



Sniffing Attack using Wireshark



Objective

The goal of this experiment is to demonstrate how credentials (username and password) submitted over an unencrypted HTTP website can be captured using **Wireshark**. The test is performed legally on a publicly available vulnerable web application.



Target Website

URL: <http://testphp.vulnweb.com>

Page Used: /login.php



Tools Used

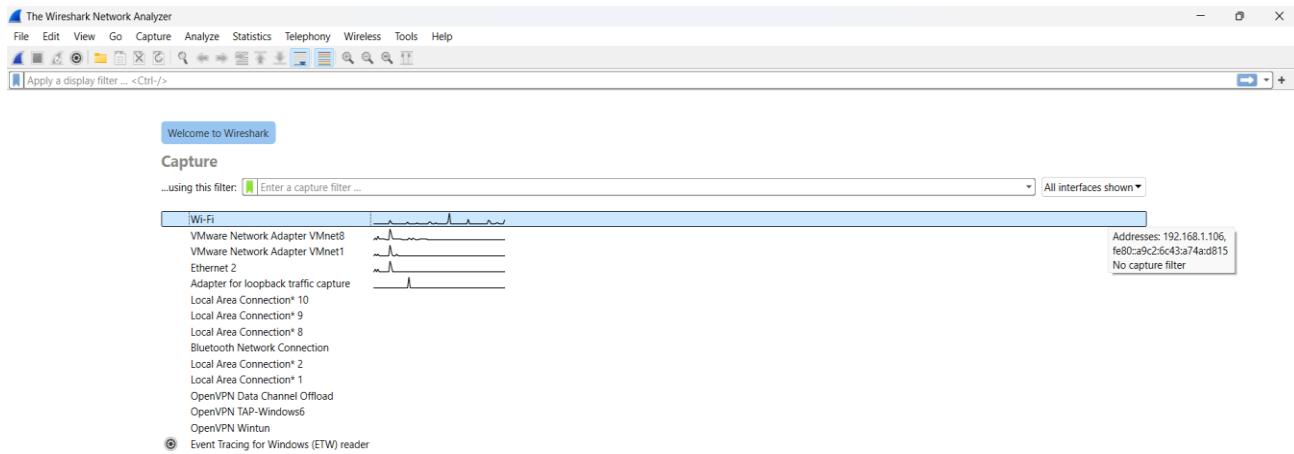
Tool	Purpose
Wireshark	Packet capture and protocol analysis
Web Browser	Used to simulate login
OS	Windows



Procedure

1. Start Wireshark

- Open Wireshark on your computer.
- Select your active network interface (e.g., Wi-Fi or Ethernet).
- Click **Start** to begin capturing packets.



Learn
[User's Guide](#) · [Wiki](#) · [Questions and Answers](#) · [Mailing Lists](#) · [SharkFest](#) · [Wireshark Discord](#) · [Donate](#)
 You are running Wireshark 4.2.6 (v4.2.6-0-g2acd1a854bab). You receive automatic updates.



Capturing from Wi-Fi

No.	Time	Source	Destination	Protocol	Length	Info
21	3.389570	23.63.111.73	192.168.1.106	TCP	54	443 → 51678 [ACK] Seq=3941 Ack=551 Win=64128 Len=0
22	3.391253	23.63.111.73	192.168.1.106	TCP	54	443 → 51678 [ACK] Seq=3941 Ack=631 Win=64128 Len=0
23	3.391253	23.63.111.73	192.168.1.106	TLSv1.3	341	Application Data
24	3.391253	23.63.111.73	192.168.1.106	TLSv1.3	341	Application Data
25	3.391253	192.168.1.106	23.63.111.73	TCP	54	51678 → 443 [ACK] Seq=745 Ack=4515 Win=64768 Len=0
26	3.391343	23.63.111.73	192.168.1.106	TLSv1.3	115	Application Data
27	3.391343	23.63.111.73	192.168.1.106	TLSv1.3	85	Application Data
28	3.391368	192.168.1.106	23.63.111.73	TCP	54	51678 → 443 [ACK] Seq=745 Ack=4607 Win=64768 Len=0
29	3.392415	192.168.1.106	23.63.111.73	TLSv1.3	85	Application Data
30	3.392570	23.63.111.73	192.168.1.106	TLSv1.3	199	Application Data
31	3.392570	23.63.111.73	192.168.1.106	TLSv1.3	85	Application Data
32	3.392613	192.168.1.106	23.63.111.73	TCP	54	51678 → 443 [ACK] Seq=776 Ack=4783 Win=64512 Len=0
33	3.393089	192.168.1.106	23.63.111.73	TCP	54	51678 → 443 [FIN, ACK] Seq=776 Ack=4783 Win=64512 Len=0
34	3.406027	23.63.111.73	192.168.1.106	TLSv1.3	78	Application Data
35	3.406027	23.63.111.73	192.168.1.106	TCP	54	443 → 51678 [FIN, ACK] Seq=4807 Ack=777 Win=64128 Len=0
36	3.406094	192.168.1.106	23.63.111.73	TCP	54	51678 → 443 [RST, ACK] Seq=777 Ack=4807 Win=0 Len=0
37	7.765990	192.168.1.106	172.64.155.209	TCP	55	51661 → 443 [ACK] Seq=1 Ack=1 Win=253 Len=1 [TCP segment of a reassembled PDU]

```
> Frame 1: 80 bytes on wire (640 bits), 80 bytes captured (640 bits) on interface '\Device\NPF_{C3BB15AF-7E0A-4D3B-AE0A-000000000000}' (ethernet II, Src: MercuryComm_a6:a9:40 (c0:25:2fa:f6:a9:40), Dst: AzureWaveTec_cc:cc:11 (50:5a:65:c1:c1:c1))
> Internet Protocol Version 4, Src: 140.82.113.25, Dst: 192.168.1.106
> Transmission Control Protocol, Src Port: 443, Dst Port: 51588, Seq: 1, Ack: 1, Len: 26
> Transport Layer Security
```

Packets: 37 - Displayed: 37 (100.0%) | Profile: Default



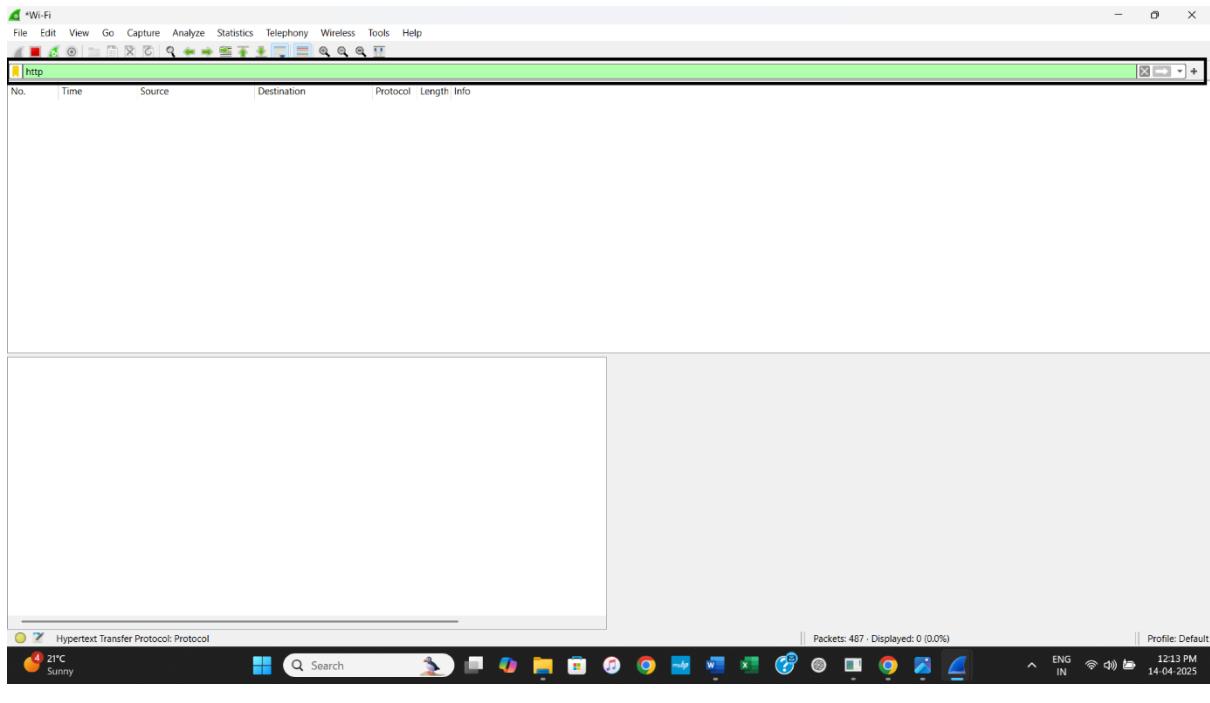
2. Apply Filter (Optional)

To only show HTTP traffic, apply the display filter:

- `http`

To focus on login attempts:

- `http.request.method == "POST"`



3. Perform Login on Target Site

- Open your web browser.
- Navigate to: <http://testphp.vulnweb.com/login.php>

A screenshot of a web browser window. The address bar shows the URL "testphp.vulnweb.com/login.php". The page content is the login interface for the Acunetix Web Vulnerability Scanner. It features a logo for "acunetix acuart" and a navigation menu with links like home, categories, artists, disclaimer, your cart, guestbook, and AJAX Demo. On the left, there's a sidebar with links for search art, browse categories, and various system links. The main form asks for login information with fields for Username and Password, and a "login" button. Below the form, it says "Signup disabled. Please use the username test and the password test." At the bottom, there's a warning message: "Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more." The system tray at the bottom of the screen shows the date and time as 14-04-2025, 12:15 PM, and various system icons.

- Enter the following credentials:
 - Username:** admin
 - Password:** admin123
- Click the **Login** button.

If you are already registered please enter your login information below:

Username : Password :

You can also [signup here](#).
Signup disabled. Please use the username **test** and the password **test**.

Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

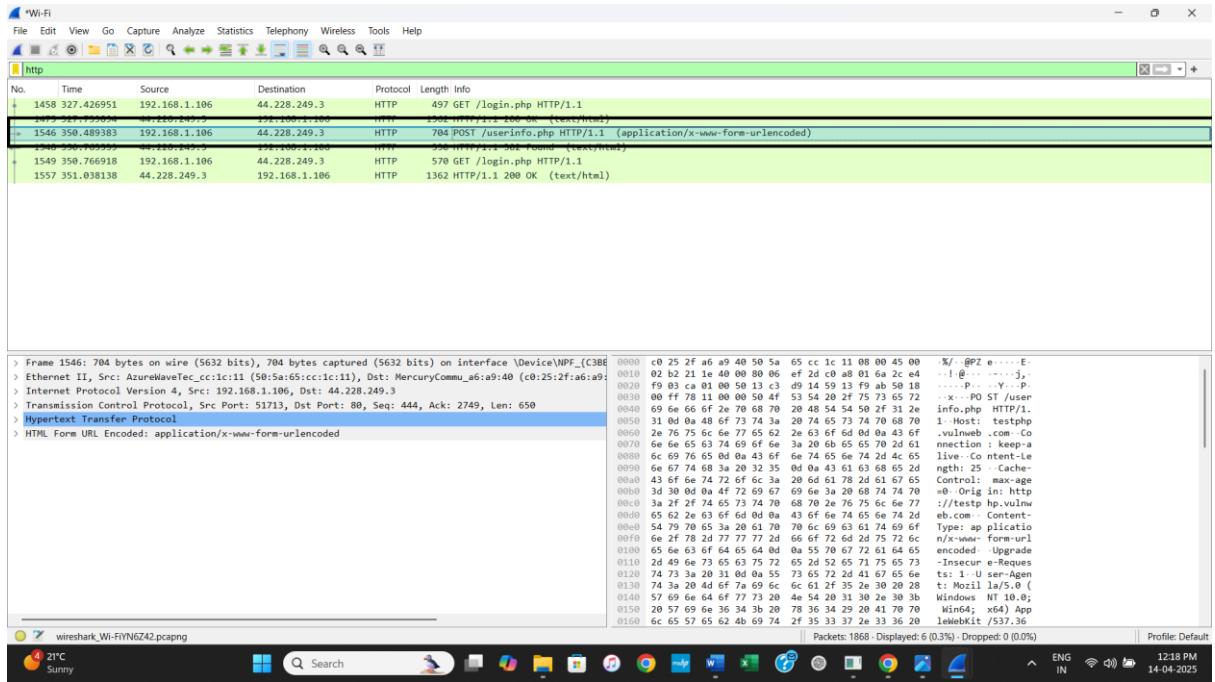
4. Stop Packet Capture

- Return to Wireshark.
- Click the **Stop** button (red square) to end the capture.

No.	Time	Source	Destination	Protocol	Length	Info
1458	327.426951	192.168.1.106	44.228.249.3	HTTP	497	GET /login.php HTTP/1.1
1475	327.739894	44.228.249.3	192.168.1.106	HTTP	1362	HTTP/1.1 200 OK (text/html)
1546	350.489383	192.168.1.106	44.228.249.3	HTTP	704	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
1548	350.763539	44.228.249.3	192.168.1.106	HTTP	330	HTTP/1.1 302 Found (text/html)
1549	350.766918	192.168.1.106	44.228.249.3	HTTP	579	GET /login.php HTTP/1.1
1557	351.038138	44.228.249.3	192.168.1.106	HTTP	1362	HTTP/1.1 200 OK (text/html)

5. Locate and Follow the POST Request

- Find the HTTP POST request to /login.php.
- Right-click on the packet → Select **Follow** → **HTTP Stream**.



6. Analyze the Captured Data

You should see data like this:

- POST /login.php HTTP/1.1**

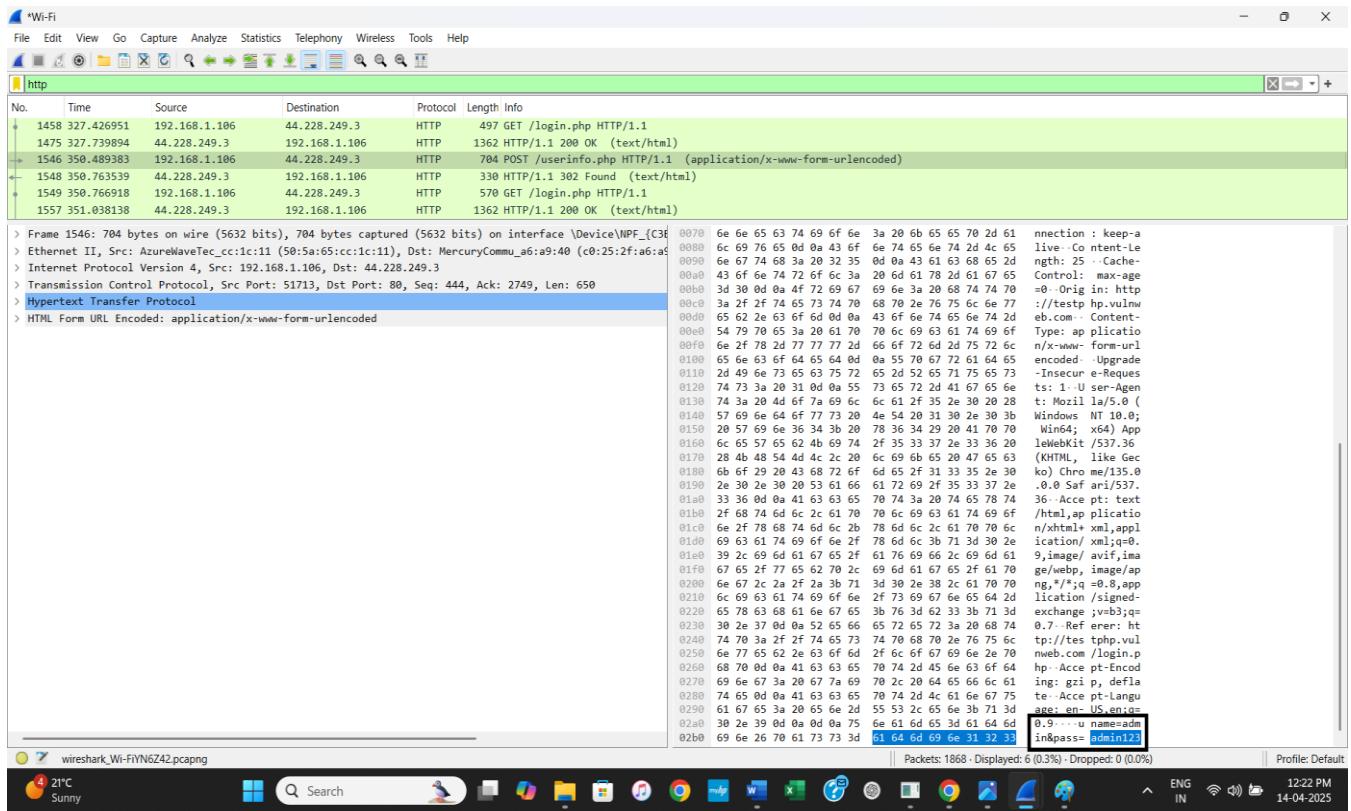
Host: testphp.vulnweb.com

User-Agent: Mozilla/5.0 ...

Content-Type: application/x-www-form-urlencoded

Content-Length: ...

- uname=admin&pass=admin123&login=login**



Captured Information

Field	Value
Username	admin
Password	admin123