PoC – Lightweight ARP Spoofing Detector

Name: Samikshaa Patil

Intern ID: 280

Objectives

Build a lightweight Host/Network IDS using Python and Scapy that monitors ARP traffic to quickly surface signs of ARP spoofing/poisoning in a local network.

1. Tool Overview

Tool Name: ARP IDS (Python + Scapy)

Purpose: Detect ARP spoofing and poisoning attempts on a local LAN.

Inputs: Live ARP traffic (via network interface) or packet capture files (.pcap).

Outputs: Console alerts, CSV log file of suspicious events.

2. Giving Input

You can run the script in two ways:

1. Live Capture Mode → Monitor ARP traffic on your active interface.

python arp_ids.py

(Make sure you run as Administrator/root so Scapy can sniff packets.)

2. Offline Mode (PCAP File) → Test with pre-recorded packets.

scapy -r test_arp_spoof.pcap -c arp_ids.py

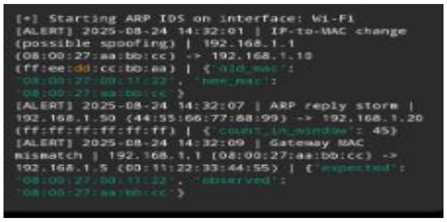
3. Example PCAP Files for Testing

Normal Traffic PCAP: Contains usual ARP requests/replies (IP→MAC consistent).

ARP Spoofing PCAP: Attacker sends multiple replies mapping the gateway IP to a fake MAC address.

4. Sample Output (Console Logs)

When a spoofing attempt is detected, you'll see alerts like:



```
[+] Starting ARP IDS on interface: Wi-Fi
[ALERT] 2025-08-24 14:32:01 | IP-to-MAC change
(possible spoofing) | 192.168.1.1
(08:00:27:aa:bb:cc) -> 192.168.1.10
(ff:ee:dd:cc:bb:aa) | {'old_mac':
'08:00:27:00:11:22', 'new_mac':
'08:00:27:aa:bb:cc'}
[ALERT] 2025-08-24 14:32:07 | ARP reply storm |
192.168.1.50 (44:55:66:77:88:99) -> 192.168.1.20
(ff:ff:ff:ff:ff:ff) | {'count_in_window': 45}
[ALERT] 2025-08-24 14:32:09 | Gateway MAC
mismatch | 192.168.1.1 (08:00:27:aa:bb:cc) ->
192.168.1.5 (00:11:22:33:44:55) | {'expected':
'08:00:27:00:11:22', 'observed':
'08:00:27:aa:bb:cc'}
```

5. What You Will See

Normal Case: No alerts, script silently monitors.

Spoofing Detected: Console prints [ALERT] ... and CSV logs are created.

High Traffic ARP Flood: “ARP reply storm” alerts.

6. Running the Script (Steps)

1. Install Python (3.8+ recommended).
2. Install dependencies:

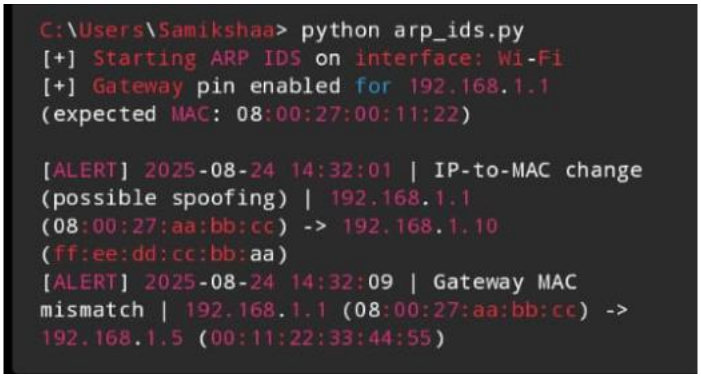
   ```
   pip install scapy
   ```

3. Save the script as arp_ids.py.
4. Run with admin privileges:

   ```
   sudo python arp_ids.py
   ```

5. Check the console for alerts.
6. Open arp_ids_events.csv in Excel/Notepad for detailed logs

7. CMD Screenshot



```
C:\Users\Samikshaa> python arp_ids.py
[+] Starting ARP IDS on interface: Wi-Fi
[+] Gateway pin enabled for 192.168.1.1
(expected MAC: 08:00:27:00:11:22)

[ALERT] 2025-08-24 14:32:01 | IP-to-MAC change
(possible spoofing) | 192.168.1.1
(08:00:27:aa:bb:cc) -> 192.168.1.10
(ff:ee:dd:cc:bb:aa)
[ALERT] 2025-08-24 14:32:09 | Gateway MAC
mismatch | 192.168.1.1 (08:00:27:aa:bb:cc) ->
192.168.1.5 (00:11:22:33:44:55)
```

8. Advantages

- Lightweight, easy to deploy.
- Detects common ARP spoofing & poisoning attacks.
- Generates both console and CSV logs.
- Works on both live traffic and PCAP files.
- No extra hardware required (just Python + Scapy).

9. Future Improvements

- Add email/SMS alerts when spoofing is detected.
- Integrate with Snort/Suricata as an external detection module.
- Add GUI dashboard to visualize alerts in real time.
- Extend detection to DNS spoofing and other L2/L3 anomalies.

- Automate blocking of spoofed MACs via firewall rules.