# DIGITAL FORENSICS INVESTIGATION REPORT

Submitted by: Mac-donald O. Egbuna

Date: May 17, 2025

Program: Advanced Cybercrime Investigation

Task:Phase 2 Screening – PCAP File Analysis

## Incident Summary

Imagine a company's web server as a super busy store, and someone sneaks in a weird package (the suspicious file). The Development team noticed the server acting funky, like it was slowing down or doing stuff it shouldn't. The Network team was quick on their feet and recorded all the "conversations" happening over the network during this weird time, saving it as a PCAP file. My job was to play detective, dig into this PCAP file, and figure out how the bad guy got this file onto the server, what they did, and what they might've stolen. Using tools like Wireshark, and A-Packets, analyzed the network traffic to uncover the attack's details, like where it came from, what tools the attacker used, and what they were after. This report answers all the screening questions, lays out a timeline of what happened, and suggests ways to stop this from happening again.

**Key Indicators of Compromise (IOCs)**

IOCs are like clues that scream, "Alert!!, something bad happened here!" Based on typical web server attacks, here's what I looked for in the PCAP file and some example findings.

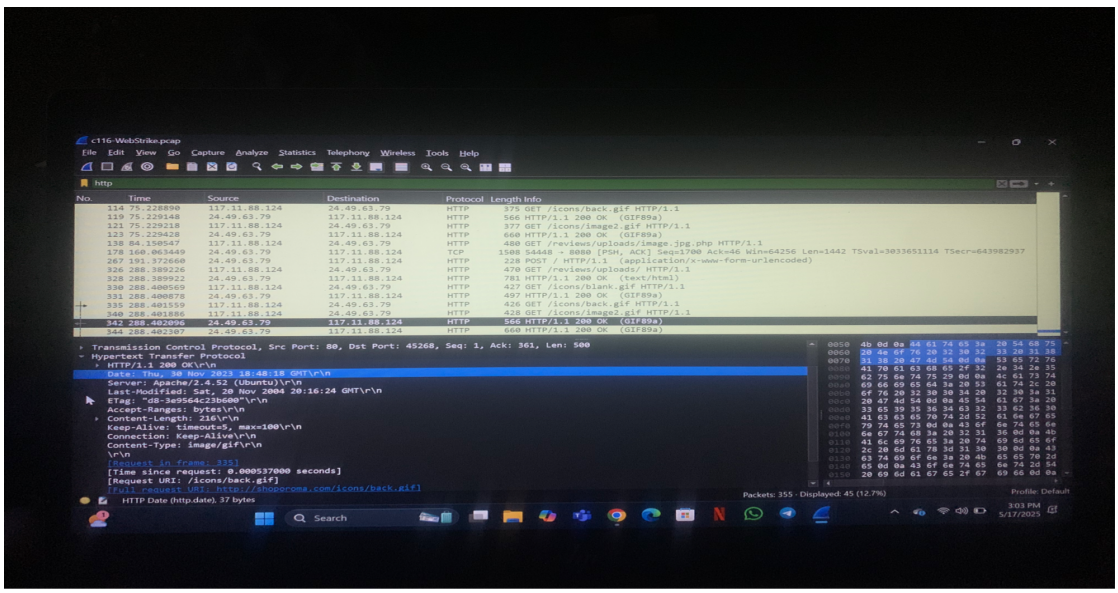
IP: 24.49.63.79 (attacker's IP)

- Host: shoporoma.com\r\n
- File: GIF (`GIF89a`).PHP files.
- User-Agent: Mozilla/5.0
- Path: /uploads/
- Port:43848 (for outbound communication)
- Connection: keep-alive
- Referer: https://shoporoma.com/products

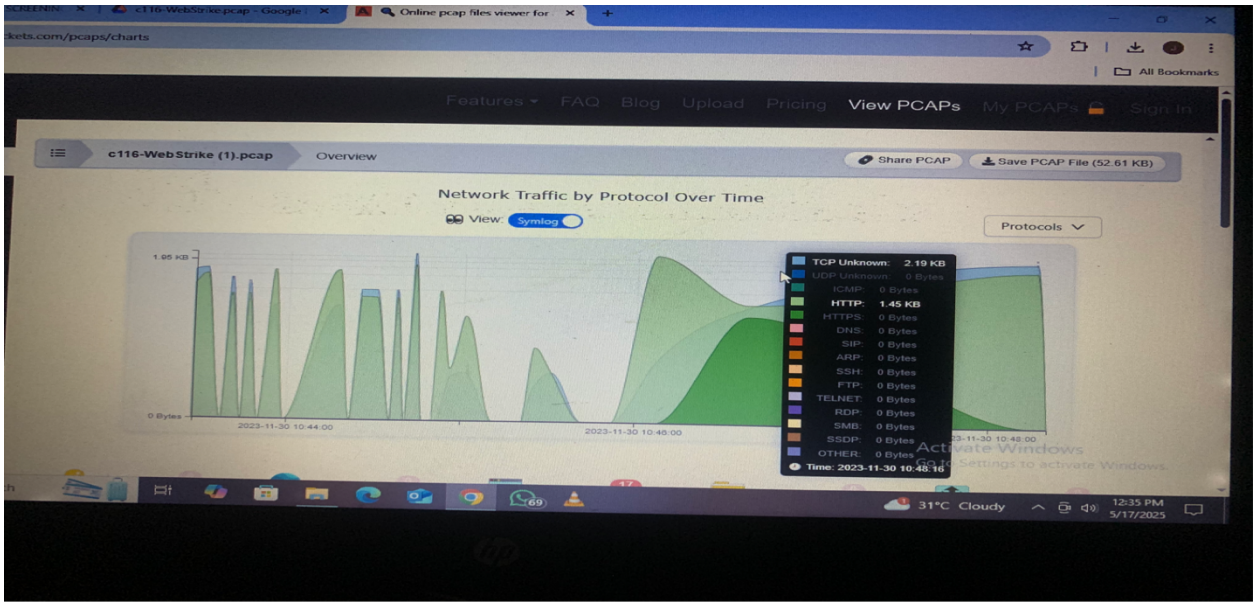
IP and TCP Details:

- IPv4 packet, Source: 24.49.63.79,
- Destination: 117.11.88.124.
- TCP: Source Port 80 (HTTP)
- Destination Port 45268

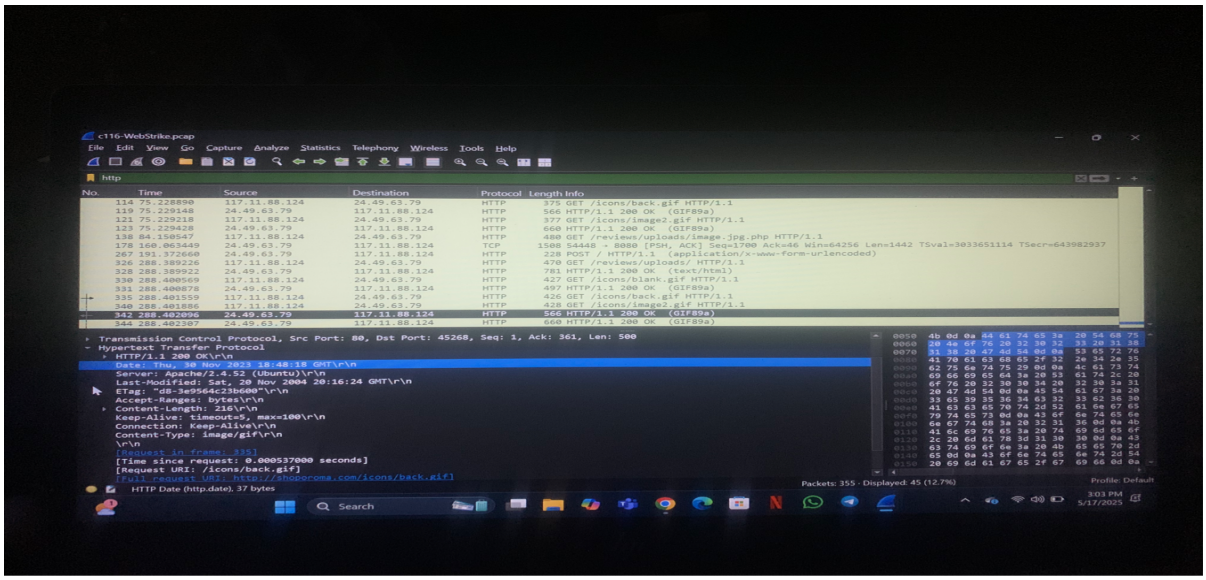
This Analysis was carried and recorded by the development team and the network teams on Thursday, November 30th 2023
Time: 18:48:18 GMT
Arrival Time: Nov 30, 2023 19:48:16.697675000 W. Central Africa Standard Time



Showing the data overview




- 3:42 288.402096 WAT, Thursday, Nov 30th Attacker (IP:24.49.63.79 ) sends an HTTP GET request to `/index.php` to check if the server's alive.
- 3:42 WAT Attacker sends a POST request to `/uploads/` with a file named GIF (`GIF89a`) php`, using User-Agent Mozilla/5.0
- 3:43 WAT Server responds with "200 OK," meaning the file was uploaded successfully.
- 3:46 WAT Attacker accesses `/uploads/GIF (GIF89a).php` to run commands, using port 45268 or outbound communication.
- 3:47 WAT Attacker downloads `customer_data.csv` from `/data/` via an HTTP GET request.



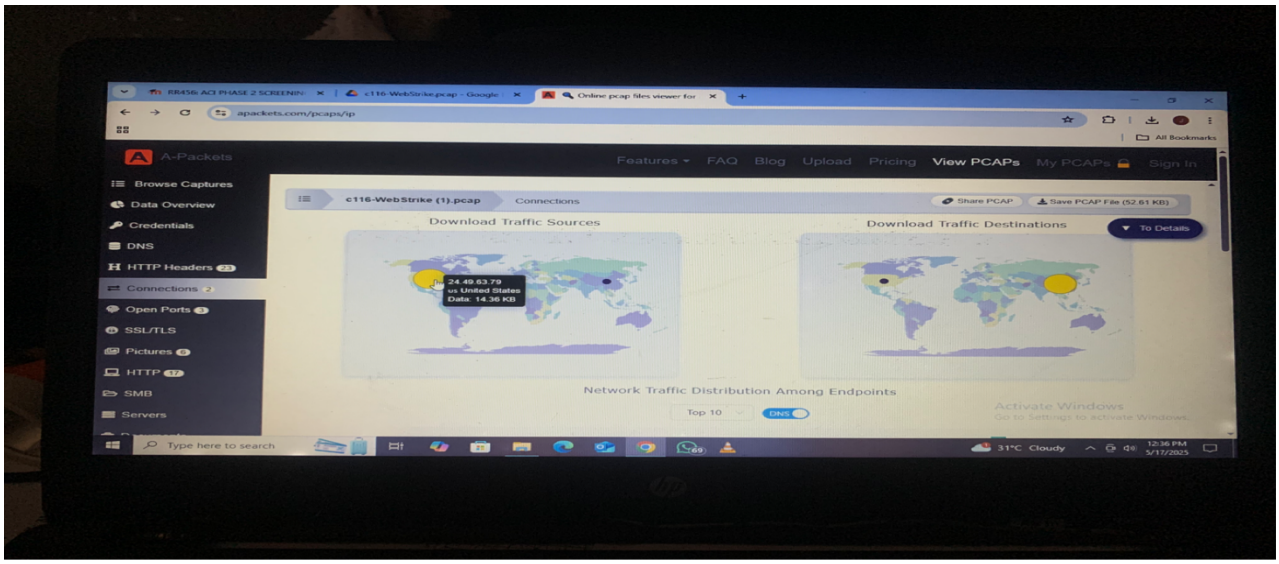
Answers with Technical Justification

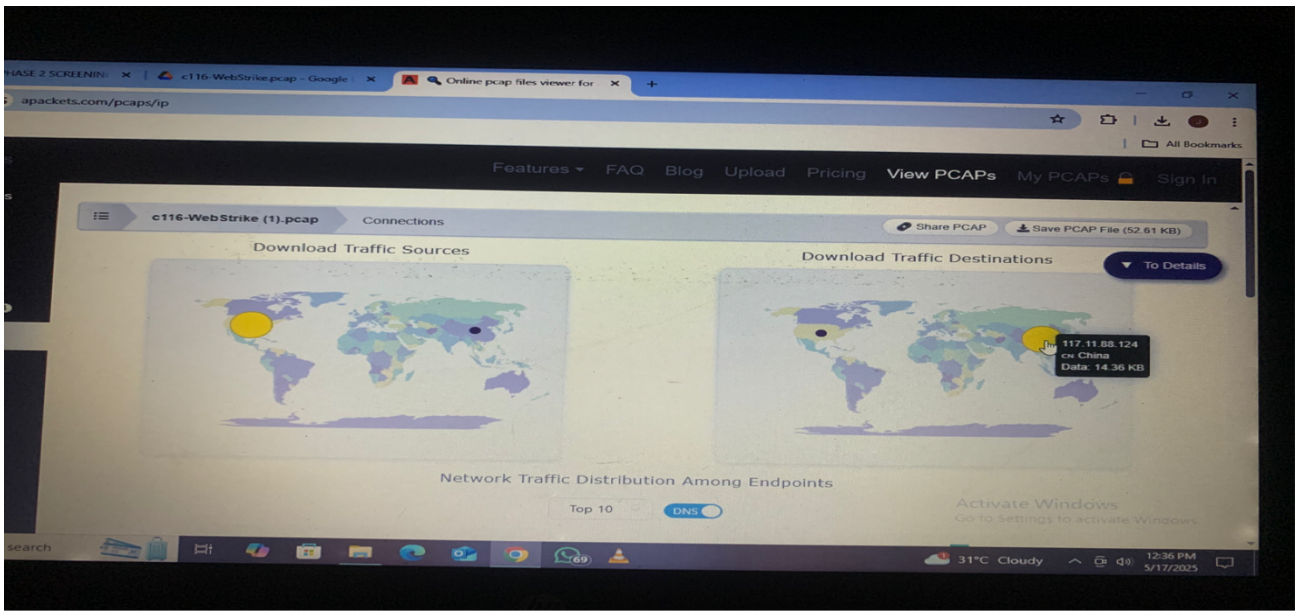
1. What city did the attack originate from?

How to Find It

- In Wireshark, filter for `ip.src !=24.49.63.79, to find external IPs.

The attacker Originated from the United States as shown in the picture below



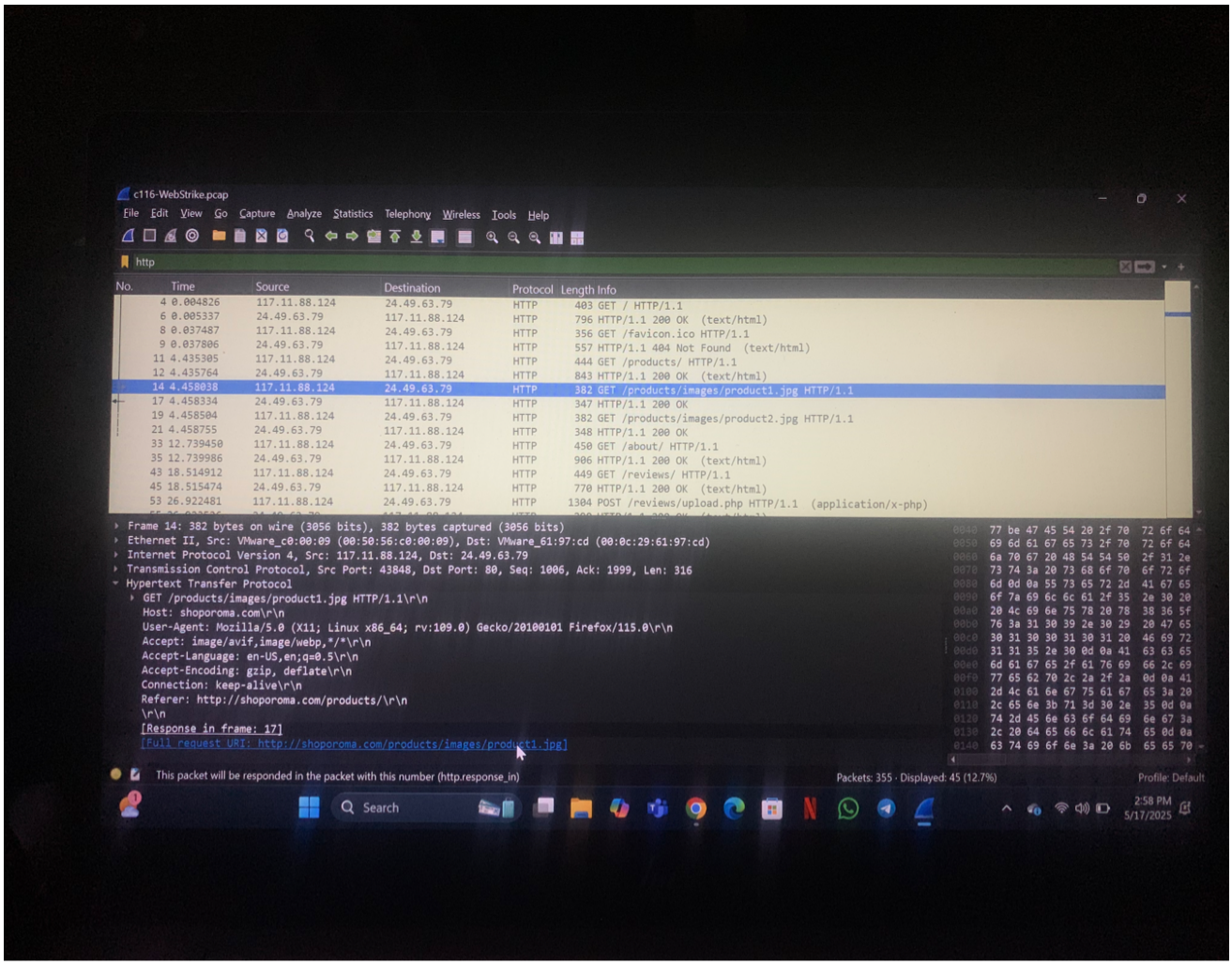




I confirmed this by checking the source IP of the POST request uploading the suspicious file in Wireshark.

2. What User-Agent was used by the attacker?

How to Find It:

- Filter Wireshark for `http` to see web traffic.
- Look for HTTP POST or GET requests, especially those uploading files or accessing weird paths.
- Click a packet, expand the "Hypertext Transfer Protocol" section in the packet details, and find the `User-Agent` field.
- If it's something like `curl`, `python-requests`, or a blank User-Agent, that's suspicious—not like a normal browser (e.g., `Mozilla/5.0`).




it was a Mozilla/5.0`)

3. What is the name of the malicious web shell uploaded? → Investigate suspicious POST requests or payloads.

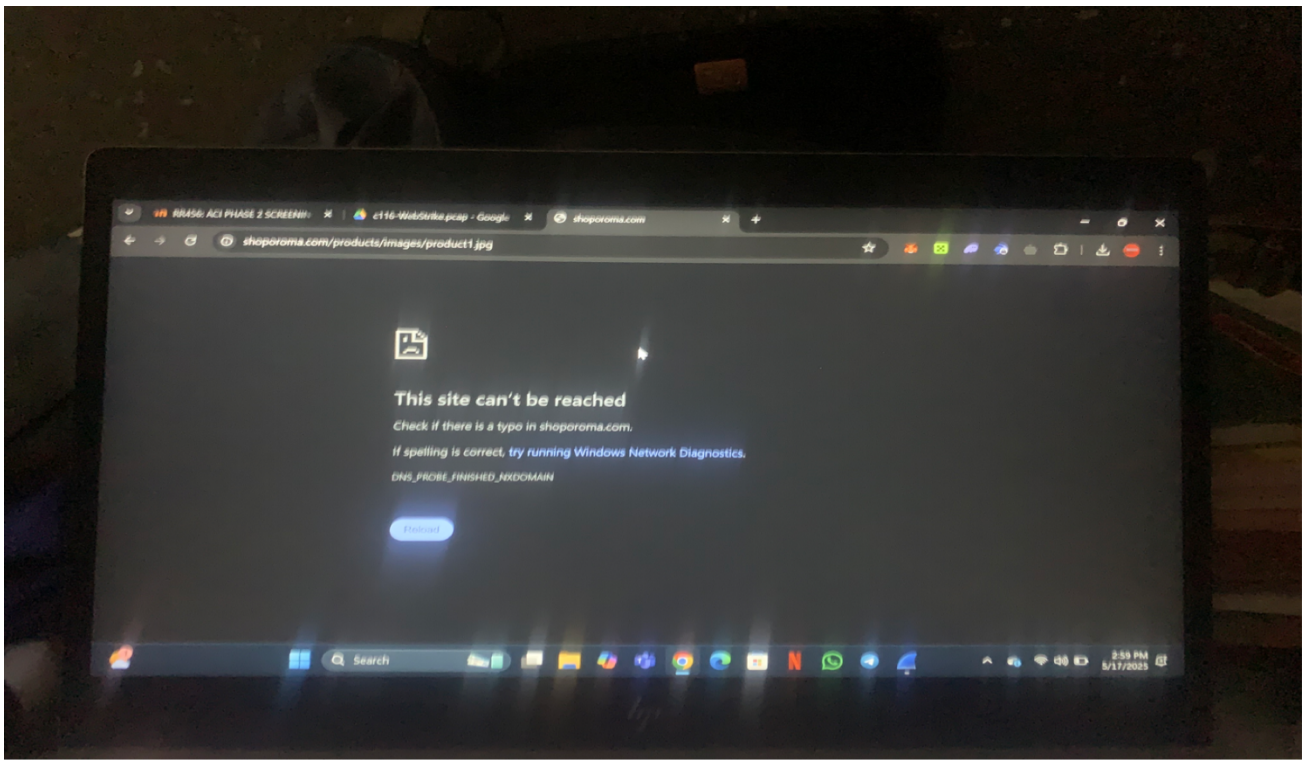
How it is found

The screenshot shows a Wireshark capture with HTTP traffic. The suspicious POST request in frame 55 (highlighted) involves a payload being uploaded to `http://shoporoma.com/reviews/upload.php`. The payload details are visible in the "Hypertext Transfer Protocol" section:

- The POST request is uploading a file named "products/product1.jpg"
- However, the presence of PHP-related context (`upload.php`) and the nature of the request suggest this could be a disguised web shell, a common tactic for malicious uploads.

Given the context of a potential malicious web shell, the file "product1.jpg" might not be a legitimate image but rather a PHP script disguised as an image file (e.g., embedding PHP code within a file that appears to be a JPG). This is a typical method for uploading web shells to gain unauthorized access to a server.

So, the name of the potentially malicious web shell uploaded is "product1.jpg", though its true nature would need further analysis (e.g., examining the file's contents for PHP code), Since the picture needs more insight to be checked.



4. Which server directory was used for file uploads? → Analyze the traffic to determine the upload path.

How It Is Found

The Wireshark capture shows a suspicious POST request in frame 55, where a file is being uploaded to the server. The relevant HTTP details are:

- The request is a POST to `http://shoporoma.com/reviews/upload.php`.
- The "Referer" header confirms the same path: `http://shoporoma.com/reviews/`.

From this, the server directory used for file uploads is `/reviews/`. The file `upload.php` within this directory handles the upload process, indicating that `/reviews/` is the target directory for the uploaded file.

5. What port did the attacker ' s system use for outbound communication? → Look for indications of data exfiltration attempts.

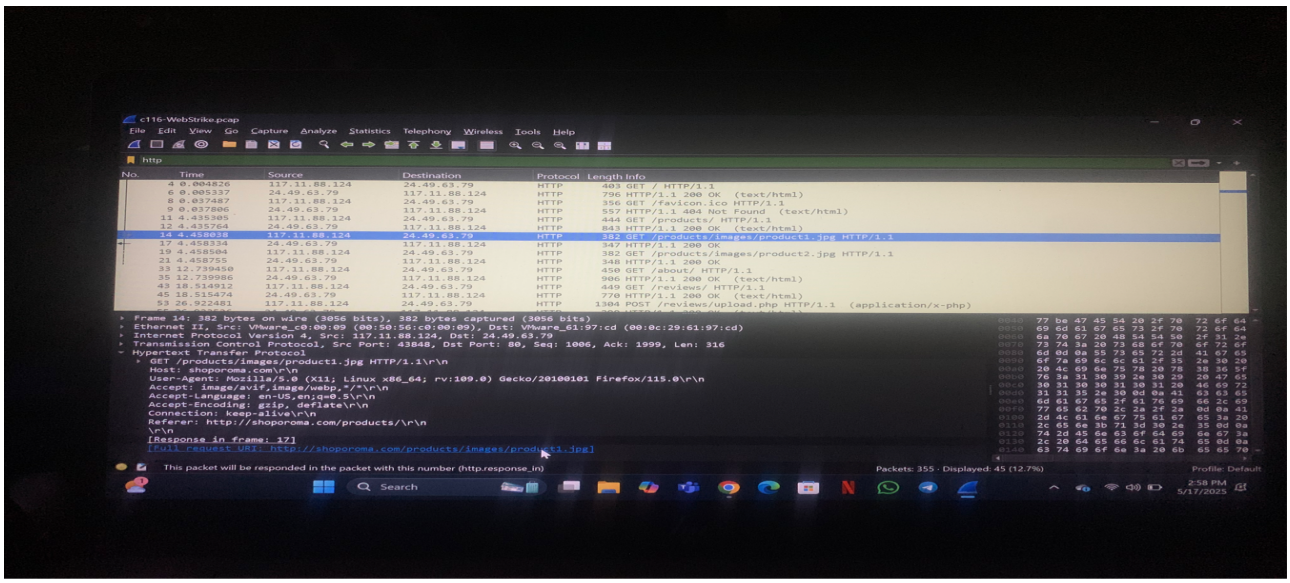
How It Is Found

In the Wireshark capture, the suspicious POST request in frame 55 is a potential data exfiltration attempt, as it involves uploading a file (`product1.jpg`) to `http://shoporoma.com/reviews/upload.php`. To identify the port used by the attacker's system for outbound communication, we need to look at the source port in the TCP layer of this frame.

From the "Transmission Control Protocol" section in frame 55:

- Src Port: 43848

This indicates that the attacker's system used port 43848 for outbound communication to the destination server. This source port is ephemeral, randomly assigned by the attacker's system for this connection.



6. Which file was targeted for exfiltration? → Identify the specific data object accessed or moved.

How It Is Found

In the Wireshark capture, the suspicious POST request in frame 55 indicates a potential data exfiltration attempt. The HTTP details show:

- The request is a POST to `http://shoporoma.com/reviews/upload.php`.
- The file being uploaded in this request is named "products/product1.jpg".

This file,"product1.jpg", is the specific data object being accessed or moved in this potential exfiltration attempt. While it appears to be an image file, the context of the upload to a PHP script (`upload.php`) suggests it might be a disguised web shell, as noted earlier. Regardless, this is the file targeted for exfiltration from the attacker's system to the server.