

PRESIDENCY UNIVERSITY

BENGALURU

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CSE3097 WEB SECURITY

LAB MANUAL

B.Tech 6th Semester A.Y (2024-25)

Instructor Incharge : Sreelatha P K

Course Credit Structure : 2-0-2 (3 Credits)

Exp 1 : Network Connection Inspection

Aim:

To analyze the network using commands

Commands for **network connection inspection** in Windows -

1. ipconfig

- Displays detailed IP configuration information for your system.
- Use ipconfig /all to view full details like IP address, subnet mask, default gateway, DNS servers, and MAC addresses.

2. ping

- Tests connectivity to a specific IP address or domain.
- Example: ping google.com sends packets to Google to check if it's reachable and measures round-trip times.

3. tracert

- Traces the route packets take to reach a specific domain or IP address.
- Example: tracert google.com shows all the intermediate devices your packets pass through.

4. netstat

- Displays active connections, open ports, and listening services on your system.
- Example: netstat -an shows all active TCP and UDP connections with their states.
- Use netstat -b to see which applications are using those connections.

5. nslookup

- Resolves domain names to IP addresses and vice versa.
- Example: nslookup google.com shows the IP address for Google's domain.

6. arp

- Displays or modifies the ARP (Address Resolution Protocol) table, which maps IP addresses to MAC addresses.
- Example: arp -a shows the current ARP table for your network.

7. route

- Displays or modifies the routing table, showing how network traffic is directed.
- Example: route print shows the current routing table for your system.

8. getmac

- Displays the MAC addresses of all network adapters on your system.
- Example: getmac shows the physical address for each network adapter.

9. netsh

- A powerful tool for managing and troubleshooting network configurations.
- Examples:
 - `netsh wlan show profile` shows saved Wi-Fi profiles.
 - `netsh interface ipv4 show config` displays IPv4 configurations.

10. systeminfo

- Displays detailed information about your system, including the OS version, hardware, and network card details.
- Example: `systeminfo` lists all system specs.

11. tasklist

- Shows all running processes on your system.
- Example: `tasklist` lists processes with their PID and memory usage.
- Combine with networking tools: `netstat -b` to see which processes are using network connections.

12. whoami

- Displays the current username and domain information for the logged-in user.

13. hostname

- Displays the hostname of the computer.

14. sc

- Manages system services.
- Example: `sc query` displays the status of all services on the system.

15. powercfg

- Manages power settings on your system.
- Example: `powercfg /energy` generates a report about energy usage and power settings.

16. wmic

- Retrieves system and hardware information via the Windows Management Instrumentation Command-line.
- Example: `wmic nic get name,macaddress` displays the name and MAC addresses of network adapters.

17. net

- Manages shared resources, users, and network connections.
- Example: `net view` shows all shared devices on your network.

18. driverquery

- Displays a list of all installed drivers and their details.
- Example: `driverquery /v` gives detailed information about each driver.

19. pathping

- Combines ping and tracert to show packet loss information for each hop.

Example: pathping google.com gives detailed stats for network performance.

EXP 2: Nessus Tool for Monitoring Network Traffic

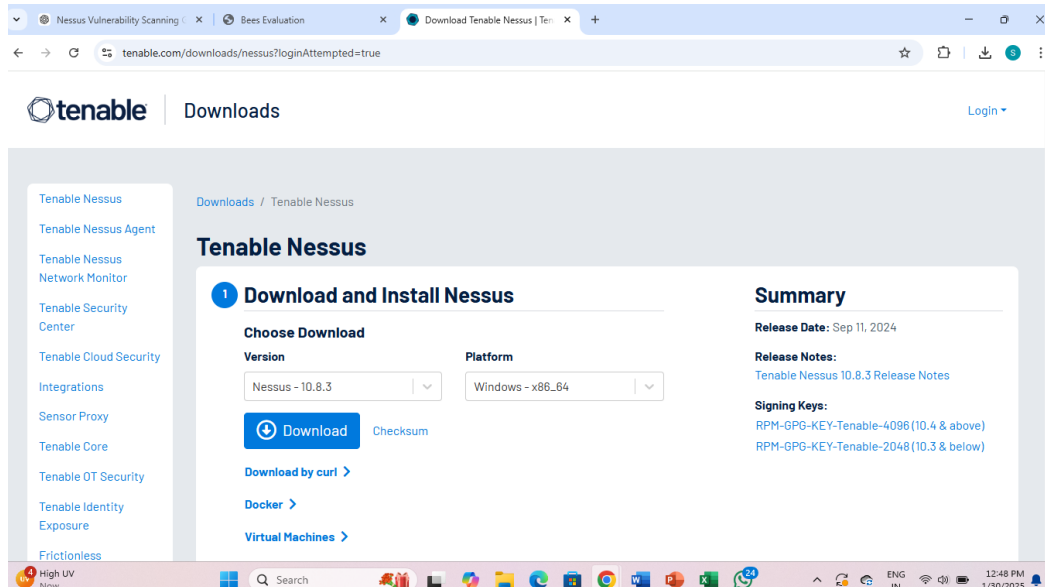
Aim:

To analyze and capture network traffic to detect anomalies, and generate report

Procedure -

1. Download Nessus:

- Visit the [Tenable Downloads page](#).
- Select the appropriate installer for your operating system (e.g., Windows, macOS, Linux distributions).



2. Install Nessus:

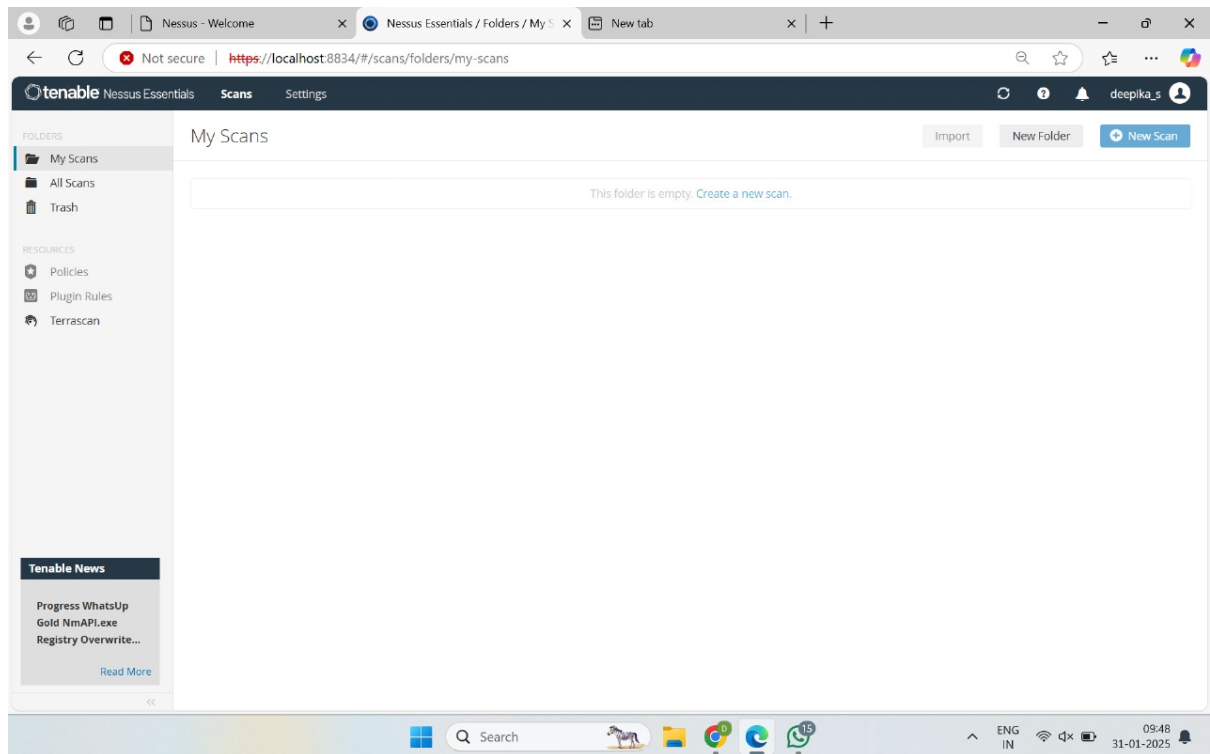
For Windows:

- Locate the downloaded installer and double-click to run it.
- Follow the InstallShield Wizard prompts:
 - Accept the license agreement.
 - Choose the installation directory or proceed with the default.
 - Click 'Install' to begin the installation process.
- Once installed, the Nessus service will start automatically.

3. Configure Nessus:

- Open your web browser and navigate to <https://localhost:8834/> (or replace localhost with your system's IP address if accessing remotely).
- In the web interface:
 - Choose the Nessus product type as **Nessus Essentials**
 - Register by entering the activation code if prompted.
 - Create an administrator account by setting a username and password.
 - Nessus will then download and install the latest plugins, which may take several minutes.

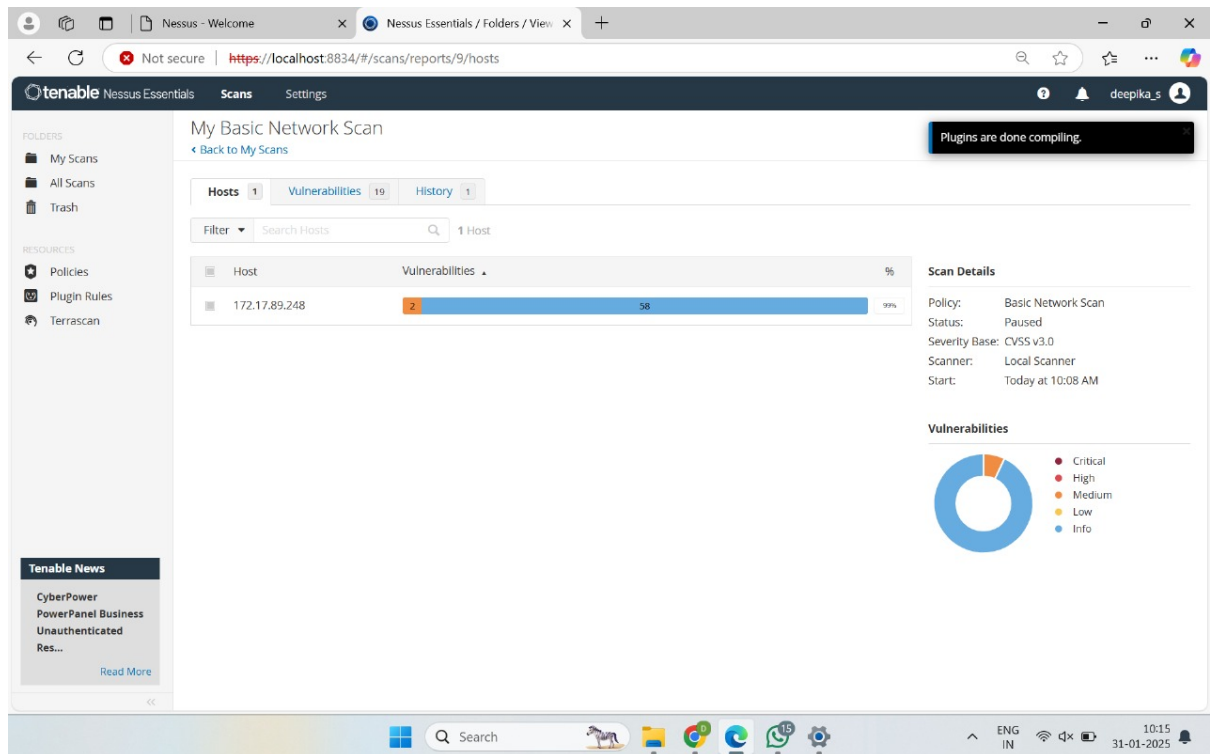
Now, Nessus is ready for use and run vulnerability scans.



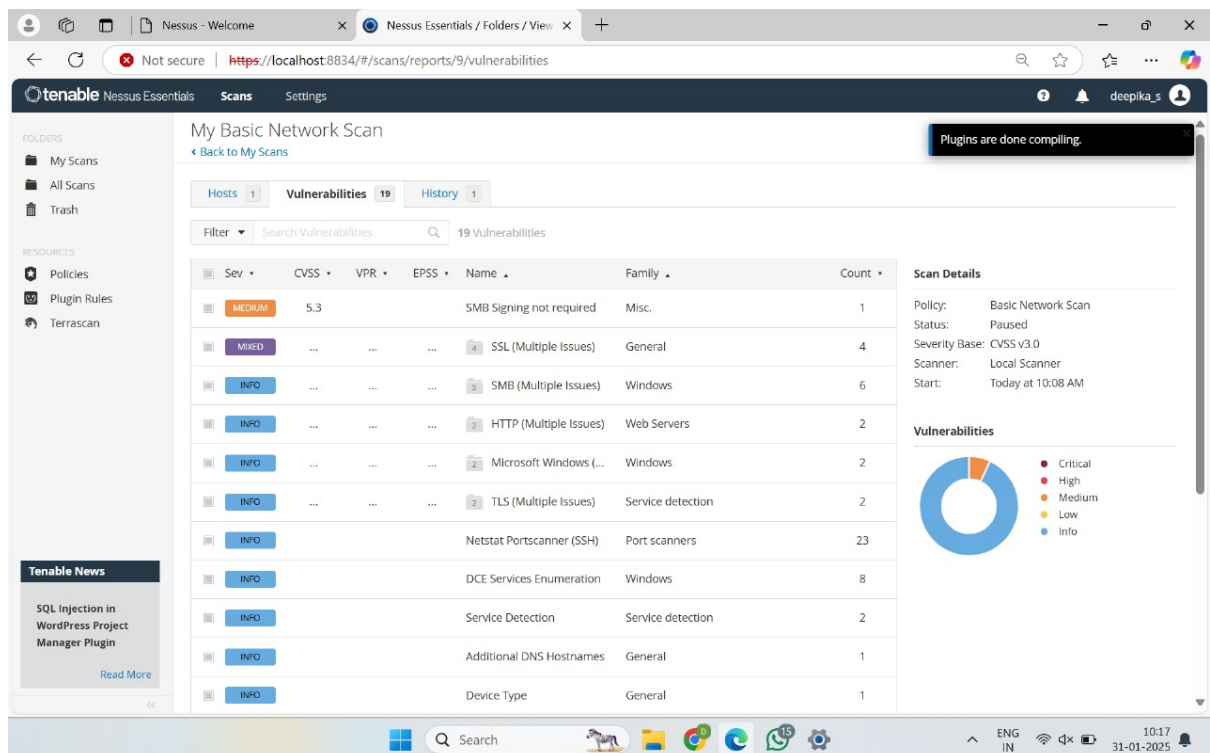
Types of Scans

i. Basic Network Scanning

- **Objective:** Identify active hosts, services, and open ports in a network.
- **Steps:**
 1. Launch the Nessus application.
 2. Create a new scan by selecting **Basic Network Scan**.
 3. Enter the target range (e.g., specific IP addresses, subnets).
 4. Configure optional settings (e.g., credentials for better access).
 5. Launch the scan and monitor progress.
 6. Review the report for active hosts, their open ports, and basic information.



Basic Network Scan



Report Generated

ii. Advanced Scanning in General Search

- **Objective:** Conduct in-depth scanning to identify detailed vulnerabilities in a target network or system.
- **Steps:**
 1. Create a new scan and choose **Advanced Scan**.
 2. Configure detailed options:
 - **Target systems:** Enter IP ranges or hostnames.
 - **Scan templates:** Customize for specific needs like compliance or web applications.
 - Enable deep packet inspection or brute force if required.
 3. Set up **specific policies or rules** for scanning (e.g., timeout limits, retries).
 4. Launch and analyze vulnerabilities, prioritizing critical ones.
 5. Review the report to identify high, medium, and low-severity issues.

iii. Policies

- **Objective:** Define specific scanning rules and configurations.
- **Common Policies:**
 - Credentialed Scans: Use valid credentials for deeper scanning.
 - Compliance Policies: Focus on specific compliance standards (e.g., PCI DSS, HIPAA).
 - Malware Scans: Target malicious files or activities.
- **Configuration:**
 - Go to the **Policies** section in Nessus.
 - Create and save policies based on your scanning needs (e.g., custom port ranges, specific vulnerabilities).

vi. Plugins

- **Objective:** Leverage Nessus plugins to detect specific vulnerabilities.
- **Steps:**
 1. Go to the **Plugins** tab in Nessus.
 2. Enable relevant plugin families:
 - Web Servers (CGI scripts, LAMP vulnerabilities).
 - Databases (MySQL, PostgreSQL).
 - APIs (REST vulnerabilities).
 - Cross-Site Scripting (XSS).
 3. Ensure plugins are updated before scanning.

EXP 3. MONITORING NETWORK TRAFFIC (WIRESHARK)

Aim:

To analyze and capture network traffic to identify patterns, detect anomalies, and assess overall network performance and security.

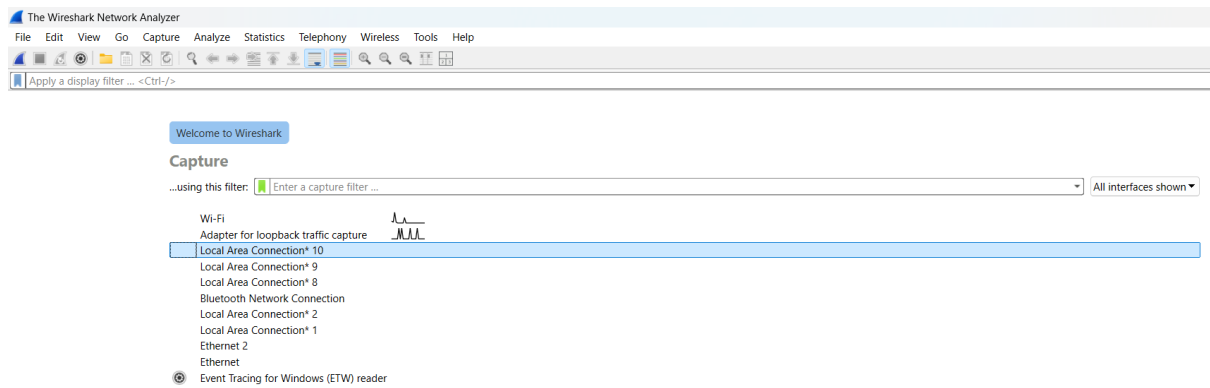
Procedure:**1. Install and Configure Wireshark**

- **Step 1.1: Download Wireshark**

- Visit the official Wireshark website: <https://www.wireshark.org/download.html>.



- Select the appropriate version for your operating system (Windows, macOS, or Linux) and download the installer.
- **Step 1.2: Install Wireshark**
 - Run the downloaded installer and follow the installation prompts.
 - On Windows, you may need to install WinPcap (or Npcap) during the Wireshark setup. These drivers are required to capture network traffic.
- **Step 1.3: Start Wireshark**
 - After installation, open Wireshark. You should see the main user interface.
- **Step 1.4: Select Network Interface**
 - In the Wireshark window, you'll see a list of available network interfaces (such as Ethernet, Wi-Fi, etc.).

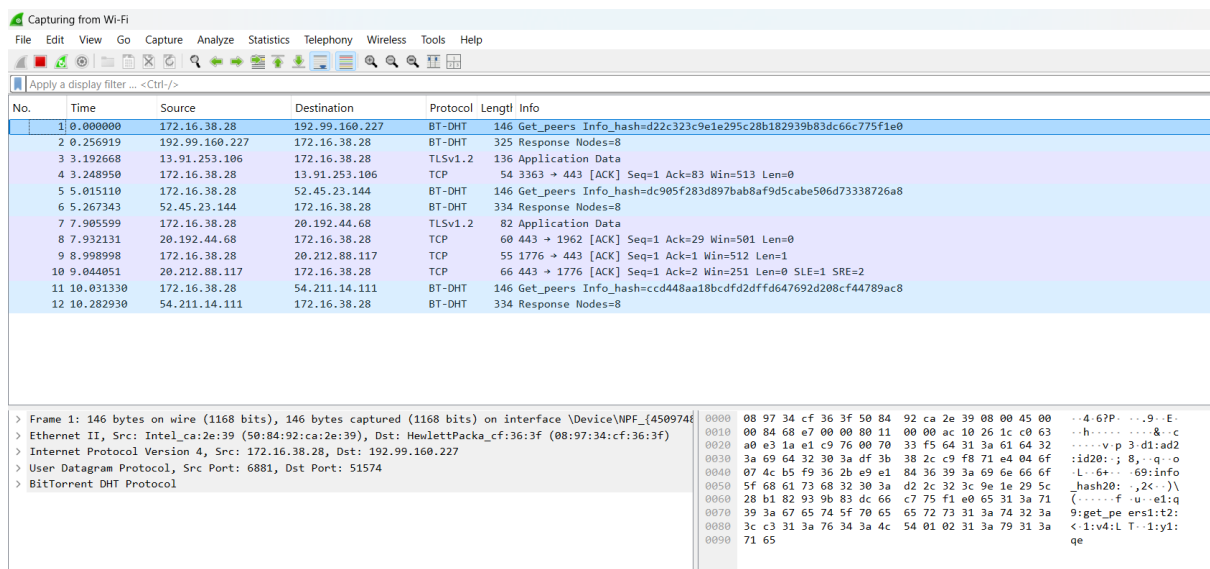


- Choose the correct network interface to capture traffic. If you're unsure, look for the one showing the most activity.

2. Start Capturing Network Traffic on the Network Interface

• Step 2.1: Start Capture

- Click on the network interface (e.g., Wi-Fi or Ethernet) from the main screen of Wireshark to begin capturing traffic.



- As Wireshark starts capturing, you will see packets begin to populate in real-time in the main window.

• Step 2.2: Monitor the Capture Process

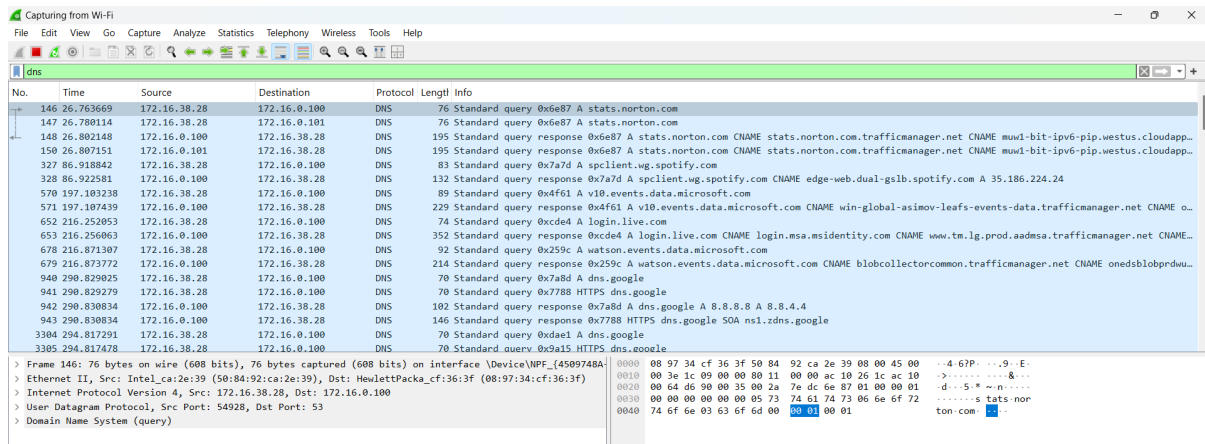
- Observe the packet count and types of traffic being captured. This will provide insight into your network's behavior.
- As packets are captured, Wireshark will display detailed information on each packet, including source and destination IP addresses, protocols, packet length, and more.

3. Use Filters to Focus on Specific Traffic (e.g., HTTP, FTP)

• Step 3.1: Apply Display Filters

- To narrow down the traffic you're interested in, you can apply display filters in the filter bar at the top of the window. For example:

- **HTTP Traffic:** To capture only HTTP traffic, use the filter http.
- **FTP Traffic:** To capture FTP traffic, use ftp.
- **DNS Traffic:** To capture DNS requests, use dns.

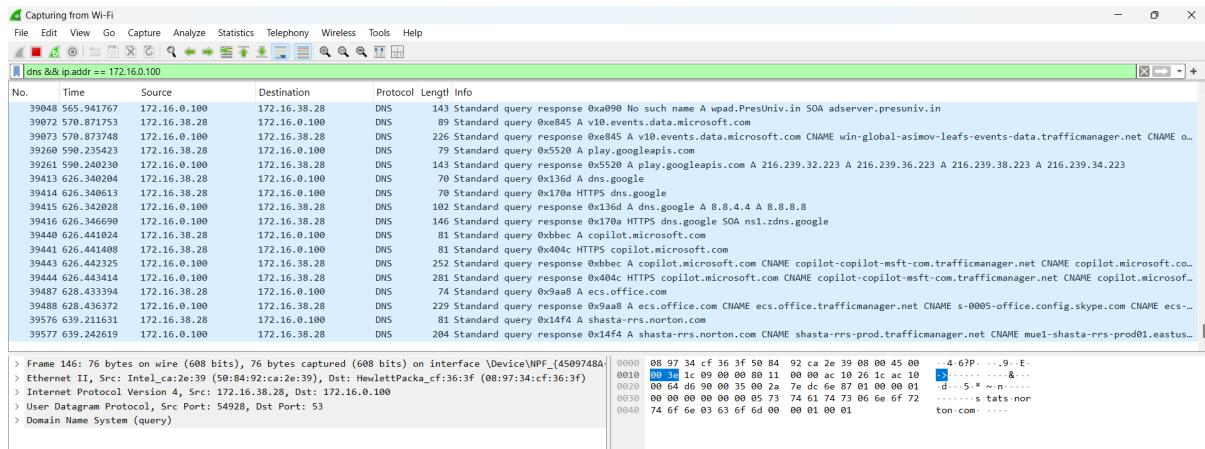


- Wireshark will now display only the packets that match your filter criteria.

• Step 3.2: Use More Advanced Filters

- You can combine filters for more advanced packet analysis. For example, if you want to capture HTTP traffic from a specific IP address, you can use:

- dns && ip.addr == 172.16.0.100

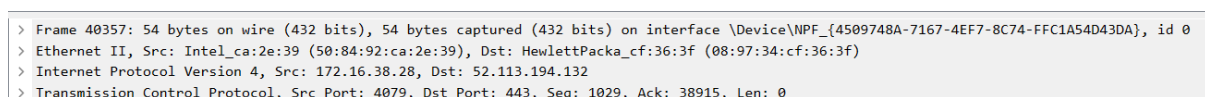


- This filter will show only dns traffic to or from the IP address 172.16.0.100.

4. Analyze Packets for Any Anomalies or Suspicious Activities

• Step 4.1: Inspect Packet Details

- Select a packet from the list, and in the lower part of the screen, Wireshark will show detailed information about the selected packet.



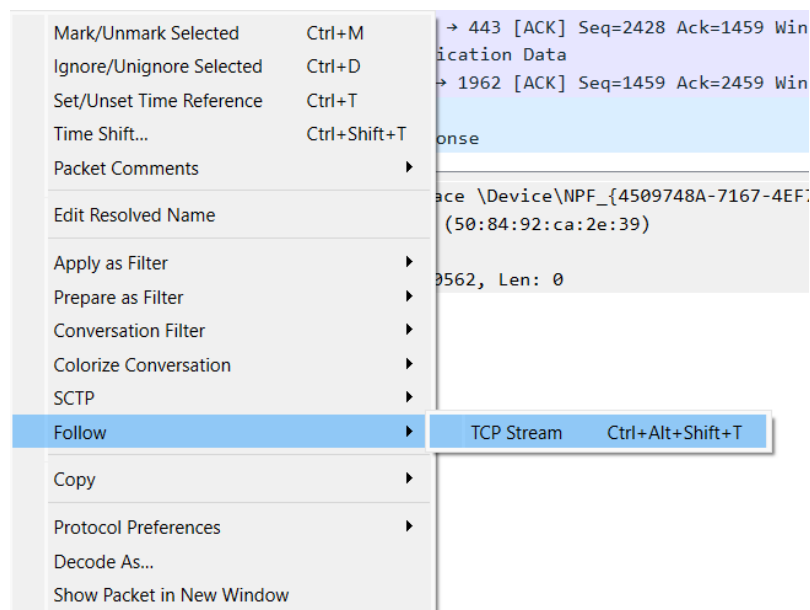
- Look at the **Protocol**, **Source**, **Destination**, **Packet Length**, and other header information for each packet.

- **Step 4.2: Identify Anomalies**

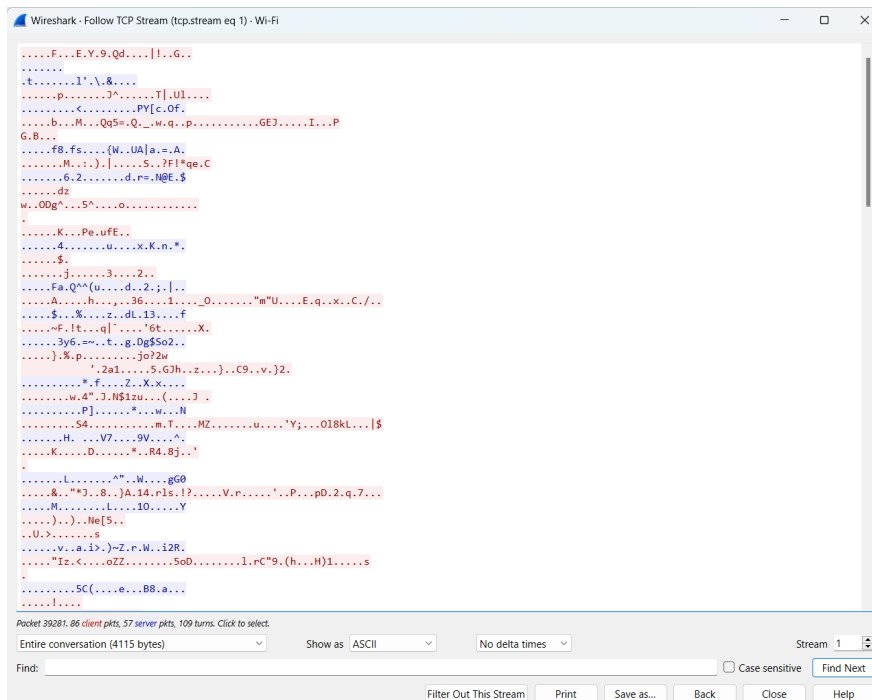
- Common anomalies to look for include:
 - **Unusual Ports:** Packets sent or received on unexpected ports.
 - **Unencrypted Credentials:** Plaintext passwords being transmitted over HTTP or FTP.
 - **Suspicious Protocols:** Unexpected traffic types, such as unusual ICMP requests or unauthorized protocols.
 - **Excessive Traffic:** High traffic on specific ports or protocols, which may indicate a denial-of-service (DoS) attack or botnet activity.
 - **Malformed Packets:** Malformed packets that don't conform to expected protocol formats, often used in exploits.

- **Step 4.3: Follow a Stream**

- Wireshark allows you to “Follow” a stream of packets in a conversation. This is helpful for analyzing a single communication, such as an HTTP request and its response.



- Right-click on a packet and choose **Follow > TCP Stream** (for TCP traffic) or **Follow > UDP Stream** (for UDP traffic). This will display the entire conversation between two endpoints.



5. Save the Capture File for Analysis

- **Step 5.1: Stop the Capture**
 - When you've captured enough data, click on the **Stop** button to end the capture session.
- **Step 5.2: Save the Capture**
 - Go to **File > Save As** to save the captured packets in a .pcap file format (Wireshark's native capture format).
 - Choose a location to save the file, and provide an appropriate name.
 - You can also choose to export the capture data in other formats, like CSV or plain text, depending on the type of analysis you intend to do.

Output:

- A captured packet trace containing network traffic, including source and destination addresses, protocols, and flags.
- Detailed analysis of specific protocols like HTTP, FTP, DNS, etc.

Result:

- **Identification of Unusual or Malicious Traffic:**
 - You may find cleartext passwords being transmitted over HTTP.
 - Uncommon ports being used by unauthorized services.

- Signs of DoS attacks or malware communication based on unexpected traffic patterns.
-

Discussion:

- **Unencrypted Credentials:** Network traffic that is not encrypted (such as HTTP traffic with unencrypted login credentials) can pose significant security risks. For example, capturing HTTP packets containing usernames and passwords exposes sensitive information to potential attackers.
 - **Wireshark as a Detection Tool:** Wireshark is an essential tool for monitoring and detecting suspicious activities in network traffic. By filtering for certain traffic types and analyzing the patterns, network administrators can spot potential vulnerabilities and anomalies that may not be apparent in traditional logs.
 - **Traffic Analysis Techniques:**
 - **Traffic volume:** Excessive amounts of traffic to or from certain ports may indicate an attack in progress (e.g., DDoS).
 - **Unexpected protocols:** Monitoring for rare or unexpected protocols can uncover malicious activity like tunneling.
-

Aim:

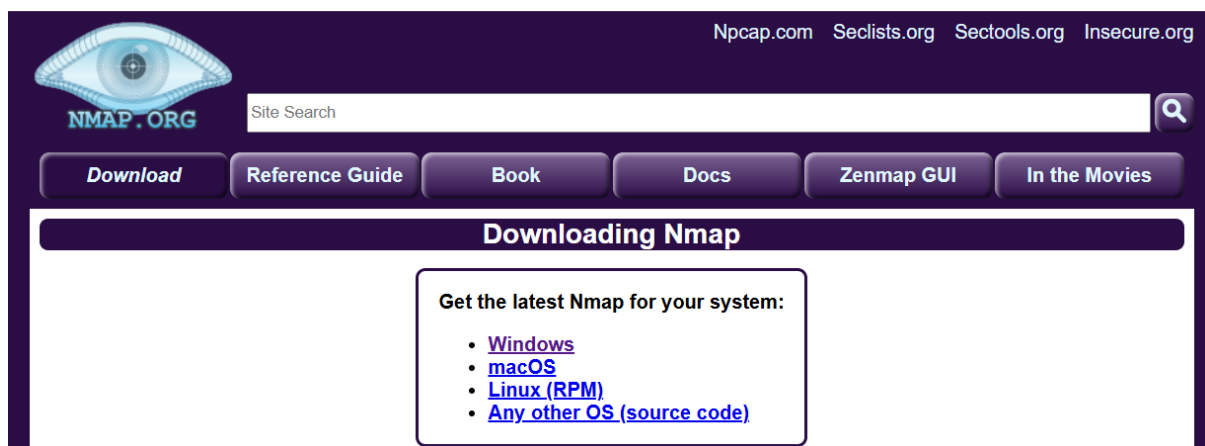
To identify active hosts and the services they are running within a network using Nmap, enabling a comprehensive understanding of the network environment.

Procedure:

1. Install Nmap on Your Machine

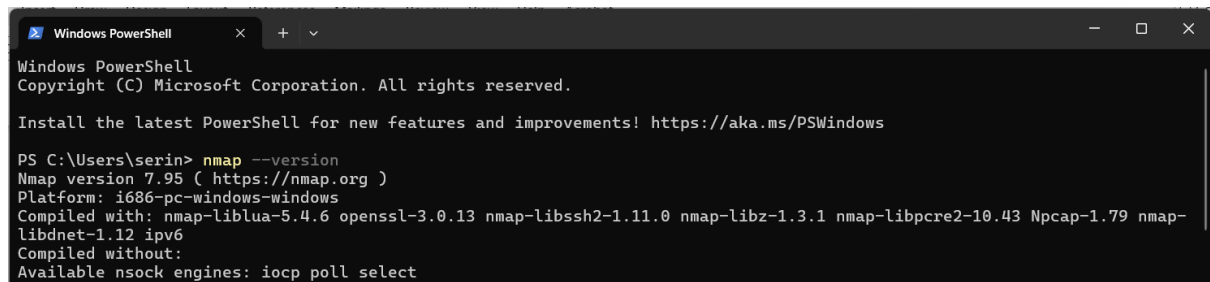
- **Step 1.1: Download Nmap**

- Go to the official Nmap website: <https://nmap.org/download.html>.



- Select the appropriate version for your operating system (Windows, macOS, or Linux).
- **Step 1.2: Install Nmap**
 - For **Windows**: Download the .exe installer and run it. Follow the on-screen instructions to complete the installation.
 - For **macOS**: Use the Homebrew package manager with the command `brew install nmap` (if you have Homebrew installed).
 - For **Linux**: You can install Nmap via your package manager. For example:
 - On **Debian/Ubuntu**: `sudo apt-get install nmap`
 - On **CentOS/Fedora**: `sudo yum install nmap`
- **Step 1.3: Verify Installation**
 - After installation, open a terminal or command prompt and type:

`nmap -version`



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\serin> nmap --version
Nmap version 7.95 ( https://nmap.org )
Platform: i686-pc-windows-windows
Compiled with: nmap-liblua-5.4.6 openssl-3.0.13 nmap-libssh2-1.11.0 nmap-libz-1.3.1 nmap-libpcap-1.10.4 nmap-libnet-1.1.2 ipv6
Compiled without:
Available nsock engines: iocp poll select
```

- This should display the Nmap version installed on your system.

2. Run a Basic Nmap Scan to Discover Active Hosts

- **Step 2.1: Open Terminal/Command Prompt**

- Open a terminal or command prompt on your system.

- **Step 2.2: Run Basic Ping Scan**

- A basic Nmap scan (ping scan) can be used to discover which hosts are up within a network range. This will simply send ICMP (ping) requests to the specified range and identify live hosts.
- Run the following command:

`nmap -sP <network-range>`

For example:

`nmap -sP 172.16.38.28/24`


```
Windows PowerShell
PS C:\Users\serin> nmap -sP 172.16.38.28/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-15 10:02 India Standard Time
Nmap scan report for 172.16.38.0
Host is up (0.0050s latency).
MAC Address: BC:03:58:45:10:C8 (Intel Corporate)
Nmap scan report for 172.16.38.1
Host is up (0.087s latency).
MAC Address: 2E:9C:40:E8:B2:37 (Unknown)
Nmap scan report for 172.16.38.3
Host is up (0.0040s latency).
MAC Address: 00:E1:8C:23:6B:67 (Intel Corporate)
Nmap scan report for 172.16.38.5
Host is up (0.077s latency).
MAC Address: 3A:3D:D3:70:60:00 (Unknown)
Nmap scan report for 172.16.38.7
Host is up (0.0040s latency).
MAC Address: F4:D1:08:A9:E8:F9 (Intel Corporate)
Nmap scan report for 172.16.38.10
Host is up (0.083s latency).
MAC Address: AE:F5:1E:CA:F3:2C (Unknown)
Nmap scan report for 172.16.38.11
Host is up (0.0080s latency).
MAC Address: B4:8C:9D:B4:D9:73 (AzureWave Technology)
Nmap scan report for 172.16.38.13
Host is up (0.069s latency).
MAC Address: A6:32:C1:EF:D3:B2 (Unknown)
Nmap scan report for 172.16.38.14
Host is up (0.0050s latency).
MAC Address: 34:60:F9:DB:3D:73 (TP-Link Limited)
Nmap scan report for 172.16.38.15
Host is up (0.010s latency).
MAC Address: 0C:96:E6:D9:7D:95 (Cloud Network Technology (Samoa) Limited)
Nmap scan report for 172.16.38.16
Host is up (0.31s latency).
MAC Address: 2A:0E:D8:E3:4A:90 (Unknown)
Nmap scan report for 172.16.38.17
Host is up (0.094s latency).
MAC Address: D8:F3:BC:65:AF:C7 (Liteon Technology)
Nmap scan report for 172.16.38.18
Host is up (0.0080s latency).
MAC Address: 78:46:5C:39:2F:1B (Cloud Network Technology Singapore PTE.)
```

- **Explanation:**

- -sP: This tells Nmap to perform a "ping sweep" to discover active hosts.
- <network-range>: Replace this with the target IP range you want to scan (e.g., 192.168.1.0/24 for a typical local network).

- **Step 2.3: Review the Output**

- Nmap will return a list of hosts that are up, along with their IP addresses and hostnames (if available).
- Example output:

Nmap scan report for 192.168.1.1

Host is up (0.0010s latency).

Nmap scan report for 192.168.1.2

Host is up (0.0008s latency).

3. Perform a Service Detection Scan

- **Step 3.1: Run a Service Version Detection Scan**

- Once you've identified live hosts, you can perform a service detection scan to identify open ports and the services running on those ports.
- Use the following command to run a version detection scan:

```
nmap -sV <target-IP>
```

For example:

```
nmap -sV 172.16.38.232
```

```
PS C:\Users\serin> nmap -sV 172.16.38.232
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-15 10:08 India Standard Time
Nmap scan report for 172.16.38.232
Host is up (0.0083s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
5000/tcp  open  rtsp    AirTunes rtspd 665.13.1
7000/tcp  open  rtsp    AirTunes rtspd 665.13.1
MAC Address: 1C:57:DC:54:35:68 (Apple)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 22.63 seconds
```

- **Explanation:**
 - -sV: This tells Nmap to detect the versions of services running on open ports.
 - <target-IP>: Replace this with the IP address of the host you want to scan (e.g., 172.16.38.232).

- **Step 3.2: Review the Service Details**

- The scan will provide a list of open ports, the services running on those ports, and the versions of those services. For example:

Nmap scan report for 192.168.1.1

Host is up (0.0010s latency).

Not shown: 998 closed ports

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Debian 4
80/tcp    open  http     Apache httpd 2.4.38
443/tcp   open  https    Apache httpd 2.4.38
```

- **Explanation:**
 - **PORT:** The port number that is open on the host.
 - **STATE:** The state of the port (open/closed).
 - **SERVICE:** The service running on the open port.

- **VERSION:** The version of the service running on the port (if detectable).

4. Document the Discovered Hosts and Their Services

- **Step 4.1: Record the Discovered Hosts**

- After scanning, create a document or spreadsheet to record the following:
 - Host IP address
 - Hostname (if available)
 - Open ports
 - Services running on each port
 - Service versions

- **Step 4.2: Use a Table for Better Organization**

- A table may help in organizing the results. Example:

IP Address	Hostname	Open Ports	Services Detected	Service Versions
192.168.1.1	router	22, 80, 443	SSH, HTTP, HTTPS	OpenSSH 7.6p1, Apache 2.4.38
192.168.1.2	server1	21, 80, 3306	FTP, HTTP, MySQL	vsftpd 3.0.3, Apache 2.4.38

Output:

- A list of active hosts within the network and the services they are running.
- Detailed information about open ports, associated services, and their versions.

Result:

- **Identification of Open Ports and Services:** This process allows you to identify which ports are open and what services are accessible on those ports.
- **Potential Vulnerabilities:** By identifying the versions of services running, you can look up known vulnerabilities associated with those versions (e.g., CVEs) to assess security risks.

Discussion:

- **Importance of Host Discovery:**
 - Host discovery helps identify active devices within a network. It is a fundamental step in any penetration testing or network assessment process.
 - Detecting all hosts on a network enables security professionals to ensure that all critical systems are being properly secured.
- **Service Enumeration:**
 - Once hosts are discovered, service enumeration is key to understanding the type of services available on the network. Open ports and services can present attack vectors for malicious users.

- For instance, outdated services or exposed services like FTP (which sends data unencrypted) could be exploited by attackers.
- By identifying service versions, security practitioners can search for any known vulnerabilities that might be exploitable.
- **Potential Attack Vectors:**
 - Unsecured services, especially those running outdated versions, can be exploited by attackers. For example, an exposed SSH service with weak authentication might be a potential attack vector.
 - Unauthorized open ports, such as those running unpatched web servers or databases, can expose sensitive information or provide entry points for further attacks.