This is a Python-based tool that monitors a local network for suspicious activities and automatically responds with countermeasures.

📌 **Features**

This system detects and responds to:

- 🕵️ ARP Spoofing

- 🚪 Port Scanning

- 🧑‍💻 MITM (Man-in-the-Middle) Attempts

All threats are logged in a local SQLite database (`logs.db`) and can trigger real-time alerts or automatic blocking (e.g., via firewall rules).

⚙️ **Key Components**

✅ Packet Sniffer

A live packet sniffer built using Python’s `scapy` library, used to capture and analyze real-time network traffic.

✅ ARP Spoofing Detection

Detects when the same IP address is associated with multiple MAC addresses — a key signature of ARP poisoning attacks. Logs and flags the anomaly immediately.

✅ Port Scan Detection

Detects high volumes of SYN packets from a single source to many ports, which typically indicates an Nmap or masscan port scanning attack.

✅ MITM Attempt Detection

Analyzes abnormal ARP behavior and traffic redirection patterns that may suggest the attacker has inserted themselves between the target and gateway.

✅ Active Response (WIP)

Automatically responds to confirmed threats by:

- Logging them to the database

- Sending terminal alerts

- Potential future support for IP blocking via `iptables` or `ufw`

🚨 **Real-Time Logging**

All suspicious events are logged with:

- Timestamp

- Source IP

- MAC address

- Type of attack

- Brief description (**Eg:** 2025-07-13T23:43:36.693024|192.168.1.102|00:0b:44:11:3a:b7|ARP Spoofing|Detected ARP Spoofing from IP 192.168.1.102 with MAC 00:0b:44:11:3a:b7)

Logs can be viewed or exported using the built-in terminal menu.

🧪 **Simulating an Attack**

To test your setup, use two devices (or VMs) on the same broadcast network:

1. 🖥️ Attacker: Kali Linux (host or VM)

2. 🧑‍💻 *Target*: Your Linux Mint system running this tool

For simulating an attack, we need to install the following on both the machines.

* Python 3
* scapy
* sqlite3
* bettercap, arpspoof, nmap, tcpdump

🔧 **VM Configuration**

If you're using a virtual machine:

- Set *Adapter 1* to *Bridged Adapter* mode

- Restart the VM

- Ensure both attacker and target are on the same subnet/broadcast

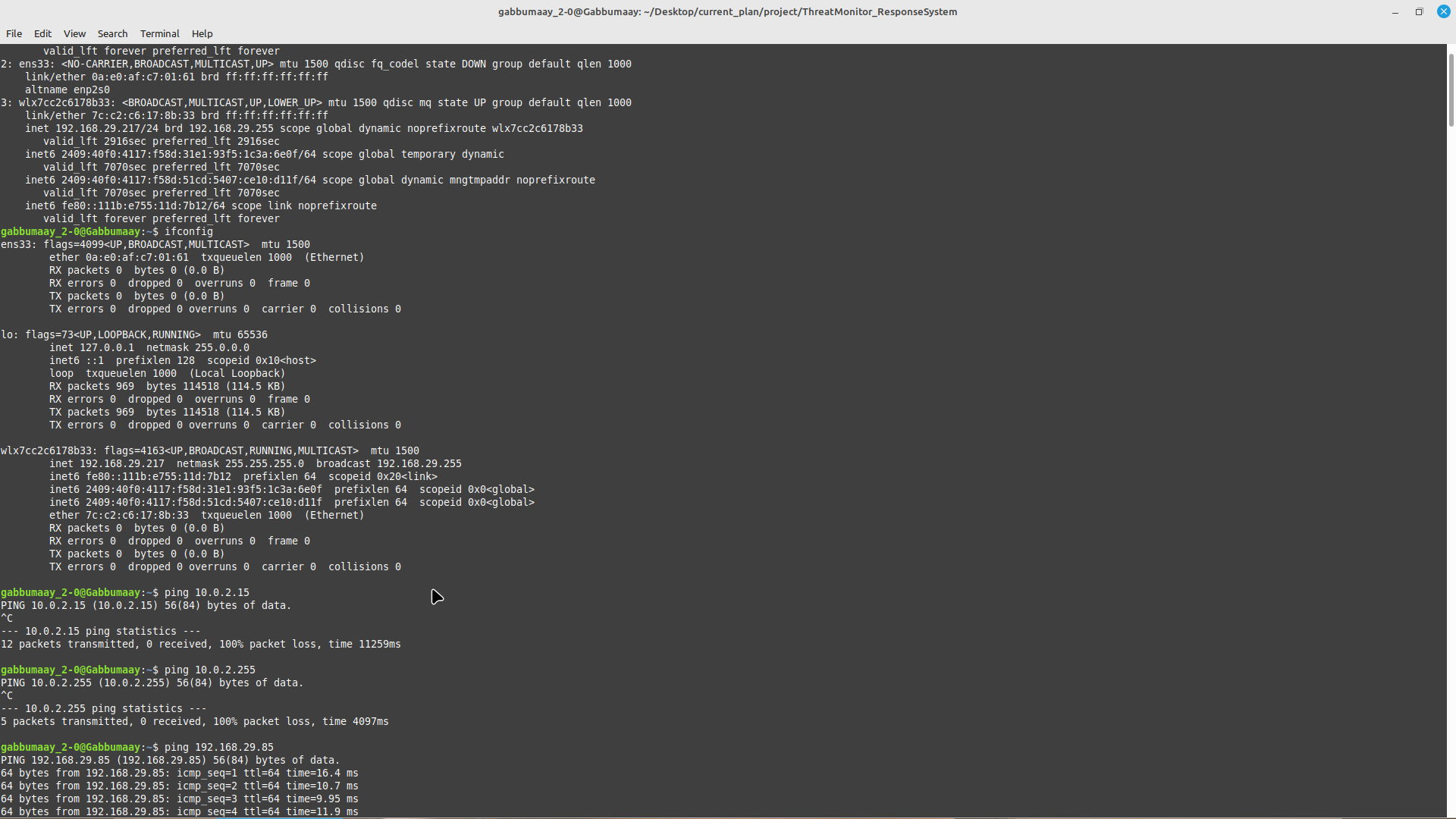
✅ Verify Network Connection

Test that the attacker and victim can see each other:

**bash**

$ping <target\_ip> # On attacker. Eg: ping 192.168.29.173.

$ping <attacker\_ip> # On target



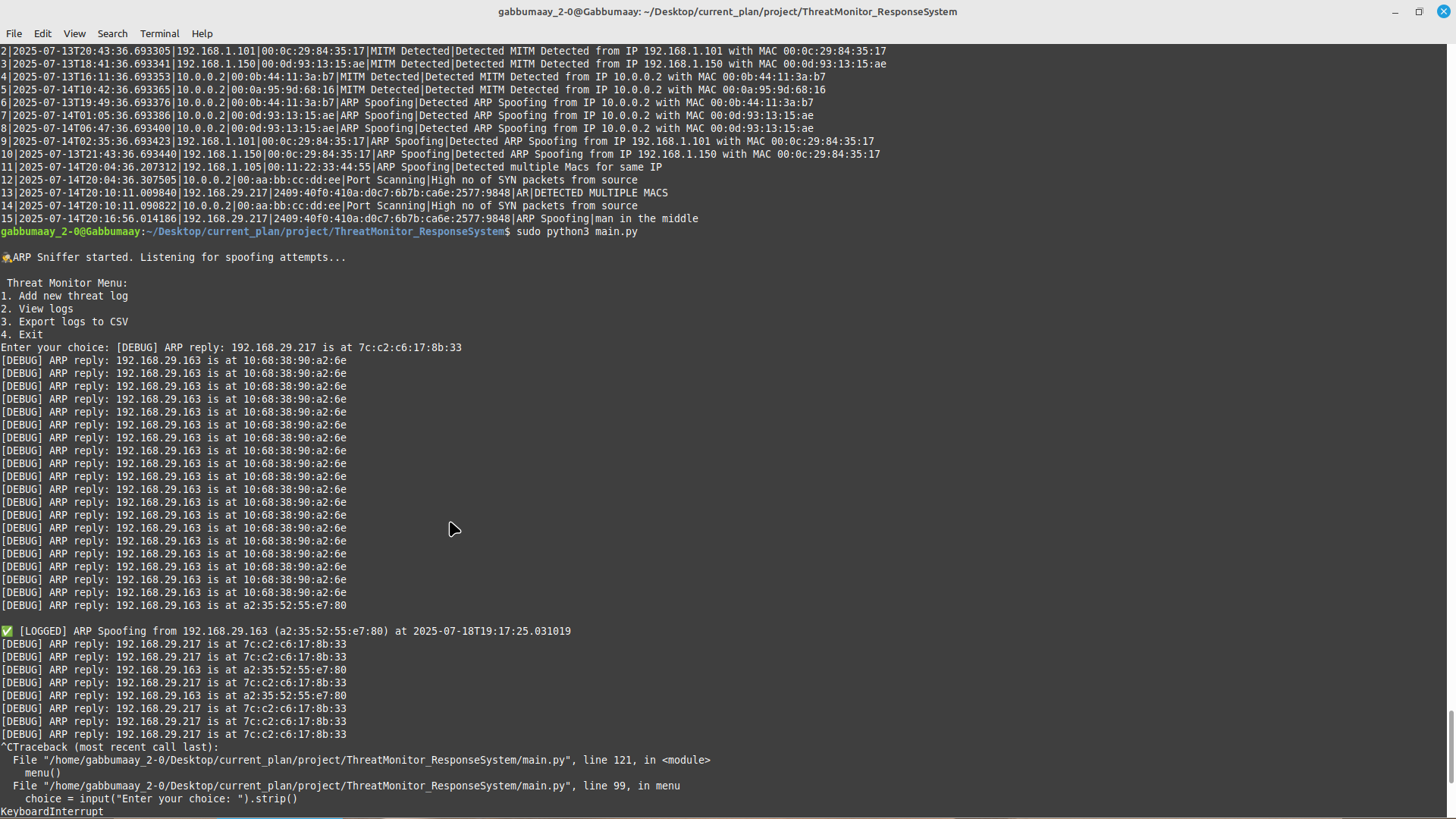
On the Attacker’s machine, we can take help from **bettercap** to perform ARP Spoofing.

* sudo bettercap -iface eth0

# inside bettercap prompt:

* set arp.spoof.targets <target\_ip>
* arp.spoof on

Before turning on the arp spoofing, run the python script(“[main.py](https://github.com/Gabbumaay/Threat-Response-System/blob/Gabbumaay-patch-1/main.py)”). You can observe change in the MAC address for the same IP and it will be logged as below.



⚠️ **Safety Guidelines & Ethical Disclaimer**

This project and report are created solely for educational and research purposes in the domain of cybersecurity. The tools, techniques, and methods described or demonstrated herein are intended to:

* Enhance understanding of network security principles
* Simulate attacks in **controlled lab** environments
* Aid in the development of defensive mechanisms against real-world threats.

This is completely against unethical use. Get the complete details of the project here( [project link](https://github.com/Gabbumaay/Threat-Response-System/tree/Gabbumaay-patch-1).)