(CYBER SECURITY INTERNSHIP REPORT)

[CodeAlpha]

DECEMBER 10, 2024

[DHRUMIL PATEL]

[CodeAlpha DECEMBER]

Task 1 (Network Sniffer):

Dhrumil Patel

Network Sniffer Documentation

INTRODUCTION

This Network Sniffer documentation provides an overview of the functionality, setup, and operation of the Python-based network sniffering packets. The network sniffer captures and analyzes the network traffic, offering insights to the how data flows through the network packets and the structure of network packets. This tool is essential for the understanding network packets behavior and troubleshooting network-related issues.

PURPOSE

The primary purpose of this network sniffering packets is:

- Capture the network traffic from a specified interface or profile.
- Analyze the Ethernet, Wi-Fi, IP, TCP, and UDP packets.
- Provide the visibility to packet-level communication for educational and diagnostic purposes.

Dhrumil Patel

KEY FUNCTIONALITIES

1. **Packet Capture:**

- Captures packets from a specified network interface.
- Requires administrative/root privileges to access raw packets.

2. **Packet Analysis:**

- Extracts Ethernet frame details (e.g., MAC addresses).
- Analyzes IP packets for source and destination addresses and protocol types.
- Examines TCP/UDP segments for port information.

3. **Real-Time Processing:**

- Processes packets in real-time and displays information to the user.

KEY FUNCTIONALITIES

1. **Python 3 installed on the system.**
2. **“Scapy” library to use python:**

   Install using pip:

   Bash Script
3. **Command to Install Scapy Library:**

   pip install scapy

4. **Npcap (for Windows users only):**

Download and install from [Npcap official website: https://nmap.org/npcap/]

Ensure "Install Npcap in WinPcap API-compatible mode" is selected during installation.

How It Works

1. **Input Interface or Input Profile:**

- The user will specifies the network interface or profile to sniff the traffic (e.g., “eth0", “Wi-Fi”)

2. **Start Sniffing Packets:**

- The `sniff()` function is to capturing the network packets in the real-time.
- Network Packets are processed using the “packet_sniffering” callback function.

3. **Display Network Packet Details:**

- **Ethernet:** Source and destination MAC addresses.
- **Wi-Fi:** Source and destination MAC addresses.
- **IP:** Source and destination IP addresses and protocol type.
- **TCP/UDP:** Source and destination ports.

4. Stop Network Sniffing Packets:

- **Press “Ctrl+C” to terminate the network sniffing process.**

Usage Instructions

1. **Step 1: Run the Python Script**

   Run the script with administrative/root privileges:

- **Linux/macOS:** bash script (“sudo python3 sniffer.py”)
- **Windows:** Open a command prompt as a Administrator and run: bash script (“python sniffer.py”)

2. **Step 2: Select Network Interface or profiles**

- Provide the name of the network interface or profile when the prompted (e.g.. “Wi-Fi”, “eth0”).

3. **Step 3: View Captured Network Packets**

- Captured the network packets are displayed in the real-time, and showing the Ethernet, Wi-Fi, IP, TCP, and UDP details.

Sample Output

1. **Network Packet has been Captured:**

- Ethernet Frame: Source MAC: 00:1A:2B:3C:4D:5E, Destination MAC: FF:FF:FF:FF:FF:FF

- IP Packet: Source IP: 192.168.1.100, Destination IP: 192.168.1.1

- Protocol: 6

- TCP Segment: Source Port: 57684, Destination Port: 80

Troubleshooting

1. **Permission Error:**

- Run the script as a Administrator or with “sudo”.

2. **WinPcap/Npcap Not Installed**

- Install the Npcap on Windows to enable the packet sniffing at Layer 2 on OSI Model.

3. **Invalid Interface or Profiles**

- Ensure the interface or profile name matches one listed in “ifconfig” (Linux/macOS) or “ipconfig” (Windows).

Conclusion

This network sniffer packets are provides valuable insights into network packet traffic, allowing the users to analyze the data flows and packet structures effectively. It is the excellent tool for the network troubleshooting and understanding the fundamentals of the networking concepts.