

## EXPERIMENT 4

**Objective 1:** Simulate a scenario where a password is transmitted in plaintext. Use wire shark to capture and analyze the packets to demonstrate the vulnerability and the importance of encryption.

### Tool Used: Wireshark

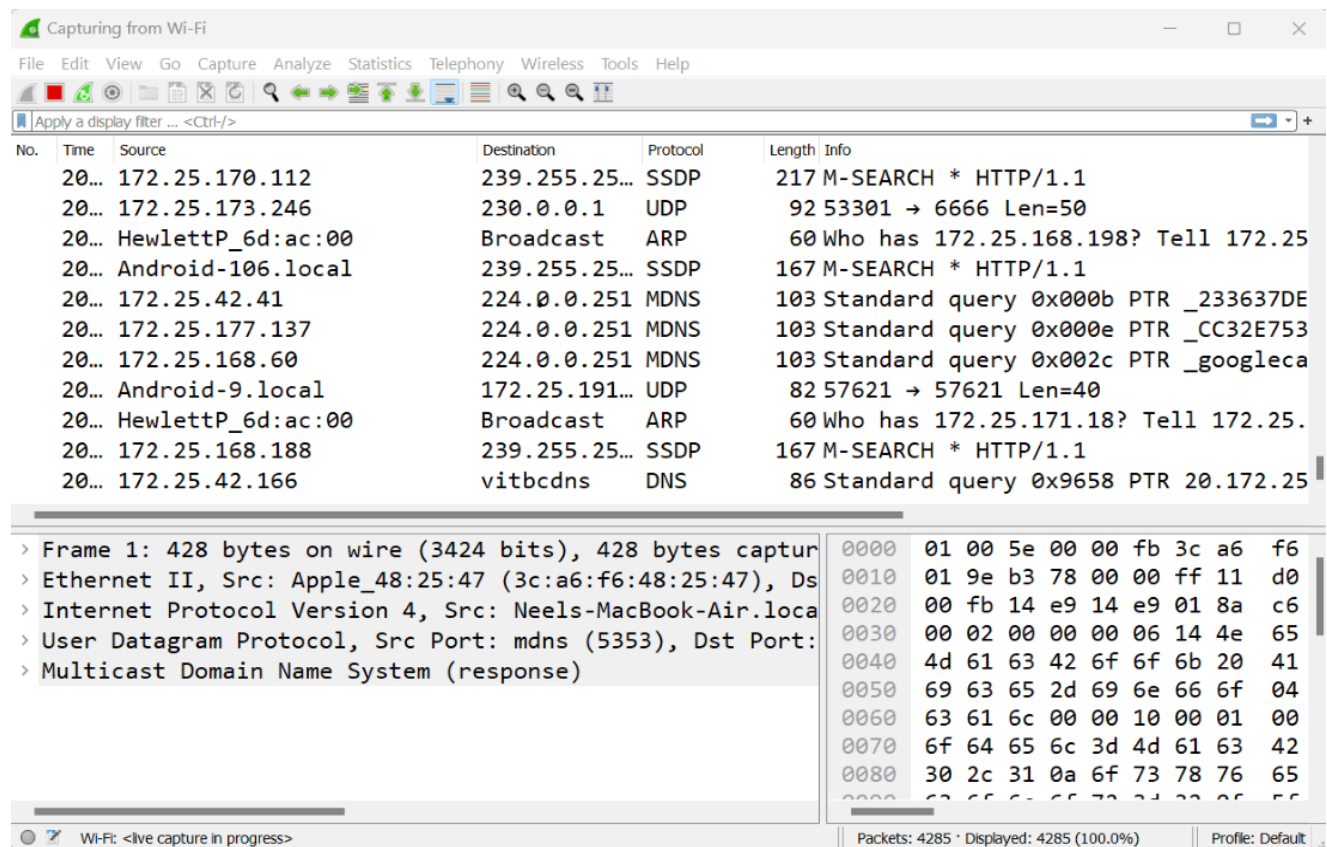
#### Password Capturing/Sniffing

Wireshark can capture not only passwords but any type of information transmitted over the network: usernames, email addresses, personal information, etc. As long as we can capture network traffic, Wireshark can sniff passing passwords.

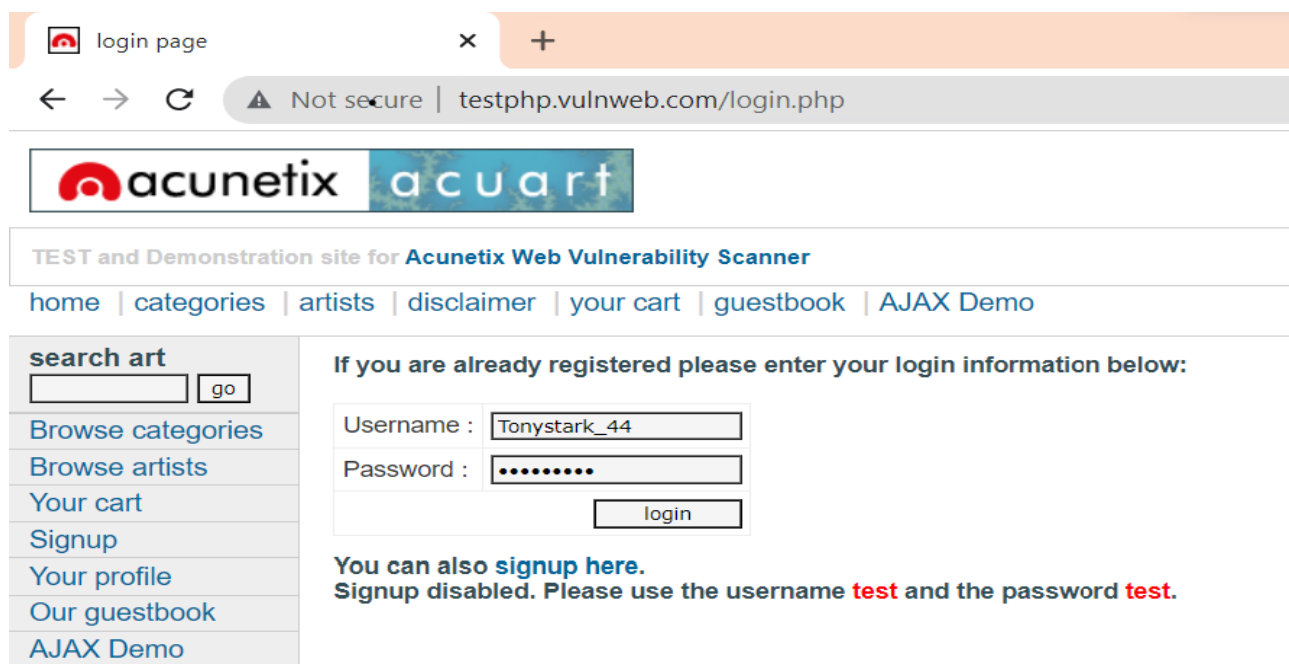
In sniffing can include passwords for various protocols such as HTTP, FTP, Telnet, etc. the captured data can be used to troubleshoot network problems, but can also be used maliciously to gain unauthorized access to sensitive information.

So, here we will see how we can capture the password using the Wireshark network capture analyzer. and see the outputs of the following steps.

**Step 1:** First of all, open your Wireshark tool in your window or in Linux virtual machine. and start capturing the network. suppose I am capturing my wireless fidelity.



**Step 2:** After starting the packet capturing we will go to the website and login the credential on that website as you can see in the image.



login page

Not secure | testphp.vulnweb.com/login.php

acunetix acuart

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

search art

go

Browse categories

Browse artists

Your cart

Signup

Your profile

Our guestbook

AJAX Demo

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

Username :

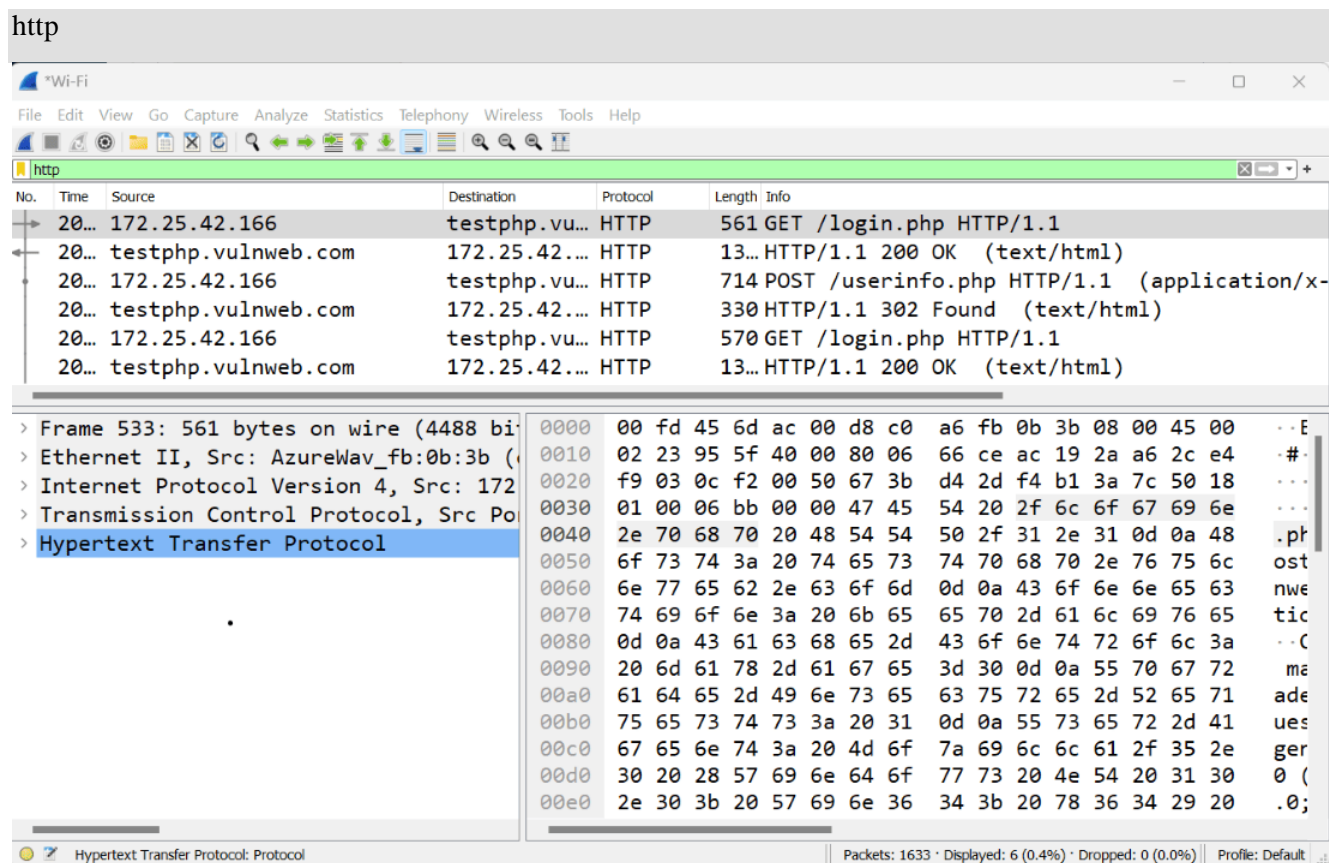
Password :

You can also [signup here](#).

Signup disabled. Please use the username **test** and the password **test**.

**Step 3:** Now after completing the login credential we will go and capture the password in Wireshark. for that we have to use some filter that helps to find the login credential through the packet capturing.

**Step 4:** Wireshark has captured some packets but we specifically looking for HTTP packets. so in the display filter bar we use some command to find all the captured HTTP packets. as you can see in the below image the green bar where we apply the filter.



http

\*Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
20...	172.25.42.166	testphp.vu...	HTTP	561	GET /login.php HTTP/1.1	
20...	testphp.vulnweb.com	172.25.42...	HTTP	13...	HTTP/1.1 200 OK (text/html)	
20...	172.25.42.166	testphp.vu...	HTTP	714	POST /userinfo.php HTTP/1.1 (application/x-...	
20...	testphp.vulnweb.com	172.25.42...	HTTP	330	HTTP/1.1 302 Found (text/html)	
20...	172.25.42.166	testphp.vu...	HTTP	570	GET /login.php HTTP/1.1	
20...	testphp.vulnweb.com	172.25.42...	HTTP	13...	HTTP/1.1 200 OK (text/html)	

> Frame 533: 561 bytes on wire (4488 bi...)

> Ethernet II, Src: AzureWav\_fb:0b:3b (...)

> Internet Protocol Version 4, Src: 172...

> Transmission Control Protocol, Src Po...

> Hypertext Transfer Protocol

0000 00 fd 45 6d ac 00 d8 c0 a6 fb 0b 3b 08 00 45 00 ..E

0010 02 23 95 5f 40 00 80 06 66 ce ac 19 2a a6 2c e4 .#.

0020 f9 03 0c f2 00 50 67 3b d4 2d f4 b1 3a 7c 50 18 ...

0030 01 00 06 bb 00 00 47 45 54 20 2f 6c 6f 67 69 6e ...

0040 2e 70 68 70 20 48 54 54 50 2f 31 2e 31 0d 0a 48 .ph

0050 6f 73 74 3a 20 74 65 73 74 70 68 70 2e 76 75 6c ost

0060 6e 77 65 62 2e 63 6f 6d 0d 0a 43 6f 6e 6e 65 63 nwe

0070 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 76 65 tic

0080 0d 0a 43 61 63 68 65 2d 43 6f 6e 74 72 6f 6c 3a ..C

0090 20 6d 61 78 2d 61 67 65 3d 30 0d 0a 55 70 67 72 ma

00a0 61 64 65 2d 49 6e 73 65 63 75 72 65 2d 52 65 71 ade

00b0 75 65 73 74 73 3a 20 31 0d 0a 55 73 65 72 2d 41 ues

00c0 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e ger

00d0 30 20 28 57 69 6e 64 6f 77 73 20 4e 54 20 31 30 0 (

00e0 2e 30 3b 20 57 69 6e 36 34 3b 20 78 36 34 29 20 .0;

Hypertext Transfer Protocol: Protocol

Packets: 1633 · Displayed: 6 (0.4%) · Dropped: 0 (0.0%) · Profile: Default

**Step 5:** So there are some HTTP packets are captured but we specifically looking for form data that the user submitted to the website. for that, we have a separate filter

As we know that there are main two methods used for submitting form data from web pages like login forms to the server. the methods are-

- GET
- POST

**Step 6:** So firstly for knowing the credential we use the first method and apply the filter for the GET methods as you can see below.

```
http.request.method == "GET"
```

The image shows a Wireshark network traffic capture. The filter bar at the top is set to `http.request.method == "GET"`. The packet list shows two HTTP GET requests to `testphp.vulnweb.com/login.php`. The first packet (No. 20) is selected, and its details are expanded. The details pane shows the Hypertext Transfer Protocol section with the following fields:

```

GET /login.php HTTP/1.1\r\n
Host: testphp.vulnweb.com\r\n
Connection: keep-alive\r\n
Cache-Control: max-age=0\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win6
Accept: text/html,application/xhtml+xml,applic
Referer: http://testphp.vulnweb.com/\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9\r\n
\r\n
[Full request URI: http://testphp.vulnweb.com/]
[HTTP request 1/3]
[Response in frame: 555]
[Next request in frame: 1268]

```

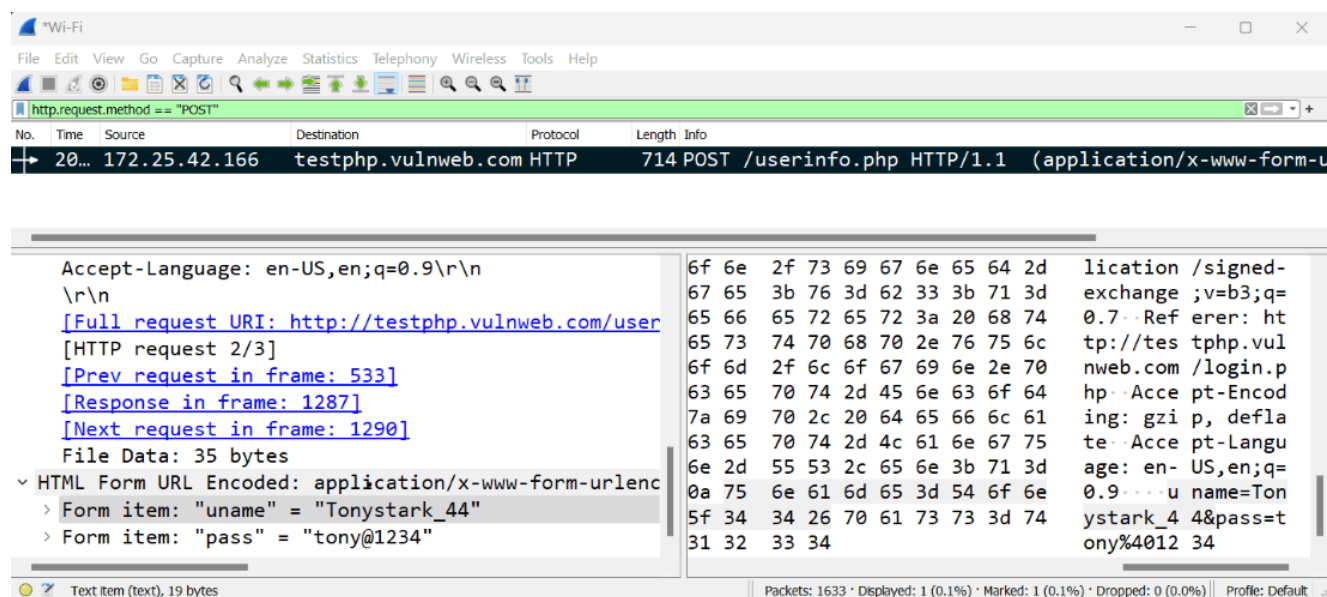
The hex data pane on the right shows the raw bytes of the selected packet, starting with `0130 33 37 2e 33 36 0d 0a 41 63 63 65 70 74 3`.

#### GET method

As you can see in the image there are two packets where the login page was requested with a GET request as well, but there is no form data submitted with a GET request.

**Step 7:** Now after checking the GET method if we didn't find the form data, then we will try the POST method for that we will apply the filter on Wireshark as you can see.

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



As you can see we have a packet with form data click on the packet with user info and the application URL encoded. and click on the down-

HTML form URL Encoded where the login credential is found. login credential as it is the same that we filed on the website in step 2.

Form item: "uname" = "Tonystark\_44"  
Form item: "pass" = "tony@1234"