TITLE: PACKET SNIFFING ANALYSIS WITH WIRESHARK

Subtitle: “Capturing and Analyzing Plaintext Credentials in Unencrypted Traffic"

Done By:

Name :Karabo Kekana,

LinkedIn: karabo(kay)kekana

Date: 2025/04/09

INTRODUCTION

HTTP transmits data in plaintext, making it vulnerable to interception. This report demonstrates how attackers can exploit this weakness using Wireshark.

TOOLS USED INCLUDE :

- **Wireshark**
- **Test Website**: http://testphp.vulnweb.com/login.php
- **Operating System**: Windows 11

METHODOLOGY

1. Setup Network

Connected to my University Wi-Fi and initiated Wireshark capture on the Wi-Fi interface

2.Traffic Capture

- **Steps**:

  1.Applied filter: http && (ip.src ==194.253.249.114 || ip.dst == 194.253.249.114)

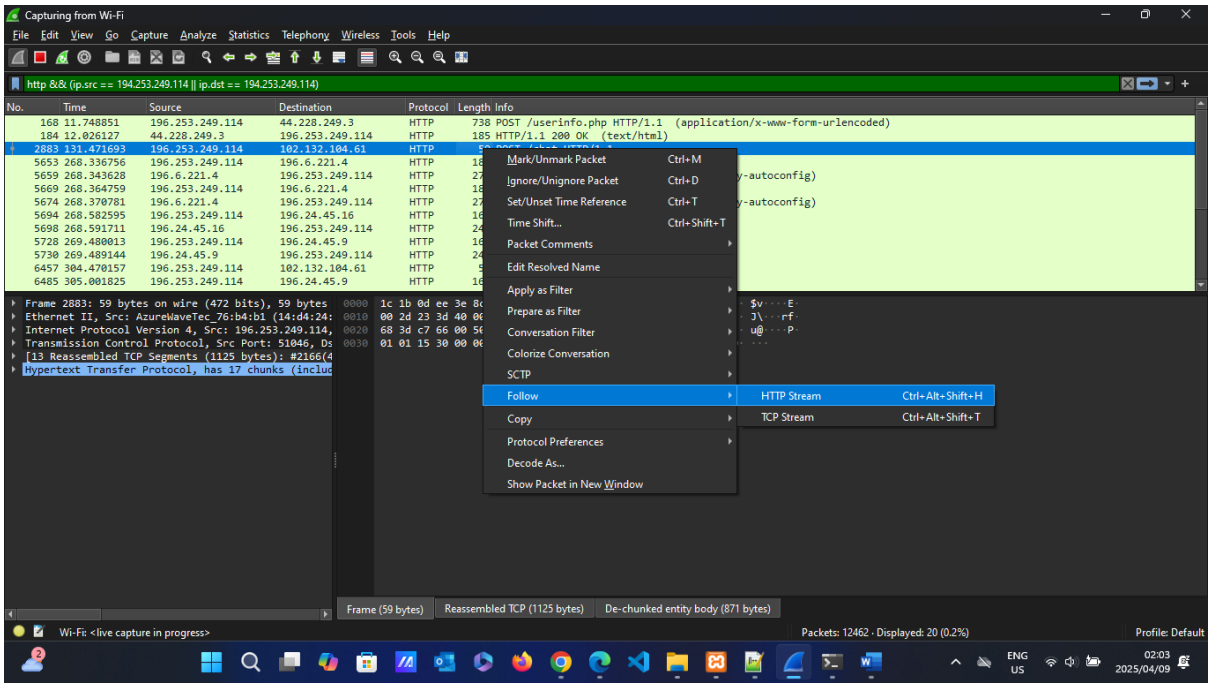

figure1: Capturing and analysis

2.Visited http://testphp.vulnweb.com.

3.Entered test credentials (test:test).

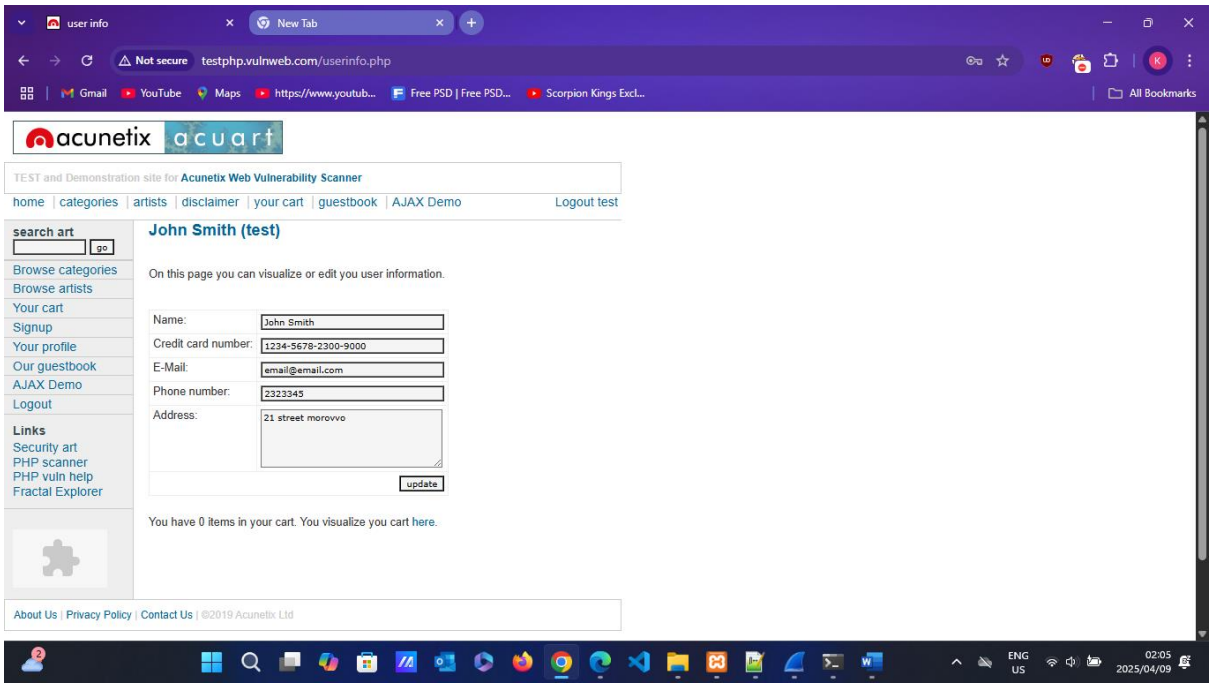

figure 2: (Website visited)

3 Analysis

- then I analysed the captured packet by right click on it then followed TCP stream to extract raw HTTP POST data. (see figure 1 also)
- Results include :

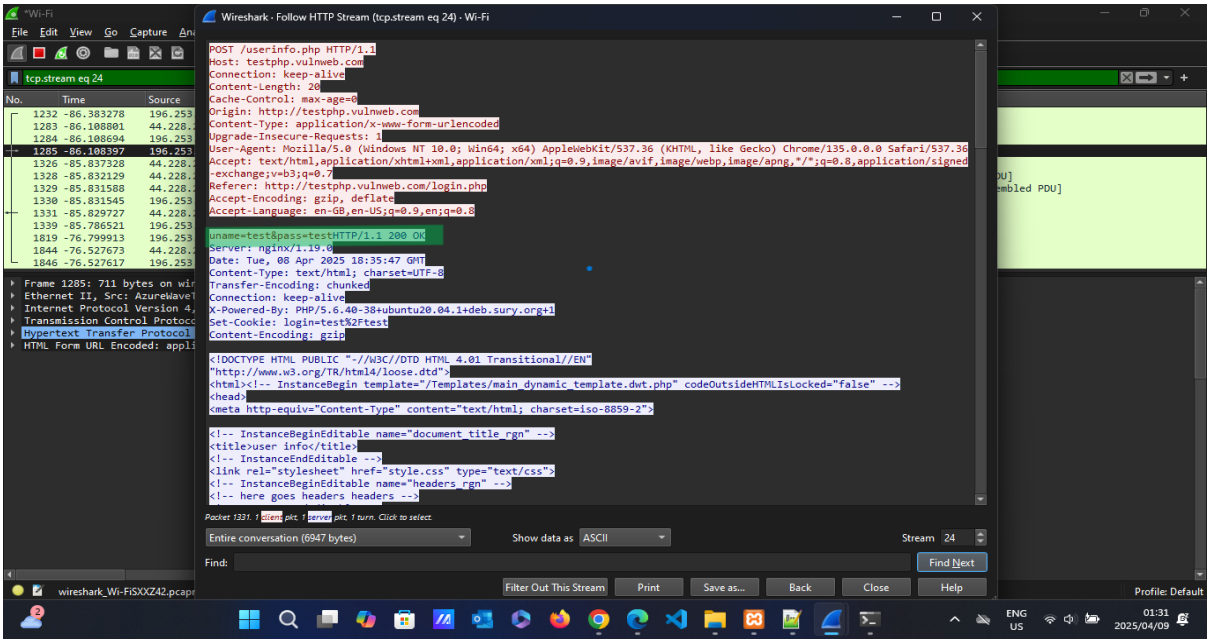

Figure 3(Results)

FINDINGS

1. HTTP Credentials Exposure

- *"Wireshark captured the username and password in plaintext (Fig 3)."*

```
POST /login.php HTTP/1.1
Host: testphp.vulnweb.com
uname=test&psw=test
```

2. DNS Queries

- *"DNS leaks revealed visited domains (e.g., testphp.vulnweb.com)."*

SECURITY RISKS

- **Key Vulnerabilities**:
  - Man-in-the-Middle (MITM) attacks.
  - Session hijacking via stolen cookies.
  - Credential theft on public Wi-Fi.

RECOMMENDATIONS

- **For Users**:
  - Always check for HTTPS (🔒 padlock icon).
  - Use a VPN on untrusted networks.
- **For Developers**:
  - Enforce HTTPS (HSTS, TLS 1.3).

REFERENCES

- Wireshark Official Docs: https://www.wireshark.org/docs/
- OWASP HTTP Risks: https://owasp.org