Wireshark Packet Analysis Report

1.

Executive Summary

This report documents the practical usage of **Wireshark** for analyzing network traffic on a controlled lab setup using **Kali Linux running on UTM**. The goal was to observe and understand real-time packet flow, identify unencrypted credentials, and learn how various protocols behave in a live environment. The target was **DVWA (Damn Vulnerable Web App)** to simulate ethical hacking and basic traffic analysis scenarios.

2.

Objective

The primary objective of this experiment was:

- To understand how packets travel over a network
- To analyze the packet structure (SYN, SYN-ACK, ACK, ARP, etc.)
- To identify unencrypted protocols such as HTTP and extract potential credentials
- To get hands-on practice with Wireshark filtering and traffic inspection for both client and server-side traffic

3.

Tool Used – Wireshark

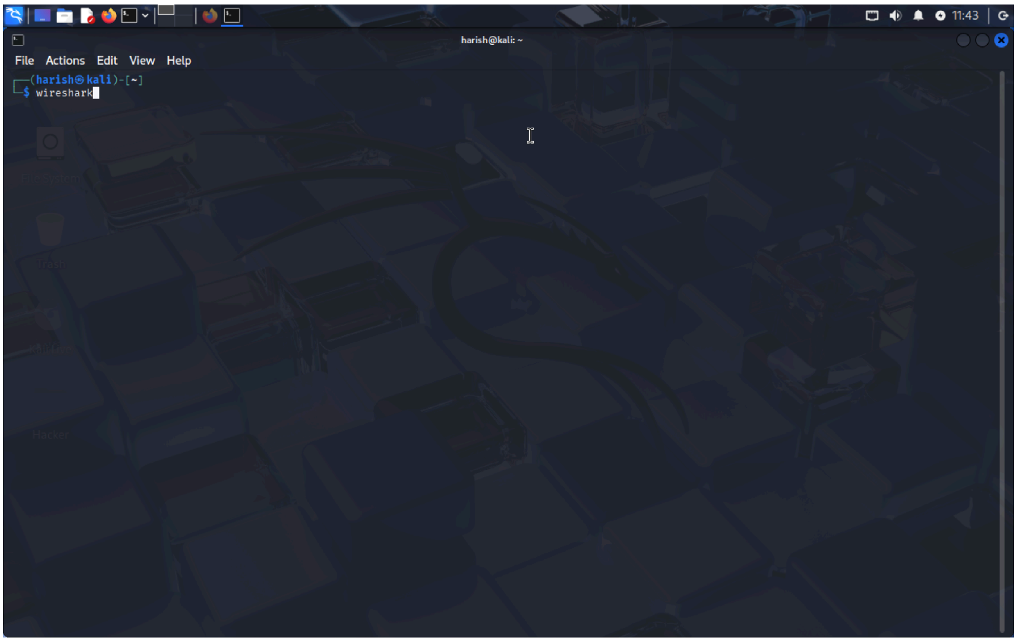
Wireshark is an open-source network protocol analyzer used for **packet sniffing and traffic analysis**. It's widely used in the cybersecurity world to:

- Analyze protocols in detail
- Troubleshoot network issues
- Detect anomalies and intrusions
- Capture sensitive data traveling unencrypted

4.

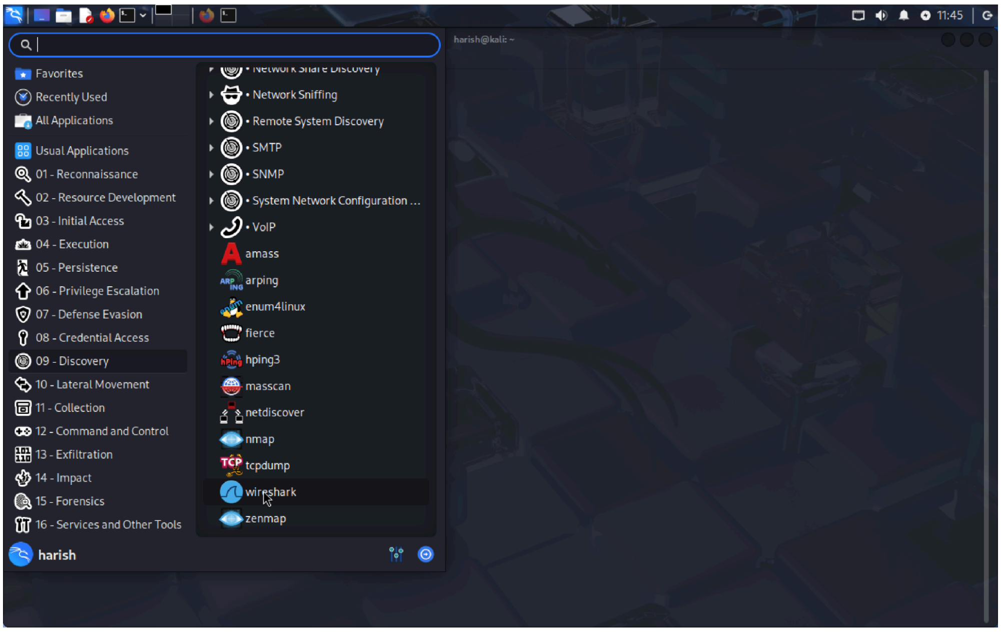
Setup and Environment

- **OS**: Kali Linux (Apple Silicon build)
- **Virtualization**: UTM (running on M1 Mac)
- **Network Interface**: Loopback (lo) – since the traffic was being generated and analyzed on the same system (localhost)
- **Target**: DVWA (localhost web app)
- **Tool**: Wireshark (pre-installed on Kali)



```
┌──(harish㉿kali)-[~]
└─$ wireshark
```

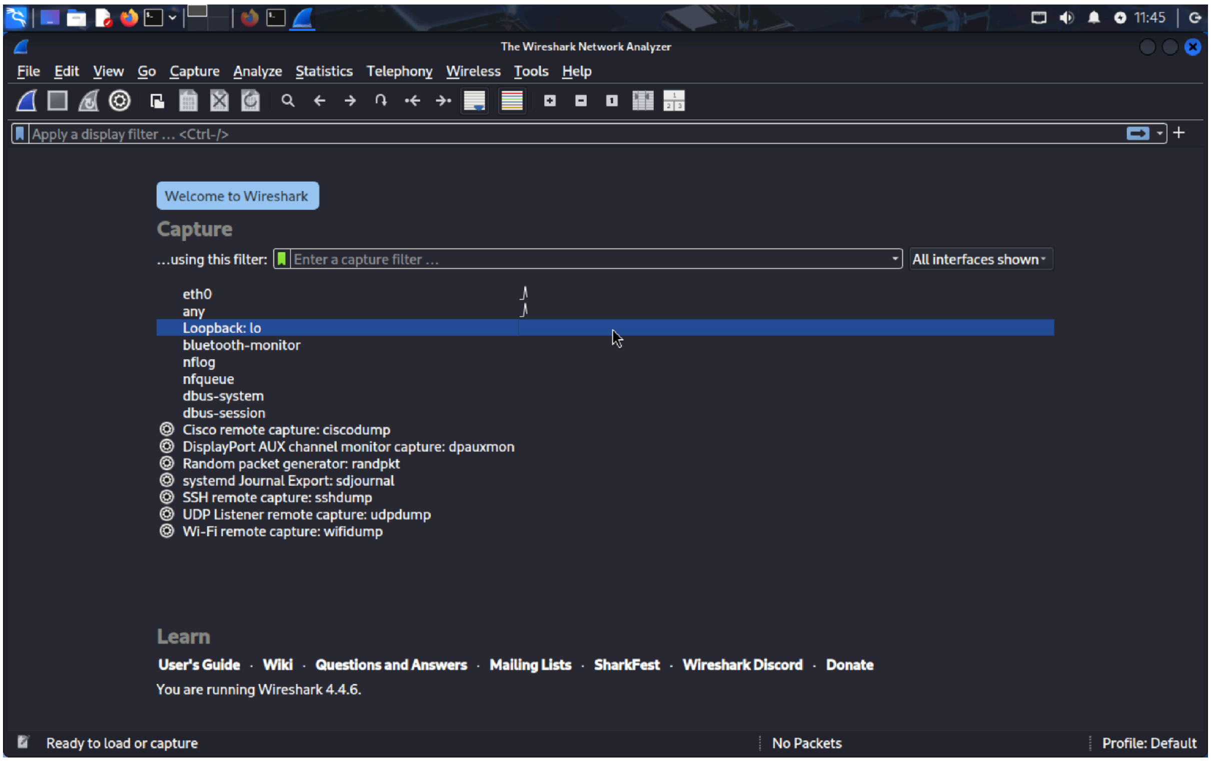
Fig: Launch WireShark CLI



5.

Steps Performed

1. Launched Wireshark and selected the **loopback interface**.
2. Enabled packet **capturing** before interacting with DVWA.
3. Opened **DVWA in the browser** and attempted login using test credentials.
4. Captured packets in real-time and applied **filters** for:
    - ○ http
    - ○ tcp.port == 80
    - ○ ip.addr == 127.0.0.1


11:45
The Wireshark Network Analyzer
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Welcome to Wireshark
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord · Donate

Apply a display filter … <Ctrl-/>

Capture

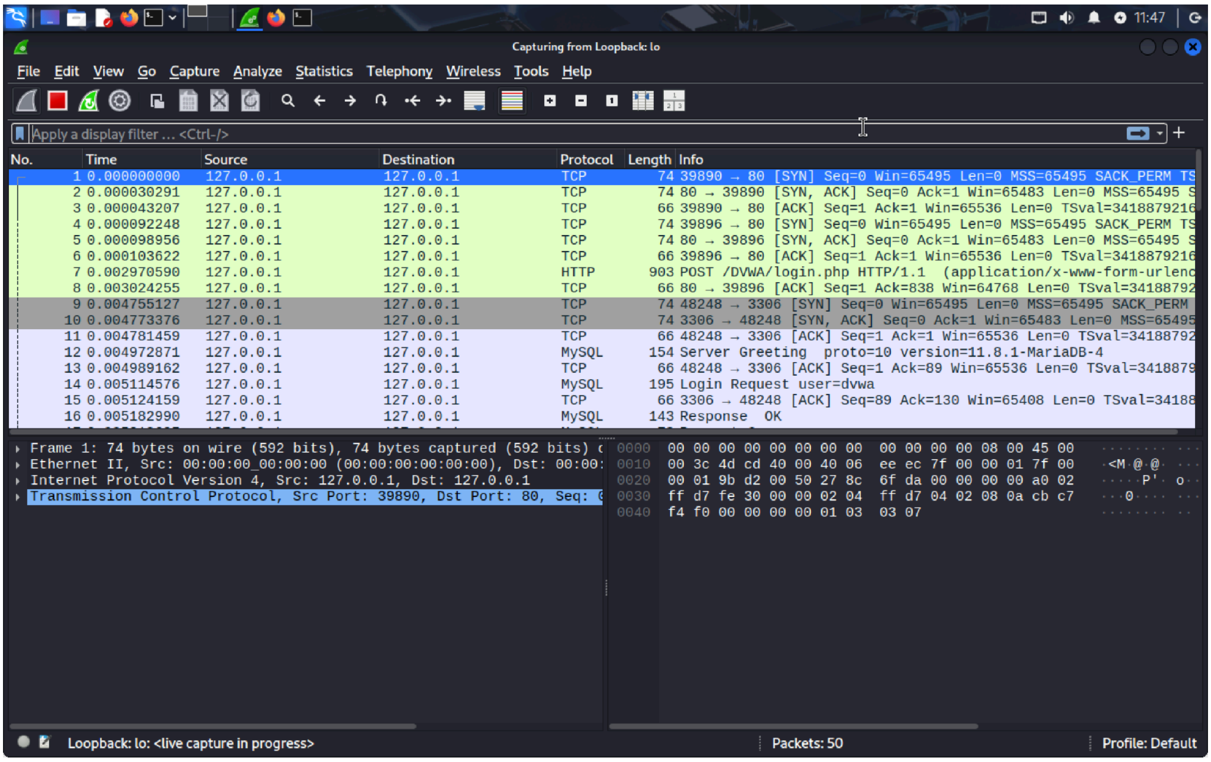
…using this filter: Enter a capture filter … All interfaces shown

eth0
any
Loopback: lo
bluetooth-monitor
nflog
nfqueue
dbus-system
dbus-session
Cisco remote capture: ciscodump
DisplayPort AUX channel monitor capture: dpauxmon
Random packet generator: randpkt
systemd Journal Export: sdjournal
SSH remote capture: sshdump
UDP Listener remote capture: udpdump
Wi-Fi remote capture: wifidump

You are running Wireshark 4.4.6.

Ready to load or capture | No Packets | Profile: Default

Fig: Choose loopback



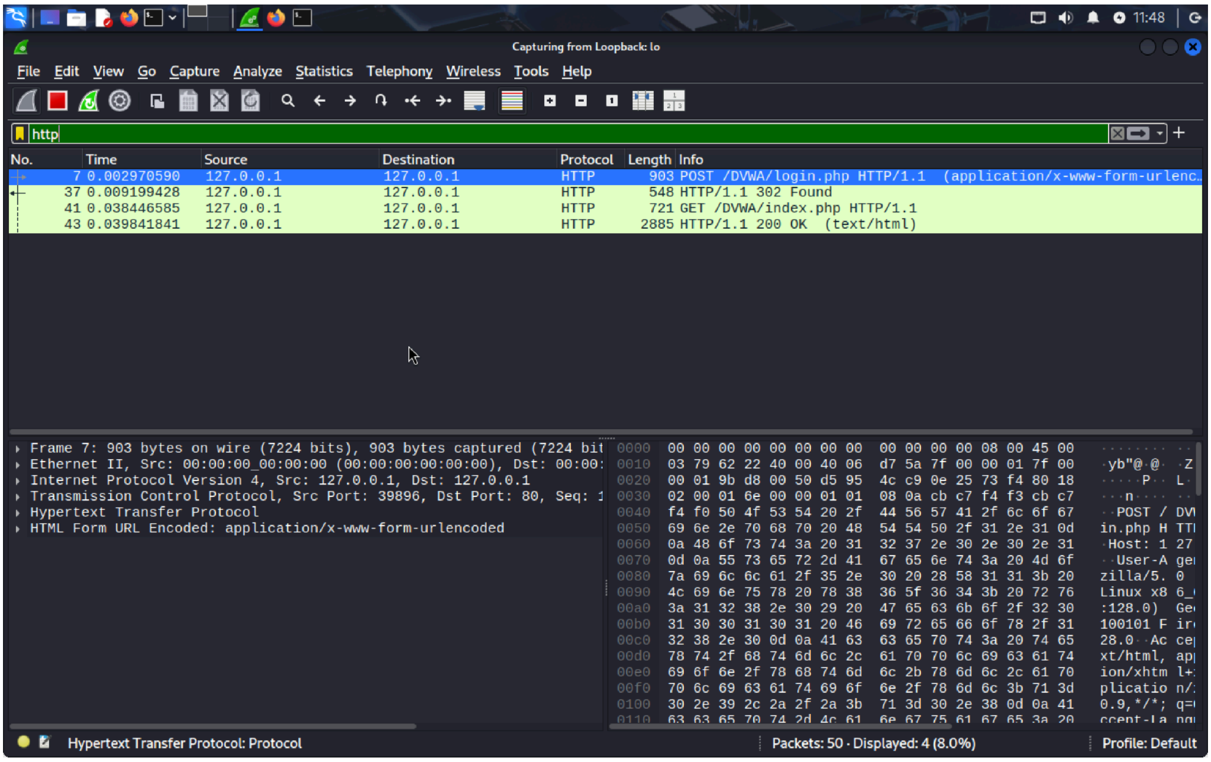
```
No.   Time         Source      Destination  Protocol Length Info
    1 0.000000000  127.0.0.1   127.0.0.1    TCP        74 39890 → 80 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TS
    2 0.000030291  127.0.0.1   127.0.0.1    TCP        74 80 → 39890 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 S
    3 0.000043207  127.0.0.1   127.0.0.1    TCP        66 39890 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3418879216
    4 0.000092248  127.0.0.1   127.0.0.1    TCP        74 39896 → 80 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TS
    5 0.000098956  127.0.0.1   127.0.0.1    TCP        74 80 → 39896 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 S
    6 0.000103622  127.0.0.1   127.0.0.1    TCP        66 39896 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3418879216
    7 0.002970590  127.0.0.1   127.0.0.1    HTTP      903 POST /DVWA/login.php HTTP/1.1  (application/x-www-form-urlenc
    8 0.003024255  127.0.0.1   127.0.0.1    TCP        66 80 → 39896 [ACK] Seq=1 Ack=838 Win=64768 Len=0 TSval=34188792
    9 0.004755127  127.0.0.1   127.0.0.1    TCP        74 48248 → 3306 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM
   10 0.004773376  127.0.0.1   127.0.0.1    TCP        74 3306 → 48248 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495
   11 0.004781459  127.0.0.1   127.0.0.1    TCP        66 48248 → 3306 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=34188792
   12 0.004972871  127.0.0.1   127.0.0.1    MySQL     154 Server Greeting  proto=10 version=11.8.1-MariaDB-4
   13 0.004989162  127.0.0.1   127.0.0.1    TCP        66 48248 → 3306 [ACK] Seq=1 Ack=89 Win=65536 Len=0 TSval=3418879
   14 0.005114576  127.0.0.1   127.0.0.1    MySQL     195 Login Request user=dvwa
   15 0.005124159  127.0.0.1   127.0.0.1    TCP        66 3306 → 48248 [ACK] Seq=89 Ack=130 Win=65408 Len=0 TSval=34188
   16 0.005182990  127.0.0.1   127.0.0.1    MySQL     143 Response  OK
```

```
▸ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) c
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 39890, Dst Port: 80, Seq: 0
```

```
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ········ ···
0010  00 3c 4d cd 40 00 40 06  ee ec 7f 00 00 01 7f 00   ·<M·@·@· ···
0020  00 01 9b d2 00 50 27 8c  6f da 00 00 00 00 a0 02   ·····P'· o··
0030  ff d7 fe 30 00 00 02 04  ff d7 04 02 08 0a cb c7   ···0···· ···
0040  f4 f0 00 00 00 00 01 03  03 07                     ········ ··
```

Loopback: lo: <live capture in progress>   Packets: 50   Profile: Default

Fig: Capturing Traffic



```
http
```

```
No.   Time         Source      Destination  Protocol Length Info
    7 0.002970590  127.0.0.1   127.0.0.1    HTTP      903 POST /DVWA/login.php HTTP/1.1  (application/x-www-form-urlenc…
   37 0.009199428  127.0.0.1   127.0.0.1    HTTP      548 HTTP/1.1 302 Found
   41 0.038446585  127.0.0.1   127.0.0.1    HTTP      721 GET /DVWA/index.php HTTP/1.1
   43 0.039841841  127.0.0.1   127.0.0.1    HTTP     2885 HTTP/1.1 200 OK  (text/html)
```

```
▸ Frame 7: 903 bytes on wire (7224 bits), 903 bytes captured (7224 bit
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 39896, Dst Port: 80, Seq: 1
▸ Hypertext Transfer Protocol
▸ HTML Form URL Encoded: application/x-www-form-urlencoded
```

```
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ········ ··
0010  03 79 62 22 40 00 40 06  d7 5a 7f 00 00 01 7f 00   ·yb"@·@· ·Z
0020  00 01 9b d8 00 50 d5 95  4c c9 0e 25 73 f4 80 18   ·····P·· L·
0030  02 00 01 6e 00 00 01 01  08 0a cb c7 f4 f3 cb c7   ···n···· ··
0040  f4 f0 50 4f 53 54 20 2f  44 56 57 41 2f 6c 6f 67   ··POST / DV
0050  69 6e 2e 70 68 70 20 48  54 54 50 2f 31 2e 31 0d   in.php H TT
0060  0a 48 6f 73 74 3a 20 31  32 37 2e 30 2e 30 2e 31   ·Host: 1 27
0070  0d 0a 55 73 65 72 2d 41  67 65 6e 74 3a 20 4d 6f   ··User-A ge
0080  7a 69 6c 6c 61 2f 35 2e  30 20 28 58 31 31 3b 20   zilla/5. 0
0090  4c 69 6e 75 78 20 78 38  36 5f 36 34 3b 20 72 76   Linux x8 6_
00a0  3a 31 32 38 2e 30 29 20  47 65 63 6b 6f 2f 32 30   :128.0)  Ge
00b0  31 30 30 31 30 31 20 46  69 72 65 66 6f 78 2f 31   100101 F ir
00c0  32 38 2e 30 0d 0a 41 63  63 65 70 74 3a 20 74 65   28.0··Ac ce
00d0  78 74 2f 68 74 6d 6c 2c  61 70 70 6c 69 63 61 74   xt/html, ap
00e0  69 6f 6e 2f 78 68 74 6d  6c 2b 78 6d 6c 2c 61 70   ion/xhtm l+
00f0  70 6c 69 63 61 74 69 6f  6e 2f 78 6d 6c 3b 71 3d   plicatio n/
0100  30 2e 39 2c 2a 2f 2a 3b  71 3d 30 2e 38 0d 0a 41   0.9,*/*; q=
0110  63 63 65 70 74 2d 4c 61  6e 67 75 61 67 65 3a 20   ccept-La ng
```

Hypertext Transfer Protocol: Protocol   Packets: 50 · Displayed: 4 (8.0%)   Profile: Default

Fig: Filtering HTTP Protocols

5. Observed:
    - **HTTP requests** transmitting login data in **plain text**
    - Packets containing **username and password**
    - TCP handshake (SYN, SYN-ACK, ACK)
    - **ARP** broadcast (who-has / is-at)
    - Traffic between client and web app on port **80**
    - Random **client-side port** used (ephemeral ports like 35234)

6.

Packet Types and What They Do

- **SYN / SYN-ACK / ACK**: Part of the TCP 3-way handshake — establishes a reliable connection between client and server.
- **ARP**: Used to resolve IP addresses into MAC addresses.
- **HTTP Packets**: Contain raw request and response data. Since HTTP is unencrypted, you can see login parameters like user=admin&pass=admin.
- **Client Port**: Temporary port assigned by the OS to maintain session with the server's fixed port (like 80 for HTTP).

7.

Filter Usage

Two-way filtering was used:

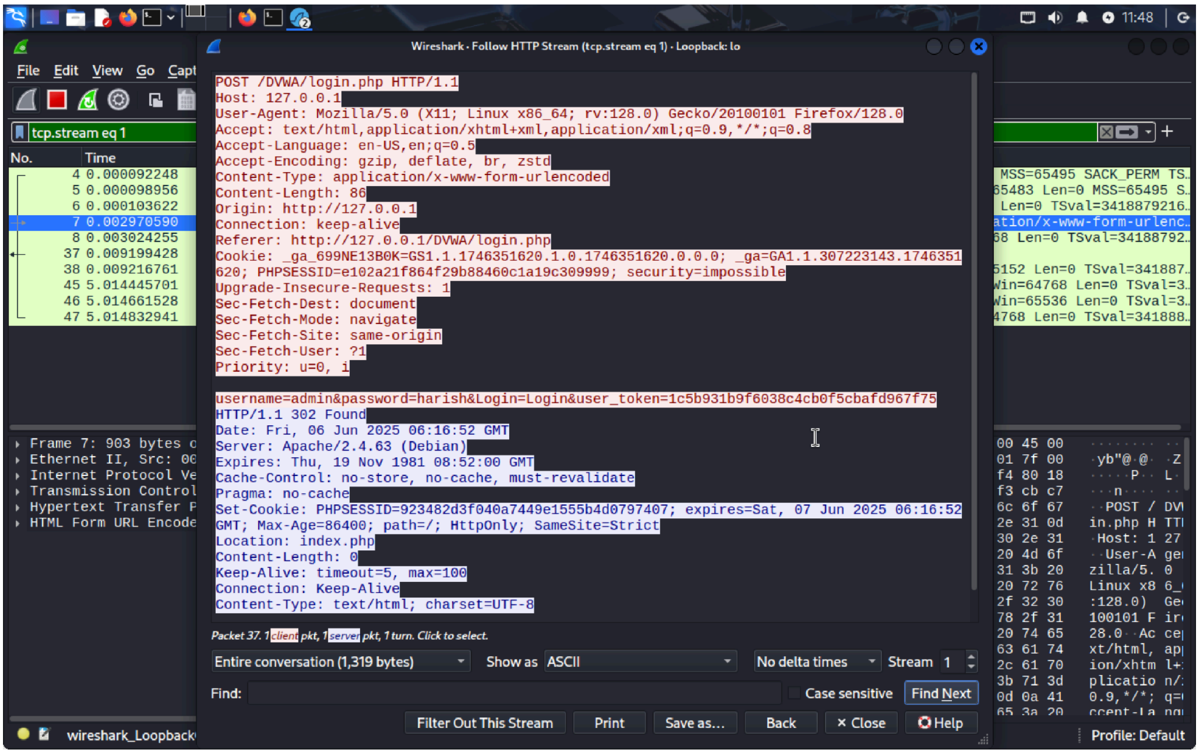
- **Host/Address-based**: ip.addr == 127.0.0.1
- **Protocol-based**: http, tcp.port == 80

  This helped in isolating traffic between the browser and the DVWA app.

8.

Findings

- HTTP login requests transmit **plaintext credentials** — clearly visible in the packet payload.



Wireshark · Follow HTTP Stream (tcp.stream eq 1) · Loopback: lo

```
POST /DVWA/login.php HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br, zstd
Content-Type: application/x-www-form-urlencoded
Content-Length: 86
Origin: http://127.0.0.1
Connection: keep-alive
Referer: http://127.0.0.1/DVWA/login.php
Cookie: _ga_699NE13B0K=GS1.1.1746351620.1.0.1746351620.0.0.0; _ga=GA1.1.307223143.1746351
620; PHPSESSID=e102a21f864f29b88460c1a19c309999; security=impossible
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i

username=admin&password=harish&Login=Login&user_token=1c5b931b9f6038c4cb0f5cbafd967f75
HTTP/1.1 302 Found
Date: Fri, 06 Jun 2025 06:16:52 GMT
Server: Apache/2.4.63 (Debian)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=923482d3f040a7449e1555b4d0797407; expires=Sat, 07 Jun 2025 06:16:52
GMT; Max-Age=86400; path=/; HttpOnly; SameSite=Strict
Location: index.php
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

Packet 37. 1 client pkt, 1 server pkt, 1 turn. Click to select.

Entire conversation (1,319 bytes) — Show as: ASCII — No delta times — Stream: 1

Display filter: tcp.stream eq 1

Fig: Findings Credentials

- Capturing loopback traffic using Wireshark is a legit way to inspect browser-to-app comms.

- TCP behavior (handshake, ACKs) and ARP mechanics were observed.

- Wireshark helps in identifying insecure communication paths and misconfigured services.

9.

Conclusion

This lab demonstrated how **Wireshark can be used to sniff, analyze, and understand** network traffic in a local environment. By targeting a vulnerable app like DVWA, it's possible to identify serious issues like **unencrypted login credentials**, observe TCP/IP operations, and understand the client-server data flow. Such skills are vital for penetration testing, traffic auditing, and network forensics.