Vedanth D

NETWORK PACKET SNIFFER USING PYTHON

Introduction:

Packet sniffing is the process of capturing data packets traveling over a network. This project demonstrates a simple Python-based packet sniffer using the Scapy library. It captures and analyzes basic information about each packet.

Requirements:

- Python 3.x
- Scapy library
- VS Code
- Administrative privileges (for sniffing)

Design:

- The script uses sniff() to capture live network packets.
- For each packet, it checks if the IP layer exists.
- It prints source/destination IPs and the protocol number.

CODE EXPLANATION :

Importing Scapy

We import sniff() from Scapy, which is used to capture packets on the network.

Defining process_packet()

A function to process each captured packet. It checks if the packet contains an IP layer.

Extracting Info

If it's an IP packet, we extract and print:

- Source IP
- Destination IP
- Protocol
- Summary of the packet

Start Sniffing

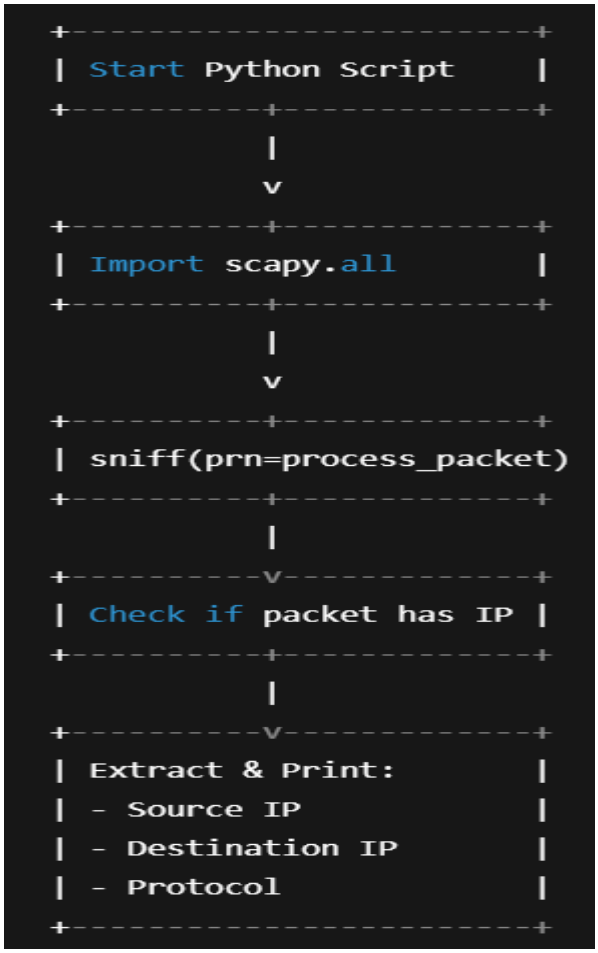
We call sniff() and tell it to use our function for each packet, and capture 10 packets in total.

```python
from scapy.all import sniff

def process_packet(packet):
    if packet.haslayer("IP"):
        ip_layer = packet["IP"]
        print("========================================")
        print(f"Packet Captured")
        print(f"Source IP: {ip_layer.src}")
        print(f"Destination IP: {ip_layer.dst}")
        print(f"Protocol: {ip_layer.proto}")
        print(f"Packet Summary: {packet.summary()}")
        print("========================================")

# Start sniffing packets
print("Starting the Network Sniffer...")
sniff(prn=process_packet, count=10)
```

DATA FLOW :



OUTPUT :

```
C:\Windows\System32>E:

E:\>cd network_sniffer_task

E:\network_sniffer_task>python sniffer.py
Starting the Network Sniffer...
========================================
Packet Captured
Source IP: 192.168.206.101
Destination IP: 192.168.206.104
Protocol: 17
Packet Summary: Ether / IP / UDP / DNS Qry b'www.bing.com.'
========================================
========================================
```

Conclusion:

The project successfully demonstrates how network packets can be captured and analyzed in real-time using Python. It provides a hands-on understanding of network-level communication.