

# Lab 24 – Using ARP command for network reconnaissance

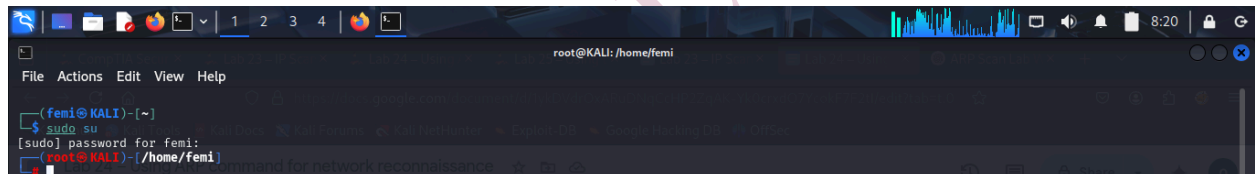
## ✓ Lab Objective:

You're learning to use the arp-scan tool to map IP addresses to MAC addresses on your local network. This is important for network reconnaissance—the process of discovering devices connected to a network.

## 🔧 Lab Preparation

### 1. Log in as Root

`sudo su -`



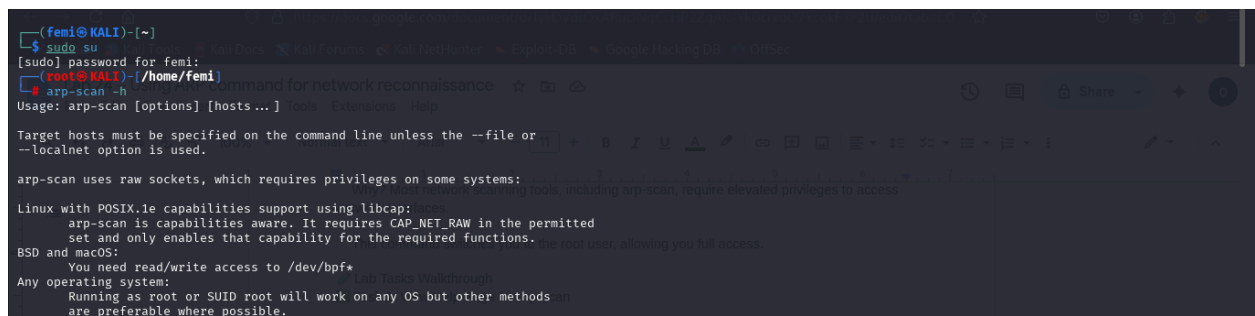
```
(femi@KALI)-[~]  
$ sudo su  
[sudo] password for femi:  
(root@KALI)-[/home/femi]
```

Why? Most network scanning tools, including arp-scan, require elevated privileges to access network interfaces.

This command switches you to the root user, allowing you full access.

## 🔧 Lab Tasks Walkthrough

### ✓ Task 1: View Help Page for arp-scan



```
(femi@KALI)-[~]  
$ sudo su  
[sudo] password for femi:  
(root@KALI)-[/home/femi]  
$ arp-scan -h  
Usage: arp-scan [options] [hosts...] Tools: Extensions: Help  
  
Target hosts must be specified on the command line unless the --file or  
--localnet option is used.  
  
arp-scan uses raw sockets, which requires privileges on some systems:  
  
Linux with POSIX.1e capabilities support using libcap: yes  
arp-scan is capabilities aware. It requires CAP_NET_RAW in the permitted  
set and only enables that capability for the required functions.  
BSD and macOS:  
You need read/write access to /dev/bpf*  
  
Any operating system:  
Running as root or SUID root will work on any OS but other methods  
are preferable where possible.
```

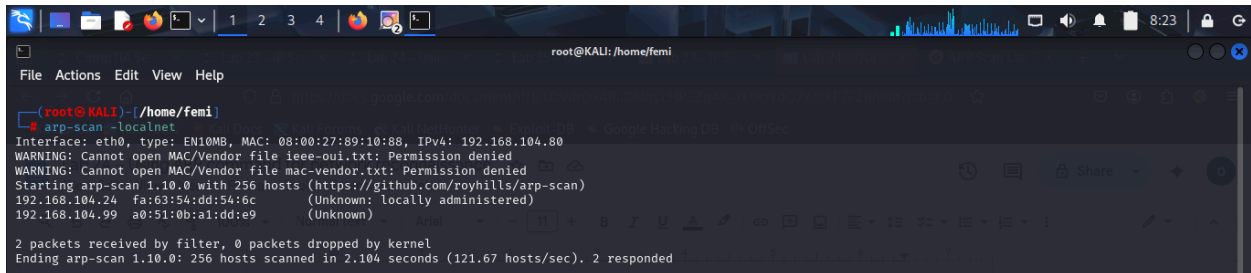
`arp-scan -h`

Why? This command shows the help menu. It provides available options, flags, and syntax for using arp-scan.

Example flags: -I for interface, --localnet for scanning local subnet, etc.

## ✓ Task 2: Scan the Local Network

arp-scan --localnet



```
root@KALI: /home/femi
File Actions Edit View Help

(root@KALI)-[/home/femi]
# arp-scan --localnet
Interface: eth0, type: EN10Mb, MAC: 08:00:27:89:10:88, IPv4: 192.168.104.80
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.104.24 fa:63:54:dd:54:6c (Unknown: locally administered)
192.168.104.99 a0:51:0b:a1:dd:e9 (Unknown)

2 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.104 seconds (121.67 hosts/sec). 2 responded
```

What it does: Scans all IPs in your current subnet (e.g., 192.168.1.0/24) using ARP requests.

Why ARP? ARP works at Layer 2 (Data Link layer). It cannot be blocked by host firewalls, making it very effective for local discovery.

Output Explanation:

IP Address — The address of the detected device.

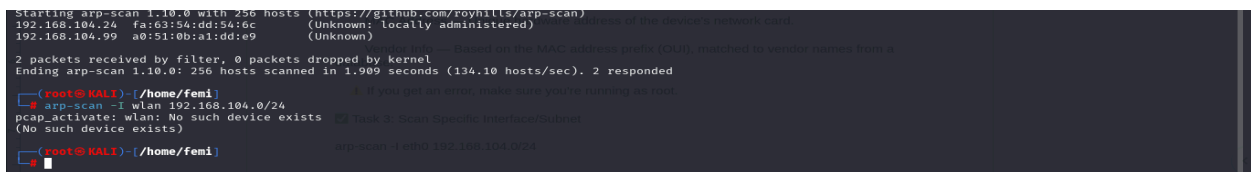
MAC Address — Unique hardware address of the device's network card.

Vendor Info — Based on the MAC address prefix (OUI), matched to vendor names from a database.

⚠ If you get an error, make sure you're running as root.

## ✓ Task 3: Scan Specific Interface/Subnet

arp-scan -I eth0 192.168.104.0/24



```
(root@KALI)-[/home/femi]
# arp-scan -I wlan 192.168.104.0/24
pcap_activate: wlan: No such device exists
(No such device exists)

(root@KALI)-[/home/femi]
# arp-scan -I eth0 192.168.104.0/24
```

-I eth0: Specifies which network interface to use (e.g., eth0, wlan0).

192.168.104.0/24: Explicitly states the subnet to scan.

Why? Useful if:

You're connected to multiple networks.

Your interface doesn't have an IP but is physically connected.

You want to scan a subnet different from the one you're in.

#### ✓ Task 4: Scan a VLAN (Trunk Port)

(Command not explicitly shown, but implied logic follows the same pattern.)

`arp-scan -I eth0.10 192.168.10.0/24`

eth0.10: Denotes VLAN 10 on the eth0 interface (802.1q tagged).

Why? On trunk ports, you might want to scan specific VLANs. VLAN tagging allows you to do this using subinterfaces.

#### ✓ Task 5: Identifying Unknown Devices & Updating Vendor DB

1. Sometimes, arp-scan might show:

Unknown

in the vendor column.

Why? The MAC address prefix isn't in the tool's local vendor database.

What to do?

Use an online MAC lookup tool (e.g., <https://macvendors.com>).

Update the database with:

`cd /usr/share/arp-scan`

`get-iab -v -u http://standards.ieee.org/develop/regauth/iab/iab.txt`

`get-oui -v -u http://standards.ieee.org/develop/regauth/oui/oui.txt`

```
(root@KALI)-[/usr/share/arp-scan]
get-oui -v -u http://standards.ieee.org/develop/regauth/oui/oui.txt
Renaming /usr/share/arp-scan/ieee-oui.txt to /usr/share/arp-scan/ieee-oui.txt.backup, you can manually add a line
Opening /usr/share/arp-scan/ieee-oui.txt for output
Processing IEEE IAB registry data from file:///var/lib/ieee-data/iab.csv
Downloaded 381908 bytes
4575 IAB entries written to /usr/share/arp-scan/ieee-oui.txt
Processing IEEE MAM registry data from file:///var/lib/ieee-data/mam.csv
Downloaded 608108 bytes
5418 MAM entries written to /usr/share/arp-scan/ieee-oui.txt
Processing IEEE OUI registry data from file:///var/lib/ieee-data/oui.csv
Downloaded 3372302 bytes
35816 OUI entries written to /usr/share/arp-scan/ieee-oui.txt
Processing IEEE OUI36 registry data from file:///var/lib/ieee-data/oui36.csv
Downloaded 557084 bytes
6079 OUI36 entries written to /usr/share/arp-scan/ieee-oui.txt
Total of 51888 MAC/Vendor mappings written to /usr/share/arp-scan/ieee-oui.txt
(root@KALI)-[/usr/share/arp-scan]
```

[Lab 101 - More Advanced Scripting with PowerShell](#)  
[Read More](#)

[Lab 106 - Introduction to scripting with PowerShell](#)  
[Read More](#)

[Lab 99 - More Advanced Python Scripting](#)  
[Read More](#)

Why update? MAC address vendor databases evolve over time. Keeping them up to date increases accuracy.

## 2. Manually Add Known Spoofed MACs

```
echo "060027      SPOOFED-NIC TECHNOLOGIES LLC" >>
/usr/share/arp-scan/mac-vendor.txt
```

Why? If you have custom or spoofed devices, you can add entries manually to keep track of them.

### ✓ Task 6: Save Scan Results for Later Analysis

```
arp-scan --localnet -W results.pcap
```

What it does: Saves scan results to a .pcap file for later analysis.

-W flag = Write to file in pcap format (used by tools like Wireshark or tcpdump).

```
tcpdump -nr results.pcap
```

What it does: Reads the .pcap file using tcpdump.

-n: Don't resolve names.

-r: Read from file instead of capturing live traffic.

Why? Great for:

Auditing

Offline analysis

Keeping records

## 🎯 What You Can Learn from ARP Scan Results

All active IPv4 devices on a local network.

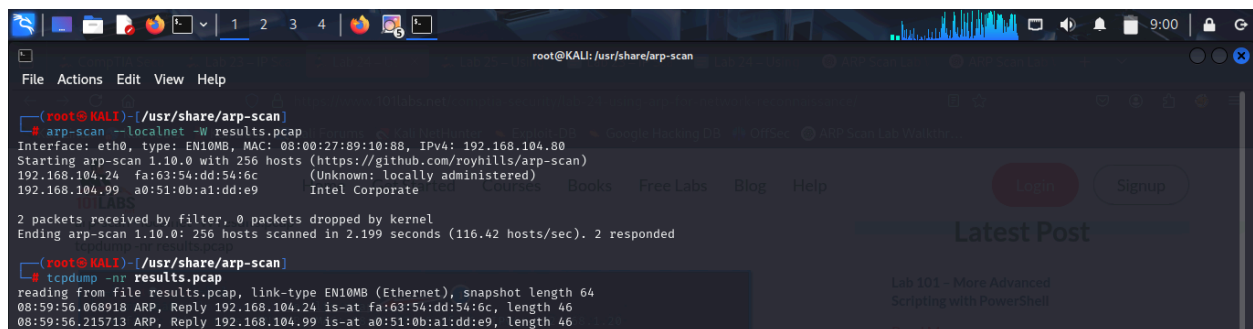
Map IP addresses to MAC addresses.

Identify vendor information.

Find IP conflicts (duplicate IPs).

Spot spoofed or unauthorized devices.

Validate VLAN segmentation and trunk configurations.



```
root@KALI: /usr/share/arp-scan
File Actions Edit View Help
root@KALI: /usr/share/arp-scan
root@KALI:~# arp-scan -localnet -W results.pcap
Interface: eth0, type: EN10MB, MAC: 08:00:27:89:10:88, IPv4: 192.168.104.80
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.104.24 fa:63:54:dd:54:6c (Unknown: locally administered)
192.168.104.99 a0:51:0b:a1:dd:e9 Intel Corporate
2 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.199 seconds (116.42 hosts/sec). 2 responded

root@KALI:~# tcpdump -nr results.pcap
reading from file results.pcap, link-type EN10MB (Ethernet), snapshot length 64
08:59:56.068918 ARP, Reply 192.168.104.24 is-at fa:63:54:dd:54:6c, length 46
08:59:56.215713 ARP, Reply 192.168.104.99 is-at a0:51:0b:a1:dd:e9, length 46
```

## END Conclusion

Using arp-scan is a fast, efficient way to conduct network reconnaissance on local IPv4 networks. It leverages the Address Resolution Protocol to map out who is connected, on what IP/MAC, and what vendor they're using.