

WebStrike Write-up

Hashim Alkhateeb

December 2025

1 Scenario

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.










Question 1: Identifying the geographical origin of the attack facilitates the implementation of geo-blocking measures and the analysis of threat intelligence. From which city did the attack originate?

To start, I opened the PCAP file with Wireshark and sorted by time. The PCAP file had two unique IP addresses so I had to distinguish who the attacker and the target were. 24.49.63.79 used port 80, so this is the web server and 117.11.80.124 is the client using random high ports. From the scenario, we know the suspicious file was found on the company's web server so 117.11.80.124 is the attacker. I looked up this IP on [iplocation.net](#) to identify the city.

Geolocation data from

IP2Location

Product: DB6, 2025-12-1

 IP ADDRESS: 117.11.88.124	 ISP: China Unicom Tianjin Province Network
 COUNTRY: China 	 ORGANIZATION: Not available
 REGION: Tianjin	 LATITUDE: 39.1422
 CITY: Tianjin	 LONGITUDE: 117.1761

Incorrect location?

[Contact IP2Location](#)


 [view map](#)

Figure 1: City name identified as Tainjin.

No.	Time	Source	Destination	Protocol	Length	Destination Port	Source Port	Info
1	0.000000	117.11.88.124	24.49.63.79	TCP	74	80	43848	43848 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=643822874 TSecr=0 WS=128
2	0.000139	24.49.63.79	117.11.88.124	TCP	74	43848	80	80 → 43848 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3033491050 TSecr=643822874
3	0.000243	117.11.88.124	24.49.63.79	TCP	66	80	43848	43848 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=643822874 TSecr=3033491050
4	0.000426	117.11.88.124	24.49.63.79	HTTP	403	80	43848	GET / HTTP/1.1
5	0.004936	24.49.63.79	117.11.88.124	TCP	66	43848	80	80 → 43848 [ACK] Seq=1 Ack=338 Win=64896 Len=0 TSval=3033491055 TSecr=643822879
6	0.005337	24.49.63.79	117.11.88.124	HTTP	796	43848	80	HTTP/1.1 200 OK (text/html)
7	0.005484	117.11.88.124	24.49.63.79	TCP	66	80	43848	43848 → 80 [ACK] Seq=338 Ack=731 Win=64128 Len=0 TSval=643822879 TSecr=3033491055
8	0.037487	117.11.88.124	24.49.63.79	HTTP	356	80	43848	GET /favicon.ico HTTP/1.1
9	0.037806	24.49.63.79	117.11.88.124	HTTP	557	43848	80	HTTP/1.1 404 Not Found (text/html)
10	0.083834	117.11.88.124	24.49.63.79	TCP	66	80	43848	43848 → 80 [ACK] Seq=628 Ack=1222 Win=64128 Len=0 TSval=643822958 TSecr=3033491088
11	4.435305	117.11.88.124	24.49.63.79	HTTP	444	80	43848	GET /products/ HTTP/1.1
12	4.435764	24.49.63.79	117.11.88.124	HTTP	843	43848	80	HTTP/1.1 200 OK (text/html)
13	4.435855	117.11.88.124	24.49.63.79	TCP	66	80	43848	43848 → 80 [ACK] Seq=1006 Ack=1999 Win=64128 Len=0 TSval=643827310 TSecr=3033495486
14	4.458038	117.11.88.124	24.49.63.79	HTTP	382	80	43848	GET /products/images/product1.jpg HTTP/1.1
15	4.458210	117.11.88.124	24.49.63.79	TCP	74	80	60240	60240 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=643827332 TSecr=0 WS=128
16	4.458304	24.49.63.79	117.11.88.124	TCP	74	60240	80	80 → 60240 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3033495508 TSecr=643827332
17	4.458334	24.49.63.79	117.11.88.124	HTTP	347	43848	80	HTTP/1.1 200 OK
18	4.458402	117.11.88.124	24.49.63.79	TCP	66	80	60240	60240 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=643827332 TSecr=3033495508
19	4.458504	117.11.88.124	24.49.63.79	HTTP	382	80	60240	GET /products/images/product2.jpg HTTP/1.1

Figure 2: Identify attacker and target IPs.

Question 2: Knowing the attacker's User-Agent assists in creating robust filtering rules. What's the attacker's Full User-Agent?

To solve this, I used CTRL + f to search the entirety of the PCAP file. I changed the search settings to *String* and specified I want to look within the *Packet details*. I then entered **agent** and clicked *find*. I found the *User-Agent* field which reveals the attacker is using a Linux OS and Firefox web browser.

The screenshot shows the Wireshark interface with a display filter set to 'agent'. The selected packet is an HTTP GET request from 117.11.88.124 to 24.49.63.79. The 'Packet details' pane is expanded to show the 'Hypertext Transfer Protocol' section, where the 'User-Agent' field is highlighted. The value of the User-Agent is 'Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0'. The 'Raw' pane shows the corresponding hex and ASCII data for the packet.

No.	Time	Source	Destination	Protocol	Length	Destination Port	Source Port	Info
51	26.922295	24.49.63.79	117.11.88.124	TCP	74	48796	80	80 → 48796 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3033517972 TSecr=643849796
52	26.922379	117.11.88.124	24.49.63.79	TCP	66	80	48796	48796 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=643849796 TSecr=3033517972
53	26.922447	117.11.88.124	24.49.63.79	HTTP	1504	80	48796	POST /reviews/upload.php HTTP/1.1 (application/x-www-form-urlencoded)

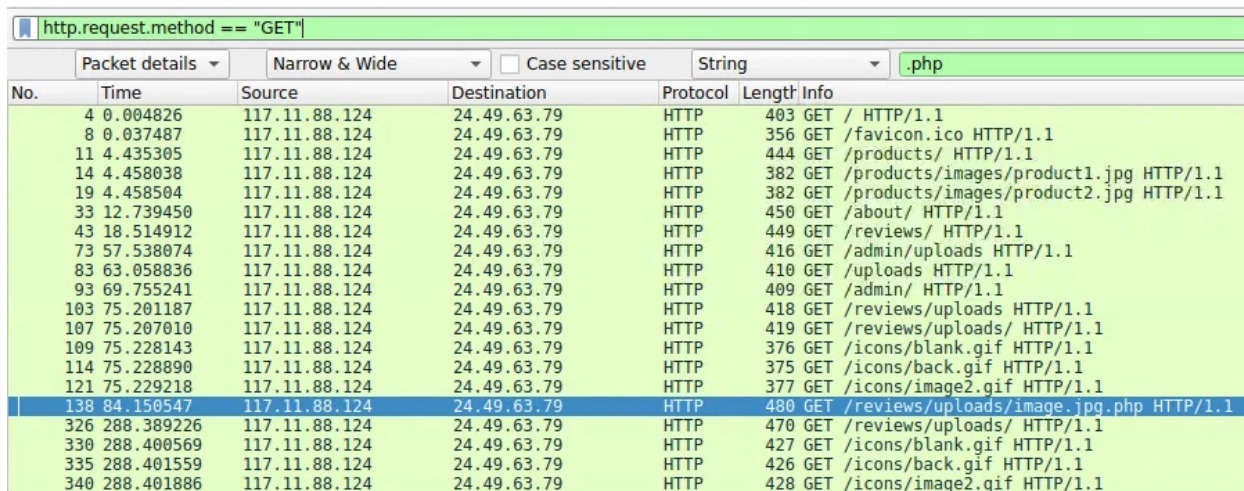
Packet details: Hypertext Transfer Protocol

- Host: shoporama.com/r/n
- User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0/r/n
- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8/r/n
- Accept-Language: en-US,en;q=0.5/r/n
- Accept-Encoding: gzip, deflate/r/n
- Content-Type: multipart/form-data; boundary=-----240702681933131672661702936221/r/n
- Origin: http://shoporama.com/r/n
- Connection: keep-alive/r/n
- Referer: http://shoporama.com/reviews/r/n
- Upgrade-Insecure-Requests: 1/r/n
- [Full request URI: http://shoporama.com/reviews/upload.php]
- [HTTP request 1/1]
- [Response in frame: 55]
- File Data: 680 bytes

Figure 3: User-agent identified as Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0

Question 3: We need to determine if any vulnerabilities were exploited. What is the name of the malicious web shell that was successfully uploaded?

I used the filter `http.request.method == "GET"` to get all http traffic using the GET method. As the results were quite small, I looked for the file name that appeared the most suspicious and it was clear as day.



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	63.058836	117.11.88.124	24.49.63.79	HTTP	410	GET /uploads HTTP/1.1
93	69.755241	117.11.88.124	24.49.63.79	HTTP	409	GET /admin/ HTTP/1.1
103	75.201187	117.11.88.124	24.49.63.79	HTTP	418	GET /reviews/uploads HTTP/1.1
107	75.207010	117.11.88.124	24.49.63.79	HTTP	419	GET /reviews/uploads/ HTTP/1.1
109	75.228143	117.11.88.124	24.49.63.79	HTTP	376	GET /icons/blank.gif HTTP/1.1
114	75.228890	117.11.88.124	24.49.63.79	HTTP	375	GET /icons/back.gif HTTP/1.1
121	75.229218	117.11.88.124	24.49.63.79	HTTP	377	GET /icons/image2.gif HTTP/1.1
138	84.150547	117.11.88.124	24.49.63.79	HTTP	480	GET /reviews/uploads/image.jpg.php HTTP/1.1
326	288.389226	117.11.88.124	24.49.63.79	HTTP	470	GET /reviews/uploads/ HTTP/1.1
330	288.400569	117.11.88.124	24.49.63.79	HTTP	427	GET /icons/blank.gif HTTP/1.1
335	288.401559	117.11.88.124	24.49.63.79	HTTP	426	GET /icons/back.gif HTTP/1.1
340	288.401886	117.11.88.124	24.49.63.79	HTTP	428	GET /icons/image2.gif HTTP/1.1

Figure 4: File Found: **image.jpg.php**

Question 4: Identifying the directory where uploaded files are stored is crucial for locating the vulnerable page and removing any malicious files. Which directory is used by the website to store the uploaded files?

I used the same method as question 3. The malicious file was located in `"/reviews/uploads/"`.

Question 5: Which port, opened on the attacker's machine, was targeted by the malicious web shell for establishing unauthorized outbound communication?

Since this questions asks for the port of the attacker, I used the filter `ip.src == 117.11.88.124` and filtered the source ports. I noticed the src ports were either high random ports or port **8080**.

ip.src == 117.11.88.124									
No.	Time	Source	Destination	Protocol	Length	Source Port	Destination Port	Info	
153	93.976854	117.11.88.124	24.49.63.79	TCP	66	8080	54448	8080 -> 54448 [ACK] Seq=17 Ack=209 Win=65824 Len=0 TSval=643916851 TSecr=3033585627	
151	93.975996	117.11.88.124	24.49.63.79	TCP	75	8080	54448	8080 -> 54448 [PSH, ACK] Seq=8 Ack=67 Win=65152 Len=0 TSval=643916850 TSecr=3033579963	
150	88.913133	117.11.88.124	24.49.63.79	TCP	66	8080	54448	8080 -> 54448 [ACK] Seq=8 Ack=67 Win=65152 Len=0 TSval=643911787 TSecr=3033579963	
148	88.912954	117.11.88.124	24.49.63.79	TCP	66	8080	54448	8080 -> 54448 [ACK] Seq=8 Ack=65 Win=65152 Len=0 TSval=643911787 TSecr=3033579963	
145	88.912834	117.11.88.124	24.49.63.79	TCP	73	8080	54448	8080 -> 54448 [PSH, ACK] Seq=1 Ack=56 Win=65152 Len=7 TSval=643911786 TSecr=3033575205	
144	84.154674	117.11.88.124	24.49.63.79	TCP	66	8080	54448	8080 -> 54448 [ACK] Seq=1 Ack=56 Win=65152 Len=0 TSval=643907028 TSecr=3033575205	
141	84.154398	117.11.88.124	24.49.63.79	TCP	74	8080	54448	8080 -> 54448 [SYN, ACK] Seq=0 Ack=1 Win=65168 Len=0 MSS=1460 SACK_PERM TSval=643907028 TSecr=3033575205	
28	9.459211	117.11.88.124	24.49.63.79	TCP	66	60240	80	60240 -> 80 [ACK] Seq=318 Ack=284 Win=64128 Len=0 TSval=643832333 TSecr=3033500509	
24	9.458918	117.11.88.124	24.49.63.79	TCP	66	60240	80	60240 -> 80 [FIN, ACK] Seq=317 Ack=283 Win=64128 Len=0 TSval=643832333 TSecr=3033495509	
22	4.458832	117.11.88.124	24.49.63.79	TCP	66	60240	80	60240 -> 80 [ACK] Seq=317 Ack=283 Win=64128 Len=0 TSval=643827333 TSecr=3033495509	
19	4.458504	117.11.88.124	24.49.63.79	HTTP	382	60240	80	GET /products/images/product2.jpg HTTP/1.1	
18	4.458402	117.11.88.124	24.49.63.79	TCP	66	60240	80	60240 -> 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=643827332 TSecr=3033495508	
15	4.458210	117.11.88.124	24.49.63.79	TCP	74	60240	80	60240 -> 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=643827332 TSecr=0 WS=128	
99	74.756399	117.11.88.124	24.49.63.79	TCP	66	59350	80	59350 -> 80 [ACK] Seq=345 Ack=494 Win=64128 Len=0 TSval=643897630 TSecr=3033568086	
97	74.756066	117.11.88.124	24.49.63.79	TCP	66	59350	80	59350 -> 80 [FIN, ACK] Seq=344 Ack=493 Win=64128 Len=0 TSval=643897630 TSecr=3033568086	
96	69.755692	117.11.88.124	24.49.63.79	TCP	66	59350	80	59350 -> 80 [ACK] Seq=344 Ack=493 Win=64128 Len=0 TSval=643892629 TSecr=3033568086	
93	69.755241	117.11.88.124	24.49.63.79	HTTP	409	59350	80	GET /admin/ HTTP/1.1	
92	68.863073	117.11.88.124	24.49.63.79	TCP	66	59350	80	59350 -> 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=643891738 TSecr=3033559914	
90	68.863612	117.11.88.124	24.49.63.79	TCP	74	59350	80	59350 -> 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=643891737 TSecr=0 WS=128	
89	68.060787	117.11.88.124	24.49.63.79	TCP	66	59340	80	59340 -> 80 [ACK] Seq=346 Ack=494 Win=64128 Len=0 TSval=643890935 TSecr=3033559111	
87	68.060534	117.11.88.124	24.49.63.79	TCP	66	59340	80	59340 -> 80 [FIN, ACK] Seq=345 Ack=493 Win=64128 Len=0 TSval=643890934 TSecr=3033554109	

Figure 5: Filtering for attacker IP address and source ports.

Question 6: Recognizing the significance of compromised data helps prioritize incident response actions. Which file was the attacker attempting to exfiltrate?

I used the filter "ip.src == 117.11.88.124 && tcp.srcport == 8080" to filter for attacker traffic using the port we found from question 5. I selected a packet and followed the tcp stream which showed the commands the attacker executed.

```

/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ uname -a
Linux ubuntu-virtual-machine 6.2.0-37-generic #38~22.
$ pwd
/var/www/html/reviews/uploads
$ ls /home
ubuntu
$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin

```

Figure 6: The attacker attempted to access /etc/passwd