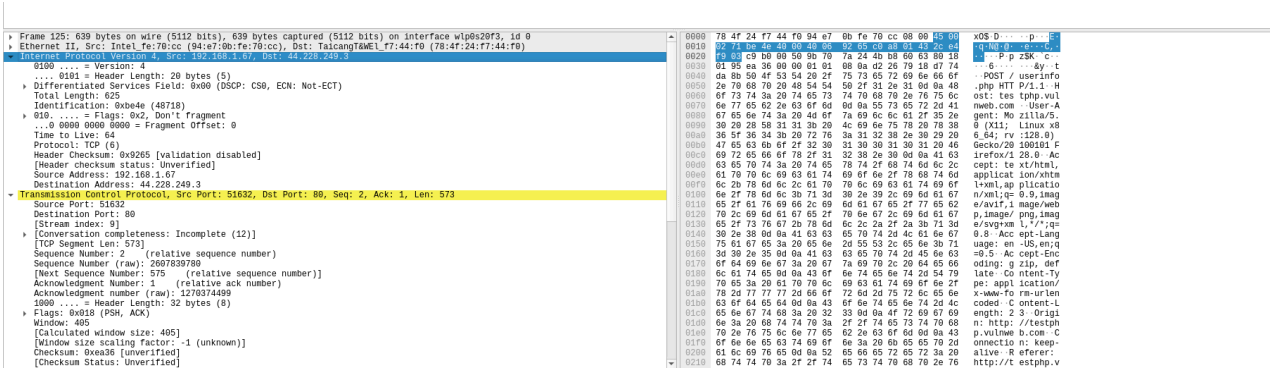
Login

1) POST



Internet Protocol Version 4
Src: 192.168.1.67
Dst: 44.228.249.3

Transmission Control Protocol
Src Port: 51632
Dst Port: 80

Hypertext Transfer Protocol

POST /userinfo.php HTTP/1.1\r\n
Host: testphp.vulnweb.com\r\n
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\r\n

Accept:text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Content-Type: application/x-www-form-urlencoded\r\n
Content-Length: 23\r\n
Origin: http://testphp.vulnweb.com\r\n
Connection: keep-alive\r\n
Referer: http://testphp.vulnweb.com/login.php\r\n
Upgrade-Insecure-Requests: 1\r\n
Priority: u=0, i\r\n
\r\n
[Full request URI: http://testphp.vulnweb.com/userinfo.php]
[HTTP request 1/2]
[Response in frame: 141]

[Next request in frame: 142]
File Data: 23 bytes

HTML Form URL Encoded: application/x-www-form-urlencoded

Form item: "uname" = "anish"
Form item: "pass" = "subedi"

Note: since no encryption method were used data send were plain text as we can see username and password clearly.

2) GET

Hypertext Transfer Protocol

GET /login.php HTTP/1.1\r\n
Host: testphp.vulnweb.com\r\n
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Referer: http://testphp.vulnweb.com/login.php\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
Priority: u=0, i\r\n
\r\n
[Full request URI: http://testphp.vulnweb.com/login.php]
[HTTP request 2/2]
[Prev request in frame: 125]
[Response in frame: 146]

Security Analysis

1. **Transport Layer Security (TLS) / Secure Sockets Layer (SSL)**
    - The communication uses HTTP on port 80, which is not encrypted. HTTPS, which operates over port 443, would be needed to secure the communication with TLS/SSL.
2. **Sensitive Data**
    - The username and password (uname=anish&pass=subedi) are sent in plaintext. This can be easily intercepted by an attacker using a network sniffer.
3. **Connection**
    - The Connection: keep-alive header indicates that the connection should be kept open for multiple requests, which is common for efficiency but does not impact security directly.

Security Measures to Consider

- **Use HTTPS:** Encrypt the communication using HTTPS to protect the data in transit.
- **Form Data Encryption:** Encrypt sensitive form data before sending it.

Summary

The analyzed packet uses plain HTTP, which does not provide any encryption, making it vulnerable to interception and eavesdropping. To secure the communication, switching to HTTPS and implementing additional security measures is essential.