Task 2 – Network Packet Sniffer

Description

A Network Packet Sniffer is a cybersecurity tool that monitors and analyzes network traffic in real time. It captures the packets traveling through a network, allowing security professionals to identify vulnerabilities, suspicious activities, or unauthorized access.

Tools Used

- Python 3
- Scapy Library (for packet capture and analysis)
- Wireshark (optional for traffic verification)
- Command Prompt / Terminal

Python Code

```
from scapy.all import sniff

def packet_callback(packet):
    print(packet.summary())

print("Starting Network Packet Sniffer...")
sniff(prn=packet_callback, count=10)
print("Sniffing complete.")
```

Code Explanation

- The sniff() function captures live network packets.
- The parameter prn=packet_callback means each captured packet is processed by the callback function.
- packet.summary() prints a short summary of each packet (source, destination, and protocol).
- The count=10 captures 10 packets (you can increase or remove this for continuous monitoring).

Output Example

Starting Network Packet Sniffer...
Ether / IP / TCP 192.168.1.5:50512 > 142.250.180.78:https S
Ether / IP / UDP 192.168.1.5:5353 > 224.0.0.251:mdns
Ether / IP / ICMP 192.168.1.5 > 8.8.8.8 echo-request

Sniffing complete.

Uses

- Detecting unauthorized devices or users on a network.
- Monitoring real-time traffic for performance and security analysis.
- Identifying network attacks (DoS, spoofing, etc.).
- Educational use to understand TCP/IP communication.

Summary

The Network Packet Sniffer is an essential tool for network administrators and cybersecurity learners. It helps visualize how data travels across networks and assists in detecting malicious or unusual activities. Using Python and Scapy makes it efficient, customizable, and perfect for practical cybersecurity projects.