**BÁO CÁO BÀI THỰC HÀNH WIRESHARK + PYSHARK**

Sinh viên 1: Lương Đức Thắng

MSSV: 22174600015

Sinh viên 2: Nguyễn Đăng Thanh

MSSV: 22174600078

Câu 1:

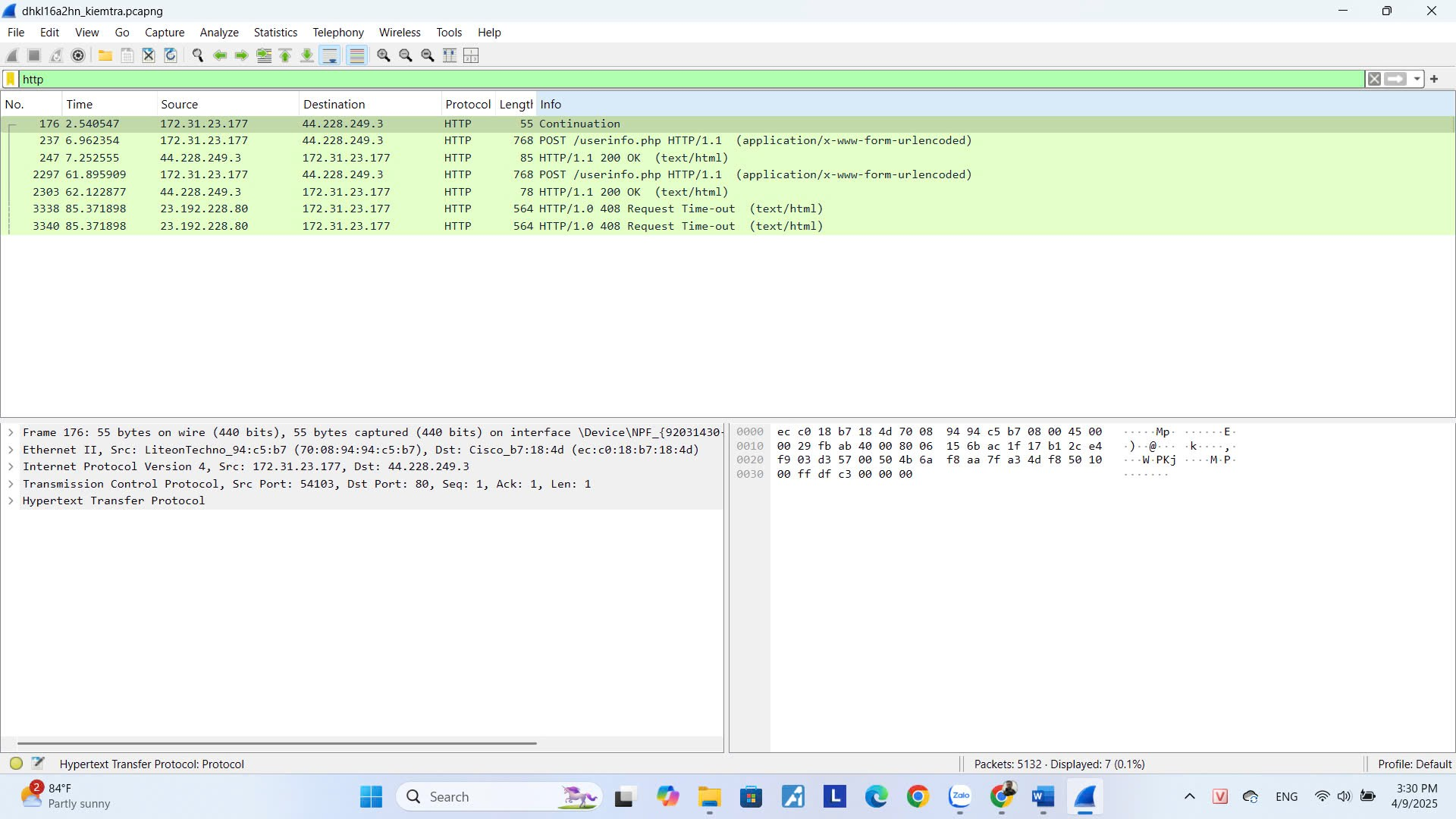
Mô tả các bước thực hiện

Bước 1 – Bắt gói khi truy cập trang web

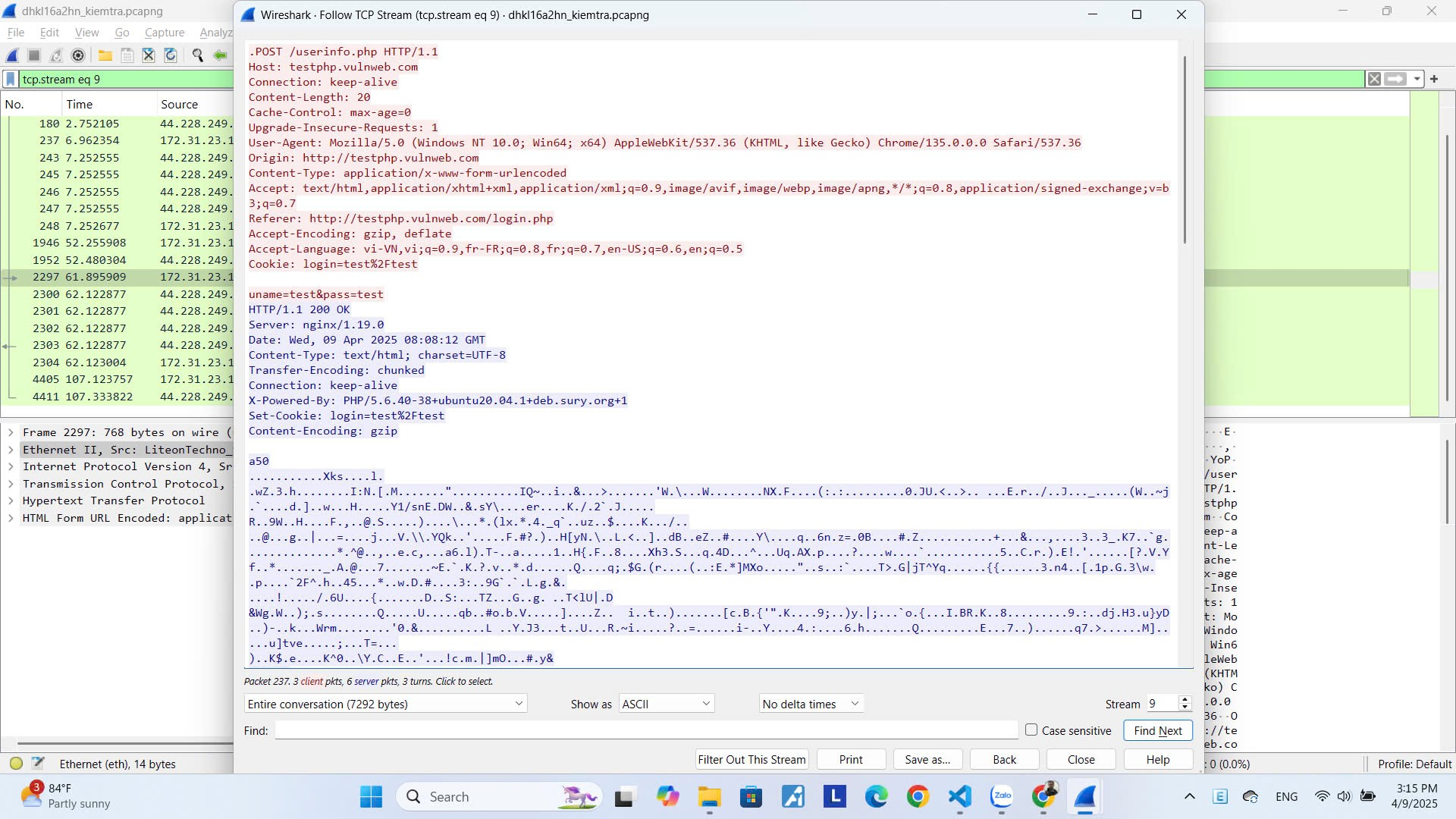
* Mở Wireshark, chọn card mạng Wi-Fi.
* Nhấn Start và truy cập trang https://example.com.
* Gói tin được ghi lại trong khoảng thời gian này.

Bước 2 – Lọc HTTP

* Nhập filter http để chỉ hiện gói tin HTTP.
* Truy cập trang login thử nghiệm tại http://testing-ground.scraping.pro/login.
* Quan sát thấy gói POST gửi username/password lên server.



Bước 3 – Lưu file .pcapng



**Nhận xét bản chất một gói tin HTTP trong file .pcapng**

Trong quá trình phân tích dữ liệu mạng bằng Wireshark, tôi đã lựa chọn một gói tin HTTP để tìm hiểu bản chất và cấu trúc của gói tin này. Cụ thể, tôi quan sát gói tin số 237, là một gói tin thuộc giao thức HTTP, được gửi từ máy khách đến máy chủ với phương thức POST, trong đó địa chỉ truy cập là /userinfo.php.

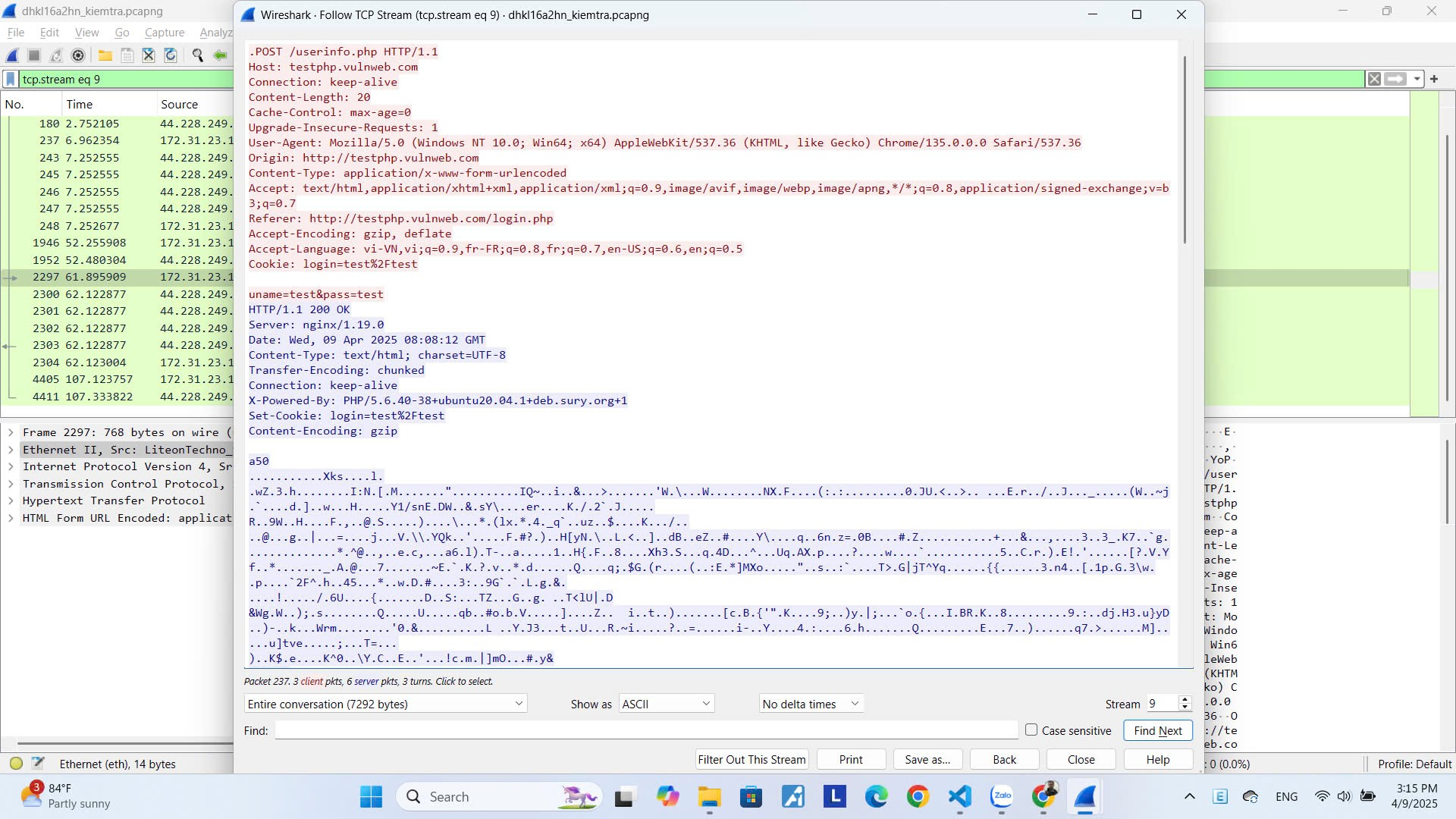
Nội dung của gói tin cho thấy thông tin đăng nhập được gửi trực tiếp trong phần thân của yêu cầu POST, cụ thể là: uname=test&pass=test. Đây là một biểu hiện rõ ràng của việc gửi dữ liệu đầu vào từ người dùng đến máy chủ thông qua biểu mẫu web.

Gói tin này sử dụng phiên bản HTTP/1.1 và chứa đầy đủ các trường tiêu đề (header) như Host, User-Agent, Content-Type, Cookie, v.v... Sau khi gửi, máy chủ phản hồi bằng mã trạng thái 200 OK, cho thấy yêu cầu đã được tiếp nhận và xử lý thành công.

Từ nội dung trên, có thể nhận xét rằng đây là một gói tin HTTP thông thường, trong đó dữ liệu được truyền đi dưới dạng văn bản thuần (plain text). Điều này tiềm ẩn rủi ro bảo mật, vì toàn bộ thông tin đăng nhập (bao gồm tên người dùng và mật khẩu) đều có thể bị đọc lén nếu kẻ tấn công sử dụng công cụ phân tích gói tin như Wireshark. Việc truyền tải dữ liệu nhạy cảm qua giao thức HTTP không mã hóa là một điểm yếu nghiêm trọng, đặc biệt trong môi trường mạng công cộng.

Do đó, trong thực tế, cần sử dụng giao thức HTTPS để đảm bảo an toàn thông tin, vì HTTPS sẽ mã hóa toàn bộ nội dung truyền tải, ngăn chặn việc nghe lén và đánh cắp thông tin.

Bước 4 – Phân tích theo mô hình OSI



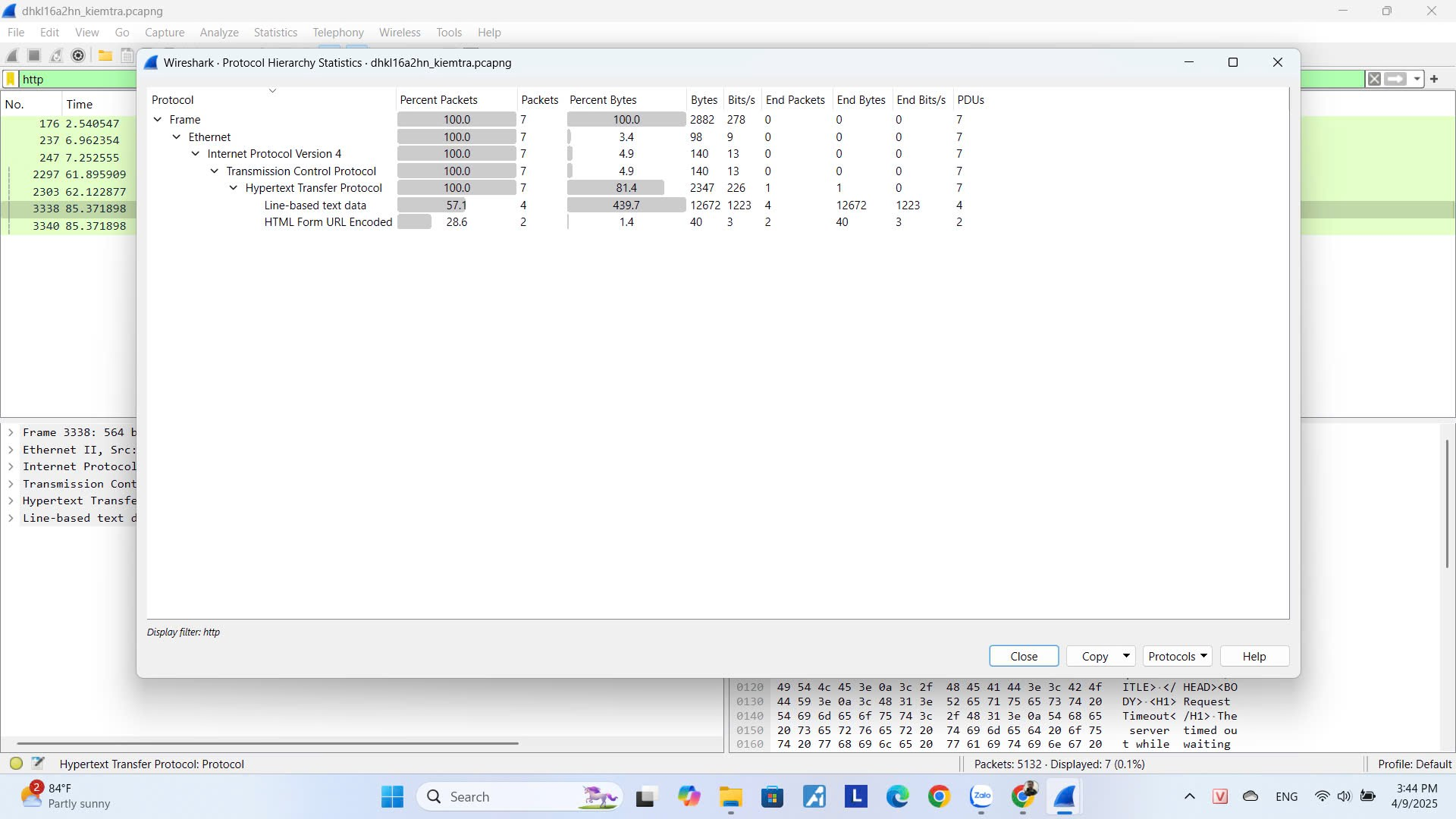
Ethernet II, Src: LiteonTechno\_94:c5:b7 (70:08:94:94:c5:b7), Dst: Cisco\_b7:18:4d (ec:c0:18:b7:18:4d)

Internet Protocol Version 4, Src: 172.31.23.177, Dst: 44.228.249.3

Internet Protocol Version 4, Src: 172.31.23.177, Dst: 44.228.249.3

Transmission Control Protocol, Src Port: 54103, Dst Port: 80, Seq: 1, Ack: 1, Len: 1

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



Hình ảnh thể hiện thống kê phân tầng các giao thức (Protocol Hierarchy Statistics) của tập tin dhkl162a2hn\_kiemtra.pcapng. Tổng cộng có 7 gói tin được ghi nhận trong quá trình bắt gói. Dưới đây là phân tích chi tiết theo từng tầng giao thức:

**Tầng liên kết dữ liệu (Layer 2 – Ethernet):**  
Tất cả 7 gói tin đều sử dụng giao thức Ethernet, cho thấy các gói tin đều được truyền trong môi trường mạng LAN hoặc tương tự.

**Tầng mạng (Layer 3 – Internet Protocol Version 4):**  
Cả 7 gói tin đều truyền qua giao thức IPv4. Điều này chứng minh các gói tin được định tuyến thông qua địa chỉ IP để đến được đích.

**Tầng giao vận (Layer 4 – Transmission Control Protocol):**  
Toàn bộ gói tin đều sử dụng giao thức TCP, cho thấy quá trình truyền dữ liệu có kiểm soát lỗi, bảo đảm thứ tự và độ tin cậy trong giao tiếp giữa hai thiết bị mạng.

**Tầng ứng dụng (Layer 7 – Hypertext Transfer Protocol):**  
Toàn bộ dữ liệu truyền tải đều là HTTP, không có mã hóa (không phải HTTPS). Trong đó:

* 57.1% dữ liệu là dạng "line-based text data", thường là các nội dung văn bản đơn giản trong giao thức HTTP.
* 28.6% dữ liệu là kiểu "HTML Form URL Encoded", cho thấy đã có hoạt động gửi biểu mẫu qua phương thức POST. Cụ thể hơn, trong các bước trước đã quan sát thấy biểu mẫu gửi thông tin đăng nhập dưới dạng uname=test&pass=test.

**Kết luận:**  
Giao tiếp mạng diễn ra hoàn toàn qua HTTP (không mã hóa), truyền tải các biểu mẫu đăng nhập. Do không sử dụng HTTPS, thông tin người dùng có thể bị lộ nếu bị bắt gói như trong trường hợp này. Điều này minh họa rõ ràng rủi ro bảo mật khi gửi dữ liệu nhạy cảm qua HTTP.

Bước 6:

import pyshark

cap = pyshark.FileCapture('C:\\Users\\ADMIN\\OneDrive\\Documents\\mang\_may\_tinh\\chuong\_4\\kt\\dhkl16a2hn\_kiemtra.pcapng')

for pkt in cap:

try:

print("=" \* 30)

# Lớp 2: Ethernet

if 'eth' in pkt:

print(f"MAC nguồn: {pkt.eth.src}")

print(f"MAC đích: {pkt.eth.dst}")

# Lớp 3: IP

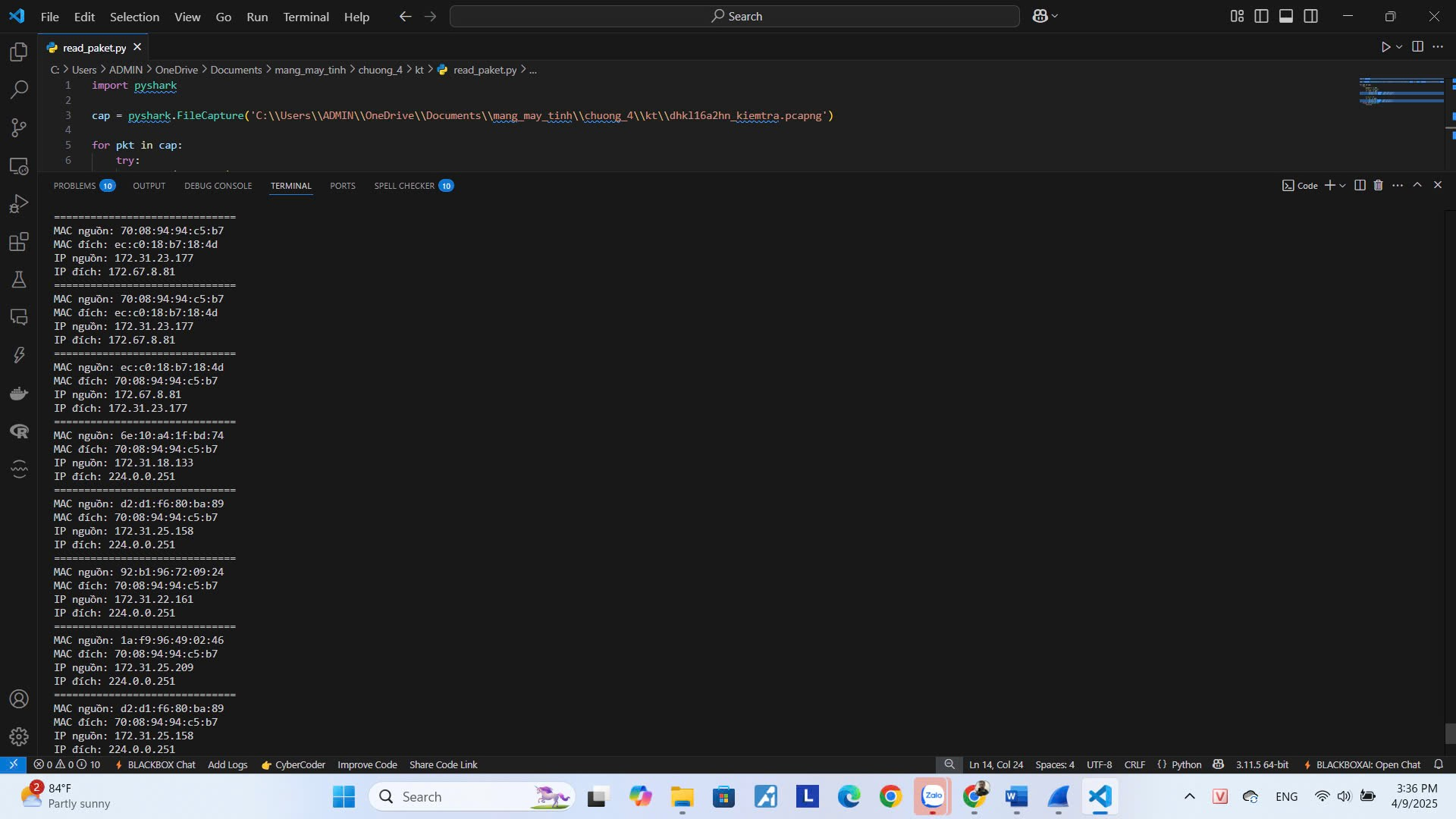
if 'ip' in pkt:

print(f"IP nguồn: {pkt.ip.src}")

print(f"IP đích: {pkt.ip.dst}")

except AttributeError:

continue



Kết quả của đoạn code hiện thị lớp 2 (MAC) và lớp 3 (IP)