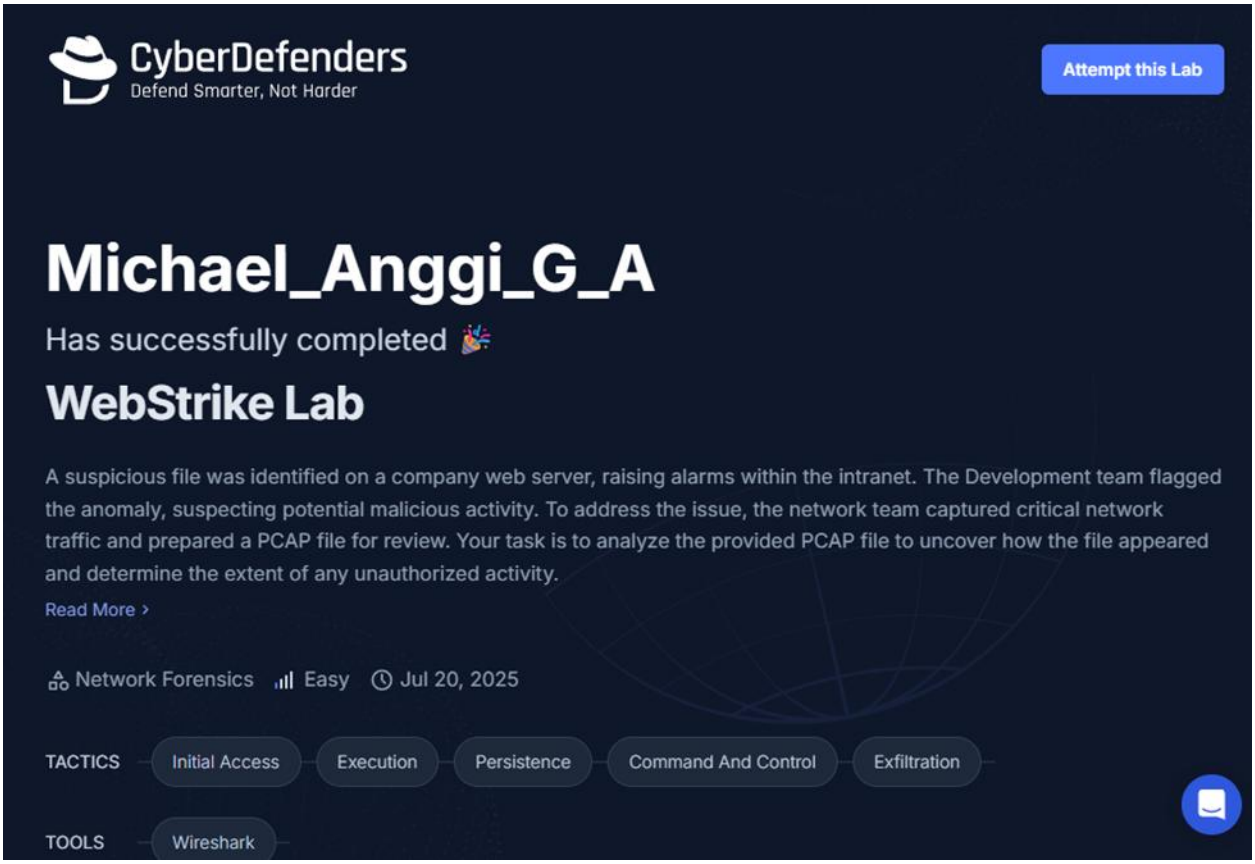
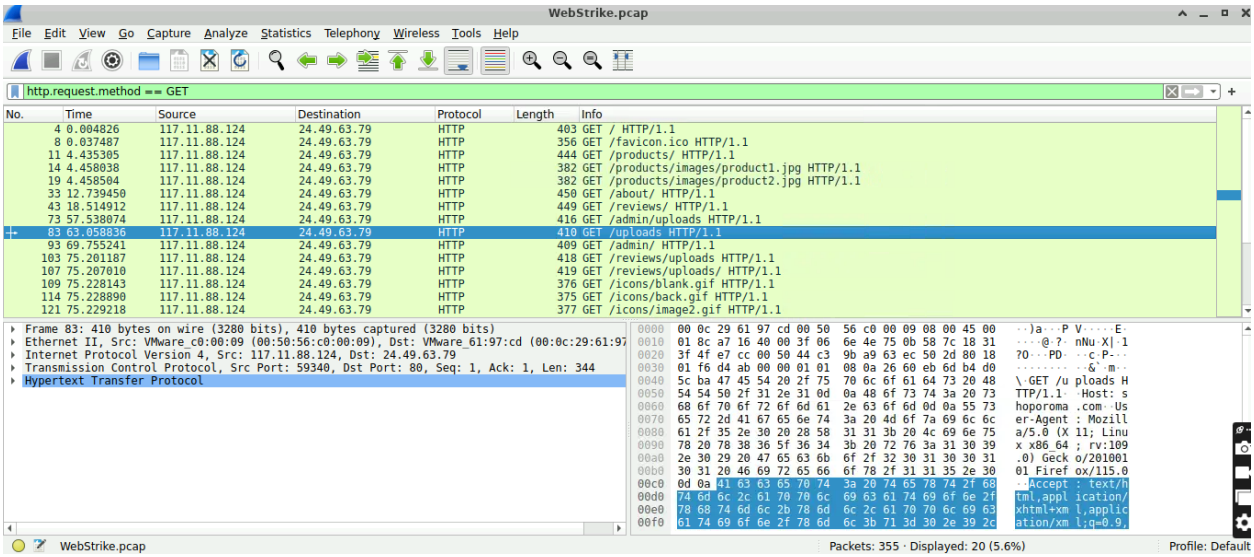
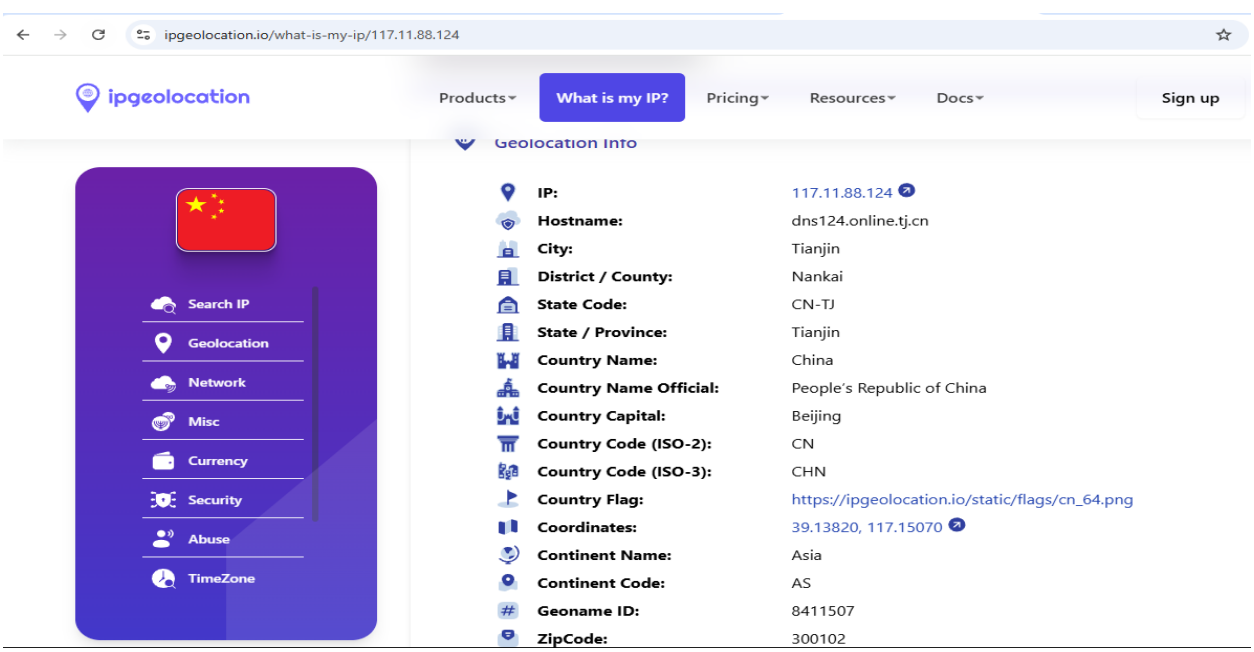
Web Shell Attack Investigation: PCAP Analysis & Threat Intelligence



Identifying the geographical origin of the attack facilitates the implementation of geo-blocking measures and the analysis of threat intelligence.

1. Look at the source and destination IP addresses in the PCAP file. Only one of them should correspond to an external entity. Have you identified which IP might be malicious?
2. Filter for HTTP GET requests using the filter: http.request.method == GET. Identify the source IP address associated with the request.
3. Use a geo-IP lookup service like https://ipgeolocation.io/ to determine the geographical location of the identified source IP.

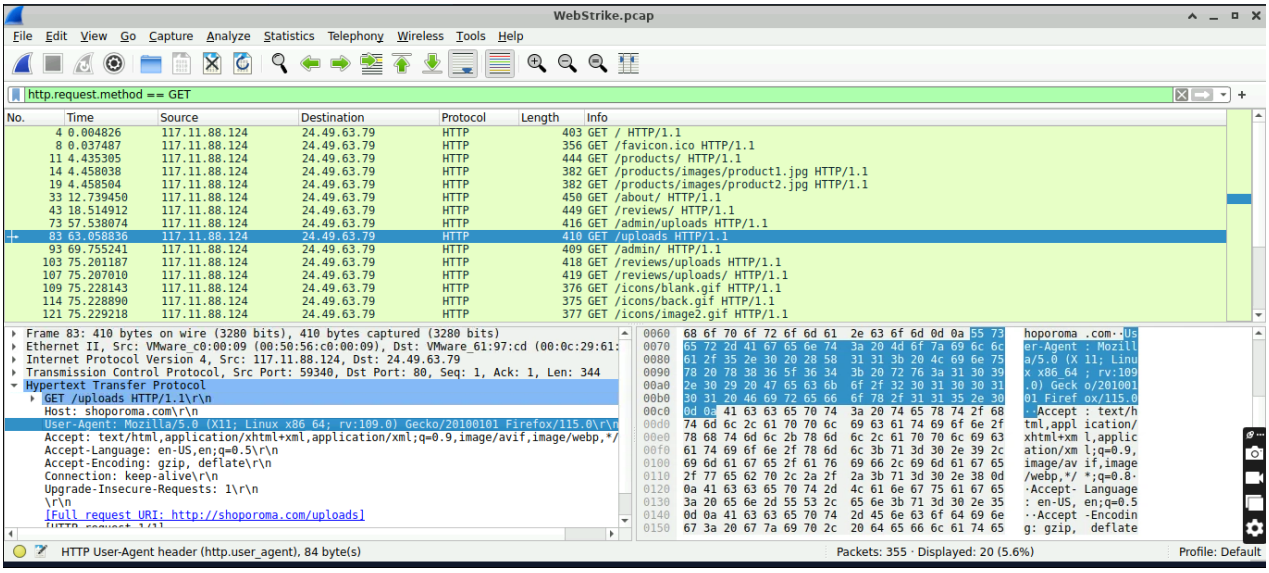




Knowing the attacker's User-Agent assists in creating robust filtering rules.

Analyze the HTTP packets in the PCAP. Check the details of requests that include User-Agent information.

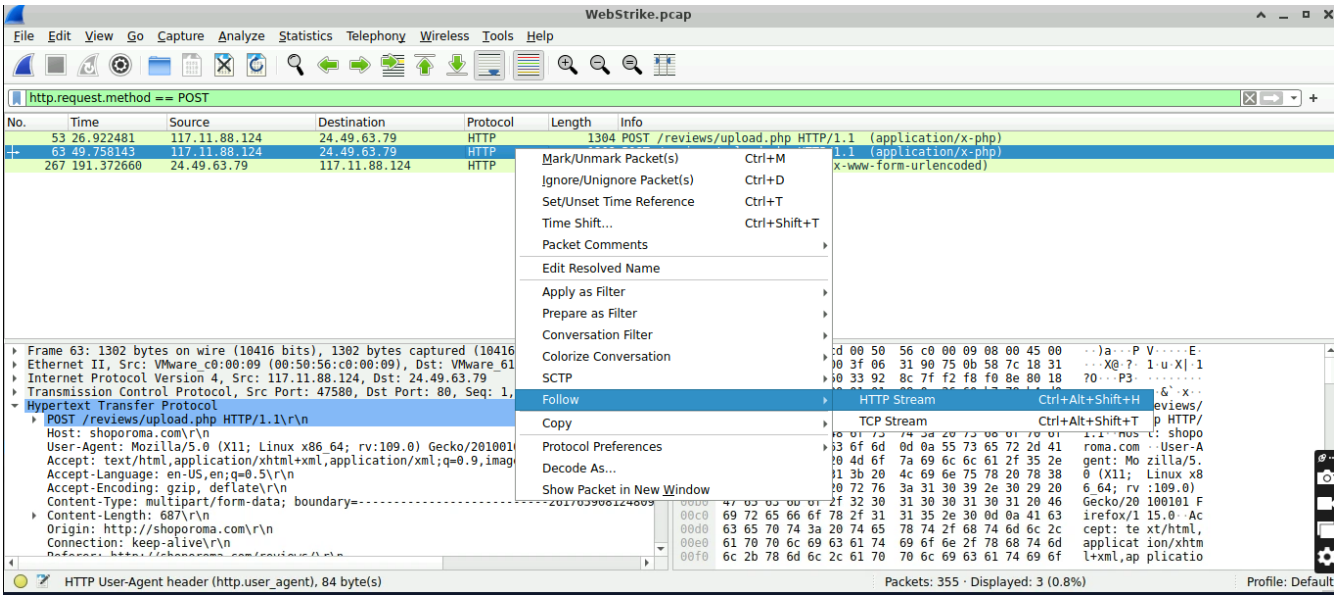
1. Filter for HTTP GET requests using http.request.method == GET. Expand the Hypertext Transfer Protocol section in a GET packet and find the User-Agent field to view the attacker's User-Agent string.

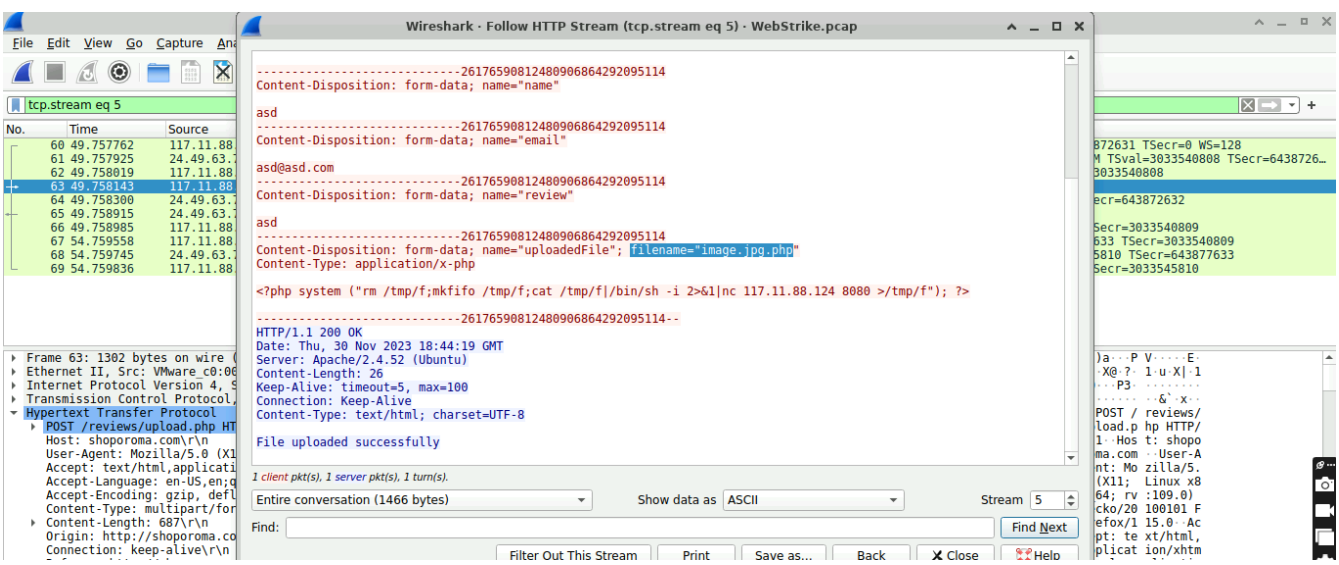


User Agent : *Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0*

A name of the malicious web shell that was successfully uploaded.

1. Focus on packets with HTTP POST requests, which are commonly used for file uploads.
2. Use the filter: http.request.method == POST. Then, analyze the HTTP POST packets by following the HTTP stream. To follow the HTTP stream, Right-click on the selected packet and select Follow > HTTP Stream to view the conversation.
3. After following the HTTP stream, observe the uploaded file name. Note that two upload attempts were made. Analyze the outcomes of each attempt to identify which file was successfully uploaded.

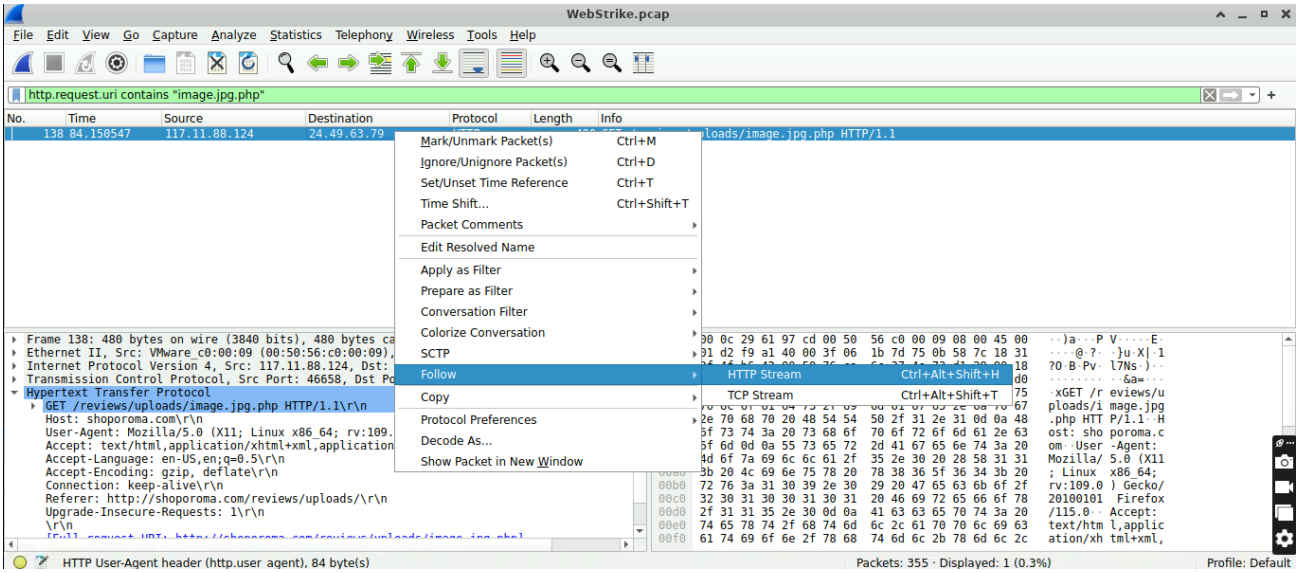


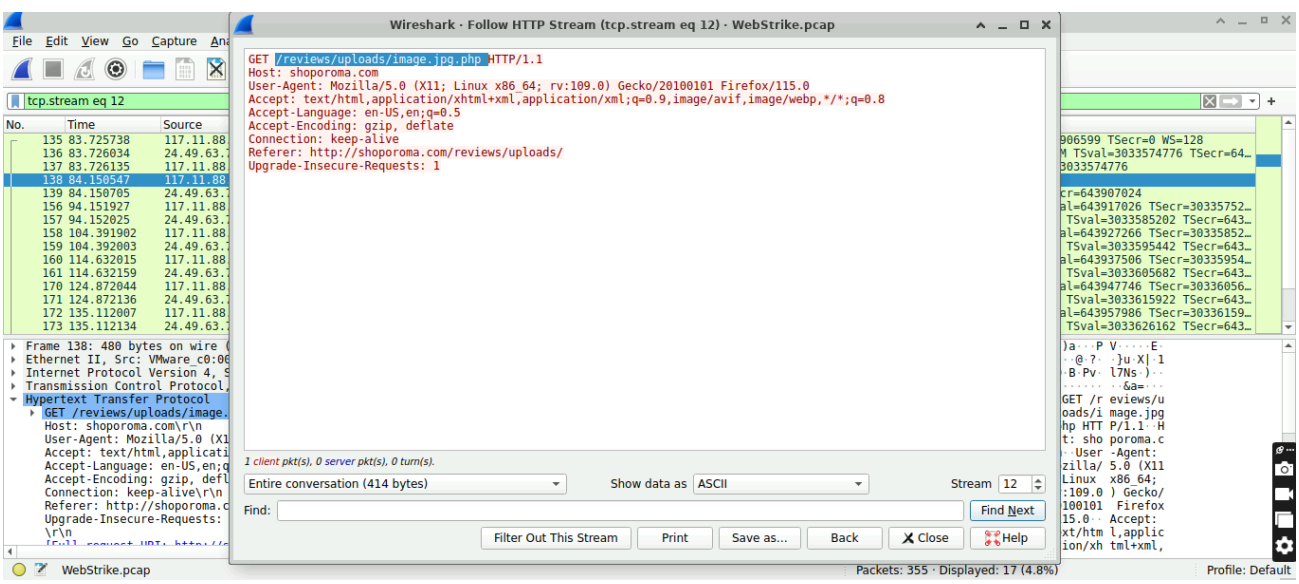


Filename : image.jpg.php

Identifying the directory where uploaded files are stored is crucial for locating the vulnerable page and removing any malicious files.

1. Look for the web shell script in HTTP POST requests to track its execution. Use the filter: http.request.method == POST
2. Apply the filter http.request.uri contains "<Uploaded_Filename>" and analyze the packet's HTTP URI to locate the upload directory.

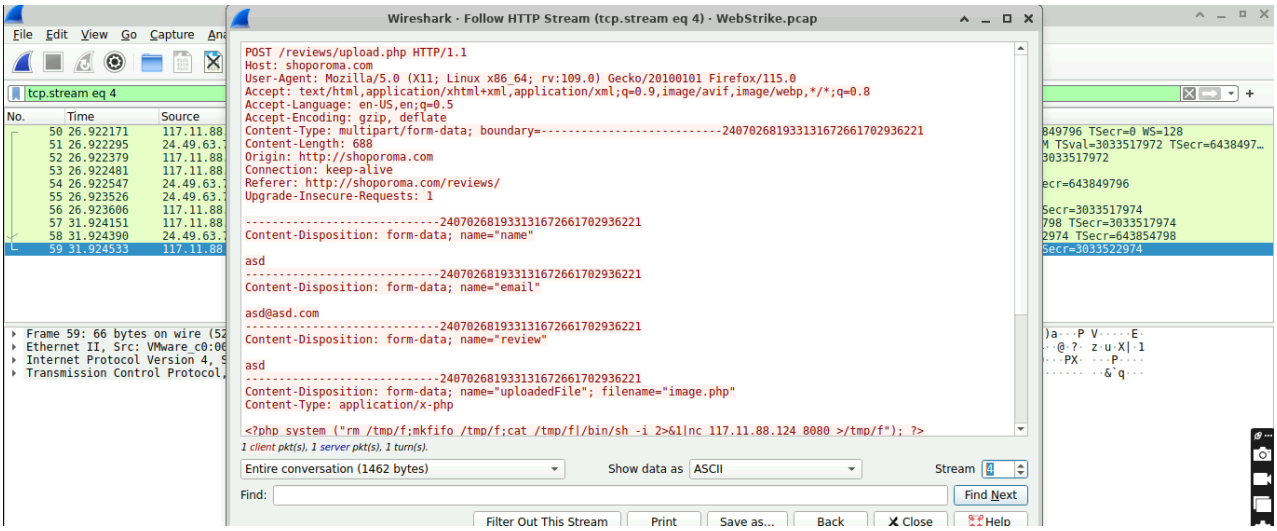


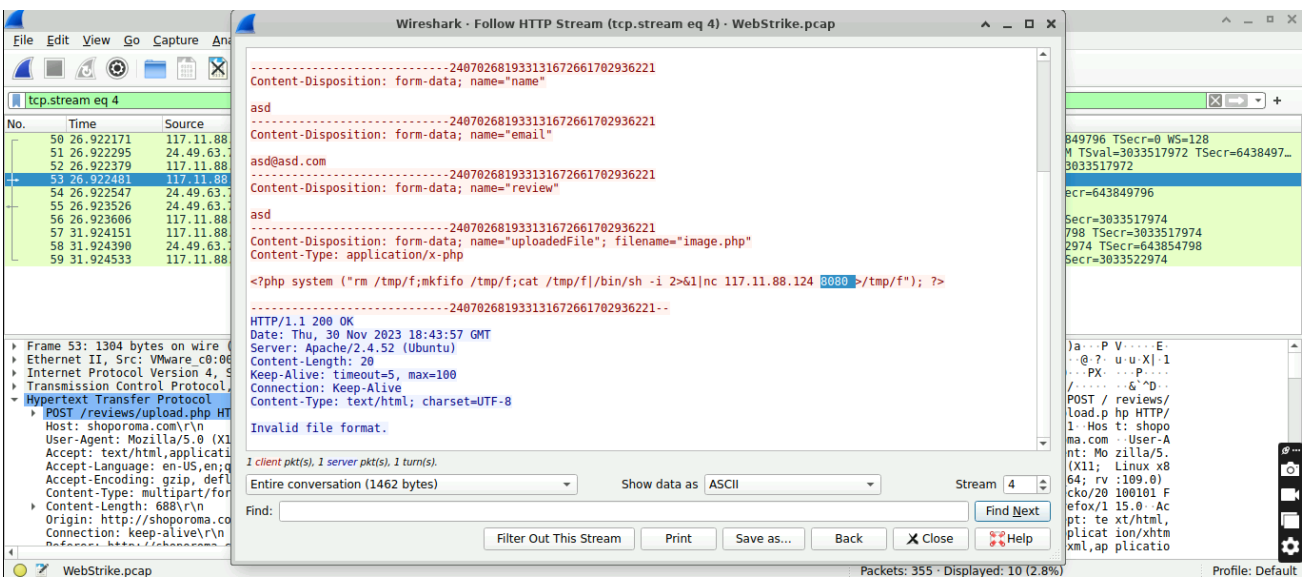


Directory : /reviews/uploads/

Which port, opened on the attacker's machine, was targeted by the malicious web shell for establishing unauthorized outbound communication?

1. Analyze the uploaded file by the attacker, focusing on POST HTTP requests.
2. Apply http.request.method == POST, right-click the POST packet, and select "Follow > HTTP Stream" to view the uploaded file content. Use tcp.stream eq 4.





Recognizing the significance of compromised data helps prioritize incident response actions. Which file was the attacker attempting to exfiltrate?

1. Check for evidence of data exfiltration in packets associated with outbound traffic. Look for POST requests or other transmissions from the server to the attacker's IP.
2. Apply the filter: tcp.dstport == <Detected_Port>. Follow the TCP stream to identify any file names or commands indicating data being exfiltrated.
3. Look for a curl -X POST command at the end of the TCP stream. The file name should be there.