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.

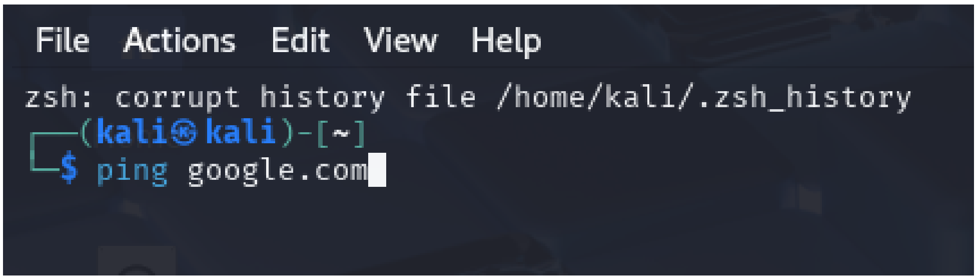
4. colorama

- Adds colors to the terminal output.

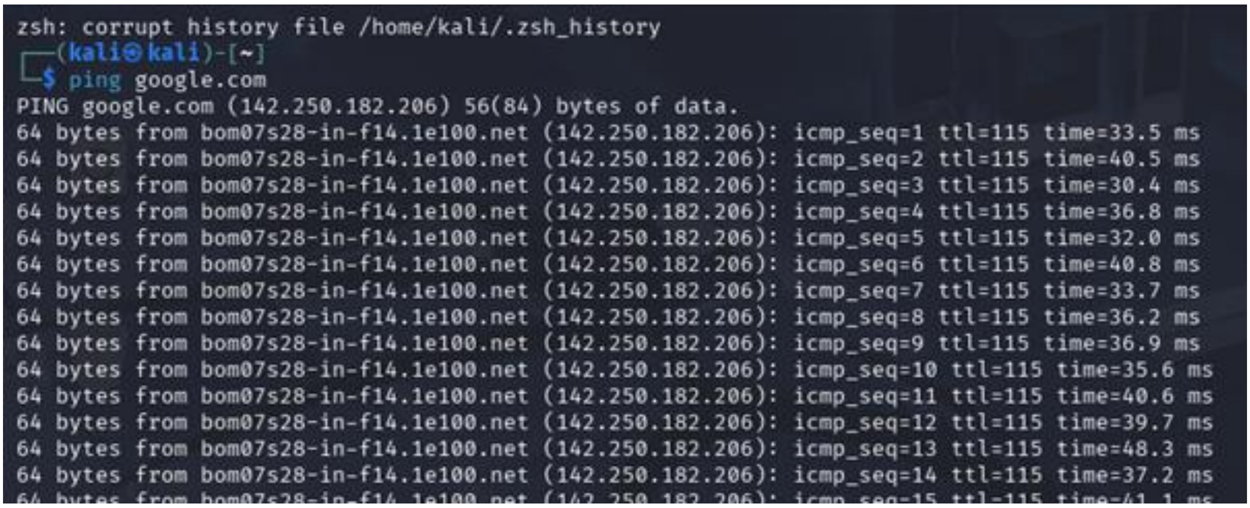
5. subprocess + re (regex)

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

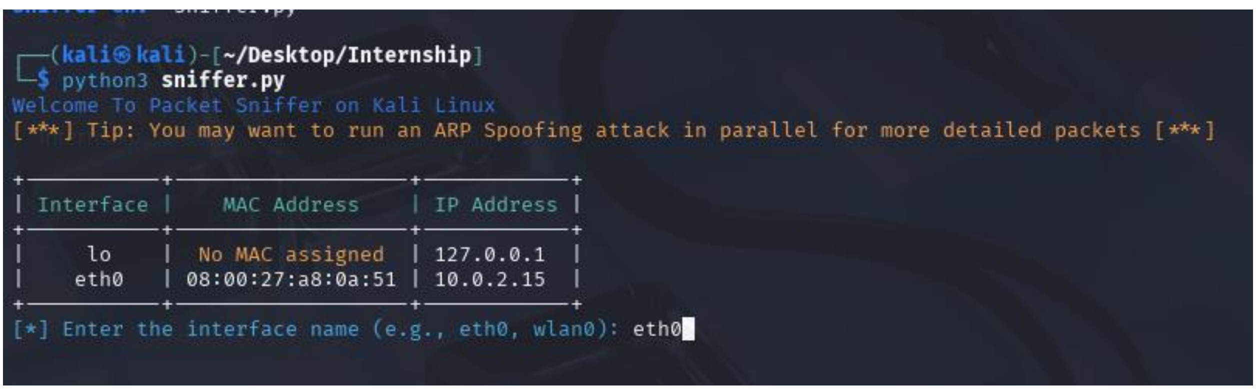
Part	What it does
ip_table()	Shows available network interfaces with IP and MAC
get_current_ip()	Pulls your IP using ifconfig
sniff(interface)	Starts capturing packets on the chosen interface
packet_callback()	Processes each packet and prints details like IPs, ports, flags, etc.



```
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ ping google.com
```

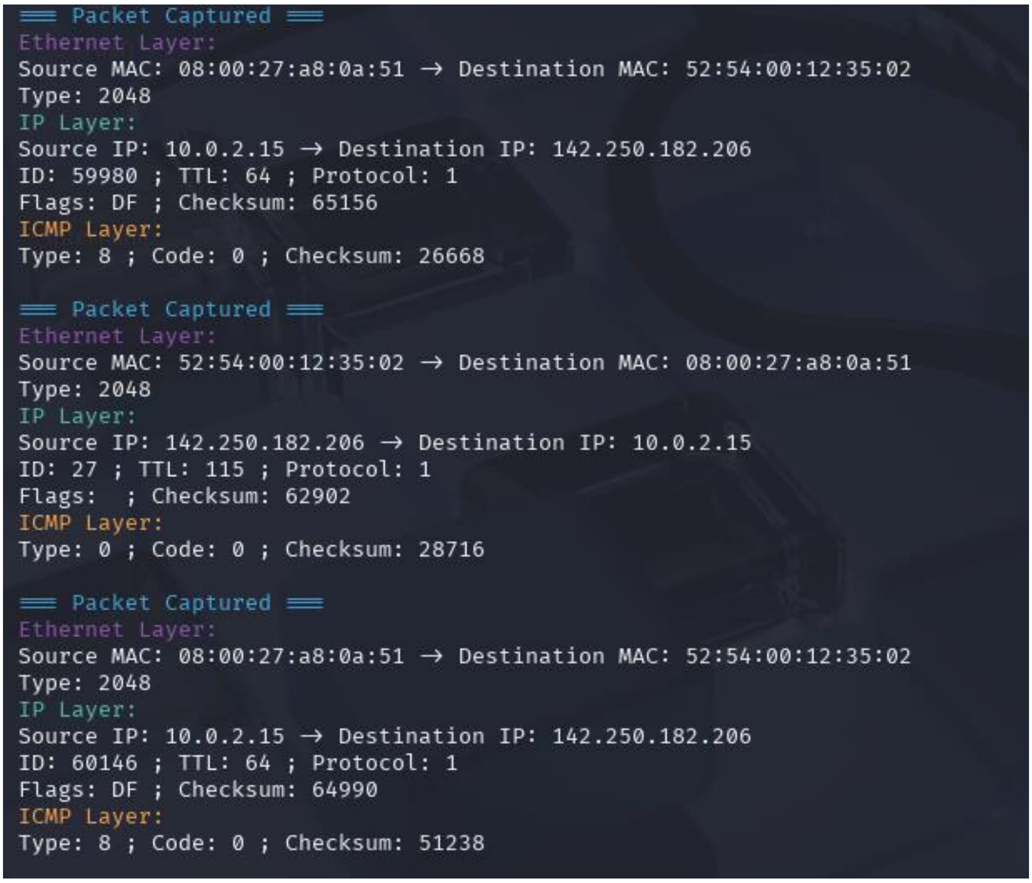


```
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ ping google.com
PING google.com (142.250.182.206) 56(84) bytes of data.
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=1 ttl=115 time=33.5 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=2 ttl=115 time=40.5 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=3 ttl=115 time=30.4 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=4 ttl=115 time=36.8 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=5 ttl=115 time=32.0 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=6 ttl=115 time=40.8 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=7 ttl=115 time=33.7 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=8 ttl=115 time=36.2 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=9 ttl=115 time=36.9 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=10 ttl=115 time=35.6 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=11 ttl=115 time=40.6 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=12 ttl=115 time=39.7 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=13 ttl=115 time=48.3 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=14 ttl=115 time=37.2 ms
64 bytes from bom07s28-in-f14.1e100.net (142.250.182.206): icmp_seq=15 ttl=115 time=41.1 ms
```



```
┌──(kali㉿kali)-[~/Desktop/Internship]
└─$ python3 sniffer.py
Welcome To Packet Sniffer on Kali Linux
[**] Tip: You may want to run an ARP Spoofing attack in parallel for more detailed packets [**]

+-----------+-------------------+------------+
| Interface |    MAC Address    | IP Address |
+-----------+-------------------+------------+
|     lo    |  No MAC assigned  | 127.0.0.1  |
|    eth0   | 08:00:27:a8:0a:51 | 10.0.2.15  |
+-----------+-------------------+------------+
[*] Enter the interface name (e.g., eth0, wlan0): eth0
```



```
══ Packet Captured ══
Ethernet Layer:
Source MAC: 08:00:27:a8:0a:51 → Destination MAC: 52:54:00:12:35:02
Type: 2048
IP Layer:
Source IP: 10.0.2.15 → Destination IP: 142.250.182.206
ID: 59980 ; TTL: 64 ; Protocol: 1
Flags: DF ; Checksum: 65156
ICMP Layer:
Type: 8 ; Code: 0 ; Checksum: 26668

══ Packet Captured ══
Ethernet Layer:
Source MAC: 52:54:00:12:35:02 → Destination MAC: 08:00:27:a8:0a:51
Type: 2048
IP Layer:
Source IP: 142.250.182.206 → Destination IP: 10.0.2.15
ID: 27 ; TTL: 115 ; Protocol: 1
Flags:  ; Checksum: 62902
ICMP Layer:
Type: 0 ; Code: 0 ; Checksum: 28716

══ Packet Captured ══
Ethernet Layer:
Source MAC: 08:00:27:a8:0a:51 → Destination MAC: 52:54:00:12:35:02
Type: 2048
IP Layer:
Source IP: 10.0.2.15 → Destination IP: 142.250.182.206
ID: 60146 ; TTL: 64 ; Protocol: 1
Flags: DF ; Checksum: 64990
ICMP Layer:
Type: 8 ; Code: 0 ; Checksum: 51238
```