> 00:0c:29:59:40:40
IP Packet: 192.168.8.2 -> 192.168.8.128
Protocol: UDP

===== New Packet Captured =====
Ethernet Frame: 00:0c:29:59:40:40 -> 00:50:56:ee:34:aa
IP Packet: 192.168.8.128 -> 192.168.8.2
Protocol: UDP

===== New Packet Captured =====
Ethernet Frame: 00:0c:29:59:40:40 -> 00:50:56:ee:34:aa
IP Packet: 192.168.8.128 -> 192.168.8.2
Protocol: UDP

===== New Packet Captured =====
Ethernet Frame: 00:50:56:ee:34:aa -> 00:0c:29:59:40:40
IP Packet: 192.168.8.2 -> 192.168.8.128
Protocol: UDP
```

At the bottom of the terminal, a message reads: "To direct input to this VM, move the mouse pointer inside or press Ctrl+G."

Page 6 of 9



National University of Computer and Emerging Sciences Islamabad Campus

4.3 Captured Packets Log File

```
kali-linux-2024.4-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
kali-linux-2024.4
Windows 7 x64
Ubuntu 64-bit
Windows 10 x64
FLARE VM
Windows 11 x64
--Desktop/captured_packets_log.txt (Read Only) - Mousepad
packet-sniffer.py
captured_packets_log.txt
1 Ether / IP / TCP 192.168.8.128:58966 > 58.65.206.9:80 S
2 Ether / IP / TCP 58.65.206.9:80 > 192.168.8.128:58966 SA / Padding
3 Ether / IP / TCP 192.168.8.128:58966 > 58.65.206.9:80 A
4 Ether / IP / TCP 192.168.8.128:58966 > 58.65.206.9:80 PA / Raw
5 Ether / IP / TCP 58.65.206.9:80 > 192.168.8.128:58966 A / Padding
6 Ether / IP / TCP 58.65.206.9:80 > 192.168.8.128:58966 PA / Raw
7 Ether / IP / TCP 192.168.8.128:58966 > 58.65.206.9:80 A
8 Ether / IP / TCP 192.168.8.128:54268 > 34.107.221.82:80 S
9 Ether / IP / TCP 192.168.8.128:58746 > 172.217.17.67:80 S
10 Ether / IP / TCP 34.107.221.82:80 > 192.168.8.128:54268 SA / Padding
11 Ether / IP / TCP 192.168.8.128:54268 > 34.107.221.82:80 A
12 Ether / IP / TCP 192.168.8.128:54268 > 34.107.221.82:80 PA / Raw
13 Ether / IP / TCP 34.107.221.82:80 > 192.168.8.128:54268 A / Padding
14 Ether / IP / TCP 172.217.17.67:80 > 192.168.8.128:58746 SA / Padding
15 Ether / IP / TCP 192.168.8.128:58746 > 172.217.17.67:80 A
16 Ether / IP / TCP 192.168.8.128:58746 > 172.217.17.67:80 PA / Raw
17 Ether / IP / TCP 172.217.17.67:80 > 192.168.8.128:58746 A / Padding
18 Ether / IP / TCP 34.107.221.82:80 > 192.168.8.128:54268 PA / Raw
19 Ether / IP / TCP 192.168.8.128:54268 > 34.107.221.82:80 A
20 Ether / IP / TCP 192.168.8.128:58966 > 58.65.206.9:80 PA / Raw
21 Ether / IP / TCP 58.65.206.9:80 > 192.168.8.128:58966 A / Padding
22 Ether / IP / TCP 172.217.17.67:80 > 192.168.8.128:58746 PA / Raw
23 Ether / IP / TCP 192.168.8.128:58746 > 172.217.17.67:80 A
24 Ether / IP / TCP 58.65.206.9:80 > 192.168.8.128:58966 PA / Raw
25 Ether / IP / TCP 192.168.8.128:58966 > 58.65.206.9:80 A
26 Ether / IP / TCP 192.168.8.128:41710 > 58.65.206.10:80 S
27 Ether / IP / TCP 58.65.206.10:80 > 192.168.8.128:41710 SA / Padding
28 Ether / IP / TCP 192.168.8.128:41710 > 58.65.206.10:80 A
29 Ether / IP / TCP 192.168.8.128:41710 > 58.65.206.10:80 PA / Raw
30 Ether / IP / TCP 58.65.206.10:80 > 192.168.8.128:41710 A / Padding
31 Ether / IP / TCP 58.65.206.10:80 > 192.168.8.128:41710 PA / Raw
32 Ether / IP / TCP 192.168.8.128:41710 > 58.65.206.10:80 A
33 Ether / IP / TCP 192.168.8.128:58746 > 172.217.17.67:80 PA / Raw
34 Ether / IP / TCP 172.217.17.67:80 > 192.168.8.128:58746 A / Padding
35 Ether / IP / TCP 192.168.8.128:41710 > 58.65.206.10:80 PA / Raw
36 Ether / IP / TCP 58.65.206.10:80 > 192.168.8.128:41710 A / Padding
37 Ether / IP / TCP 58.65.206.10:80 > 192.168.8.128:41710 PA / Raw
38 Ether / IP / TCP 192.168.8.128:41710 > 58.65.206.10:80 A
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
Links
4:57 PM
```

```
kali-linux-2024.4-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
kali-linux-2024.4
Windows 7 x64
Ubuntu 64-bit
Windows 10 x64
FLARE VM
Windows 11 x64
--Desktop/captured_packets_log.txt (Read Only) - Mousepad
packet-sniffer.py
captured_packets_log.txt
380 Ether / IP / UDP 24.36.137.203:80 > 192.168.8.128:54541 / Raw
381 Ether / IP / UDP 192.168.8.128:54541 > 24.36.137.203:80 / Raw
382 Ether / IP / UDP 24.36.137.203:80 > 192.168.8.128:54541 / Raw
383 Ether / IP / UDP / DNS Qry b'example.com.'
384 Ether / IP / UDP / DNS Ans b'example.com.'
391 Ether / IP / UDP / DNS Ans 23.192.228.84
392 Ether / IP / UDP / DNS Ans 2000.1406.1406.21:1726
393 Ether / IP / UDP 192.168.8.128:49598 > 96.7.128.175:80 / Raw
394 Ether / IP / UDP 192.168.8.128:49598 > 96.7.128.175:80 / Raw
395 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
396 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
397 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
398 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
399 Ether / IP / UDP 192.168.8.128:49598 > 96.7.128.175:80 / Raw
400 Ether / IP / UDP 192.168.8.128:49598 > 96.7.128.175:80 / Raw
401 Ether / IP / UDP 192.168.8.128:49598 > 96.7.128.175:80 / Raw
402 Ether / IP / UDP 192.168.8.128:49598 > 96.7.128.175:80 / Raw
403 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
404 Ether / IP / UDP 192.168.8.128:49598 > 96.7.128.175:80 / Raw
405 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
406 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
407 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
408 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
409 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
410 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
411 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
412 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
413 Ether / IP / UDP 192.168.8.128:49598 > 96.7.128.175:80 / Raw
414 Ether / IP / UDP 192.168.8.128:49598 > 96.7.128.175:80 / Raw
415 Ether / IP / UDP / DNS Qry b'firefox.settings.services.mozilla.com.'
416 Ether / IP / UDP / DNS Ans b'prod.settings.prod.services.mozilla.com.'
417 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
418 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
419 Ether / IP / UDP 192.168.8.128:49598 > 96.7.128.175:80 / Raw
420 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
421 Ether / IP / UDP 192.168.8.128:49598 > 96.7.128.175:80 / Raw
422 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
423 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
424 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
425 Ether / IP / UDP 192.168.8.128:49598 > 96.7.128.175:80 / Raw
426 Ether / IP / UDP 96.7.128.175:80 > 192.168.8.128:49598 / Raw
427 Ether / IP / UDP / HTTPTagman / SMO_header / Trun b'\\MAILSLOT\\BROWSE' HostAnnouncement for b'DESKTOP-8637583'
428
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
Links
4:57 PM
```




5. Ethical Considerations

- The packet sniffer was exclusively used in an isolated, personal test lab setup.
- No traffic from unauthorized or external networks was captured.
- Monitor mode and promiscuous mode were enabled only on personally owned hardware.
- Capturing packets on public or unauthorized networks is illegal and unethical.
- The primary focus remained on educational and ethical hacking practices, in line with course guidelines.

6. Future Improvements

- **User Interface:**
 - Implement a graphical user interface (GUI) using Tkinter or PyQt5 for better usability.
- **Advanced Protocol Analysis:**
 - Reconstruct full HTTP sessions and extract login data (for ethical demonstration only).
- **Deep Packet Inspection:**
 - Analyze packet payloads in detail for specific application data.
- **Database Integration:**
 - Store captured packets and metadata into a SQL/NoSQL database for better searchability.
- **Real-Time Alerts:**
 - Notify user if suspicious traffic patterns are detected (e.g., ARP spoofing attempts).
- **Wireless Management Frame Detection:**
 - Detect Wi-Fi probe requests, beacon frames, and disassociation frames for wireless reconnaissance.

• CONCLUSION

The packet sniffer tool was successfully developed and demonstrated. It captured real-time traffic from both wired and wireless interfaces, parsed key layers, displayed critical information, and logged the packet summaries. The project helped reinforce key networking concepts like OSI layers, protocols, and packet structures. All operations were conducted ethically within isolated testing environments. Future improvements can make the tool more robust, user-friendly, and powerful for advanced network security analysis.



National University of Computer and Emerging Sciences Islamabad Campus

- **References**

- Scapy Documentation. (n.d.). Retrieved from: <https://scapy.readthedocs.io/>
- Kurose, J. F., & Ross, K. W. (2017). Computer Networking: A Top-Down Approach (7th Edition). Pearson.

End of Report