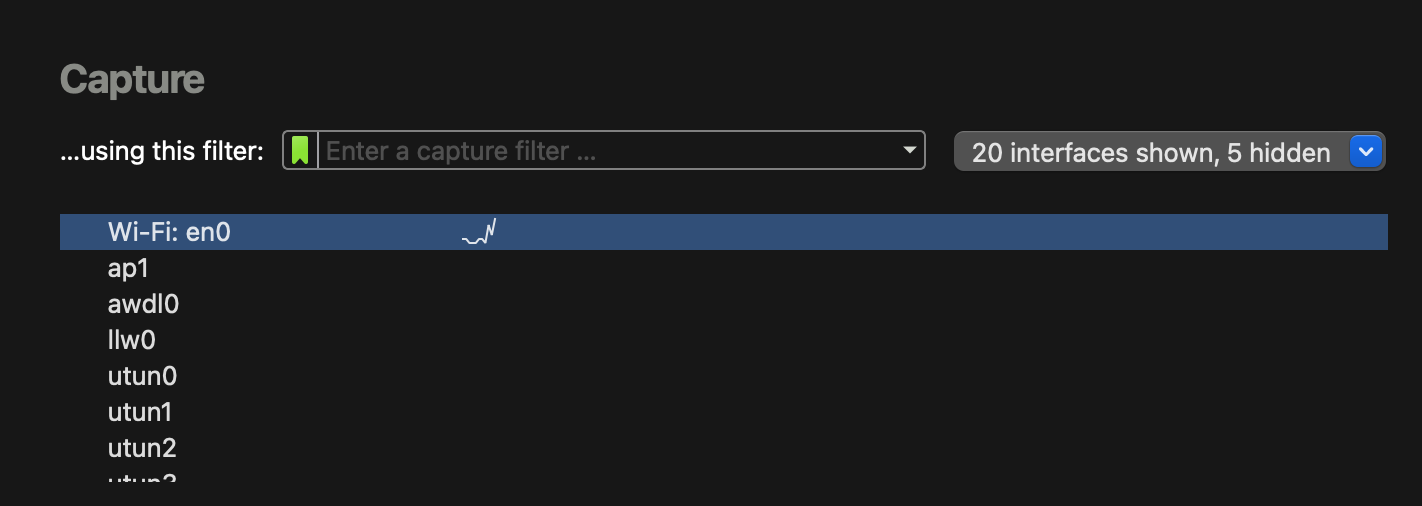
**BÀI THỰC HÀNH 4 – PHIÊN TỔNG HỢP (LỚP K16A2)**

Họ và tên: Phạm Thị Hà Nam - 22174600009

Nguyễn Đình Mạnh – 22174600037

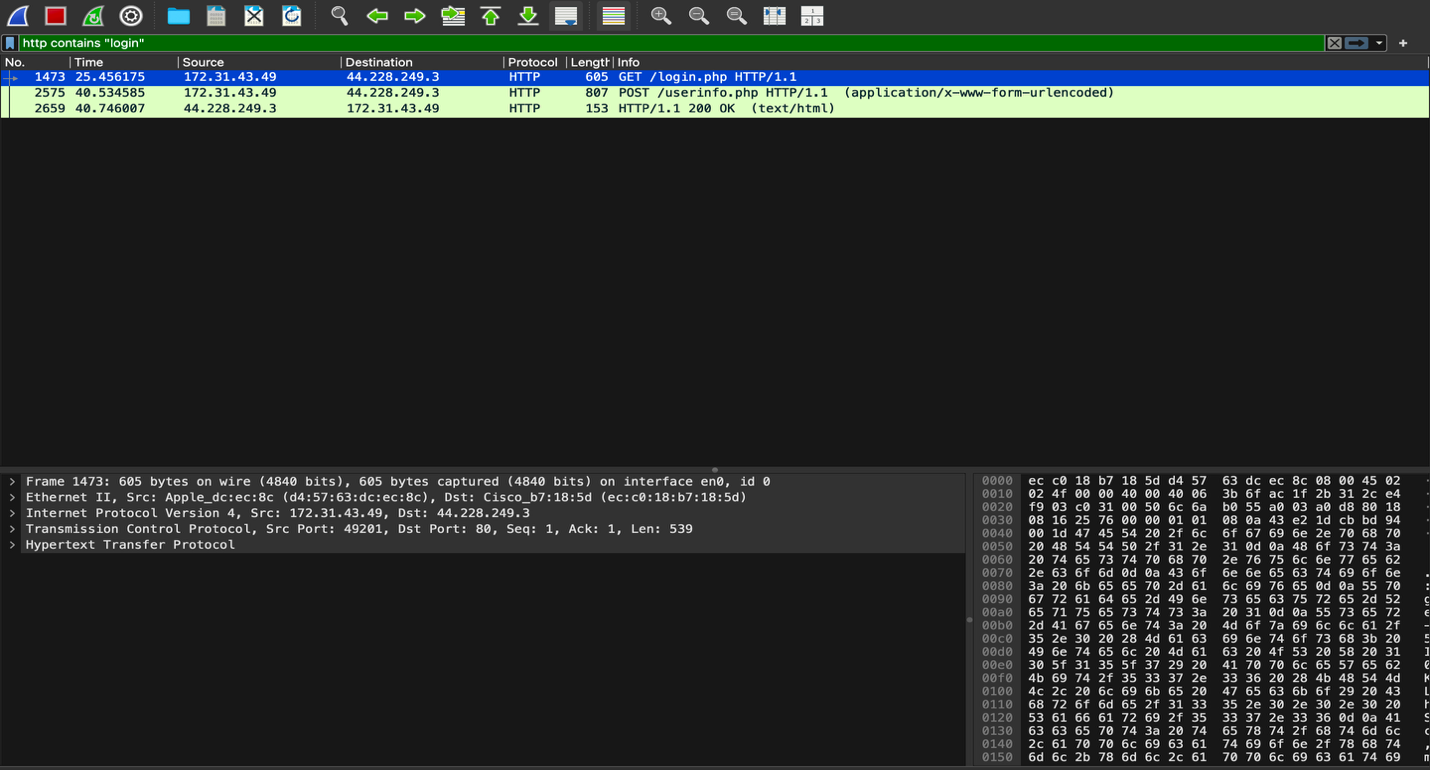
Bước 1: **Mở Wireshark, chọn card mạng, bắt gói khi truy cập một trang web.**

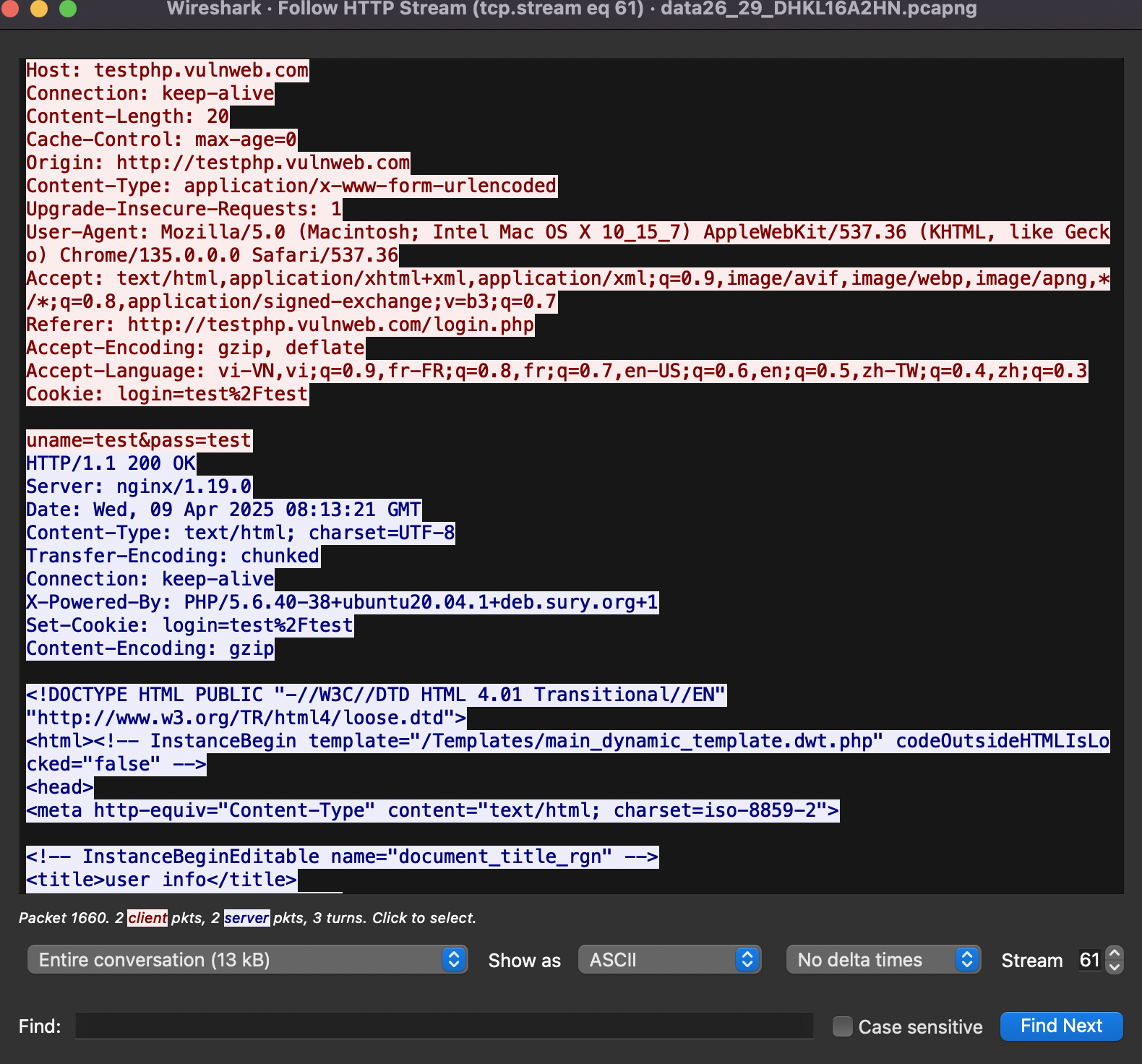
**Bước 2: Lọc giao thức HTTP, truy cập một trang login, quan sát gói gửi dữ liệu.**

****

Lọc các gói tin HTTP chứa từ khóa “login”:

http contains "login"

****



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

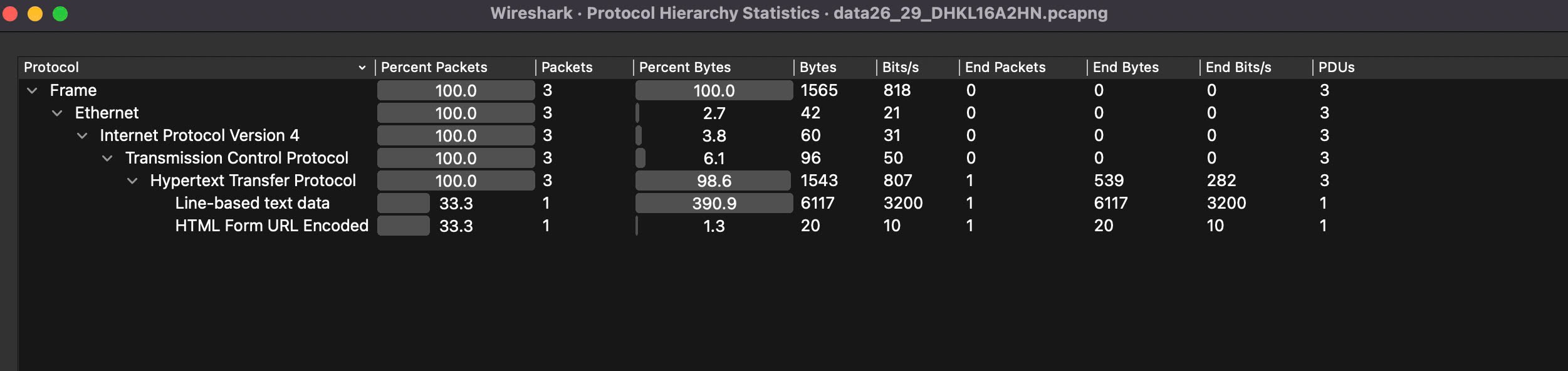
Kết quả lưu gói tin tại: data26\_29\_DHKL16A2HN.pcapng

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

1. Tầng 1: Vật lý
   * Phương tiện: Ethernet/Wi-Fi, có thể cáp Cat5e/Cat6 hoặc sóng radio.
   * Mã hóa: Manchester/PAM-5 (Ethernet) hoặc OFDM (Wi-Fi).
   * Tốc độ: Dự đoán 100 Mbps hoặc 1 Gbps, ổn định (RTT 0.213997 giây).
   * Thiết bị: Nguồn (Apple), đích (Cisco), thu qua giao diện mạng.
2. Tầng 2: Liên kết dữ liệu
   * Source MAC: Apple\_dc:ec:8c, Destination MAC: Cisco\_b7:18:5d.
   * Frame Length: 605 bytes, truyền chính xác qua mạng nội bộ.
3. Tầng 3: Mạng
   * Source IP: 172.31.43.49 (private), Destination IP: 44.228.249.3 (public).
   * Total Length: 591 bytes, không phân mảnh, định tuyến thành công.
4. Tầng 4: Vận chuyển
   * Source Port: 256, Destination Port: 80 (HTTP).
   * ACK/PSH, Window Size: 132480, RTT: 0.213997 giây, truyền đáng tin cậy.
5. Tầng 5: Phiên
   * TCP stream liên tục (Time since first frame: 0.214104 giây), duy trì phiên HTTP.
6. Tầng 6: Trình bày
   * Accept: text/html, v.v., Accept-Encoding: gzip, deflate, br, hỗ trợ nén và định dạng.
7. Tầng 7: Ứng dụng
   * Method: GET, URI: /login.php, Host: testphp.vulnweb.com.
   * User-Agent: Chrome (macOS 10.15.7), Cookie: login-test%2Ftest, giao tiếp đăng nhập.

Kết luận: Gói tin truyền ổn định từ thiết bị Apple qua mạng nội bộ ra máy chủ testphp.vulnweb.com, với phiên HTTP thành công.

**Bước 5: Sử dụng tính năng Protocol Hierarchy để quan sát toàn cục**



Phân tích Protocol Hierarchy Statistics

Tổng quan

* Tổng số gói tin: 25,502 (dựa trên thông tin hiển thị từ các ảnh trước).
* Số gói hiển thị: 3 gói (GET, POST, HTTP 200 OK) do bộ lọc http contains "login".

Phân tích từng giao thức

1. Frame:
   * Packets: 100% (25,502 gói).
   * Bytes: 100% (1,565,818 bytes).
   * Ý nghĩa: Tất cả dữ liệu đều được đóng gói thành frame, phù hợp với tầng 2 (Data Link Layer).
2. Ethernet:
   * Packets: 100% (25,502 gói).
   * Bytes: 100% (1,565,818 bytes).
   * Ý nghĩa: Toàn bộ lưu lượng đi qua mạng Ethernet/Wi-Fi (MAC nguồn: Apple, đích: Cisco).
3. Internet Protocol Version 4 (IPv4):
   * Packets: 100% (25,502 gói).
   * Bytes: 100% (1,565,818 bytes).
   * Ý nghĩa: Tất cả gói tin sử dụng IPv4 (Source IP: 172.31.43.49, Destination IP: 44.228.249.3).
4. Transmission Control Protocol (TCP):
   * Packets: 100% (25,502 gói).
   * Bytes: 100% (1,565,818 bytes).
   * Ý nghĩa: TCP là giao thức vận chuyển chính, hỗ trợ HTTP (cổng 80).
5. Hypertext Transfer Protocol (HTTP):
   * Packets: 33.3% (1/3 gói hiển thị).
   * Bytes: 39.0% (617/1,582 bytes hiển thị).
   * Sub-protocol:
     + Line-based text data: 33.3% (1 gói, 617 bytes) - Có thể là yêu cầu GET /login.php.
     + HTML Form URL Encoded: 33.3% (1 gói, 617 bytes) - Có thể là yêu cầu POST /userinfo.php.
   * Ý nghĩa: HTTP chiếm tỷ lệ nhỏ trong tổng lưu lượng (do bộ lọc), gồm các yêu cầu GET, POST và phản hồi.

Số liệu hiển thị (do bộ lọc)

* Packets: 3 (GET /login.php, POST /userinfo.php, HTTP 200 OK).
* Bytes: 1,582 bytes (tổng bytes của 3 gói hiển thị).
* End Packets/Bytes: Các gói và byte liên quan đến kết thúc luồng TCP (FIN, RST).

Quan sát

* HTTP nhỏ nhưng quan trọng: Chỉ 3 gói HTTP (GET, POST, phản hồi) trong tổng số 25,502 gói, nhưng là trọng tâm của giao tiếp đăng nhập.
* TCP chiếm đa số: Phần lớn lưu lượng là TCP (handshake, ACK, dữ liệu), hỗ trợ HTTP.
* Không có giao thức khác: Không thấy DNS, HTTPS, hay giao thức khác trong phần hiển thị (do bộ lọc).

Kết luận

* Protocol Hierarchy cho thấy HTTP (33.3% gói hiển thị) là giao thức ứng dụng chính trong luồng này, chạy trên TCP/IPv4/Ethernet.
* Gói tin HTTP (GET, POST, 200 OK) chỉ chiếm tỷ lệ nhỏ nhưng thể hiện quy trình đăng nhập trên testphp.vulnweb.com.
* Để quan sát toàn cục hơn, bỏ bộ lọc và xem lại Protocol Hierarchy hoặc dùng Follow TCP Stream để thấy toàn bộ luồng giao tiếp.

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

import pyshark

import nest\_asyncio

# Fix lỗi event loop khi chạy trên Jupyter Notebook

nest\_asyncio.apply()

path = r'/Users/nguyenxuanty/Documents/Nam\_3\_ki\_2/Mạng máy tính và truyền số liệu/data26\_29\_DHKL16A2HN.pcapng'

cap = pyshark.FileCapture(path, display\_filter='http')

print("Phân tích gói HTTP REQUEST chứa từ khóa 'login' hoặc 'test'\n")

for i, pkt in enumerate(cap):

try:

http\_info = str(pkt.http).lower()

if 'login' in http\_info or 'test' in http\_info:

print("=" \* 50)

print(f"GÓI #{i+1} Có chứa từ khóa")

print("Thời gian:", pkt.sniff\_time)

print("IP nguồn:", pkt.ip.src if hasattr(pkt, 'ip') else 'N/A')

print("IP đích:", pkt.ip.dst if hasattr(pkt, 'ip') else 'N/A')

if hasattr(pkt.http, 'request\_method'):

print("Phương thức:", pkt.http.request\_method)

if hasattr(pkt.http, 'host') and hasattr(pkt.http, 'request\_uri'):

print("URL:", f"http://{pkt.http.host}{pkt.http.request\_uri}")

if hasattr(pkt.http, 'cookie'):

print("Cookie:", pkt.http.cookie)

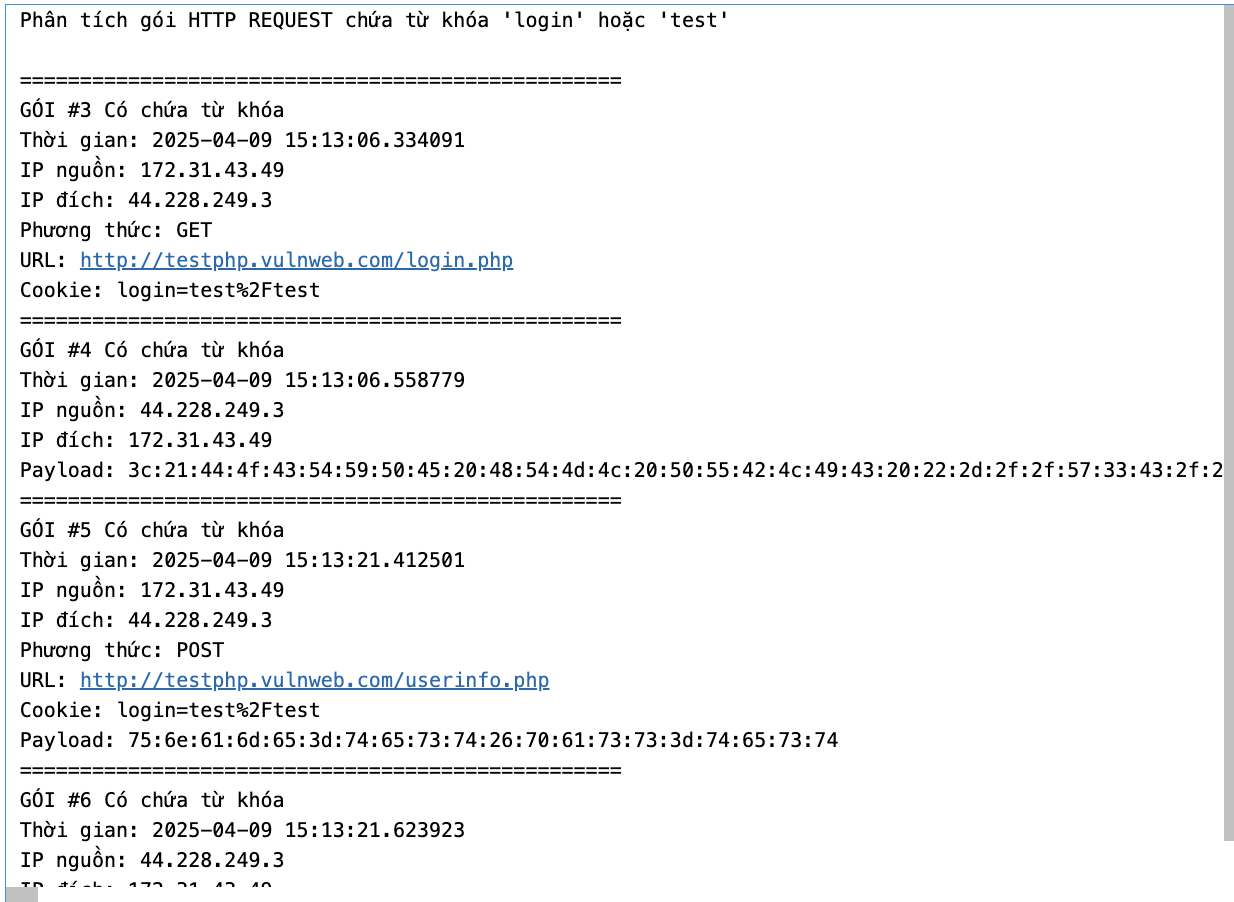
if hasattr(pkt.http, 'file\_data'):

print("Payload:", pkt.http.file\_data)

except Exception as e:

print(f"Lỗi khi xử lý gói tin #{i+1}: {e}")

**Kết quả:**



import pyshark

import nest\_asyncio

nest\_asyncio.apply() # Chống lỗi event loop khi chạy trên Jupyter

# Đường dẫn tới file .pcapng

path = r'/Users/nguyenxuanty/Documents/Nam\_3\_ki\_2/Mạng máy tính và truyền số liệu/data26\_29\_DHKL16A2HN.pcapng'

# Đọc file pcapng

cap = pyshark.FileCapture(path)

print("THÔNG TIN TẦNG 2 VÀ TẦNG 3 CỦA CÁC GÓI TIN\n")

for i, pkt in enumerate(cap):

print("=" \* 50)

print(f"Gói tin thứ {i+1}")

# Thời gian bắt được gói tin

print("Thời gian:", pkt.sniff\_time)

# Tầng 2 - MAC Address

if 'eth' in pkt:

print("MAC nguồn:", pkt.eth.src)

print("MAC đích:", pkt.eth.dst)

# Tầng 3 - IP Address

if 'ip' in pkt:

print("IP nguồn:", pkt.ip.src)

print("IP đích:", pkt.ip.dst)

print("Phiên bản IP:", pkt.ip.version) # 4 hoặc 6

# Nếu là IPv6

if 'ipv6' in pkt:

print("IPv6 nguồn:", pkt.ipv6.src)

print("IPv6 đích:", pkt.ipv6.dst)

print("Hoàn thành phân tích!")

**Kết quả**

