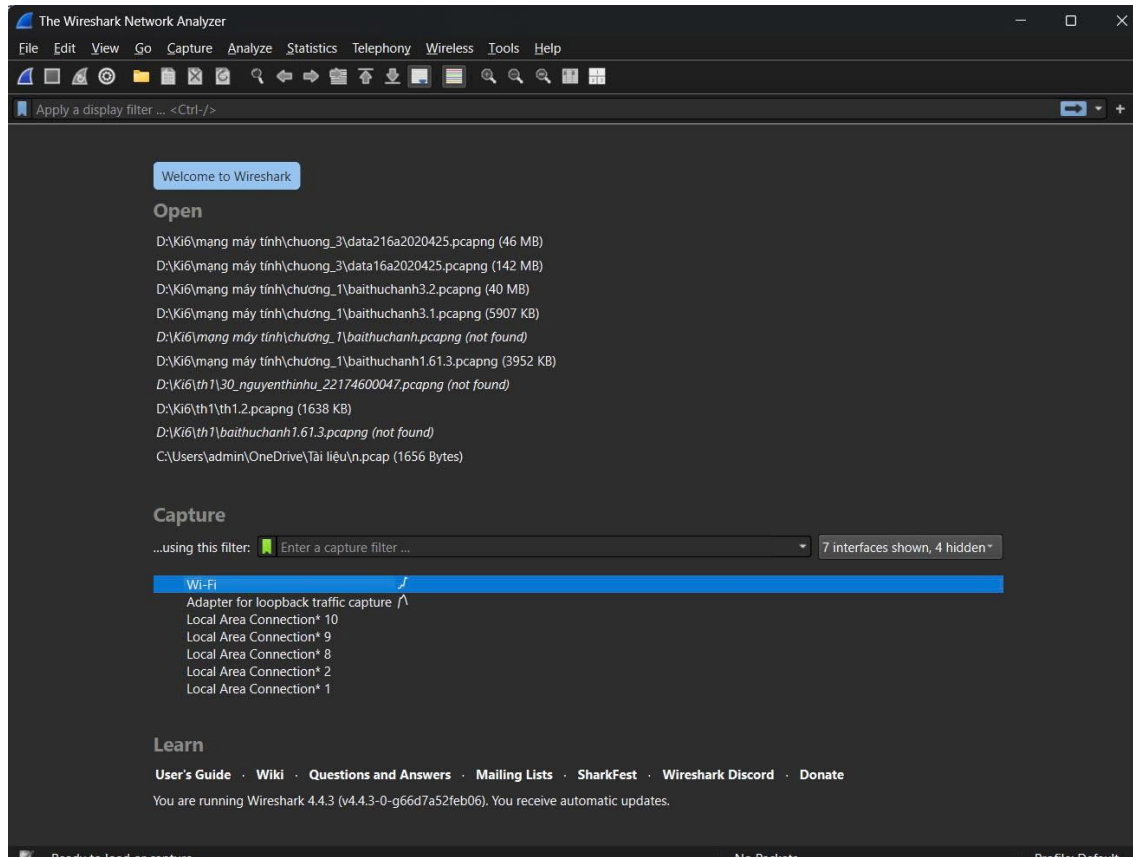



Họ và tên: Nguyễn Mạnh Tiến (22174600066) – Nguyễn Thị Như (22174600047)  
Lớp: DHKL16A2HN


## KIỂM TRA THỰC HÀNH 4

**Bước 1: Mở Wireshark, chọn card mạng, bắt gói khi truy cập một trang web.**



## Bước 2: Lọc giao thức HTTP, truy cập một trang login, quan sát gói gửi dữ liệu.

 acunetix

 acuart

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

[home](#) | [categories](#) | [artists](#) | [disclaimer](#) | [your cart](#) | [guestbook](#) | [AJAX Demo](#) | [Logout test](#)

search art

go

Browse categories

Browse artists

Your cart

Signup

Your profile

Our guestbook

AJAX Demo

Logout


Links

Security art

PHP scanner

PHP vuln help

Fractal Explorer



hai" (test)

On this page you can visualize or edit you user information.

Name:

Credit card number:

E-Mail:

Phone number:

Address:

update

You have 0 items in your cart. You visualize you cart here.

[About Us](#) | [Privacy Policy](#) | [Contact Us](#) | ©2019 Acunetix Ltd

**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.

No.	Time	Source	Destination	Protocol	Length	Info
570	14.800007	172.20.10.3	44.228.249.3	HTTP	547	GET /login.php HTTP/1.1
1197	47.627814	172.20.10.3	44.228.249.3	HTTP	749	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
1329	55.968883	172.20.10.3	44.228.249.3	HTTP	749	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
1373	56.218509	44.228.249.3	172.20.10.3	HTTP	163	HTTP/1.1 200 OK (text/html)

Frame 1197: 749 bytes on wire (5992 bits), 749 bytes captured (5992 bits) on interface \Device\NPF... (08C29E03-97EA-4A57-9E0E-692699B3BC3F), Ethernet II, Src: AzureWaveTec... (34:6f:24:5d:c8:31), Dst: 46:4a:db:05:11:64 (46:4a:db:05:11:64)

Internet Protocol Version 4, Src: 172.20.10.3, Dst: 44.228.249.3

Transmission Control Protocol, Src Port: 57687, Dst Port: 80, Seq: 1, Ack: 1, Len: 695

Hypertext Transfer Protocol

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

0000 46 4a db 05 11 64 34 6f 24 5d c8 31 08 00 45 00 F3 ddo \$} ~  
0010 02 df b3 76 40 00 00 06 60 a3 ac 14 0a 03 2c e4 . ve h  
0020 f9 03 e1 57 00 50 ee 1b 61 67 63 1d bc 26 50 18 . W P agc  
0030 00 ff 3f 3c 00 00 50 4f 53 54 20 2f 75 73 65 72 . ? . PO ST  
0040 69 6e 66 6f 2e 70 68 70 20 48 54 5a 50 2f 31 2e info.php HT  
0050 31 0d 0a 48 6f 73 74 3a 20 74 65 73 74 70 68 70 1 Host: te  
0060 2e 76 75 6c 6e 77 65 62 2e 63 6f 6d 0d 0a 43 6f .vulnweb .co  
0070 6e 6e 65 63 74 69 6f 6e 3a 20 0b 65 65 70 2d 61 nnection : k  
0080 6c 69 76 65 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 65 llver Co nten  
0090 6e 67 74 68 3a 20 32 30 0d 0a 43 61 63 68 65 3d ngth: 20 . C  
00a0 43 6f 6e 74 72 6f 6c 3a 20 6d 61 78 2d 61 67 65 Control: ma  
00b0 3d 30 0d 0a 4f 72 69 6f 69 6e 3a 20 68 74 74 70 =0 Orig in:  
00c0 3a 2f 2f 74 65 73 74 70 68 70 2e 76 75 6c 6e 77 .//testp hp:  
00d0 65 62 2e 63 6f 6d 0d 0a 43 6f 6e 74 65 6e 74 2d eb.com Con  
00e0 54 79 70 65 3a 20 61 70 70 6c 69 63 61 74 69 6f Type: ap pli  
00f0 6e 2f 78 2d 77 77 2d 66 6f 72 6d 2d 75 72 6c n/x-www- forc  
0100 65 6e 63 6f 64 65 64 0a 55 70 67 72 61 64 65 encoded Up  
0110 2d 49 6e 73 65 63 75 72 65 2d 52 65 71 75 65 73 -Insecur e-R  
0120 74 73 3a 20 31 0d 0a 55 73 65 72 2d 41 67 65 6e ts: 1 U ser  
0130 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 t: Mozilla/ ~  
0140 57 69 6e 64 6f 77 73 20 4e 54 20 31 30 2e 30 3b Windows NT ~

## Bước 3: Lưu file kết quả bắt gói (.pcapng).

Kết quả lưu gói tin tại: data\_nhu\_mtien\_16a2.pcapng

## Bước 4: Mở lại file đã lưu, phân tích theo từng tầng trong mô hình OSI.

Tầng 1: Physical (Vật lý)

- Không hiển thị rõ trong ảnh, nhưng tầng này chịu trách nhiệm truyền bit dưới dạng tín hiệu điện, quang hoặc vô tuyến trên đường truyền vật lý (cáp, sóng,...).

## Tầng 2: Data Link (Liên kết dữ liệu)

Thông tin từ dòng Ethernet II:

- Source MAC: 34:6f:24:5d:c8:31 (AzureWaveTec\_5d:c8:31)
- Destination MAC: 46:4a:db:05:11:64
- Type: IPv4 (0x0800)

→ Đây là thông tin khung Ethernet, đảm bảo truyền dữ liệu từ một máy vật lý đến máy kế tiếp trong mạng.

## Tầng 3: Network (Mạng)

Thông tin từ dòng Internet Protocol Version 4 (IPv4):

- Source IP: 172.20.10.3
- Destination IP: 44.228.249.3
- TTL (Time To Live): 128
- Protocol: TCP (6)
- Header checksum: 0x697b

→ Gói tin sử dụng giao thức IPv4 để định tuyến từ IP nguồn đến IP đích.

## Tầng 4: Transport (Giao vận)

Thông tin từ dòng Transmission Control Protocol (TCP):

- Source Port: 57664
- Destination Port: 80 (HTTP)
- Sequence Number: 1
- Acknowledgment Number: 1
- Flags: PSH, ACK
- Window Size: 65280
- TCP Segment Len: 493 bytes

→ Sử dụng TCP, là giao thức hướng kết nối, đảm bảo dữ liệu đến đúng thứ tự và không bị mất.

## Tầng 5: Session (Phiên)

- Không có trường rõ ràng đại diện, nhưng thông qua TCP connection có thể hiểu là đã thiết lập một phiên giữa client và server thông qua cổng 80.

## Tầng 6: Presentation (Trình bày)

- Không có mã hóa hay nén dữ liệu phức tạp → HTTP đơn giản, trình bày dưới dạng văn bản ASCII (plain-text)
- Ví dụ: GET /login.php HTTP/1.1\r\n, Cookie: login=test%2Ftest\r\n

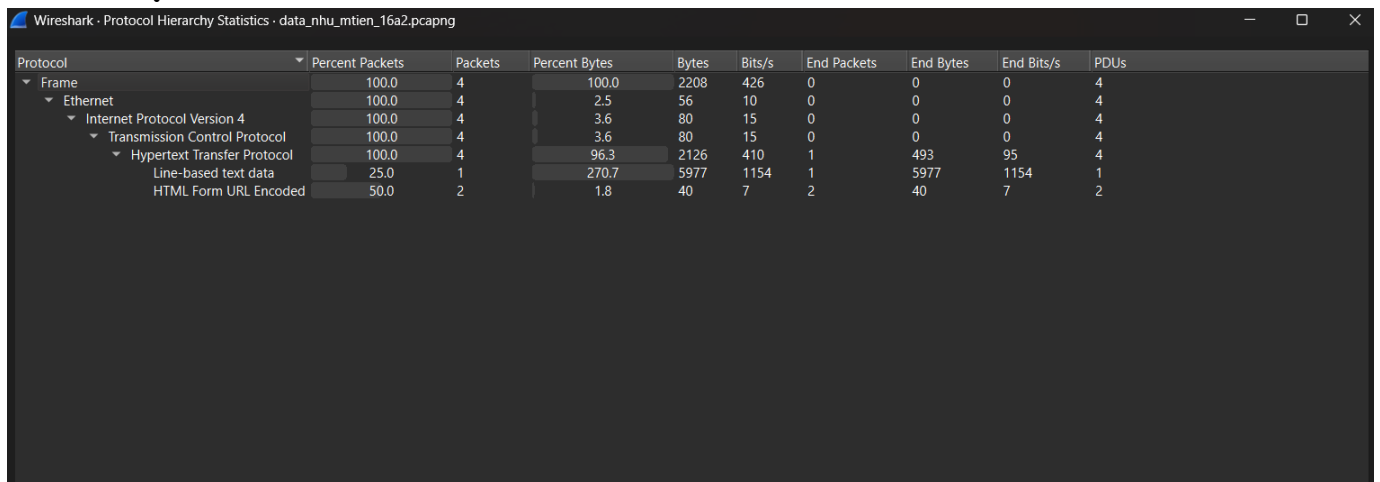
## Tầng 7: Application (Ứng dụng)

Thông tin từ phần Hypertext Transfer Protocol (HTTP):

- Method: GET /login.php
- Host: testphp.vulnweb.com
- User-Agent: Trình duyệt Chrome trên Windows 10
- Cookie: login=test%2Ftest

→ Ứng dụng HTTP dùng để gửi yêu cầu truy cập trang /login.php, có kèm cookie.

## Bước 5: Sử dụng tính năng Protocol Hierarchy hoặc Follow TCP Stream để quan sát toàn cục.



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	4	100.0	2208	426	0	0	0	4
Ethernet	100.0	4	2.5	56	10	0	0	0	4
Internet Protocol Version 4	100.0	4	3.6	80	15	0	0	0	4
Transmission Control Protocol	100.0	4	3.6	80	15	0	0	0	4
Hypertext Transfer Protocol	100.0	4	96.3	2126	410	1	493	95	4
Line-based text data	25.0	1	270.7	5977	1154	1	5977	1154	1
HTML Form URL Encoded	50.0	2	1.8	40	7	2	40	7	2

Thống kê "Protocol Hierarchy Statistics" từ Wireshark:

### Tổng quan

- Tổng số gói tin: 4 (100%).
- Tổng số byte: 2208 bytes.
- Tất cả các gói đều sử dụng: Ethernet → IPv4 → TCP → HTTP.

### Phân tầng giao thức

#### 1. Frame / Ethernet

- 100% số gói sử dụng Ethernet.
- Tổng số byte: 2208 bytes.

#### 2. IPv4

- Tất cả các gói đều là IPv4 (4 gói, 100%).
- Tổng số byte: 80 bytes (3.6% tổng lưu lượng).

### 3. TCP

- Tất cả các gói đều sử dụng TCP (4 gói, 100%).
- Tổng số byte: 80 bytes (3.6%).

### 4. HTTP (Hypertext Transfer Protocol)

- 4 gói HTTP (100%).
- Chiếm phần lớn dung lượng: 2126 bytes (96.3%).

Bên trong HTTP:

- Line-based text data:
  - 1 gói (25%)
  - 5977 bytes (270.7%) — con số này cao vì có thể là payload lớn nằm trong cùng 1 gói hoặc tính cả dữ liệu phân mảnh, nhưng do chỉ có 4 gói nên tỉ lệ bị đẩy cao.
- HTML Form URL Encoded:
  - 2 gói (50%)
  - 40 bytes (1.8%)

### Các chỉ số khác

- End Packets và End Bytes cho biết gói tin cuối cùng của phiên kết thúc như thế nào.
  - Có 1 gói kết thúc với Line-based text data (kèm theo 1154 bits).
  - 2 gói kết thúc với HTML Form URL Encoded (tổng cộng 40 bytes).
- PDUs: Có 4 đơn vị dữ liệu giao thức (Protocol Data Units), tương ứng với 4 gói tin đã bắt được.

### Bước 6: Viết mã Python dùng thư viện PyShark để truy xuất thông tin tầng 2 và tầng 3 từ file .pcapng.

```
import pyshark
```

```
# Đường dẫn đến file .pcapng
```

```
path= r"D:/Ki6/mạng máy tính/chuong_3/data_nhu_mtien_16a2.pcapng"
```

```
# Mở file pcapng
```

```
cap = pyshark.FileCapture(path, use_json=True)
```

```
# Danh sách lưu thông tin tầng 2 và 3
```

```
layer2_3_info = []
```

```
for packet in cap:
```

```
    try:
```

```
        # Lấy thông tin tầng 2 (MAC)
```

```
        eth_src = packet.eth.src if hasattr(packet, 'eth') else 'N/A'
```

```
        eth_dst = packet.eth.dst if hasattr(packet, 'eth') else 'N/A'
```

```
        # Lấy thông tin tầng 3 (IP)
```

```
        ip_src = packet.ip.src if hasattr(packet, 'ip') else 'N/A'
```

```
        ip_dst = packet.ip.dst if hasattr(packet, 'ip') else 'N/A'
```

```
        layer2_3_info.append({
```

```
            'eth_src': eth_src,
```

```
            'eth_dst': eth_dst,
```

```
            'ip_src': ip_src,
```

```
            'ip_dst': ip_dst
```

```
        })
```

```
    except AttributeError:
```

```
        continue
```

```
cap.close()
```

```
# In ra 10 dòng đầu tiên
```

```
for i, info in enumerate(layer2_3_info[:10], start=1):
```

```
    print(f"Gói {i}:")
```

```
    print(f"  MAC nguồn: {info['eth_src']}")
```

```
    print(f"  MAC đích: {info['eth_dst']}")
```

```
    print(f"  IP nguồn: {info['ip_src']}")
```

```
    print(f"  IP đích: {info['ip_dst']}\n")
```

on.exe d:/Ki6/SQLforKHDL/kiemtra2.py

Gói 1:

MAC nguồn: 34:6f:24:5d:c8:31

MAC đích: 46:4a:db:05:11:64

IP nguồn: N/A

IP đích: N/A

Gói 2:

MAC nguồn: 34:6f:24:5d:c8:31

MAC đích: 46:4a:db:05:11:64

IP nguồn: N/A

IP đích: N/A

Gói 3:

MAC nguồn: 34:6f:24:5d:c8:31

MAC đích: 46:4a:db:05:11:64

IP nguồn: N/A

IP đích: N/A

Gói 4:

MAC nguồn: 46:4a:db:05:11:64

MAC đích: 34:6f:24:5d:c8:31

IP nguồn: N/A

IP đích: N/A

Gói 5:

MAC nguồn: 46:4a:db:05:11:64

MAC đích: 34:6f:24:5d:c8:31

IP nguồn: N/A

IP đích: N/A