**🔹 Task Name: Find Local IP Address and IP Range**

**🔹 Objective**

**To identify the system’s local IP address, subnet mask, and calculate the corresponding IP address range using the ifconfig command in Kali Linux.**

**🔹 Tool Used**

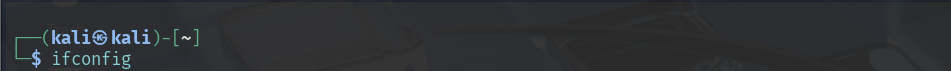
**Terminal in Kali Linux (ifconfig command)**

**🔹 Steps I Followed**

1. **Opened Terminal in Kali Linux.**
2. **Ran the command:**

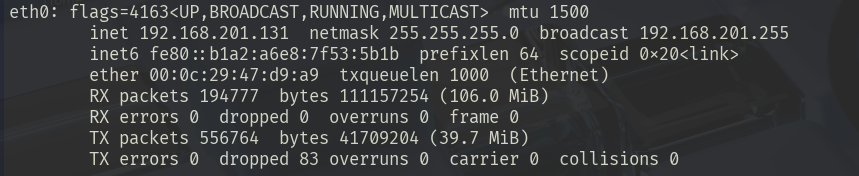
**COMMAND:**

**Ifconfig**

****

1. **Observed the output under the interface eth0.**

**🔹 Output**

****

**🔹 Findings**

| **Parameter** | **Value** |
| --- | --- |
| **Local IP Address** | **192.168.201.131** |
| **Subnet Mask** | **255.255.255.0** |
| **Broadcast Address** | **192.168.201.255** |
| **Network Interface** | **eth0** |

**🔹 IP Range Calculation**

* **Subnet Mask: 255.255.255.0 → /24**
* **Network Address: 192.168.201.0**
* **Usable IP Range:  
  192.168.201.1 to 192.168.201.254**

**🔹 Result**

**IP Range: 192.168.201.1 to 192.168.201.254**

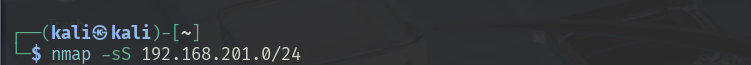
**🔹 Task Name: Perform TCP SYN Scan Using Nmap**

**🔹 Objective**

To perform a **TCP SYN scan** (-sS) using **Nmap** on the local network 192.168.201.0/24 to discover active devices and their open TCP ports.

**🔹 Command Used**

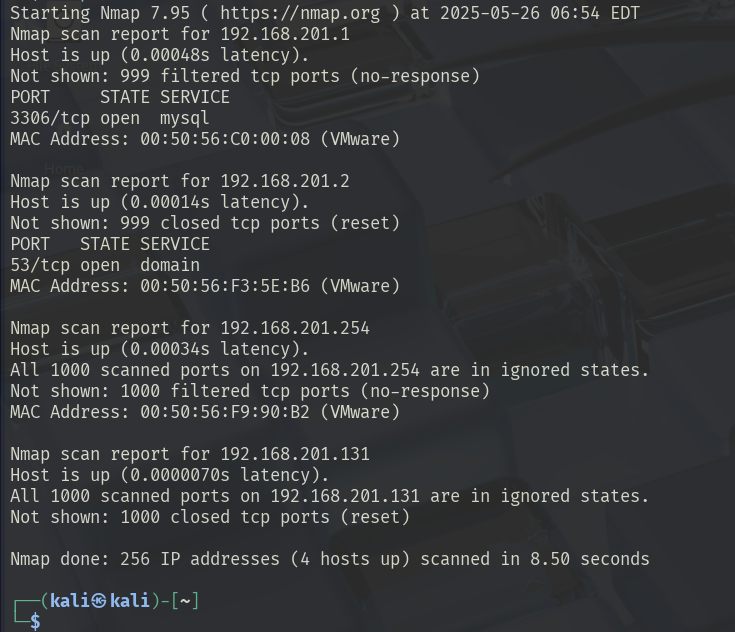
nmap -sS 192.168.201.0/24



**🔹 Explanation:**

* **Nmap**: A tool used to discover devices and services on a network.
* -sS: This flag performs a **TCP SYN scan**, also known as a **stealth scan**.
  + It sends a **SYN packet** to initiate a connection but doesn't complete it (hence *stealthy*).
  + If the target replies with **SYN-ACK**, the port is considered **open**.
  + If it replies with **RST**, the port is **closed**.
  + If there is no reply, the port is **filtered** (possibly blocked by a firewall).
* 192.168.201.0/24: Scans all IPs from .1 to .254 in your local subnet.

**🔹 Network Range Scanned**

****

**192.168.201.0/24** — 254 possible IPs  
(Since your system IP is 192.168.201.131, this is the correct range.)

**🔹 Scan Result:**

| **IP Address** | **Status** | **Open Ports** | **Services** | **MAC Address** |
| --- | --- | --- | --- | --- |
| 192.168.201.1 | Host Up | 3306 | MySQL | 00:50:56:C0:00:08 (VMware) |
| 192.168.201.2 | Host Up | 53 | DNS (domain) | 00:50:56:F3:5E:B6 (VMware) |
| 192.168.201.254 | Host Up | None | — | 00:50:56:F9:90:B2 (VMware) |
| 192.168.201.131 | Host Up | None | — | (Your own IP) |

**4 Hosts Detected as Active**

**Open Ports Found:**

| **IP Address** | **Open Port** | **Service** |
| --- | --- | --- |
| 192.168.201.1 | 3306 | MySQL |
| 192.168.201.2 | 53 | DNS |

**Task Name: TCP SYN Scan and Wireshark Packet Analysis**

**Tools Used**

* **Nmap** – for performing TCP SYN scan.
* **Wireshark** – for capturing and analyzing network packets.

**1. Local IP Address and IP Range**

To find the local IP address and subnet, we used the following command:

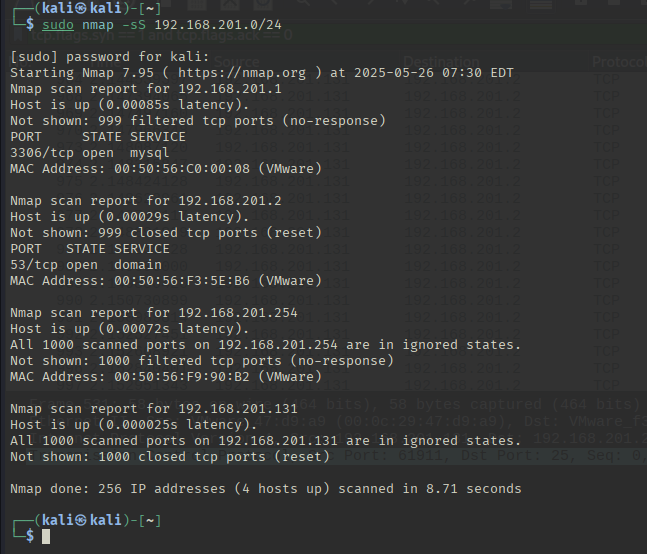
ipconfig

**Result:**

* IP Address: 192.168.201.131
* Subnet Mask: 255.255.255.0
* Therefore, **IP Range**: 192.168.201.0/24

**2. Performing TCP SYN Scan using Nmap**

nmap -sS 192.168.201.0/24



**Results:**

| **IP Address** | **Open Port(s)** | **Service** |
| --- | --- | --- |
| 192.168.201.1 | 3306/tcp | MySQL |
| 192.168.201.2 | 53/tcp | DNS |
| 192.168.201.254 | None | — |
| 192.168.201.131 | None | — |

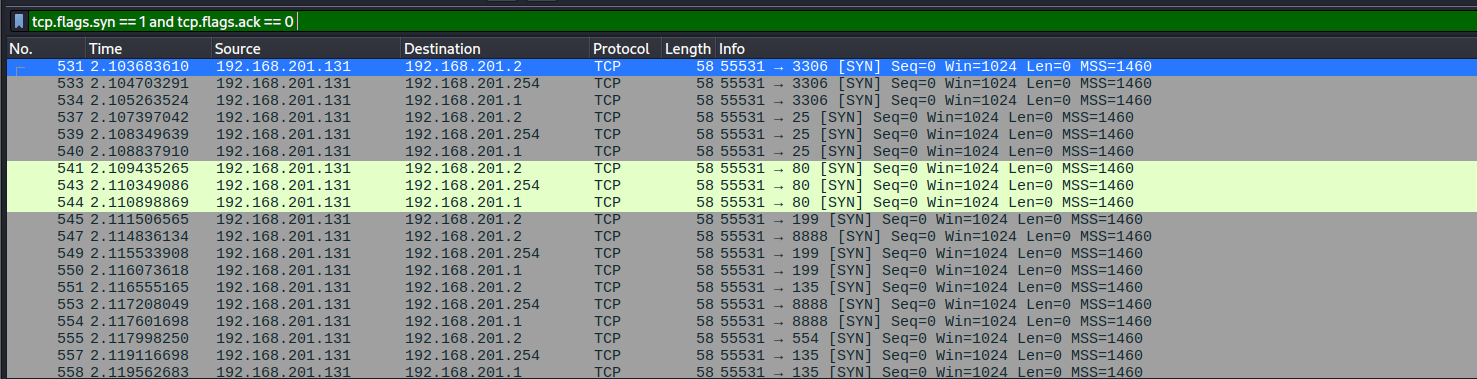
Out of 256 hosts scanned, **4 hosts were up**.

**3. Capturing Packets with Wireshark**

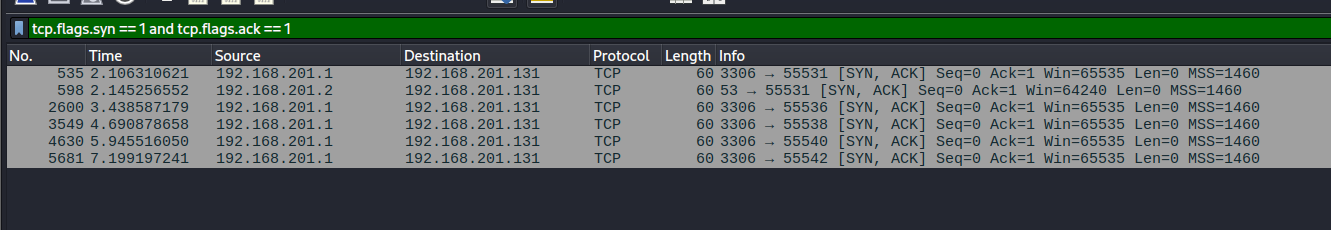
**Steps:**

1. Open Wireshark.
2. Select your network interface (eth0 or ens33 depending on setup).
3. Run the nmap -sS scan in the terminal while Wireshark is capturing.
4. Stop the capture once the scan completes.
5. Apply display filter in Wireshark:

tcp.flags.syn == 1 and tcp.flags.ack == 0



tcp.flags.syn == 1 and tcp.flags.ack == 1



**4. Research: Common Services Running on Open Ports**

During the Nmap TCP SYN scan, the following ports were identified as open:

**Port 3306 – MySQL**

* **Description**: MySQL is a widely-used open-source relational database management system.
* **Common Use**: Used to store and manage data in web applications, internal tools, and enterprise systems.
* **Default Configuration**: Listens on port 3306 for TCP connections.

**Port 53 – DNS (Domain Name System)**

* **Description**: DNS is a service that translates human-friendly domain names (like google.com) into IP addresses.
* **Common Use**: Core part of the internet infrastructure; both recursive and authoritative DNS servers operate on this port.
* **Protocols**: Uses both TCP and UDP (Nmap scan found TCP port 53 open).

**5. Identify Potential Security Risks from Open Ports**

**Port 3306 (MySQL) – Risks**

| **Risk** | **Description** |
| --- | --- |
| **Unsecured Database Access** | If exposed externally, attackers may brute-force or guess MySQL login credentials. |
| **Data Leakage** | Poorly configured databases may allow data to be dumped or exported by attackers. |
| **Outdated Software Vulnerabilities** | Older MySQL versions may have known vulnerabilities that can be exploited. |
| **No IP Restrictions** | If no firewall or binding settings are enforced, anyone on the network can attempt connection. |

**Mitigations:**

* Restrict port 3306 access to trusted IPs only.
* Use strong MySQL passwords and disable remote root login.
* Regularly update MySQL to the latest stable version.
* Implement host-based firewall rules (e.g., iptables, ufw).

**Port 53 (DNS) – Risks**

| **Risk** | **Description** |
| --- | --- |
| **DNS Amplification Attacks** | Open recursive DNS servers can be used for DDoS amplification. |
| **Zone Transfer Information Leak** | If improperly secured, DNS servers may allow attackers to request all domain records. |
| **Internal Reconnaissance** | Attackers on the local network can use the DNS server to map internal domains. |

**Mitigations:**

* Disable recursion on authoritative DNS servers if not required.
* Block zone transfers from unauthorized IPs.
* Restrict access to the DNS server on port 53 using firewall rules.
* Monitor DNS query logs for unusual activity.