ARP Spoofing + Man in the Middle Attack

Project Demonstration

Course: CSE 406

Anik Saha (2005001)

Jaber Ahmed Deeder (2005023)

Supervisor:

Abdur Rashid Tushar

Lecturer, CSE, BUET

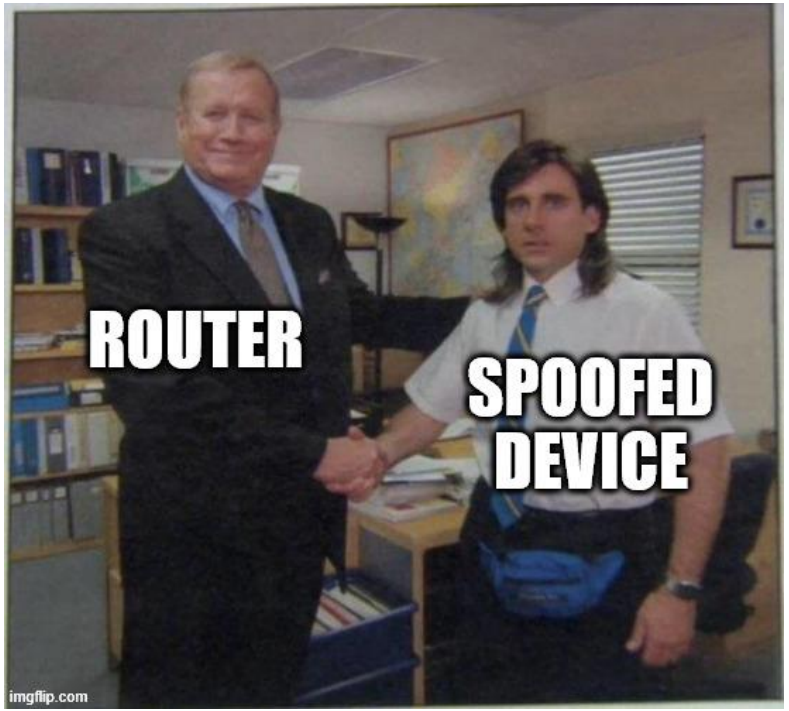
Introduction

- **ARP = Address Resolution Protocol**.
- It's a network protocol used to find the **physical (MAC) address** associated with an **IP address** on a local network.
- Devices maintain their ARP tables as cache for quick translation

But ARP blindly trusts replies! Any device can send fake ARP messages.

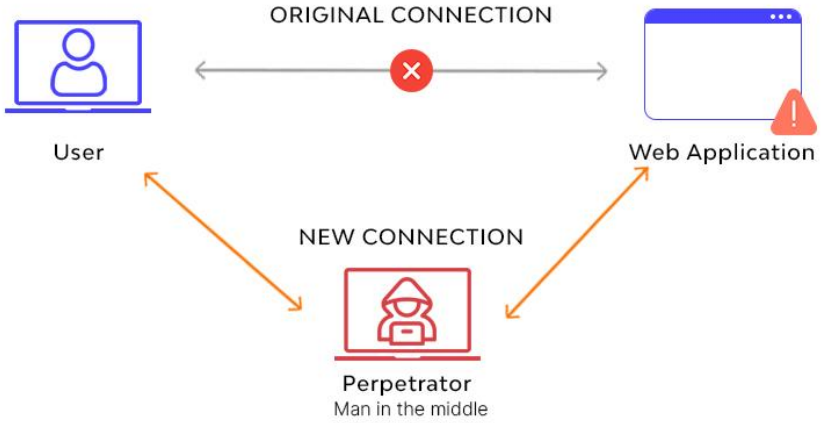
Introduction

- **Spoofing** means **pretending to be a different device** on a network (in this context)
- This can be used to intercept, modify, or block network traffic.

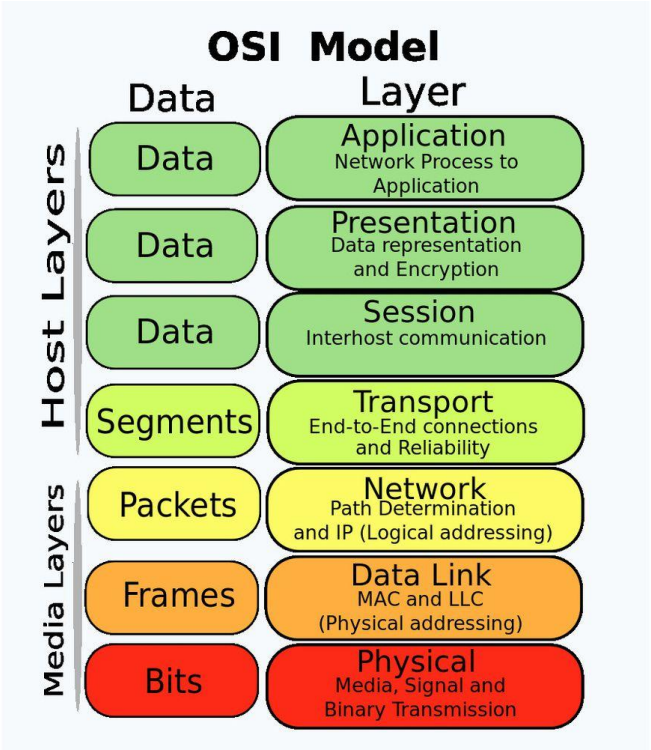


Introduction

ARP Spoofing is a network attack where an attacker sends **fake ARP replies** to devices in a LAN tricking the devices into updating their ARP tables with the **attacker's MAC address** instead of the real one.

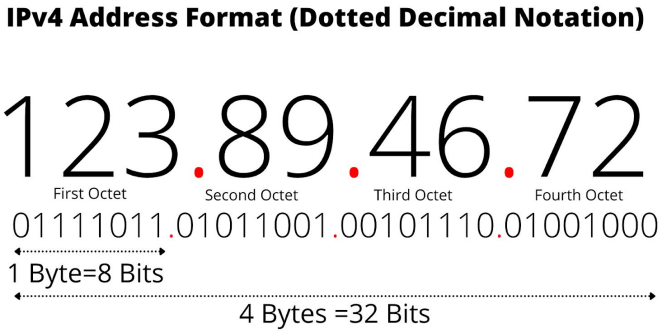


OSI Model


OSI Model
Data
Layer
Host Layers
Media Layers
Data
Application
Network Process to Application
Data
Presentation
Data representation and Encryption
Data
Session
Interhost communication
Segments
Transport
End-to-End connections and Reliability
Packets
Network
Path Determination and IP (Logical addressing)
Frames
Data Link
MAC and LLC (Physical addressing)
Bits
Physical
Media, Signal and Binary Transmission

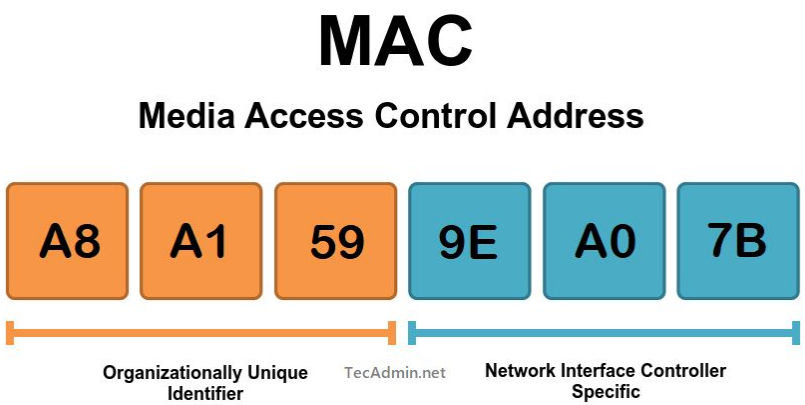
IP Address

- Logical address assigned by software/network admin.
- Used for routing packets **across different networks** (like the internet).
- Works at **Layer 3 (Network)** in OSI model.



MAC Address

- Physical, hardware address unique to a network interface (usually fixed).
- Used for communication **within the same local network (LAN)**.
- Works at **Layer 2 (Data Link)** in OSI model..



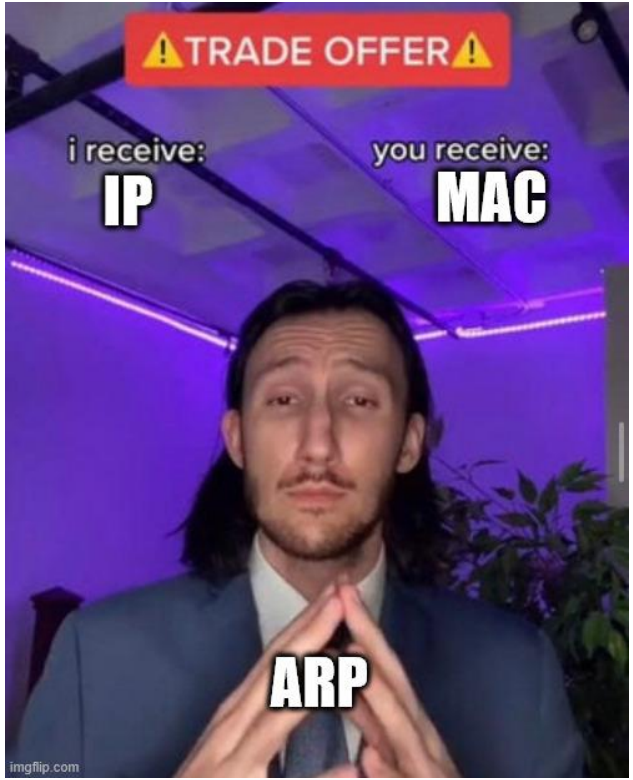
ARP

- The protocol that **links IP addresses to MAC addresses** within a local network.

ARP

ARP is responsible for resolving IP addresses to MAC addresses within a local network. It works like this:

- When a device knows the IP but not the MAC, ARP sends a broadcast request asking:
  “Who has this IP? Tell me your MAC.”
- The device with that IP replies with its MAC address.
- The sender stores this info in its **ARP table** for faster communication next time.



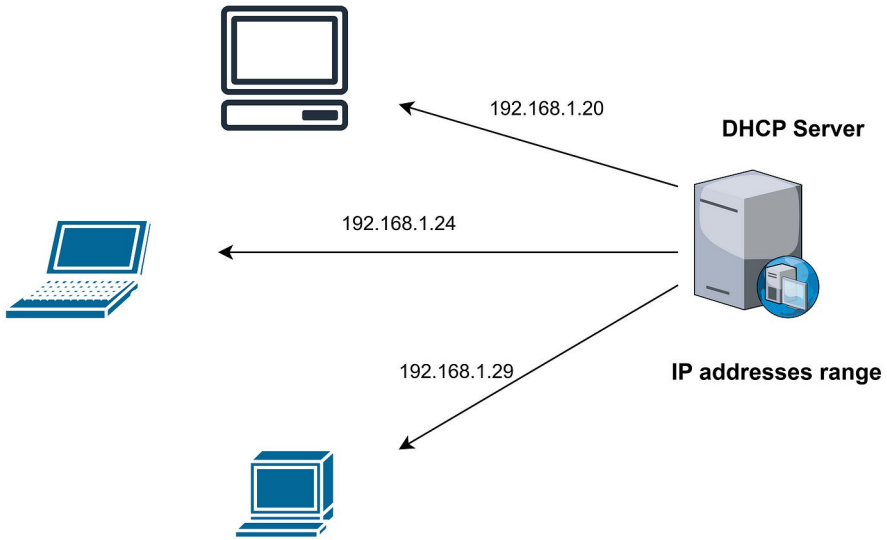
How communication happens

When a device wants to send data:

1. It prepares an IP packet with the destination IP.
2. If the destination is within the local network (same subnet), it needs the destination MAC address.
3. If the destination is outside the LAN (e.g., a website), the packet is sent to the router's MAC address instead.
4. To get the MAC address, the device checks its **ARP cache** (a table of known IP↔MAC mappings).
5. If no entry exists, it sends a **broadcast ARP request** to the network:
   "Who has IP 192.168.0.1?"
6. The correct device replies with its MAC address, and the sender stores this in the ARP cache and uses it to send the packet.

Example: A Wi-Fi Network

- Each device connects to a router or access point.
- The router assigns each device a **local IP address** (e.g., 192.168.0.105) via DHCP.
- The router itself usually has a default gateway IP (e.g., 192.168.0.1).
- ARP tables map the assigned IP addresses to device MAC



ARP Flaws

ARP is **completely stateless and unauthenticated**. Devices accept ARP replies even if they didn't ask for them. This design flaw opens the door for attacks.

ARP Flaws

- It doesn't verify the source of replies.
- It accepts any ARP reply and updates its ARP cache.
- Devices accept ARP replies even if they didn't ask for them. This design flaw opens the door for attacks.

An attacker can send fake ARP replies to poison the cache of other devices. For example:

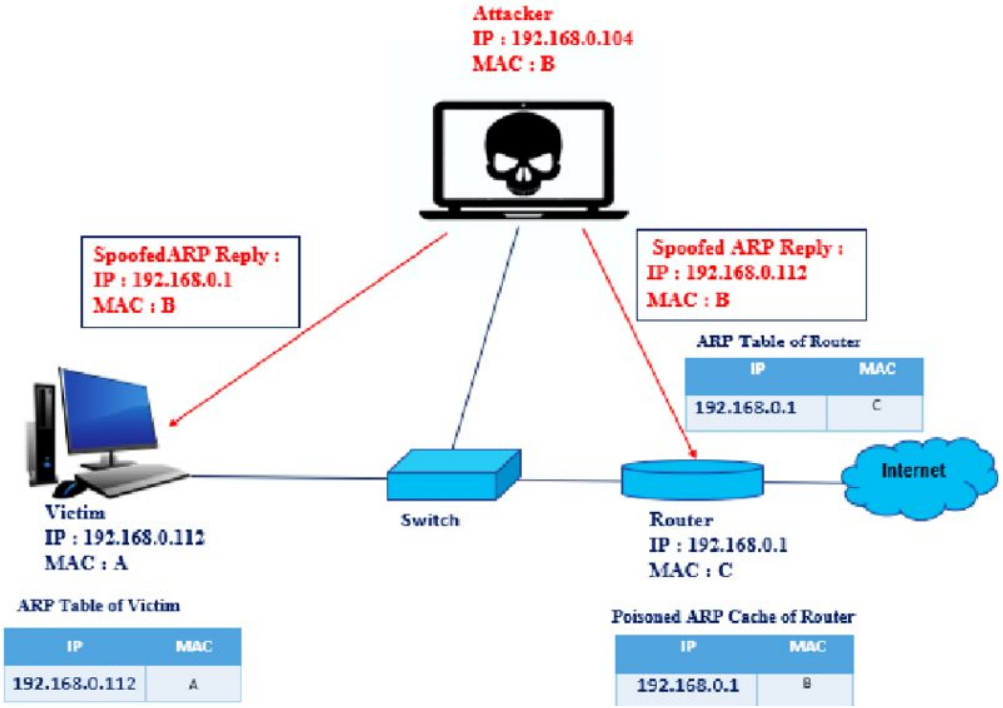
- Tell the victim: "The router's IP is at **my** MAC address."
- Tell the router: "The victim's IP is at **my** MAC address."

Now both the router and the victim will send traffic through the attacker, who is now **in the middle** of the communication.

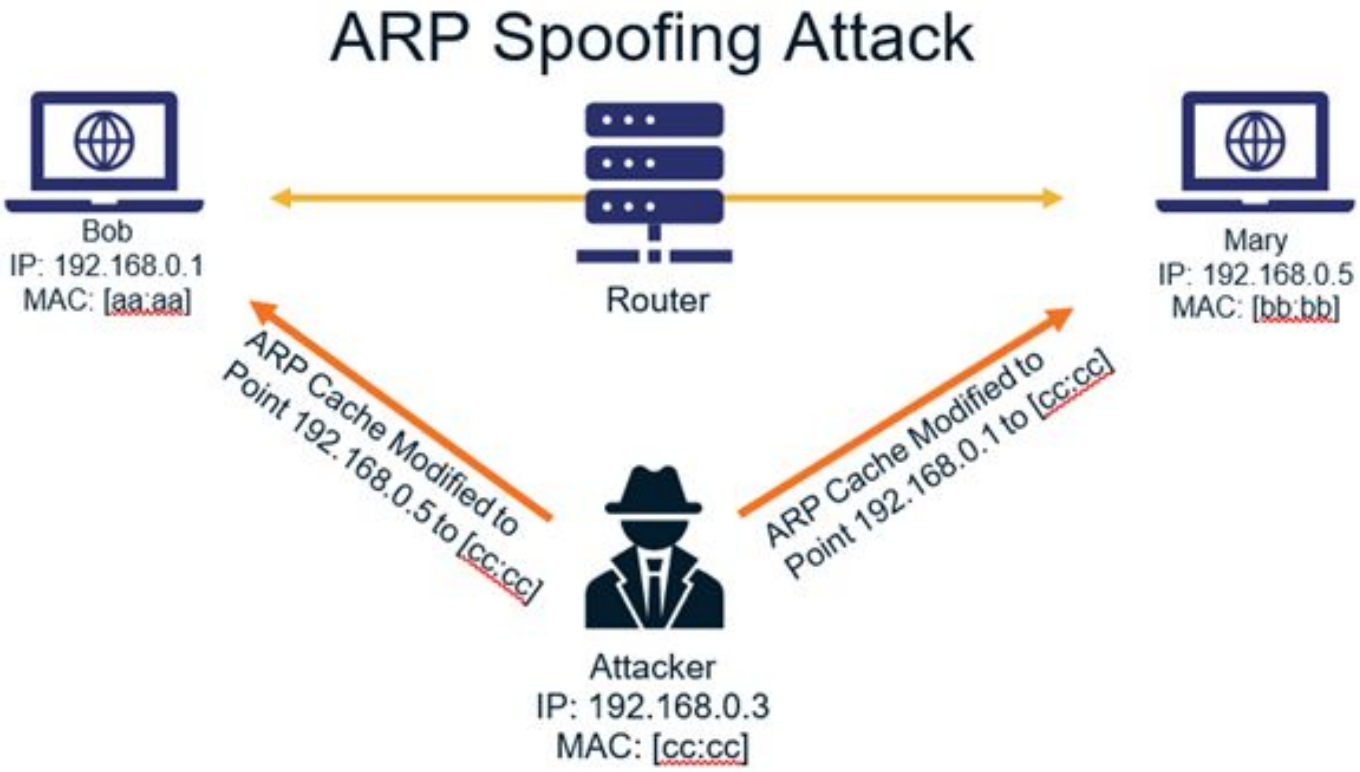
The Attack

In an ARP spoofing attack:

- The attacker sends forged ARP replies to both the **victim** and the **router**.
- The victim believes the attacker is the router.
- The router believes the attacker is the victim.



The Attack

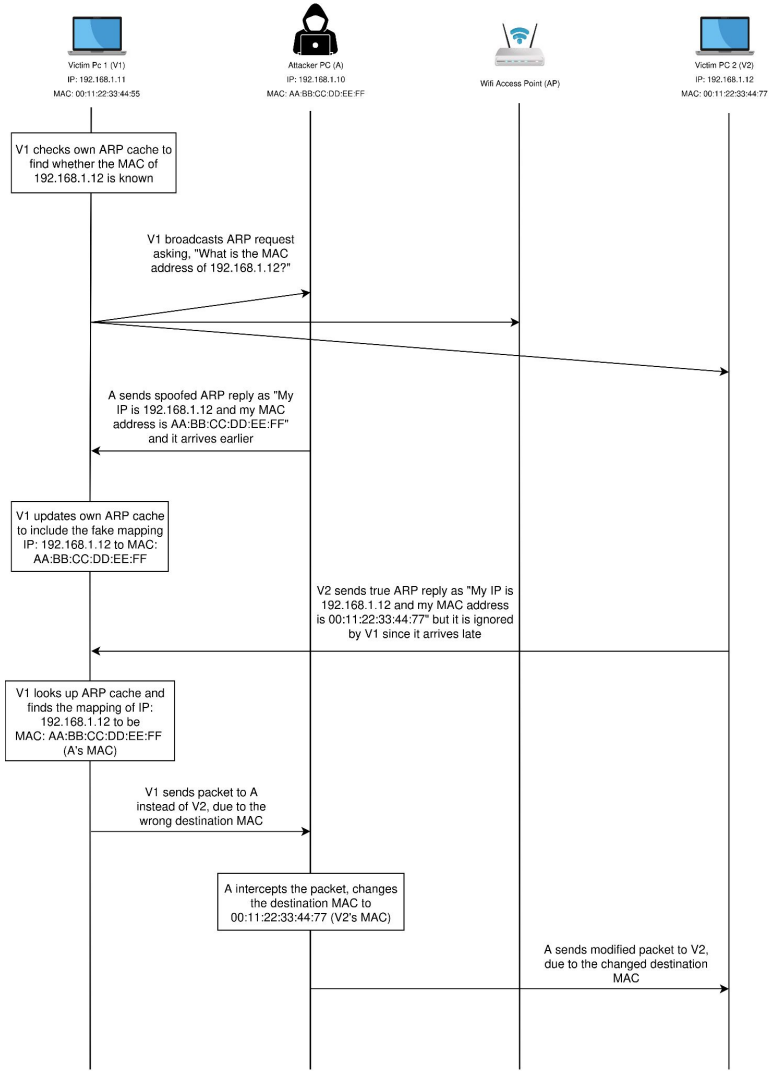

ARP Spoofing Attack
Bob
IP: 192.168.0.1
MAC: [aa:aa]
Router
Mary
IP: 192.168.0.5
MAC: [bb:bb]
ARP Cache Modified to Point 192.168.0.5 to [cc:cc]
ARP Cache Modified to Point 192.168.0.1 to [cc:cc]
Attacker
IP: 192.168.0.3
MAC: [cc:cc]

The Attack

An attacker can:

- Intercept traffic
- Modify packets (e.g., change a website)
- Steal credentials from HTTP logins
- Inject malicious payloads (phishing/malware)
- Hijack sessions (if HTTPS is not used)

Attack Timing Diagram


Victim Pc 1 (V1)
IP: 192.168.1.11
MAC: 00:11:22:33:44:55
Attacker PC (A)
IP: 192.168.1.10
MAC: AA:BB:CC:DD:EE:FF
Wifi Access Point (AP)
Victim PC 2 (V2)
IP: 192.168.1.12
MAC: 00:11:22:33:44:77
V1 checks own ARP cache to find whether the MAC of 192.168.1.12 is known
V1 broadcasts ARP request asking, "What is the MAC address of 192.168.1.12?"
A sends spoofed ARP reply as "My IP is 192.168.1.12 and my MAC address is AA:BB:CC:DD:EE:FF" and it arrives earlier
V1 updates own ARP cache to include the fake mapping IP: 192.168.1.12 to MAC: AA:BB:CC:DD:EE:FF
V2 sends true ARP reply as "My IP is 192.168.1.12 and my MAC address is 00:11:22:33:44:77" but it is ignored by V1 since it arrives late
V1 looks up ARP cache and finds the mapping of IP: 192.168.1.12 to be MAC: AA:BB:CC:DD:EE:FF (A's MAC)
V1 sends packet to A instead of V2, due to the wrong destination MAC
A intercepts the packet, changes the destination MAC to 00:11:22:33:44:77 (V2's MAC)
A sends modified packet to V2, due to the changed destination MAC

Steps of the Attack

Step-1: Enable IP routing

Step-2: Initiate arp spoofing between two devices (eg. gateway and victim). Packets sent between these two devices will go through the attacker.

Step-3: Use netfilterqueue to sniff, modify or drop packets!

Impacts

- Packet Sniffing (HTTP/TCP/UDP) can lead to sensitive information disclosure like **passwords, emails, phone numbers, access tokens, form data** etc
- Showing false information by intercepting packets.
- Delivering malwares by showing fake download buttons

Impacts

- HTTP packet modification be used to launch **phishing, forced malware downloads, script injections, browser exploitations (eg. BeeF)** etc
- Dropping packets can make internet unuseable (eg. **NetCut** tool)
- Intercepting DHCP responses can lead to **DNS spoofing**
- Sometimes attacks can be escalated to even bypass HTTPs (**SSLStrip**)

Attack Modules

- TCP Socket Messaging
  - Packet Monitoring
  - Message Alteration - Replacement of Words
  - Packet Dropping
- HTTP
  - Packet Monitoring
  - Injection of malicious HTML
  - Packet Dropping
- DNS
  - Response URL Redirection

Defenses

Switch!

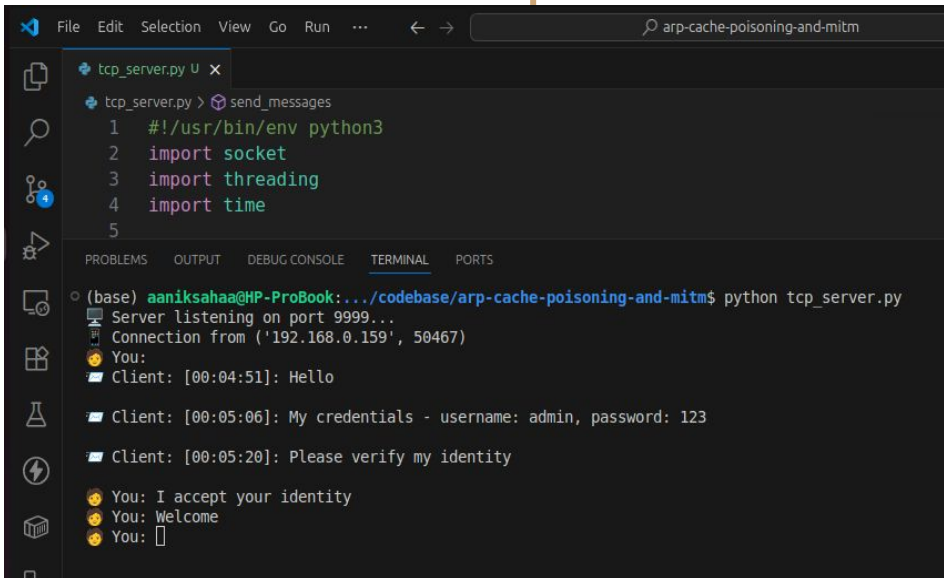
- **MAC Binding**: Lock MAC addresses to specific physical ports
- **Limit MAC Count**: Restrict number of MACs per port (prevents spoof flooding)
- **Sticky MACs**: Learn and retain valid MACs automatically

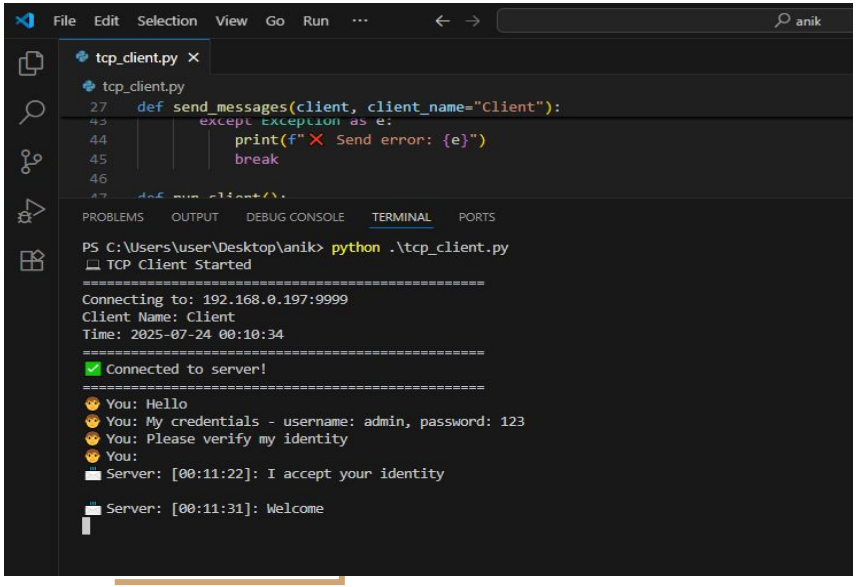
Defenses

- Create a list of trusted IP and MAC address pairs
- Continuously check ARP Table and detect attack
- The attack is detected when we find a mismatch with a trusted mapping
- At That point,
    a. We instantly update the cache with known mapping
    b. And also send gratuitous ARP replies so that others in network also get informed of it

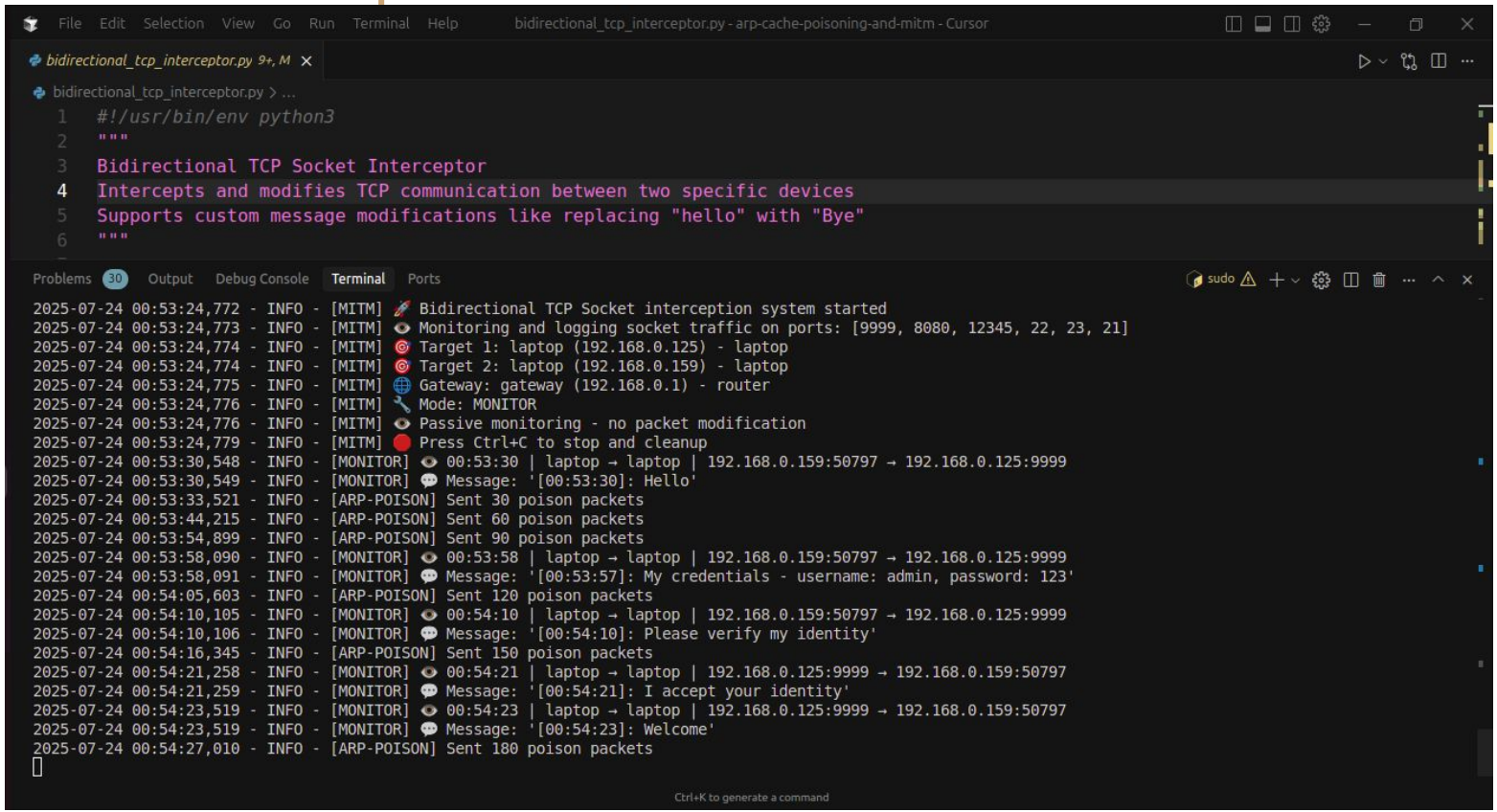
Observed Results

TCP Socket Messaging - Normal Operation

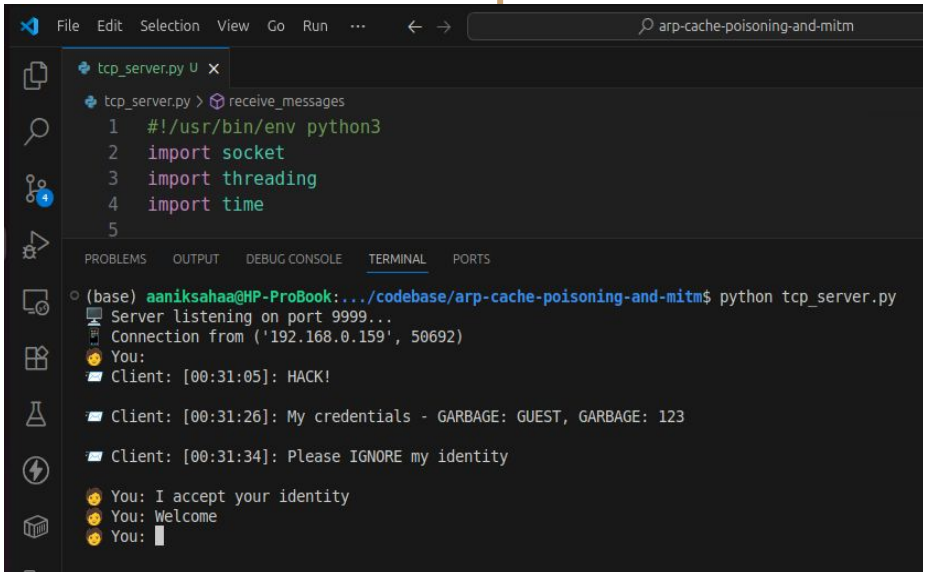


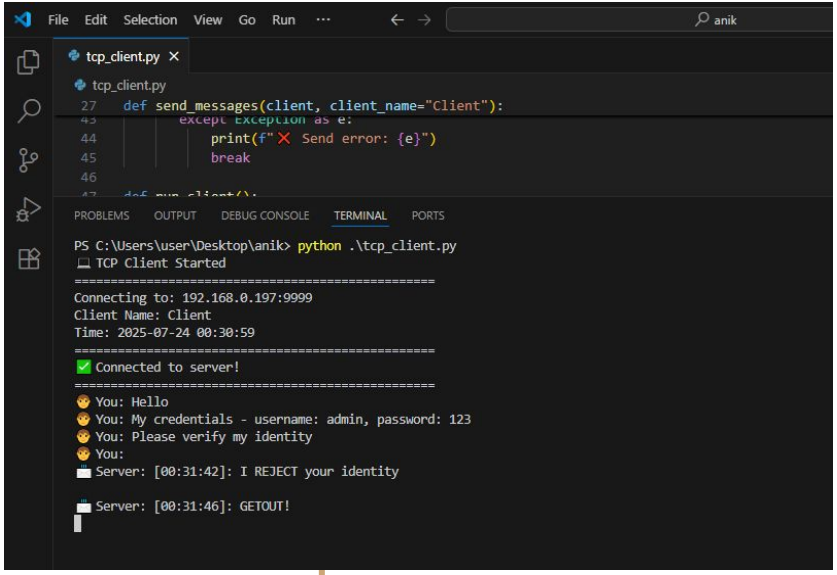


TCP Socket Messaging - Packet Monitoring

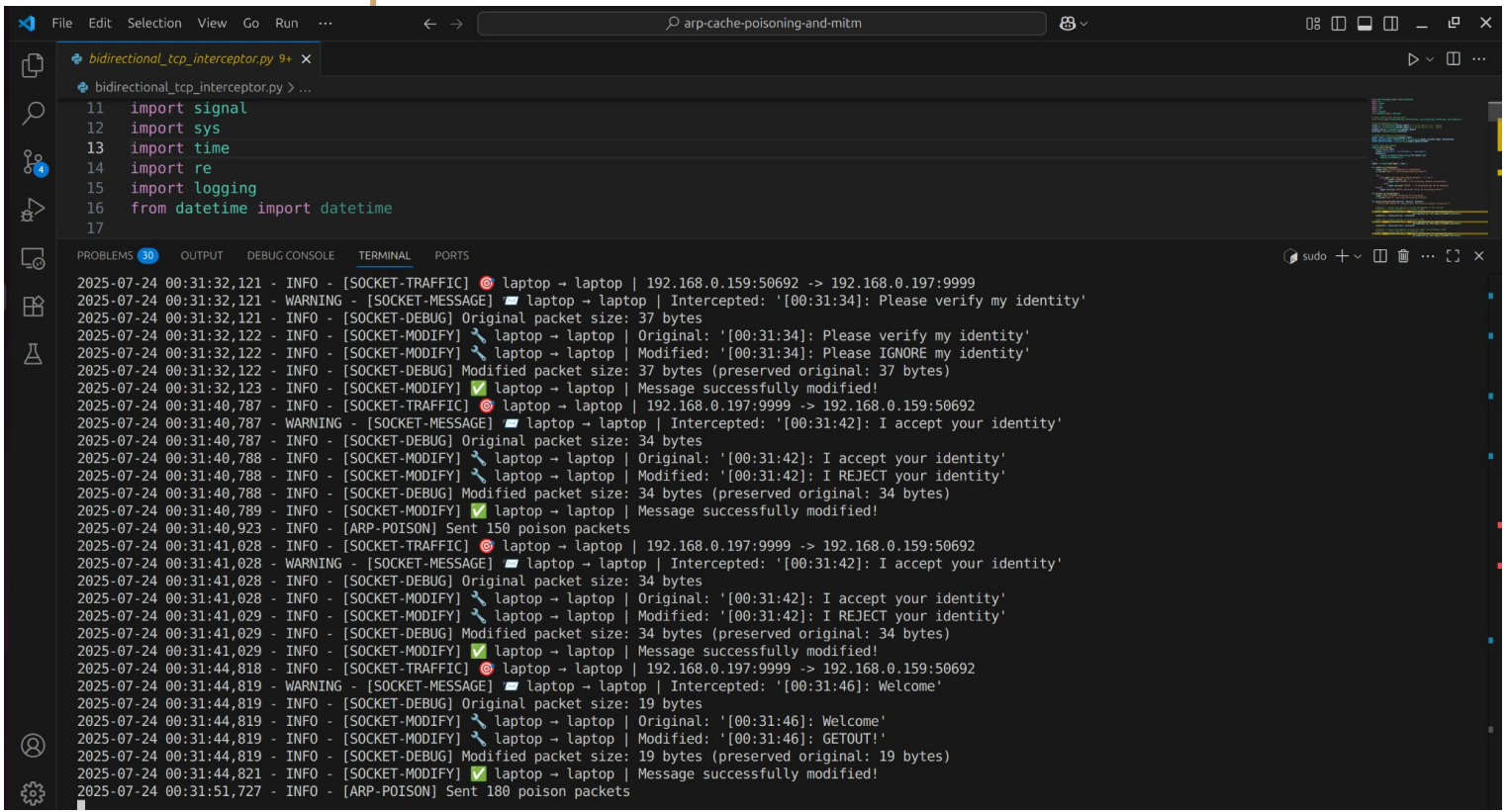


TCP Socket Messaging - Message Alteration

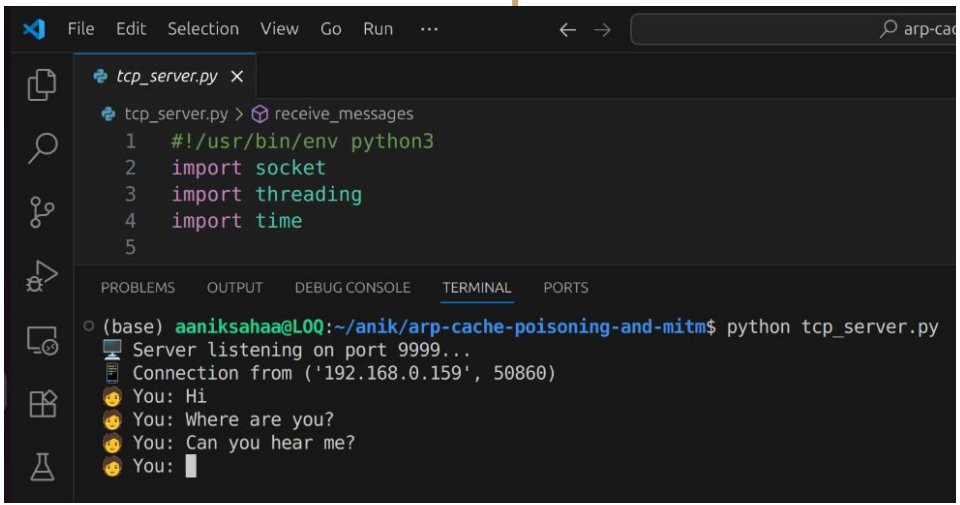


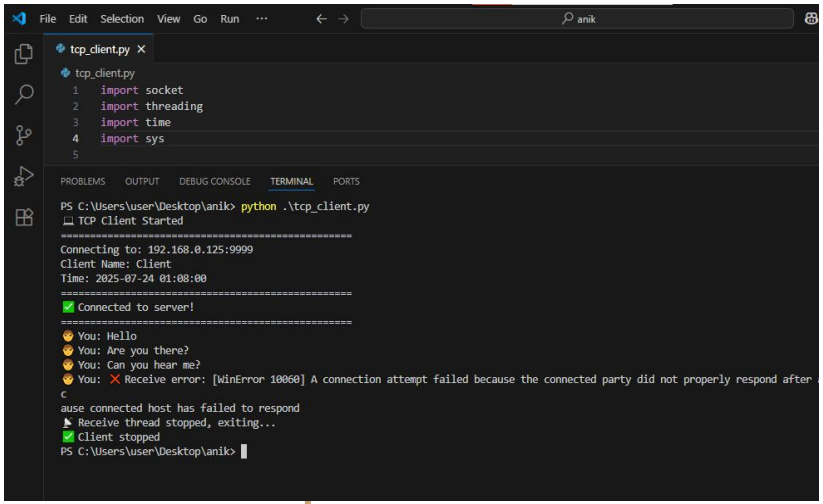


TCP Socket Messaging - Message Alteration

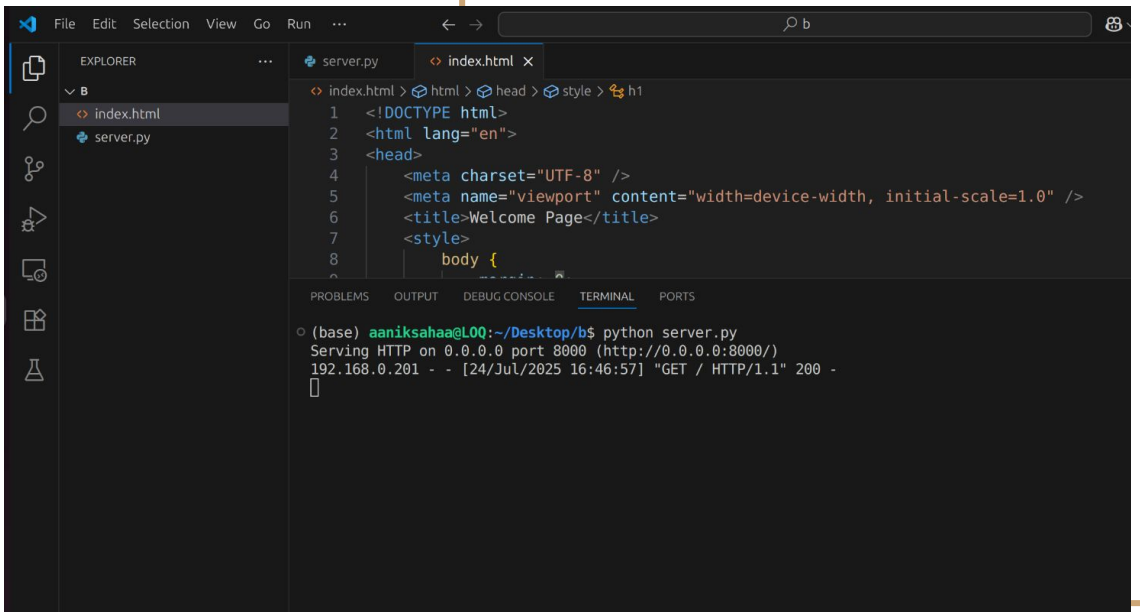


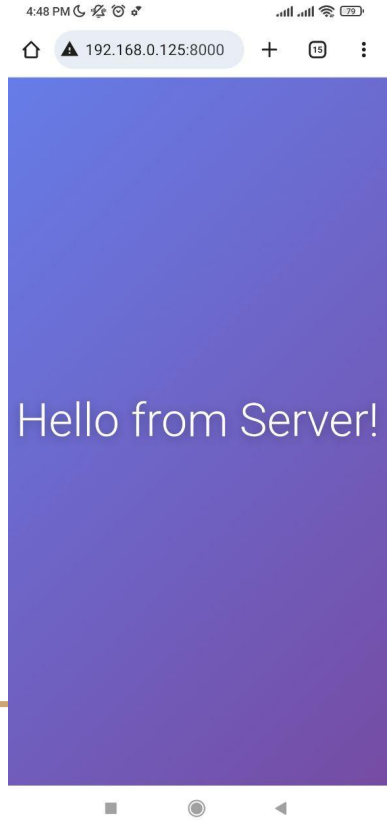
TCP Socket Messaging - Packet Dropping



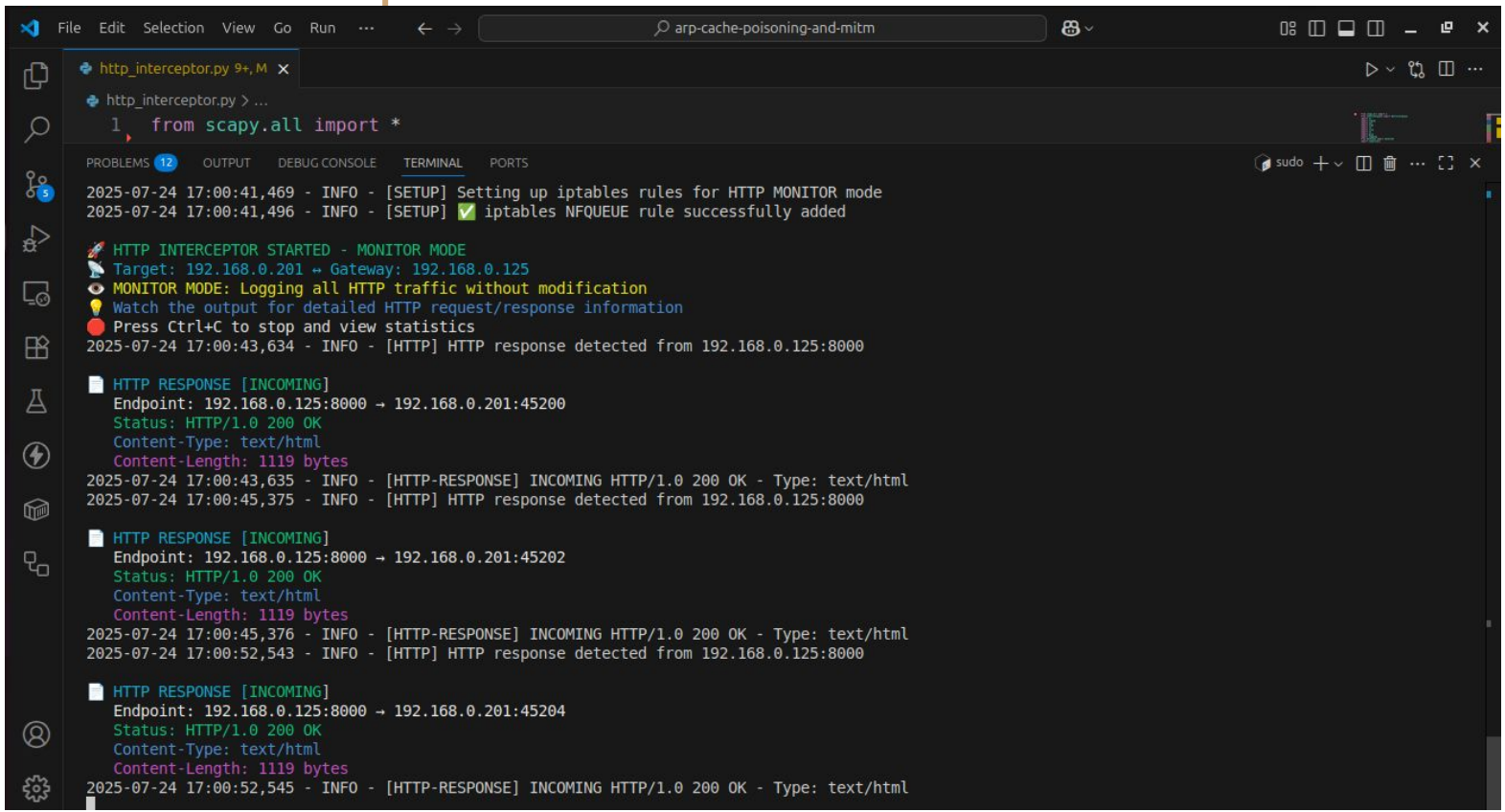


HTTP - Normal Operation

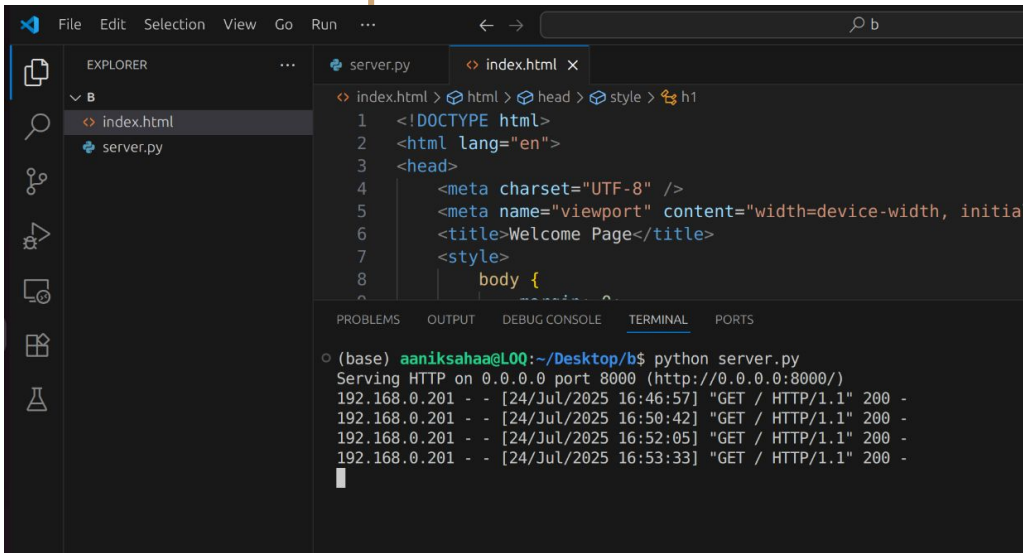


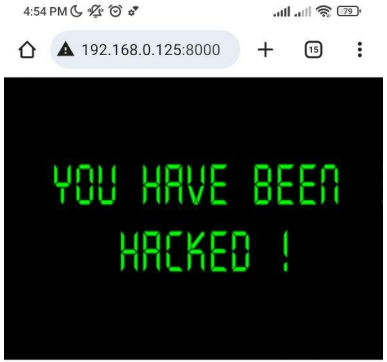


HTTP - Packet Monitoring

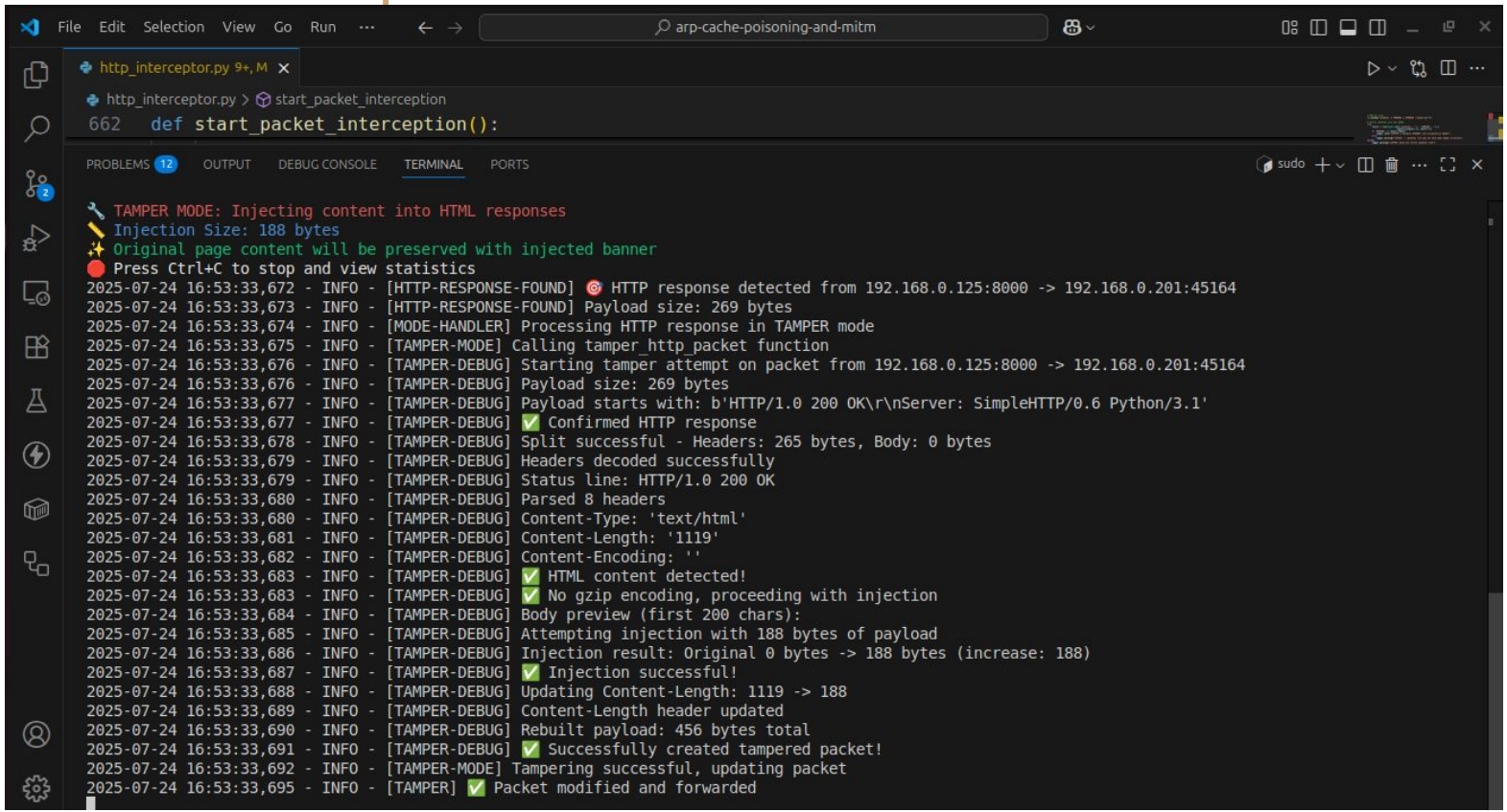


HTTP - Injection of Malicious HTML

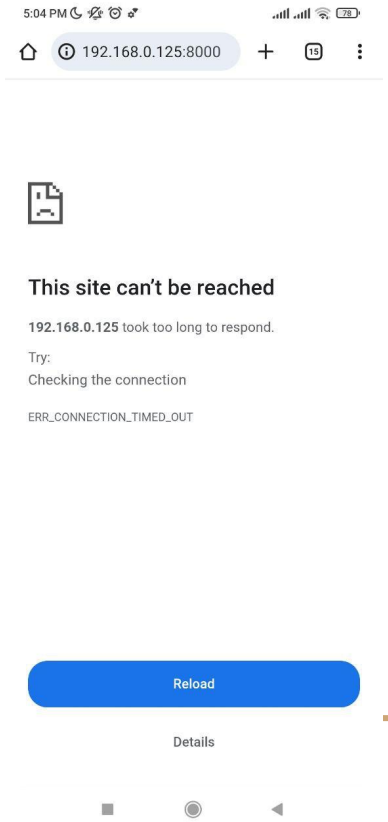




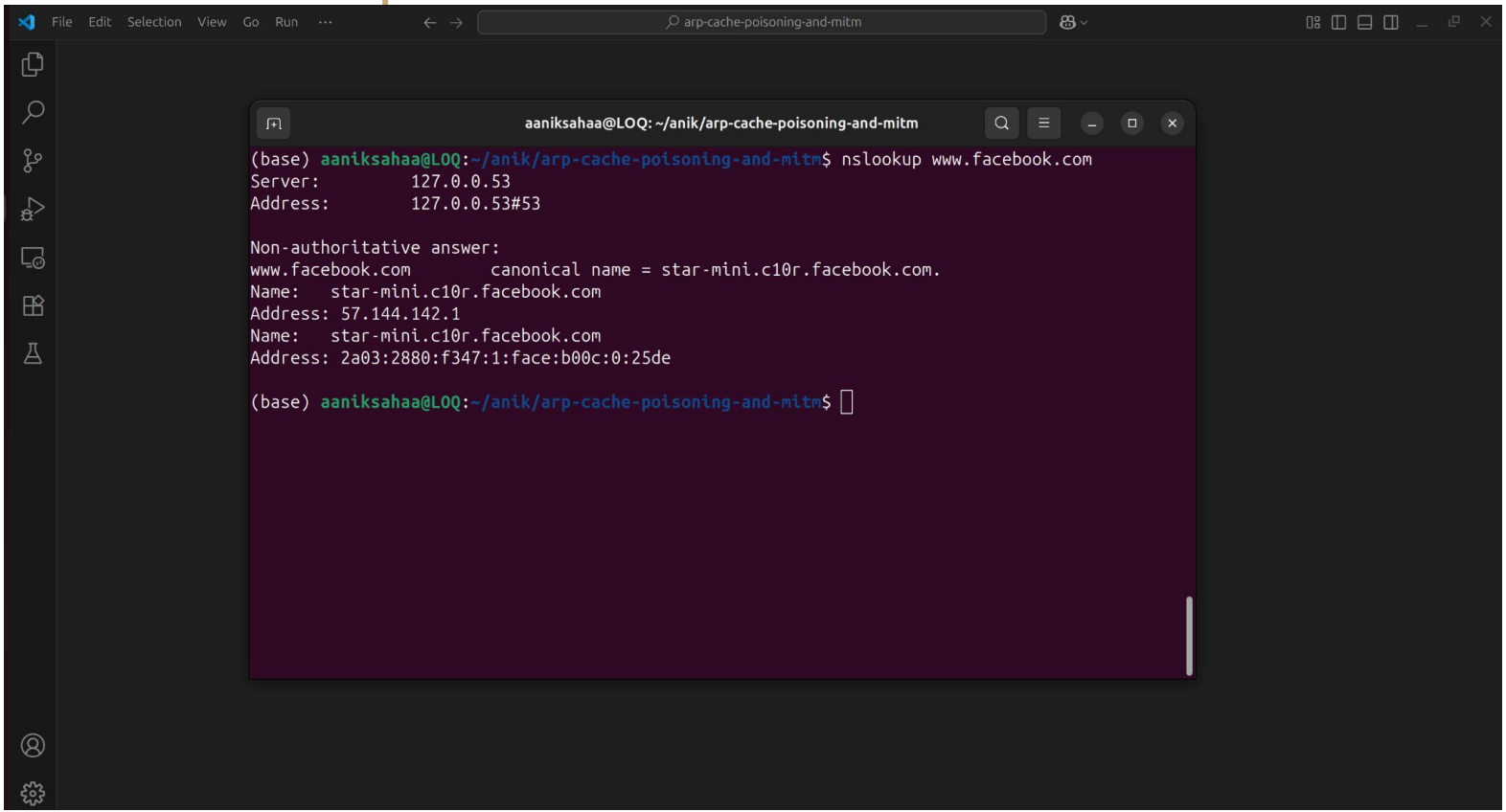
HTTP - Injection of Malicious HTML



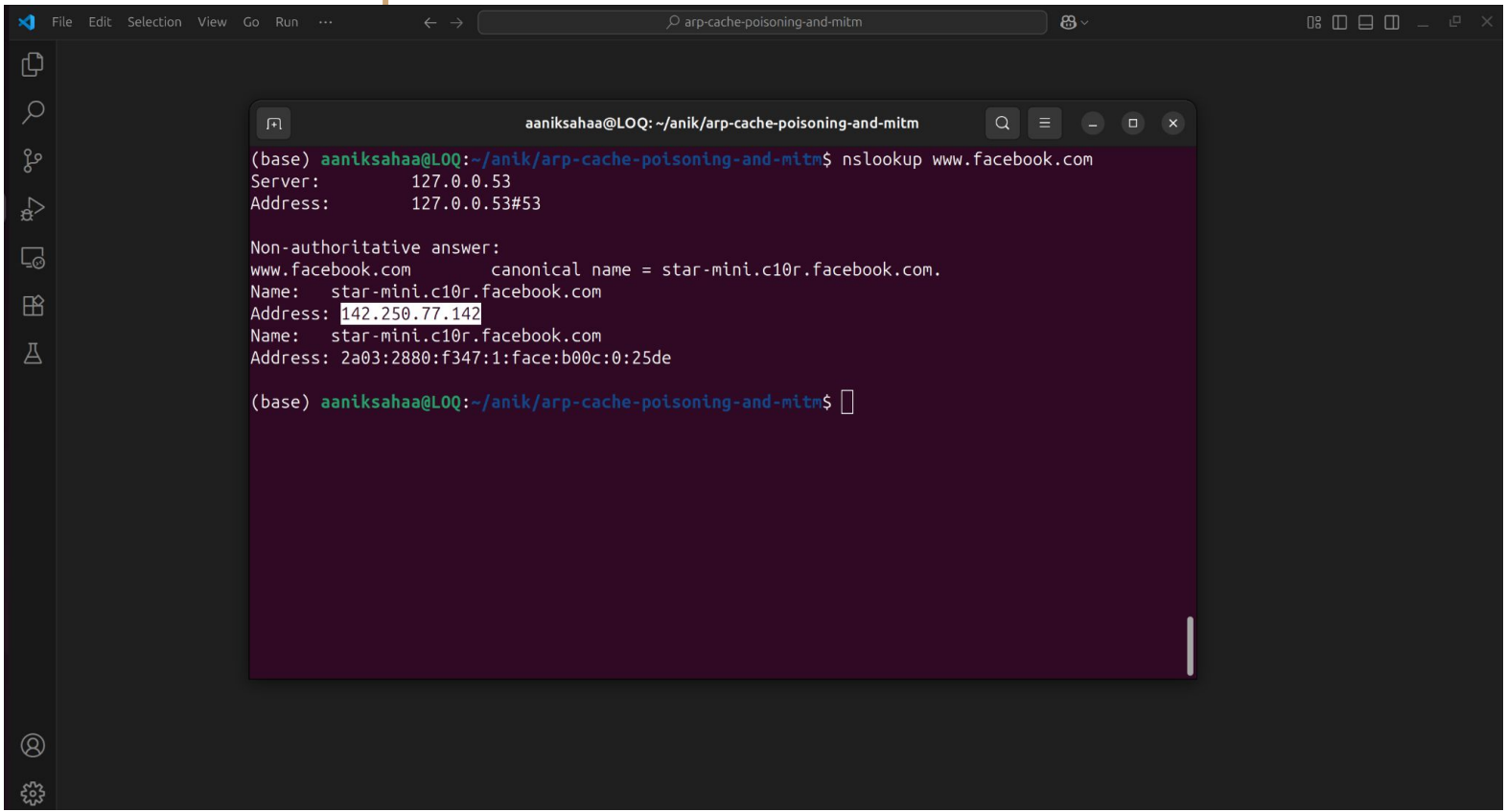
HTTP - Packet Dropping



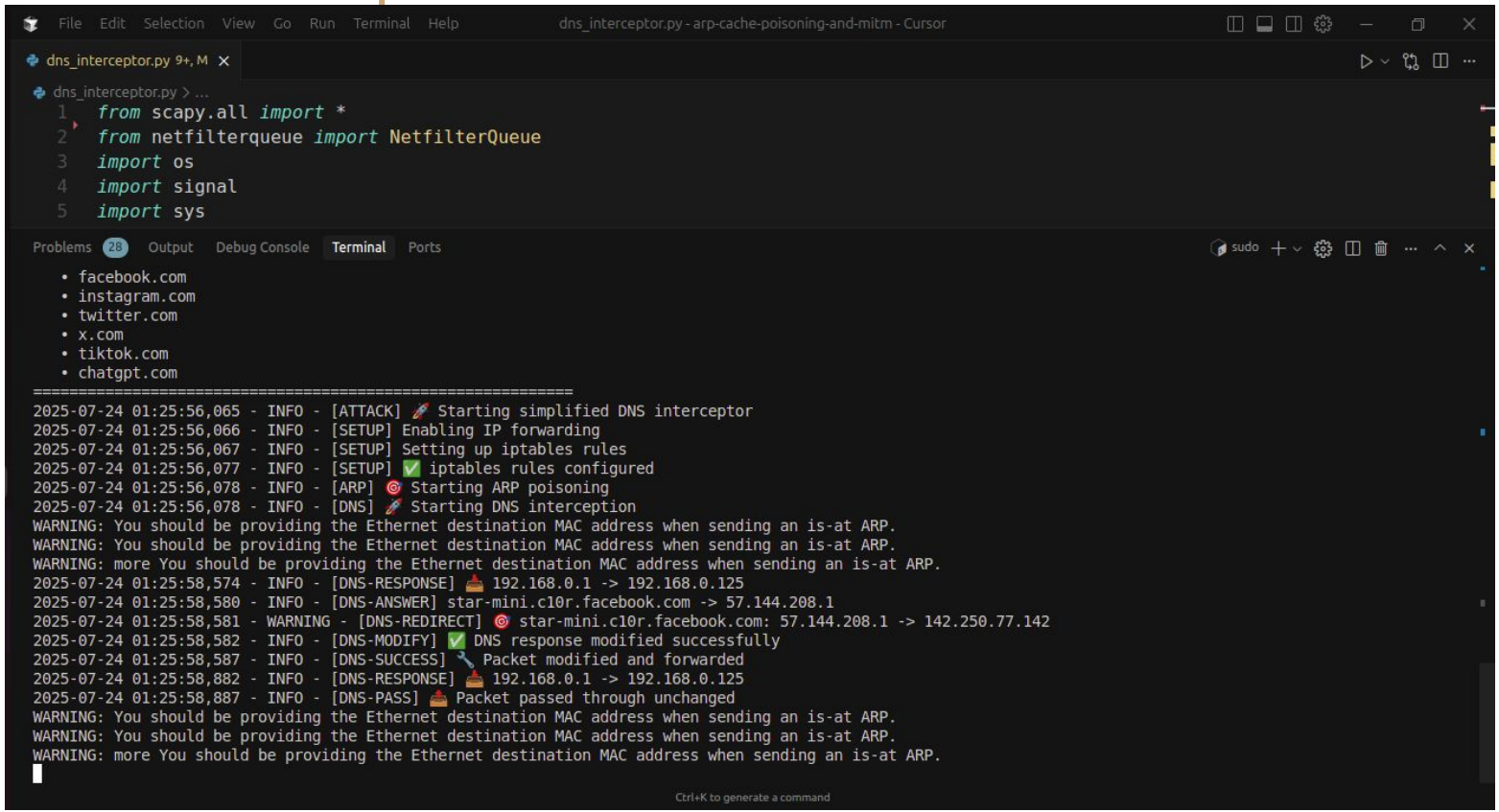
DNS - Normal Operation



DNS - Response URL Redirection



DNS - Response URL Redirection



Thank you!