**Basic Network Sniffer**

Build a network sniffer in Python that captures and analyzes network traffic. This project will help you understand how data flows on a network and how network packets are structured.

* **What Is This Program?**

This is a packet sniffer. It listens to network traffic on a specific interface (like eth0 or wlan0) and shows you detailed info about each packet that passes through.

You’ll see:

* Source & destination IP and MAC addresses
* Protocols (TCP, UDP, ICMP, etc.)
* Port numbers
* And more...

It’s similar to Wireshark, but in the terminal and written in Python.

* **Libraries Used**

**1. scapy**

* A Python library used for network packet crafting and sniffing.
* It reads packets like a pro (used in hacking, testing, and security tools).

**2. psutil**

* Helps get information about your system’s network interfaces (like IP, MAC).

**3. prettytable**

* Just used to display output in table format.

**4. colorama**

* Adds colors to the terminal output.

**5. subprocess + re (regex)**

* Runs Linux commands (ifconfig) and pulls MAC/IP addresses from the result.

| **Part** | **What it does** |
| --- | --- |
| ip\_table() | Shows available network interfaces with IP and MAC |
| get\_current\_ip() | Pulls your IP using ifconfig |
| sniff(interface) | Starts capturing packets on the chosen interface |
| packet\_callback() | Processes each packet and prints details like IPs, ports, flags, etc. |







