

Assignment 8

Title: Configure and Demonstrate Use of Vulnerability Assessment Tool like Wireshark

Name: Shubhankar jakate

PRN : 22310371

Roll No = 382019

Aim:

To configure and demonstrate the use of a vulnerability assessment and network analysis tool such as **Wireshark**.

Objectives:

1. To understand how network packets are captured and analyzed.
2. To identify vulnerabilities and suspicious network behavior.
3. To learn how to use Wireshark for real-time traffic monitoring and protocol analysis.

Tools / Software Required:

- Wireshark (Latest version)
- Operating System: Windows
- Network Connection: Wi-Fi

Theory:

Wireshark is an open-source network protocol analyzer used for capturing and inspecting data packets flowing through a network.

It helps in identifying vulnerabilities, diagnosing network issues, and monitoring real-time traffic.

Wireshark captures packets at the network interface and displays detailed information such as source and destination IP addresses, protocols, ports, and payload data.

It supports various protocols like **TCP, UDP, HTTP, HTTPS, DNS, ARP, and ICMP**.

Key Features:

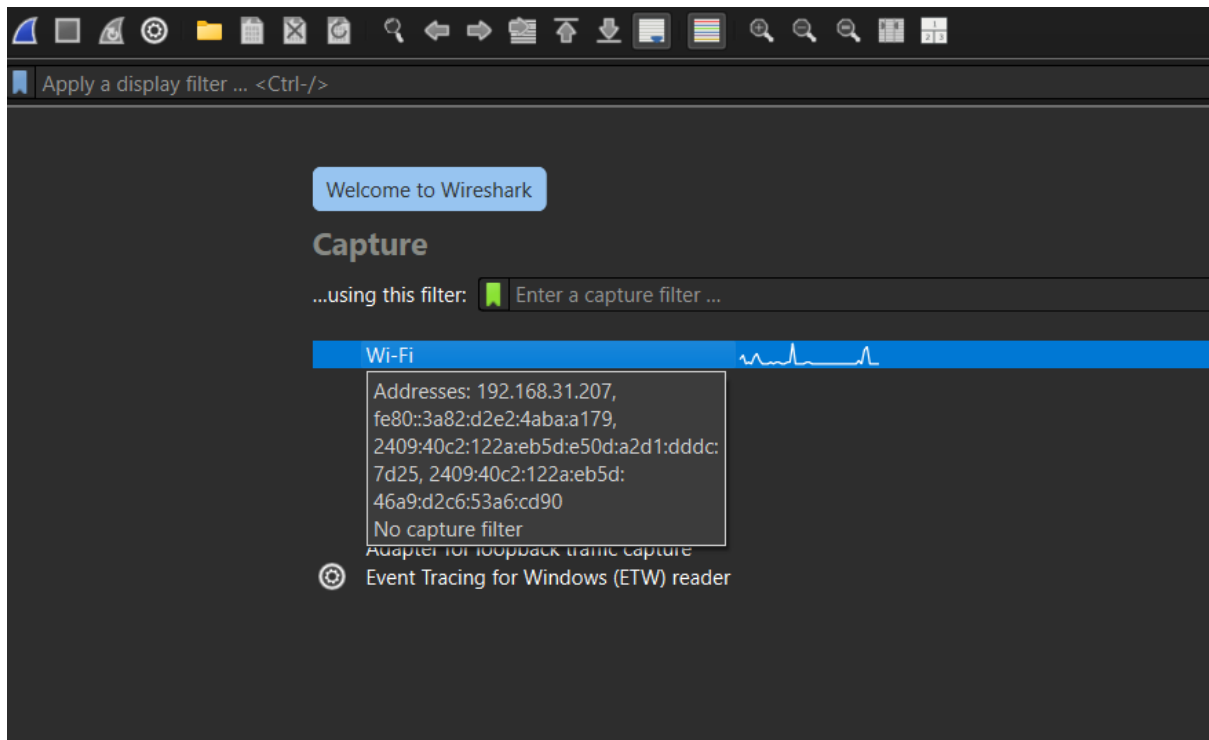
- Real-time packet capture and offline analysis
- Deep inspection of hundreds of protocols
- Filtering tools for specific traffic
- Color-coding for better packet visualization
- Export data for further analysis

Procedure:

1. Install Wireshark from <https://www.wireshark.org/>.
2. Launch Wireshark and select the Wi-Fi network interface.
3. Click on **Start Capturing Packets** (blue shark fin icon).
4. Observe packets being captured in real-time.
5. Apply filters like ip, http, or tcp.port == 80 to focus on specific packets.
6. Select a packet to view details such as source, destination, and protocol information.
7. Identify potential vulnerabilities like unencrypted HTTP traffic or unknown connections.
8. Stop capture and save the session as .pcap file for later analysis.

Screenshots:

Screenshot shows network interface selection



Screenshot shows live packet capture

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::7ebf:77ff:fe3...	ff02::1	ICMPv6	150	Router Advertisement from 7c:bf:77:31:8c:cb
2	3.073409	2404:6800:4009:807:...	2409:40c2:122a:eb5d:...	UDP	143	443 → 63030 Len=81
3	3.075177	2409:40c2:122a:eb5d:...	2404:6800:4009:807:...	UDP	98	63030 → 443 Len=36
4	4.408110	2409:40c2:122a:eb5d:...	2603:1040:f02:e:1b7	TLSv1.2	124	Application Data
5	4.608456	2603:1040:f02:e:1b7	2409:40c2:122a:eb5d:...	TLSv1.2	113	Application Data
6	4.657101	2409:40c2:122a:eb5d:...	2603:1040:f02:e:1b7	TCP	74	52263 → 443 [ACK] Seq=51 Ack=40 Win=254 Len=0
7	5.762785	192.168.31.207	172.188.155.25	TCP	55	60665 → 443 [ACK] Seq=1 Ack=1 Win=254 Len=1
8	5.836727	172.188.155.25	192.168.31.207	TCP	66	443 → 60665 [ACK] Seq=1 Ack=2 Win=303 Len=0 SLE=1 SRE=2
9	8.192265	fe80::7ebf:77ff:fe3...	2409:40c2:122a:eb5d:...	ICMPv6	86	Neighbor Solicitation for 2409:40c2:122a:eb5d:e50d:a2d1:dddc:7d25 fr
10	8.192390	2409:40c2:122a:eb5d:...	fe80::7ebf:77ff:fe3...	ICMPv6	86	Neighbor Advertisement 2409:40c2:122a:eb5d:e50d:a2d1:dddc:7d25 (sol,
11	9.002804	2409:40c2:122a:eb5d:...	2a03:2880:f288:1ca:...	TLSv1.2	144	Application Data
12	9.047756	2a03:2880:f288:1ca:...	2409:40c2:122a:eb5d:...	TCP	74	443 → 53992 [ACK] Seq=1 Ack=71 Win=1567 Len=0
13	9.318881	2a03:2880:f288:1ca:...	2409:40c2:122a:eb5d:...	TLSv1.2	146	Application Data
14	9.369757	2409:40c2:122a:eb5d:...	2a03:2880:f288:1ca:...	TCP	74	53992 → 443 [ACK] Seq=71 Ack=73 Win=253 Len=0
15	10.854664	2404:6800:4009:807:...	2409:40c2:122a:eb5d:...	UDP	1286	443 → 63030 Len=1224
16	10.854883	2404:6800:4009:807:...	2409:40c2:122a:eb5d:...	UDP	97	443 → 63030 Len=35
17	10.868317	2409:40c2:122a:eb5d:...	2404:6800:4009:807:...	UDP	99	63030 → 443 Len=37
18	17.003796	192.168.31.207	192.168.31.1	DNS	75	Standard query 0x091b HTTPS ssl.gstatic.com
19	17.005325	192.168.31.207	192.168.31.1	DNS	75	Standard query 0x1bbf AAAA ssl.gstatic.com
20	17.006588	192.168.31.207	192.168.31.1	DNS	75	Standard query 0xfa9c A ssl.gstatic.com

▶ Frame 1: 150 bytes on wire (1200 bits), 150 bytes captured (1200 bits) on interface \Device\NPF_{2B094CE...} 0000 33 33 00 00 00 01 7c bf 77 31 8c
 ▶ Ethernet II, Src: Speedtech_31:8c:cb (7c:bf:77:31:8c:cb), Dst: IPv6mcast_01 (33:33:00:00:00:01) 0010 14 3b 00 60 3a ff fe 80 00 00 00
 ▶ Internet Protocol Version 6, Src: fe80::7ebf:77ff:fe31:8ccb, Dst: ff02::1 0020 77 ff fe 31 8c cb ff 02 00 00 00
 ▶ Internet Control Message Protocol v6 0030 00 00 00 00 00 01 85 00 64 74 40
 0040 00 00 00 00 00 00 01 01 7c bf 77
 0050 00 00 00 00 05 dc 03 04 40 c0 00
 0060 54 5e 00 00 00 00 24 09 40 c2 12
 0070 00 00 00 00 00 19 03 00 00 00
 0080 40 c2 12 2a eb 5d 7e bf 77 ff fe
 0090 00 00 00 00 75 30

Observations:

Sr. No.	Source IP	Destination IP	Protocol	Info / Observation
1	192.168.31.207	172.188.155.25	TCP	Normal HTTP request

2	192.168.31.207	8.8.8.8	ICMP	Ping request detected
3	192.168.31.207	192.168.31.1	DNS	DNS query for domain resolution

Result:

Successfully configured and used **Wireshark** to capture and analyze network packets. Observed different protocols and identified potential vulnerabilities in unencrypted network traffic.

Conclusion:

Wireshark is a powerful tool for monitoring, analyzing, and identifying vulnerabilities within network traffic. It provides detailed insights into network communication, which is essential for cybersecurity and performance troubleshooting.