

# Code Alpha Project

## Task – 1

### Basic Network Sniffer

Creating a basic network sniffer in Python involves using a library like Scapy to capture and analyze network packets. This basic network sniffer captures IP packets and extracts information such as source and destination IP addresses, as well as protocol details. It also checks if the packet contains TCP or UDP information and prints the corresponding source and destination ports.

Keep in mind that building network tools should be done responsibly and in compliance with relevant laws and policies.

#### 1. Command – apt install python3

```
File Actions Edit View Help
(kali@kali)~$ sudo su
[sudo] password for kali:
(kali@kali)~$ apt install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.11.2-1+b1).
python3 set to manually installed.
The following packages were automatically installed and are no longer required:
  bluez-firmware firmware-ath9k-htc firmware-atheros firmware-brcn80211 firmware-intel-sound firmware-iwlwifi firmware-libertas firmware-realtek firmware-sof-signed firmware-ti-connectivity
  firmware-zd1211 kali-linux-firmware
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
```

#### 2. Pip install scapy

```
(root@kali)~/home/kali$ pip install scapy
Requirement already satisfied: scapy in /usr/lib/python3/dist-packages (2.5.0)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

#### 3. Vi sniffer.py

```
(root@kali)~/home/kali$ vi sniffer.py

import scapy.all as scapy

def sniff(interface):
    scapy.sniff(iface=interface, store=False, prn=process_packet)

def process_packet(packet):
    if packet.haslayer(scapy.IP):
        ip_src = packet[scapy.IP].src
        ip_dst = packet[scapy.IP].dst
        protocol = packet[scapy.IP].proto

        print(f"IP Source: {ip_src}, IP Destination: {ip_dst}, Protocol: {protocol}")

    if packet.haslayer(scapy.TCP):
        src_port = packet[scapy.TCP].sport
        dst_port = packet[scapy.TCP].dport
        print(f"TCP Source Port: {src_port}, TCP Destination Port: {dst_port}")

    elif packet.haslayer(scapy.UDP):
        src_port = packet[scapy.UDP].sport
        dst_port = packet[scapy.UDP].dport
        print(f"UDP Source Port: {src_port}, UDP Destination Port: {dst_port}")

    print("\n")

# Specify the network interface to sniff on (e.g., "eth0" for Ethernet, "wlan0" for Wi-Fi)
network_interface = "eth0"

# Start sniffing
sniff(network_interface)
```

```
root@kali: /home/kali
File Actions Edit View Help
root@kali: /home/kali
python3 ./sniffer.py
IP Source: 192.168.217.132, IP Destination: 192.168.217.2, Protocol: 17
UDP Source Port: 39670, UDP Destination Port: 53

IP Source: 192.168.217.132, IP Destination: 192.168.217.2, Protocol: 17
UDP Source Port: 39670, UDP Destination Port: 53

IP Source: 192.168.217.2, IP Destination: 192.168.217.132, Protocol: 17
UDP Source Port: 53, UDP Destination Port: 39670

IP Source: 192.168.217.2, IP Destination: 192.168.217.132, Protocol: 17
UDP Source Port: 53, UDP Destination Port: 39670

IP Source: 192.168.217.132, IP Destination: 142.250.183.3, Protocol: 17
UDP Source Port: 57183, UDP Destination Port: 443

IP Source: 142.250.183.3, IP Destination: 192.168.217.132, Protocol: 17
UDP Source Port: 443, UDP Destination Port: 57183

IP Source: 192.168.217.132, IP Destination: 142.250.183.3, Protocol: 17
UDP Source Port: 57183, UDP Destination Port: 443

IP Source: 192.168.217.132, IP Destination: 142.250.183.3, Protocol: 6
TCP Source Port: 39346, TCP Destination Port: 443
```