Cybersecurity Project: Packet Sniffer using Scapy

Project Title: Basic Packet Sniffer using Scapy for Network Monitoring

Code Breakdown & Explanation

1. Importing Required Modules

- Scapy is a powerful Python library used for packet crafting, sniffing, and analysis.

- IP, TCP, UDP, ICMP represent core network protocols for packet inspection.

2. Packet Processing Function

- Called for every captured packet via sniff().

3. Check for IP Layer

- Filters out non-IP packets (like ARP).

4. Extract IP Info

- Extracts source/destination IP and protocol number.

5. Convert Protocol Number to Name

- Maps 1=ICMP, 6=TCP, 17=UDP for human readability.

6. Print Basic Packet Info

- Displays captured packet details clearly.

7. Extract and Print Payload (TCP/UDP)

- Checks for payload in TCP/UDP and attempts to decode it.

- Displays human-readable text or notes non-text data.

8. Handle ICMP Packets

- Simple message indicating ICMP (e.g., ping).

9. Start Sniffing

- sniff() captures packets using BPF filter "ip".

- process_packet handles each packet.

Cybersecurity Use Cases

Use Case	How It Helps
Network Monitoring	Inspect real-time traffic
Intrusion Detection	Identify unusual or malicious traffic
Learning Tool	Understand packet structure and flow
Debugging	Analyze broken connections or packet loss
IDS/IPS Foundation	Can be extended into intrusion detection systems

Project Extension Ideas

- Save output to a log file

- Add packet timestamps or count

- **Detect suspicious IPs or payloads**
- **Apply regex to search for patterns (e.g., credentials)**
- **GUI for visualization**