Assignment no : 8

Title: Configure and Demonstrate use of vulnerability assessment tool like Wireshark

---

- **Introduction to Wireshark**

Wireshark is a software tool used to monitor the network traffic through a network interface. It is the most widely used network monitoring tool today. Wireshark is loved equally by system administrators, network engineers, network enthusiasts, network security professionals and black hat hackers.

The extent of its popularity is such, that experience with Wireshark is considered as a valuable/essential trait in a computer networking-related professional.

There are many reasons why Wireshark is so popular:

1. It has a great GUI as well as a conventional CLI(T Shark).
2. It offers network monitoring on almost all types of network standards (ethernet, wlan, Bluetooth etc)
3. It is open-source with a large community of backers and developers.
4. All the necessary components for monitoring, analysing and documenting the network traffic are present. It is free to use.

The basic features of Wireshark are:

Packet Monitor: This segment visually shows the packets flowing inside the network. There are color codes for each type of packet. The packets are shown with the following information:
1. Source address
2. Destination address
3. Packet type
4. Hex dump of the packet
5. Contents of the packet in text
6. Source port(if applicable)
7. Destination port(if applicable)

Wireshark installation:

Windows:

- You can do a proper installation or run Wireshark as a portable app on your windows system. To download the installation executable or the portable app go to Wireshark Downloads
- Run the executable and follow on-screen instructions to complete the installation.

What is Sniffing?

Sniffing is a process of monitoring and capturing all data packets passing through given network. Sniffers are used by network/system administrator to monitor and troubleshoot network traffic. Attackers use sniffers to capture data packets containing sensitive information such as password, account information etc. Sniffers can be hardware or software installed in the system. By placing a packet sniffer on a network in promiscuous mode, a malicious intruder can capture and analyze all of the network traffic.

There are two types:

Active Sniffing:

Sniffing in the switch is active sniffing. A switch is a point to point network device. The switch regulates the flow of data between its ports by actively monitoring the MAC address on each port, which helps it pass data only to its intended target. In order to capture the traffic between target sniffers has to actively inject traffic into the LAN to enable sniffing of the traffic. This can be done in various ways.

Passive Sniffing:

This is the process of sniffing through the hub. Any traffic that is passing through the non-switched or unbridged network segment can be seen by all machines on that segment. Sniffers operate at the data link layer of the network. Any data sent across the LAN is actually sent to each and every machine connected to the LAN. This is called passive since sniffers placed by the attackers passively wait for the data to be sent and capture them.

Steps:

1. Install Wireshark

1. Download Wireshark from https://www.wireshark.org/download.html.
2. Install the software along with **Npcap** (Windows) or the required capture libraries for Linux/Mac.
3. Launch Wireshark from your Start Menu or Applications folder.

2. Select a Network Interface

1. On the Wireshark home screen, you will see a list of network interfaces (e.g., Wi-Fi, Ethernet).
2. Choose the interface you want to monitor (usually the one with active traffic).
3. Double-click the interface name to start capturing packets.

3. Start Capturing Packets

1. Once the capture starts, Wireshark will display packets in real-time.
2. Each packet is shown with:
    - **Time**
    - **Source and Destination IP addresses**
    - **Protocol**

- **Packet Info**

4. Apply Display Filters

To focus on specific types of traffic, enter filters in the **Display Filter** bar:

- http → Show only HTTP traffic.
- tcp.port == 443 → Show HTTPS traffic.
- ip.addr == 192.168.1.1 → Show traffic for a specific IP.

5. Analyze Packet Details

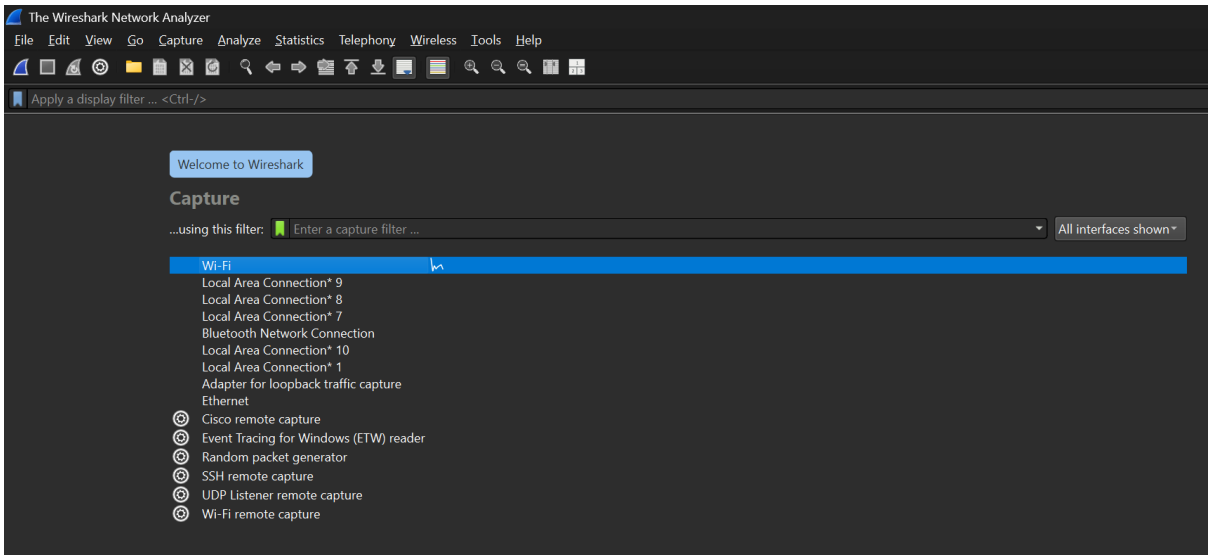
1. Click on any packet to see details in the **Packet Details** pane.
2. Expand sections to view protocol layers (Ethernet, IP, TCP/UDP, HTTP, etc.).
3. Identify important fields like HTTP requests, DNS queries, and TCP handshake steps.

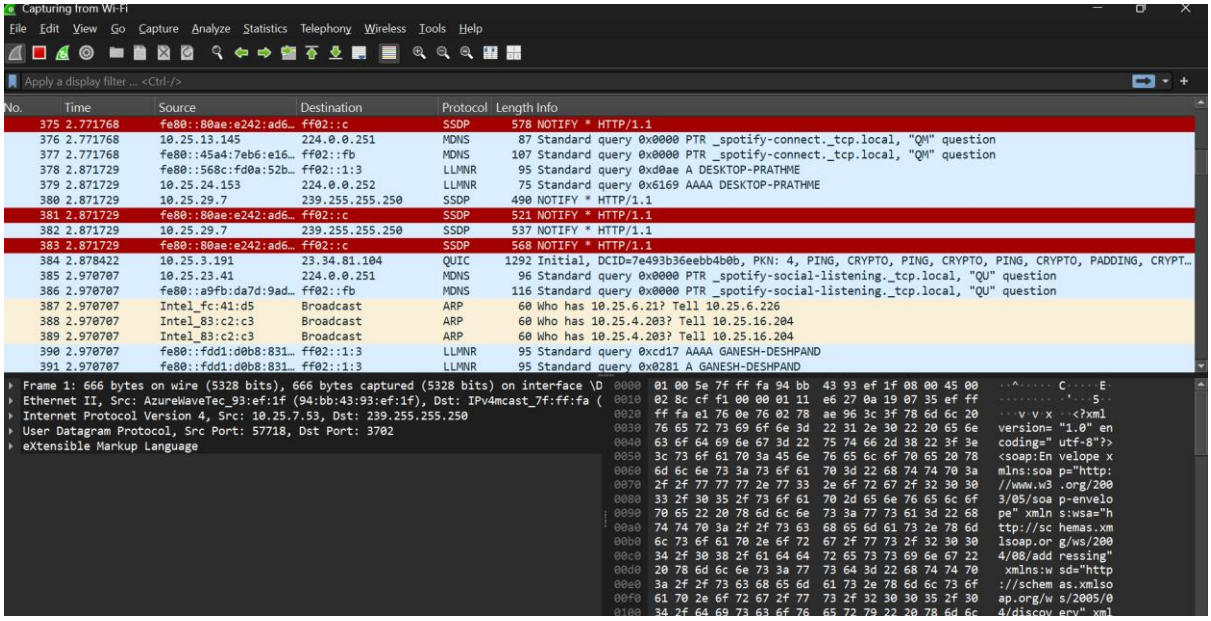
6. Stop and Save Capture

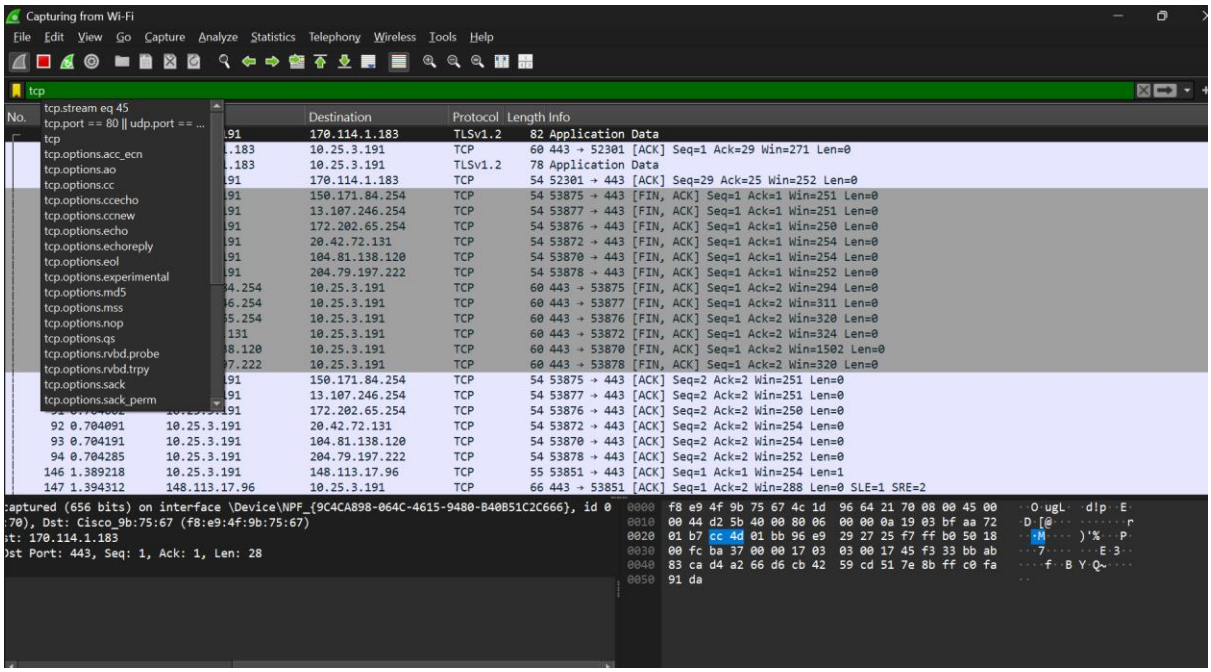
1. Click the **red square Stop button** in the toolbar to stop capturing.
2. Go to **File → Save As**.
3. Save the file with a .pcapng extension for later analysis.

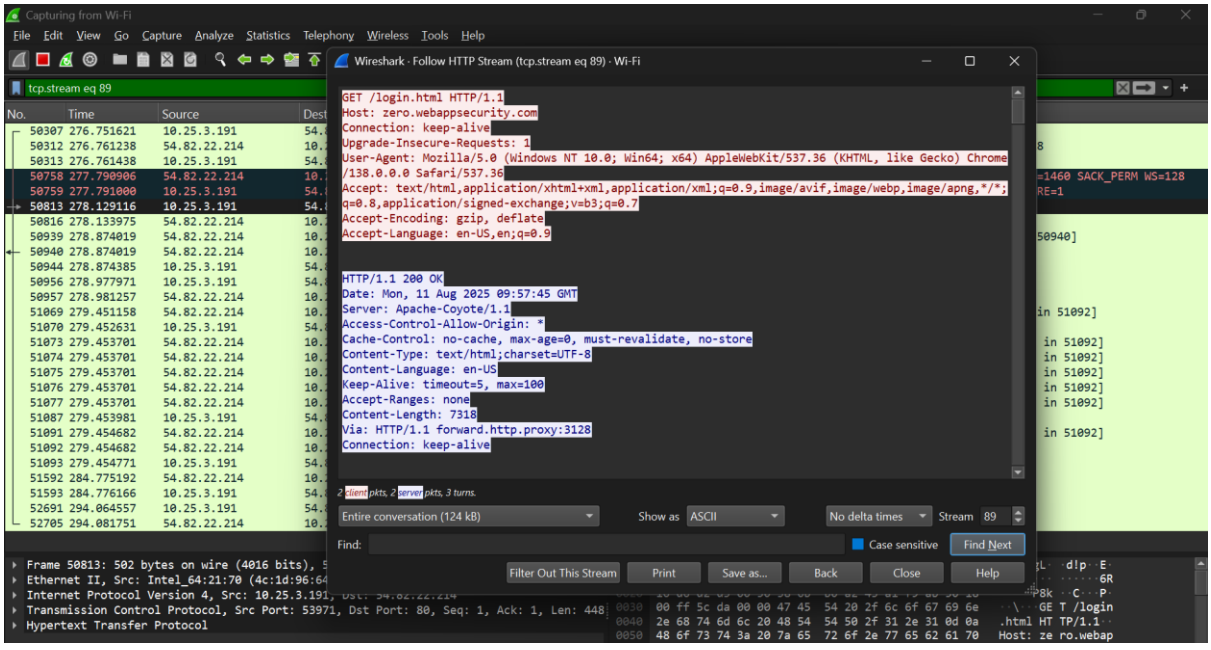
7. Optional: Export Specific Packets

- Use **File → Export Specified Packets** to save only selected traffic.
- Useful for sharing examples without giving away sensitive information.




Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
375 2.771768 fe80::80ae:e242:ad6… ff02::c SSDP 578 NOTIFY * HTTP/1.1
376 2.771768 10.25.13.145 224.0.0.251 MDNS 87 Standard query 0x0000 PTR _spotify-connect._tcp.local, "QM" question
377 2.771768 fe80::45a4:7eb6:e16… ff02::fb MDNS 107 Standard query 0x0000 PTR _spotify-connect._tcp.local, "QM" question
378 2.871729 fe80::568c:fd0a:52b… ff02::1:3 LLMNR 95 Standard query 0xd0ae A DESKTOP-PRATHME
379 2.871729 10.25.24.153 224.0.0.252 LLMNR 75 Standard query 0x6169 AAAA DESKTOP-PRATHME
380 2.871729 10.25.29.7 239.255.255.250 SSDP 490 NOTIFY * HTTP/1.1
381 2.871729 fe80::80ae:e242:ad6… ff02::c SSDP 521 NOTIFY * HTTP/1.1
382 2.871729 10.25.29.7 239.255.255.250 SSDP 537 NOTIFY * HTTP/1.1
383 2.871729 fe80::80ae:e242:ad6… ff02::c SSDP 568 NOTIFY * HTTP/1.1
384 2.878422 10.25.3.191 23.34.81.104 QUIC 1292 Initial, DCID=7e493b36eebb4b0b, PKN: 4, PING, CRYPTO, PING, CRYPTO, PING, CRYPTO, PADDING, CRYPT…
385 2.970707 10.25.23.41 224.0.0.251 MDNS 96 Standard query 0x0000 PTR _spotify-social-listening._tcp.local, "QU" question
386 2.970707 fe80::a9fb:da7d:9ad… ff02::fb MDNS 116 Standard query 0x0000 PTR _spotify-social-listening._tcp.local, "QU" question
387 2.970707 Intel_fc:41:d5 Broadcast ARP 60 Who has 10.25.6.21? Tell 10.25.6.226
388 2.970707 Intel_83:c2:c3 Broadcast ARP 60 Who has 10.25.4.203? Tell 10.25.16.204
389 2.970707 Intel_83:c2:c3 Broadcast ARP 60 Who has 10.25.4.203? Tell 10.25.16.204
390 2.970707 fe80::fdd1:d0b8:831… ff02::1:3 LLMNR 95 Standard query 0xcd17 AAAA GANESH-DESHPAND
391 2.970707 fe80::fdd1:d0b8:831… ff02::1:3 LLMNR 95 Standard query 0x0281 A GANESH-DESHPAND
Frame 1: 666 bytes on wire (5328 bits), 666 bytes captured (5328 bits) on interface \D
Ethernet II, Src: AzureWaveTec_93:ef:1f (94:bb:43:93:ef:1f), Dst: IPv4mcast_7f:ff:fa (
Internet Protocol Version 4, Src: 10.25.7.53, Dst: 239.255.255.250
User Datagram Protocol, Src Port: 57718, Dst Port: 3702
eXtensible Markup Language


Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp
tcp.stream eq 45
tcp.port == 80 || udp.port == ...
tcp
tcp.options.acc_ecn
tcp.options.ao
tcp.options.cc
tcp.options.ccecho
tcp.options.ccnew
tcp.options.echo
tcp.options.echoreply
tcp.options.eol
tcp.options.experimental
tcp.options.md5
tcp.options.mss
tcp.options.nop
tcp.options.qs
tcp.options.rvbd.probe
tcp.options.rvbd.trpy
tcp.options.sack
tcp.options.sack_perm
No. Destination Protocol Length Info
170.114.1.183 TLSv1.2 82 Application Data
10.25.3.191 TCP 60 443 → 52301 [ACK] Seq=1 Ack=29 Win=271 Len=0
10.25.3.191 TLSv1.2 78 Application Data
170.114.1.183 TCP 54 52301 → 443 [ACK] Seq=29 Ack=25 Win=252 Len=0
150.171.84.254 TCP 54 53875 → 443 [FIN, ACK] Seq=1 Ack=1 Win=251 Len=0
13.107.246.254 TCP 54 53877 → 443 [FIN, ACK] Seq=1 Ack=1 Win=251 Len=0
172.202.65.254 TCP 54 53876 → 443 [FIN, ACK] Seq=1 Ack=1 Win=250 Len=0
20.42.72.131 TCP 54 53872 → 443 [FIN, ACK] Seq=1 Ack=1 Win=254 Len=0
104.81.138.120 TCP 54 53870 → 443 [FIN, ACK] Seq=1 Ack=1 Win=254 Len=0
204.79.197.222 TCP 54 53878 → 443 [FIN, ACK] Seq=1 Ack=1 Win=252 Len=0
10.25.3.191 TCP 60 443 → 53875 [FIN, ACK] Seq=1 Ack=2 Win=294 Len=0
10.25.3.191 TCP 60 443 → 53877 [FIN, ACK] Seq=1 Ack=2 Win=311 Len=0
10.25.3.191 TCP 60 443 → 53876 [FIN, ACK] Seq=1 Ack=2 Win=320 Len=0
10.25.3.191 TCP 60 443 → 53872 [FIN, ACK] Seq=1 Ack=2 Win=324 Len=0
10.25.3.191 TCP 60 443 → 53870 [FIN, ACK] Seq=1 Ack=2 Win=1502 Len=0
10.25.3.191 TCP 60 443 → 53878 [FIN, ACK] Seq=1 Ack=2 Win=320 Len=0
150.171.84.254 TCP 54 53875 → 443 [ACK] Seq=2 Ack=2 Win=251 Len=0
13.107.246.254 TCP 54 53877 → 443 [ACK] Seq=2 Ack=2 Win=251 Len=0
172.202.65.254 TCP 54 53876 → 443 [ACK] Seq=2 Ack=2 Win=250 Len=0
92 0.704091 10.25.3.191 20.42.72.131 TCP 54 53872 → 443 [ACK] Seq=2 Ack=2 Win=254 Len=0
93 0.704191 10.25.3.191 104.81.138.120 TCP 54 53870 → 443 [ACK] Seq=2 Ack=2 Win=254 Len=0
94 0.704285 10.25.3.191 204.79.197.222 TCP 54 53878 → 443 [ACK] Seq=2 Ack=2 Win=252 Len=0
146 1.389218 10.25.3.191 148.113.17.96 TCP 55 53851 → 443 [ACK] Seq=1 Ack=1 Win=254 Len=1
147 1.394312 148.113.17.96 10.25.3.191 TCP 66 443 → 53851 [ACK] Seq=1 Ack=2 Win=288 Len=0 SLE=1 SRE=2
captured (656 bits) on interface \Device\NPF_{9C4CA898-064C-4615-9480-B40B51C2C666}, id 0
70), Dst: Cisco_9b:75:67 (f8:e9:4f:9b:75:67)
st: 170.114.1.183
Dst Port: 443, Seq: 1, Ack: 1, Len: 28


Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.stream eq 89
No. Time Source
50307 276.751621 10.25.3.191
50312 276.761238 54.82.22.214
50313 276.761438 10.25.3.191
50758 277.790906 54.82.22.214
50759 277.791000 10.25.3.191
50813 278.129116 10.25.3.191
50816 278.133975 54.82.22.214
50939 278.874019 54.82.22.214
50940 278.874019 54.82.22.214
50944 278.874385 10.25.3.191
50956 278.977971 10.25.3.191
50957 278.981257 54.82.22.214
51069 279.451158 54.82.22.214
51070 279.452631 10.25.3.191
51073 279.453701 54.82.22.214
51074 279.453701 54.82.22.214
51075 279.453701 54.82.22.214
51076 279.453701 54.82.22.214
51077 279.453701 54.82.22.214
51087 279.453981 10.25.3.191
51091 279.454682 54.82.22.214
51092 279.454682 54.82.22.214
51093 279.454771 10.25.3.191
51592 284.775192 54.82.22.214
51593 284.776166 10.25.3.191
52691 294.064557 10.25.3.191
52705 294.081751 54.82.22.214
Wireshark · Follow HTTP Stream (tcp.stream eq 89) · Wi-Fi
GET /login.html HTTP/1.1
Host: zero.webappsecurity.com
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
HTTP/1.1 200 OK
Date: Mon, 11 Aug 2025 09:57:45 GMT
Server: Apache-Coyote/1.1
Access-Control-Allow-Origin: *
Cache-Control: no-cache, max-age=0, must-revalidate, no-store
Content-Type: text/html;charset=UTF-8
Content-Language: en-US
Keep-Alive: timeout=5, max=100
Accept-Ranges: none
Content-Length: 7318
Via: HTTP/1.1 forward.http.proxy:3128
Connection: keep-alive
2 client pkts, 2 server pkts, 3 turns.
Entire conversation (124 kB)
Show as ASCII
No delta times
Stream 89
Find:
Case sensitive
Find Next
Filter Out This Stream
Print
Save as...
Back
Close
Help
Frame 50813: 502 bytes on wire (4016 bits),
Ethernet II, Src: Intel_64:21:70 (4c:1d:96:64
Internet Protocol Version 4, Src: 10.25.3.191,
Transmission Control Protocol, Src Port: 53971, Dst Port: 80, Seq: 1, Ack: 1, Len: 448
Hypertext Transfer Protocol