**🛡️Basic Network Packet Sniffer using Python & Scapy**

A simple yet powerful **network packet sniffer** built with Python and Scapy for real-time monitoring of IP packets, protocol types, and payload data. This tool is useful for cybersecurity learning, traffic analysis, and ethical hacking practices.  
  
**📌 Table of Contents**

* [Project Overview](#project-overview)
* [Features](#features)
* [Technologies Used](#technologies-used)
* [System Requirements](#system-requirements)
* [Installation & Setup](#installation--setup)
* [Running the Sniffer](#running-the-sniffer)
* [Sample Output](#sample-output)
* [Learning Outcomes](#learning-outcomes)
* [Legal Disclaimer](#legal-disclaimer)

**🚀 Project Overview**

This project captures real-time network traffic using the scapy library and extracts:

* Source & Destination IP Addresses
* Protocol type (TCP, UDP, ICMP)
* Payload data (if available)

It helps learners understand how data flows through a network, and how cybersecurity tools like Wireshark work at the packet level.

**✨ Features**

* Real-time packet capturing
* Protocol detection: TCP, UDP, ICMP
* Payload extraction (Raw data)
* Console-based output
* Extendable and customizable
* **🧰 Technologies Used**

1. Python 3.x

2. Scapy

3. Npcap

4. Terminal

**⚙️ System Requirements**

* ✅ Python 3.8 or above
* ✅ scapy installed (pip install scapy)
* ✅ Npcap installed on Windows (Download Npcap)
* ✅ Administrator privileges to run the sniffer

🔧 Installation & Setup

Step 1: Clone the Repository

git clone https://github.com/SMOZHIVARMAN/Basic\_Network\_Packet\_Sniffer.git

cd packet\_sniffer

Step 2: Install Dependencies

pip install scapy

**Step 3: Install Npcap (For Windows Users Only)**

Download from: https://npcap.com/#download  
During installation:

* ✅ Enable “WinPcap API-compatible mode”
* ✅ Run as Admin

**▶️ Running the Sniffer**

**Windows:**

Open **Command Prompt as Administrator**:

python packet\_sniffer.py

**Linux/macOS:**

Use **sudo** in terminal:

sudo python3 packet\_sniffer.py

🧪 Sample Output

Starting packet capture...... Press Ctrl+C to stop !!

Source: 192.168.1.10 -> Destination: 142.250.195.68 | Protocol: TCP

Payload: b'GET / HTTP/1.1\r\nHost: google.com\r\n...'

--------------------------------------------------------------------------------

Source: 192.168.1.10 -> Destination: 8.8.8.8 | Protocol: ICMP

--------------------------------------------------------------------------------

**📘 Learning Outcomes**

* Understand TCP/IP and OSI layers
* Learn how packet sniffers work
* Extract IP, protocol, and payload info from packets
* Practice raw socket-level programming using Python
* Learn how tools like Wireshark, tcpdump, and IDS work under the hood

**⚠️ Legal Disclaimer**

This tool is created strictly for **educational and ethical use only**.  
Do **NOT** use it on public networks or systems without proper authorization.  
Unauthorized use of packet sniffers can be considered **illegal** in many regions.

**📎 License**

This project is open source and available under the MIT License.

**🙌 Contributions Welcome!**

Feel free to fork this repository and add:

* Logging functionality
* GUI using Tkinter/PyQt
* .pcap saving support
* Filters by port/protocol