# Experiment 2: Nmap tool

## Eshan
## K027

**Aim:** To Install and use NMAP to for gathering information.

**Learning Outcomes:**
After completion of this experiment, student should be able to

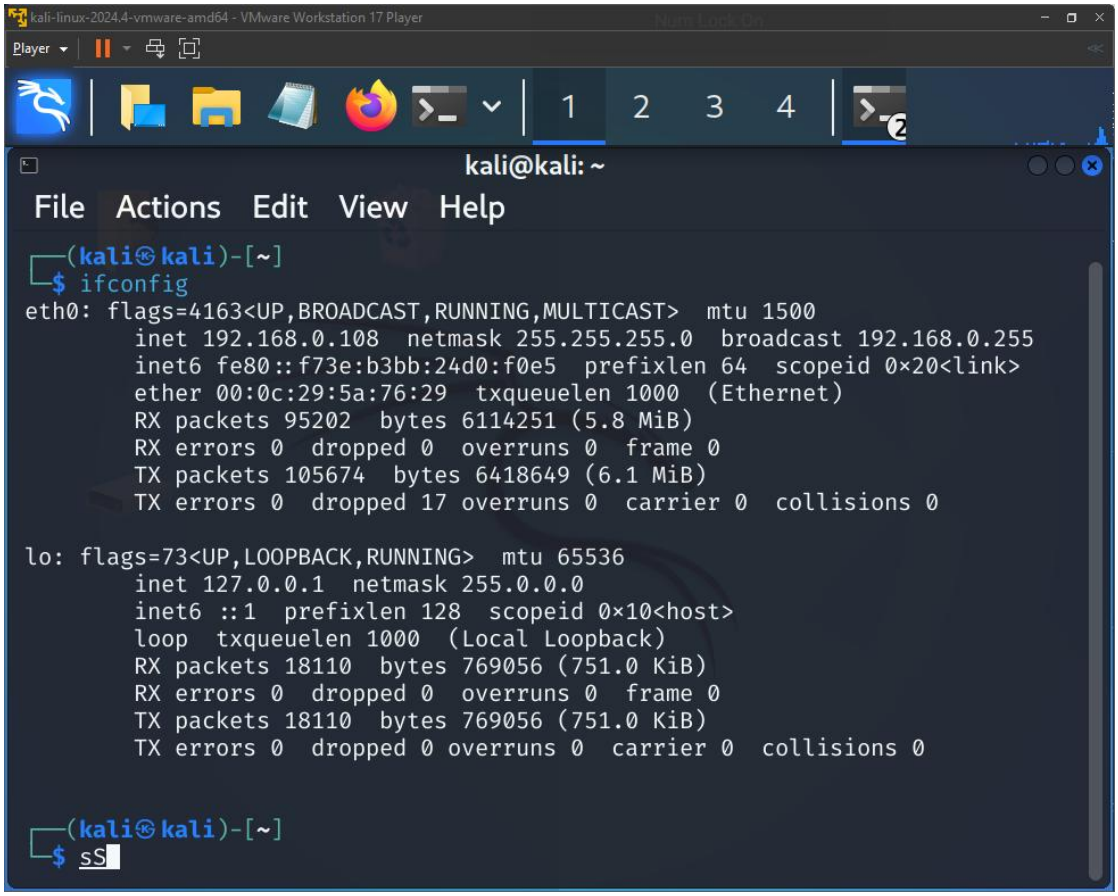
1. Perform port scanning
2. Identify services running on the target system
3. Identify OS available of the target system.

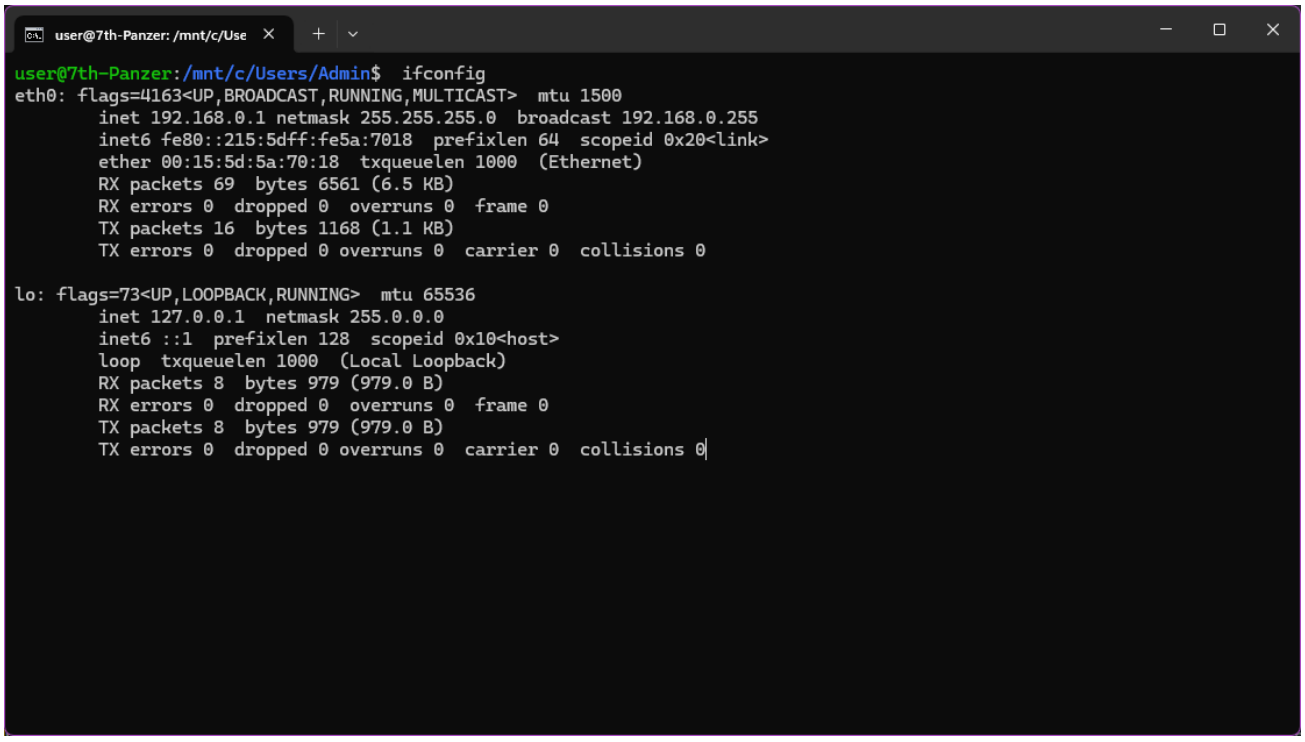
**Theory:**

**Nmap** is a security scanner used to discover hosts and services on a computer network, thus creating a "map" of the network. To accomplish its goal, Nmap sends specially crafted packets to the target host and then analyzes the responses. The software provides a number of features for probing computer networks, including host discovery and service and operating system detection. Nmap is a free open source tool that quickly and efficiently performs ping sweeps, port scanning, service identification, IP address detection, and operating system detection. Nmap has the benefit of scanning of large number of machines in a single session. It's supported by many operating systems, including Unix, Windows, and Linux.

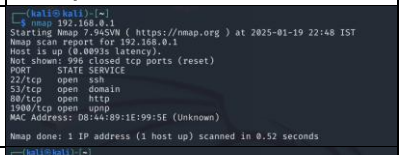
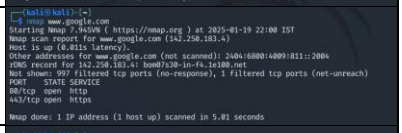
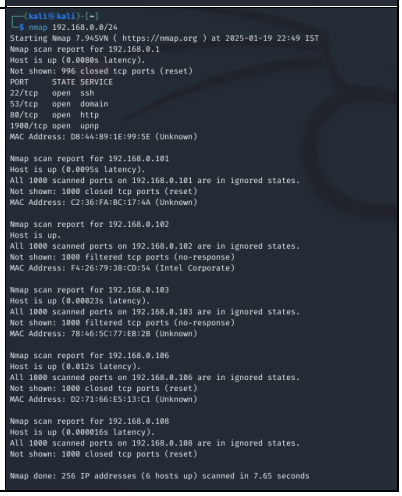
**Setup:**
To perform this lab, you will need two host systems. One system will be running nmap where as other system will be running wireshark for capturing packet.

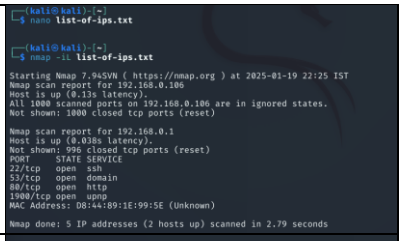
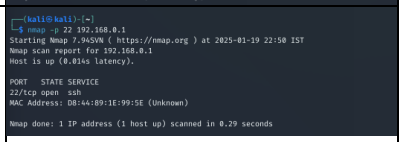
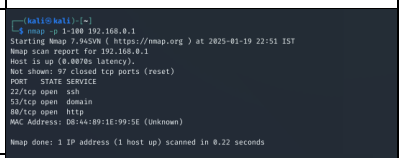
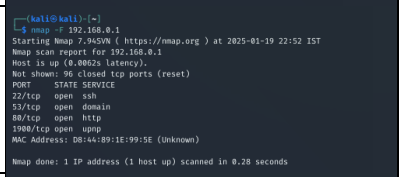
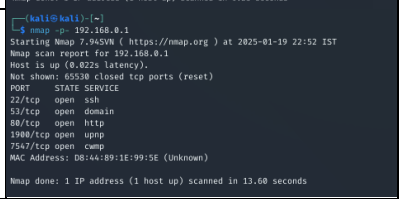
**Procedure:**
1. Open virtual box.
2. Start SEEDUbuntu1 VM.
3. Start SEEDUbuntu2 VM.
4. Note the IP of SEEDUbuntu1 and 2 using *ifconfig* command.


kali-linux-2024.4-vmware-amd64 - VMware Workstation 17 Player
Player
1
2
3
4
kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.0.108 netmask 255.255.255.0 broadcast 192.168.0.255
inet6 fe80::f73e:b3bb:24d0:f0e5 prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:5a:76:29 txqueuelen 1000 (Ethernet)
RX packets 95202 bytes 6114251 (5.8 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 105674 bytes 6418649 (6.1 MiB)
TX errors 0 dropped 17 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 18110 bytes 769056 (751.0 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 18110 bytes 769056 (751.0 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
(kali㉿kali)-[~]
$ sS


user@7th-Panzer: /mnt/c/Use
user@7th-Panzer:/mnt/c/Users/Admin$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.0.1 netmask 255.255.255.0 broadcast 192.168.0.255
inet6 fe80::215:5dff:fe5a:7018 prefixlen 64 scopeid 0x20<link>
ether 00:15:5d:5a:70:18 txqueuelen 1000 (Ethernet)
RX packets 69 bytes 6561 (6.5 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 16 bytes 1168 (1.1 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 8 bytes 979 (979.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 8 bytes 979 (979.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

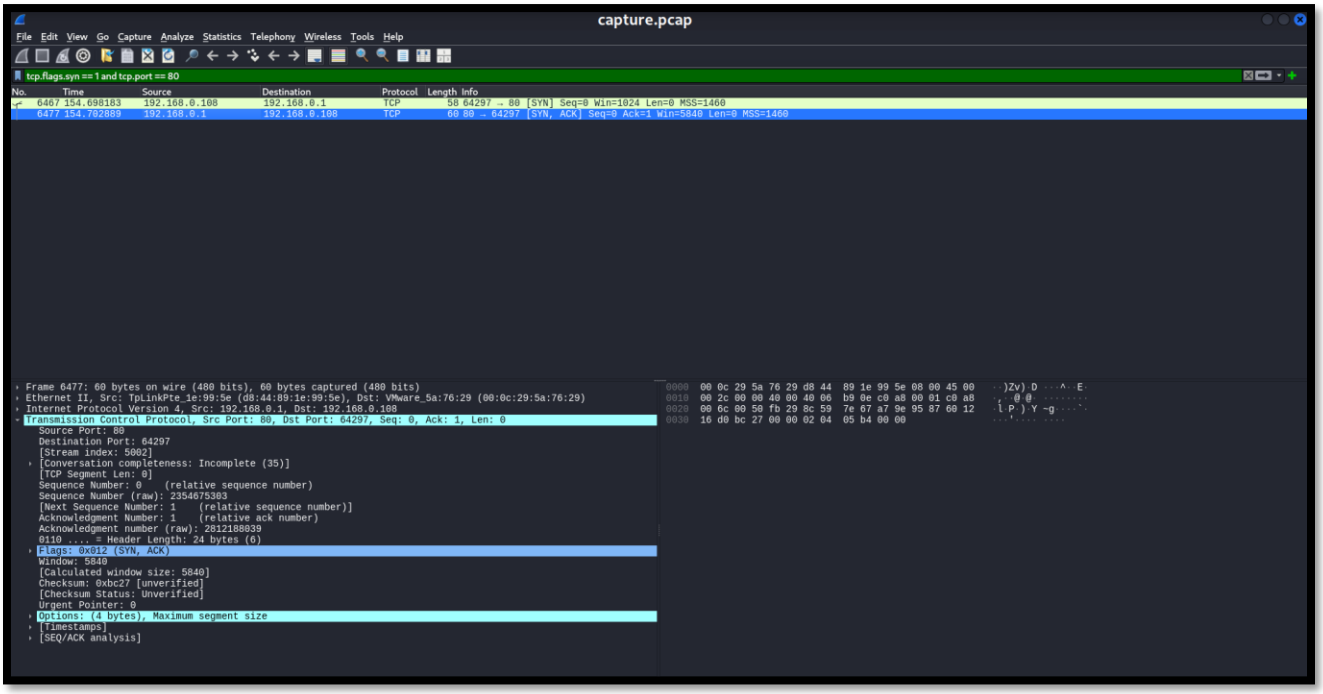
5. Verify the connectivity between SEEDUbuntu1 and 2 using *ping* command.
6. Verify installation of nmap on SEEDUbuntu1 VM.
7. Verify installation of wireshark on SEEDUbuntu2 VM.
8. Start packet capturing using wireshark on SEEDUbuntu2 VM.
9. Execute following nmap commands and document the output.

| Description | Nmap command | Output |
|---|---|---|
| Scan a single IP | nmap 192.168.0.1 |  |
| Scan a host | nmap www.google.com |  |
| Scan a range of IPs | nmap 192.168.0.1-150 |  |
| Scan a subnet | nmap 192.168.0.0/24 |  |

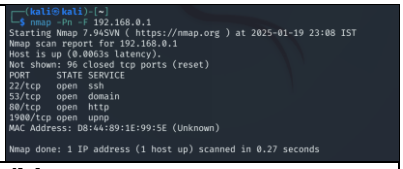
| | | |
|---|---|---|
| Scan targets from a text file | nmap -iL list-of-ips.txt |  |
| Scan a single Port | nmap -p 22 192.168.0.1 |  |
| Scan a range of ports | nmap -p 1-100 192.168.0.1 |  |
| Scan 100 most common ports (Fast) | nmap -F 192.168.0.1 |  |
| Scan all 65535 ports | nmap -p- 192.168.0.1 |  |
| Scan a single Port | nmap -p 22 192.168.0.1 |  |
| Note: Replace IP address with that of SEEDUbuntu 2 VM | | |

10. Perform port scanning using various types of scanning techniques as mentioned in the table below:
    a. Before start of any scan, do the following:
        i. Start packet capturing on SEEDUbuntu2 VM
        ii. Set filter to ip.addr= = SEEDUbuntu1 VM IP
        iii. Execute nmap commands on SEEDUbuntu1 VM
        iv. Follow TCP stream for at least one open port and one closed port in each case and note the flag status.

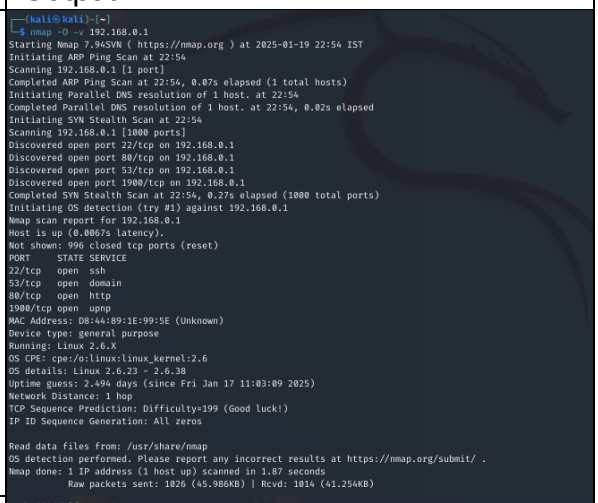
Open Port Flag




| Scanning technique | Command | Output |
|---|---|---|
| Scan using TCP connect | nmap -sT 192.168.0.1 | (kali㉿kali)-[~]<br>$ nmap -sT 192.168.0.1<br>Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-19 23:03 IST<br>Nmap scan report for 192.168.0.1<br>Host is up (0.014s latency).<br>Not shown: 996 closed tcp ports (conn-refused)<br>PORT STATE SERVICE<br>22/tcp open ssh<br>53/tcp open domain<br>80/tcp open http<br>1900/tcp open upnp<br>MAC Address: D8:44:89:1E:99:5E (Unknown)<br><br>Nmap done: 1 IP address (1 host up) scanned in 0.36 seconds |
| Scan using TCP SYN scan (default) | nmap -sS 192.168.0.1 | (kali㉿kali)-[~]<br>$ nmap -sS 192.168.0.1<br>Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-19 23:03 IST<br>Nmap scan report for 192.168.0.1<br>Host is up (0.011s latency).<br>Not shown: 996 closed tcp ports (reset)<br>PORT STATE SERVICE<br>22/tcp open ssh<br>53/tcp open domain<br>80/tcp open http<br>1900/tcp open upnp<br>MAC Address: D8:44:89:1E:99:5E (Unknown)<br><br>Nmap done: 1 IP address (1 host up) scanned in 0.53 seconds |
| Scan UDP ports | nmap -sU -p 123,161,162 192.168.0.1 | (kali㉿kali)-[~]<br>$ nmap -sU -p 123,161,162 192.168.0.1<br>Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-19 23:04 IST<br>Nmap scan report for 192.168.0.1<br>Host is up (0.0043s latency).<br><br>PORT STATE SERVICE<br>123/udp closed ntp<br>161/udp closed snmp<br>162/udp closed snmptrap<br>MAC Address: D8:44:89:1E:99:5E (Unknown)<br><br>Nmap done: 1 IP address (1 host up) scanned in 0.31 seconds |
| Scan using FIN flag | nmap –sF 192.168.0.1 | (kali㉿kali)-[~]<br>$ nmap -sF 192.168.0.1<br>Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-19 23:04 IST<br>Nmap scan report for 192.168.0.1<br>Host is up (0.0045s latency).<br>All 1000 scanned ports on 192.168.0.1 are in ignored states.<br>Not shown: 1000 open\|filtered tcp ports (no-response)<br>MAC Address: D8:44:89:1E:99:5E (Unknown)<br><br>Nmap done: 1 IP address (1 host up) scanned in 21.48 seconds |
| Null Scan | nmap –sN 192.168.0.1 | (kali㉿kali)-[~]<br>$ nmap -sN 192.168.0.1<br>Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-19 23:06 IST<br>Nmap scan report for 192.168.0.1<br>Host is up (0.0050s latency).<br>All 1000 scanned ports on 192.168.0.1 are in ignored states.<br>Not shown: 1000 open\|filtered tcp ports (no-response)<br>MAC Address: D8:44:89:1E:99:5E (Unknown)<br><br>Nmap done: 1 IP address (1 host up) scanned in 21.48 seconds |
| XMAS scan | nmap –sX 192.168.0.1 | (kali㉿kali)-[~]<br>$ nmap -sX 192.168.0.1<br>Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-19 23:07 IST<br>Nmap scan report for 192.168.0.1<br>Host is up (0.0047s latency).<br>All 1000 scanned ports on 192.168.0.1 are in ignored states.<br>Not shown: 1000 open\|filtered tcp ports (no-response)<br>MAC Address: D8:44:89:1E:99:5E (Unknown)<br><br>Nmap done: 1 IP address (1 host up) scanned in 21.47 seconds |
| Ping Scan | nmap -sP 192.168.0.1 | (kali㉿kali)-[~]<br>$ nmap -sP 192.168.0.1<br>Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-19 23:07 IST<br>Nmap scan report for 192.168.0.1<br>Host is up (0.0045s latency).<br>MAC Address: D8:44:89:1E:99:5E (Unknown)<br>Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds |
| ACK scan | nmap -sA 192.168.0.1 | (kali㉿kali)-[~]<br>$ nmap -sA 192.168.0.1<br>Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-19 23:07 IST<br>Nmap scan report for 192.168.0.1<br>Host is up (0.014s latency).<br>All 1000 scanned ports on 192.168.0.1 are in ignored states.<br>Not shown: 1000 unfiltered tcp ports (reset)<br>MAC Address: D8:44:89:1E:99:5E (Unknown)<br><br>Nmap done: 1 IP address (1 host up) scanned in 0.54 seconds |

| | | |
|---|---|---|
| Scan selected ports - ignore discovery | nmap -Pn -F 192.168.0.1 |  |
| Note: Replace IP address with that of SEEDUbuntu 2 VM | | |

11. Perform the above mentioned scan on your host machine and note the difference in output.
12. Execute following commands for OS detection.

| Command | Output |
|---|---|
| nmap -O -v 192.168.0.1 |  |
| nmap -sV –O -v 192.168.0.1 |  |

| | |
|---|---|
| nmap -A 192.168.0.1 |  |

13. Interpret the result. (Refer online documentation at http://nmap.org/docs.html )

The result from Nmap scans shows the network's structure, listing the open, closed, and filtered ports along with their respective services. By interpreting the output:

- **Open ports**: Highlight services that are running and accessible, which could indicate a potential vulnerability.
- **Closed ports**: Respond to probes but do not have an active service.
- **Filtered ports**: Are blocked by firewalls, meaning Nmap cannot determine their status. Use the information provided in Nmap's output (e.g., 22/tcp open ssh) to understand what services the system is offering.

PORT STATE SERVICE
22/tcp open ssh
53/tcp open domain
80/tcp open http
1900/tcp open upnp

7547/tcp open  cwmp

14. Identify which ports are open
    22,53,80,1900

15. Identify various services available on open ports.
    Ssh,domain,http,upnp,cwmp

16. Identify OS installed on the target system
    OS CPE: cpe:/o:linux:linux_kernel:2.6
    OS details: Linux 2.6.23 - 2.6.38

17. Document your result.

    The assignment focuses on using Nmap, a security scanner, to gather network information, including host discovery, service identification, and OS detection. Nmap sends crafted packets to target hosts and analyzes their responses. Key objectives include performing port scans, identifying services on open ports, and detecting the OS of a target system.

    The experiment involves two Linux machines, one running Nmap and the other running Wireshark to capture packets. Various Nmap commands are executed to scan single IPs, ranges, subnets, and specific ports, with the results documented. Advanced scanning techniques, including TCP Connect, SYN, UDP, FIN, Null, Xmas, Ping, and ACK scans, are explored to identify open, closed, and filtered ports.

    Additional tasks include OS detection using commands like nmap -O and service identification via version detection (nmap -sV). The assignment highlights interpreting scan outputs, identifying vulnerabilities, and leveraging Nmap Scripting Engine (NSE) to automate tasks like service detection and vulnerability assessment.

    The experiment emphasizes network security insights, showing how tools like Nmap can map network structures and identify potential vulnerabilities in open ports or services

**Review question:**

1. What is the difference between open, filtered and unfiltered port?

   Ports are endpoints for communication between devices over a network. Their state determines how they respond to network scans or connection attempts.

   - **Open Port**:
     - An open port actively listens for incoming connections. This means a service or application is actively running and accepting requests on this port.
     - Example: Port 80 (HTTP) is open when a web server (like Apache or Nginx)

is running.
    - Implication: It could be a potential entry point for attackers if not secured properly.
- **Filtered Port**:
    - A filtered port means that Nmap cannot determine whether the port is open or closed because a firewall or other security device is preventing communication.
    - Instead of a direct response, the firewall drops or rejects the probing packets.
    - Implication: This indicates the presence of security mechanisms, such as firewalls or intrusion prevention systems.
- **Unfiltered Port**:
    - An unfiltered port is accessible but Nmap cannot determine whether it is open or closed.
    - It means the port is reachable, but Nmap didn't get a response indicating an active service.
    - Implication: Unfiltered ports may require additional investigation as they could potentially be used for communication.

2. What are the different scans possible with Nmap?

Nmap offers a variety of scanning techniques to identify open ports, services, and vulnerabilities. Here are the main types of scans:

1. **TCP Connect Scan (-sT)**:
    - The simplest scan where Nmap completes the full TCP handshake.
    - Useful when you don't have root privileges.
    - Drawback: It is easily detectable by security devices.
2. **SYN Scan (-sS)**:
    - Also known as a "stealth scan," it only sends SYN packets and waits for a response.
    - If a SYN-ACK is received, the port is open; if RST is received, the port is closed.
    - Advantages: Fast and less likely to be detected compared to a TCP connect scan.
3. **UDP Scan (-sU)**:
    - Scans for open UDP ports.
    - Because UDP is connectionless, responses like ICMP "port unreachable" are used to determine closed ports.
    - Slower than TCP scans due to the lack of reliable responses.
4. **ACK Scan (-sA)**:
    - Used to check whether a port is filtered or unfiltered.
    - Does not determine whether a port is open or closed.
    - Typically used for mapping firewall rules.

5. **FIN Scan (-sF)**:
    - Sends a FIN packet (without initiating a handshake).
    - If there is no response, the port is open or filtered; if RST is received, the port is closed.
    - Effective against systems that adhere strictly to RFC 793.
6. **Xmas Scan (-sX)**:
    - Sends packets with FIN, PSH, and URG flags set.
    - Similar to the FIN scan, often used to detect open ports on systems that comply with RFC 793.
7. **NULL Scan (-sN)**:
    - Sends packets with no flags set.
    - The absence of a response indicates an open or filtered port.
    - Useful for bypassing certain firewall rules.
8. **Idle Scan (-sI)**:
    - An advanced technique that uses a third-party host (zombie) to perform scans.
    - Extremely stealthy and allows for anonymized scans.
9. **Ping Scan (-sn)**:
    - Used to identify live hosts without performing port scans.
    - Does not send probes to specific ports.
10. **Version Detection Scan (-sV)**:
    - Identifies the versions of services running on open ports.
    - Helps in identifying vulnerabilities associated with specific versions.
11. **OS Detection Scan (-O)**:
    - Attempts to determine the operating system running on the target host.
    - Uses TCP/IP stack fingerprinting.
12. **Script Scan (-sC)**:
    - Executes default Nmap Scripting Engine (NSE) scripts to detect vulnerabilities, gather information, or perform specific checks.
13. **Comprehensive Scan**:
    - Combines multiple options, like -sS, -sV, -O, and -sC, to perform an in-depth analysis of a target.

3. Explain NSE script with an example.

The **Nmap Scripting Engine (NSE)** is one of the most powerful features of Nmap, allowing users to automate a variety of network scanning tasks, such as:

- Service detection
- Vulnerability assessment
- Exploitation
- Information gathering

NSE uses scripts written in the Lua programming language, and these scripts allow Nmap to perform more than just basic port scanning, enabling it to gather detailed information about hosts, services, and potential vulnerabilities.

Example

```
description = "Checks if port 22 (SSH) is open."

author = "Your Name"
```

```
license = "Same as Nmap"
categories = {"discovery"}

-- Rule: Run this script only if port 22 is open
portrule = function(host, port)
    return port.number == 22 and port.protocol == "tcp" and port.state ==
     "open"
end

-- Action: What to do if the rule is satisfied
action = function(host, port)
    return "Port 22 is open and ready for SSH."
end
```