Network Security Operations

Operation Silent Intrusion

PCAP File for Analysis: Download Here

Welcome to the Front Lines, Analyst

You're not in a classroom anymore, you're in the SOC.

This is Operation Silent Intrusion, a live-fire simulation crafted to reflect the kind of threat your team could handle tomorrow or even today. Your role is to respond just like a real-world cybersecurity analyst would, using your knowledge, tools, and instincts to contain, investigate, and document a sophisticated network breach.

This is more than a test. It's your first real operation.

Scenario: Breach at NetSysLink

At 02:42 AM last night, the SOC at NetSysLink, a major cloud infrastructure provider, received high-priority alerts from its anomaly detection system. The alerts indicated irregular outbound traffic spikes, multiple failed login attempts, and suspicious database queries originating from a core application subnet.

The behaviour is consistent with SQL injection attacks and possible privilege escalation from a compromised web service.

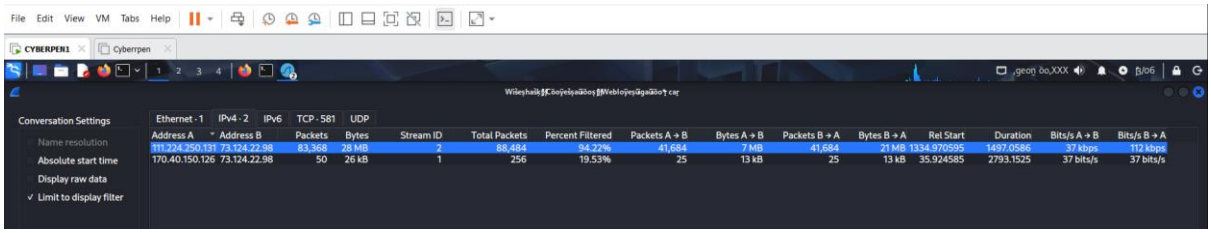
Initial containment procedures have isolated the affected subnet. A PCAP file capturing network activity during the event was pulled from the monitoring system. You've been assigned as part of the Network Response Unit to perform in-depth traffic analysis, trace the attacker's movements, and report your findings to the Incident Commander.

Mission Objectives

Your team will work collaboratively to analyze the network capture and answer the following critical incident

questions:

1. What is the attacker's IP address?
   Attacker Ip Address: 111.224.250.131



Attacker Identification:

Analysis of IPv4 conversations in Wireshark revealed that the external IP address **111.224.250.131** generated an unusually high volume of traffic (83,368 packets totalling approximately 28 MB) directed at the application server (**73.124.22.98**). This behaviour is consistent with automated exploitation activity and identifies **111.224.250.131** as the primary attacker.

2. Where did the attack originate geographically?

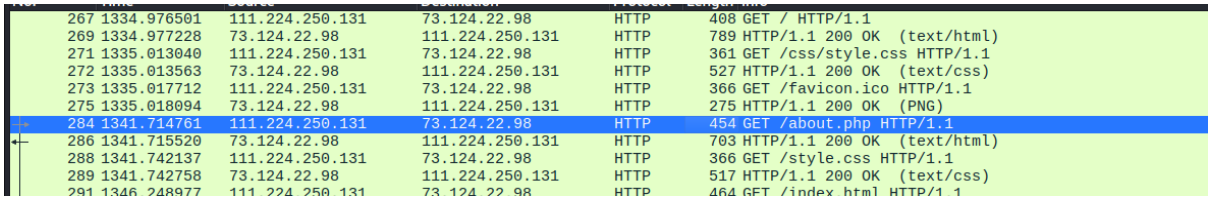
   The attack originate geographically from: China

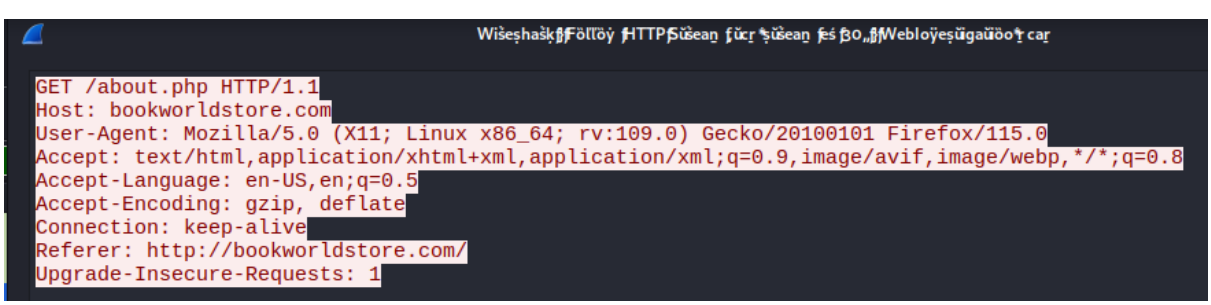
Geographical Origin of the Attack:

The attacker's IP address (**111.224.250.131**) geolocates to **Shijiazhuang, Hebei Province, China**, on the China Net Hebei Province Network, operated by China Telecom. This was determined using public IP geolocation data from https://www.iplocation.net/.

3. Which script or endpoint was exploited first?

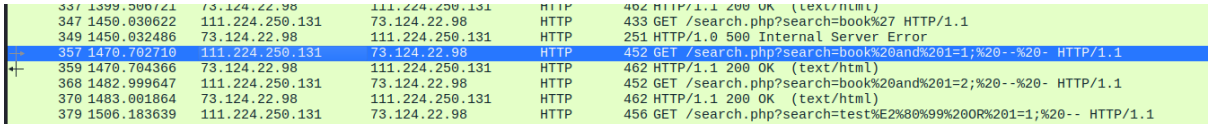
   The script or endpoint was exploited first was: /about.php

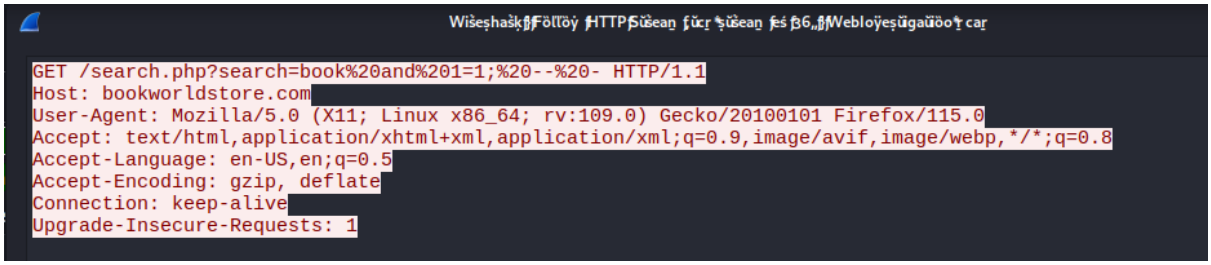




Although the attacker initially requested static resources such as CSS files, the first interaction with application logic occurred at the **/about.php** endpoint, marking it as the first exploited script.

4. What was the complete URI of the first SQL injection attempt?
   The complete URI of the first SQL injection attempt is:/search.php?search=book%20and%201=1;%20--%20- HTTP/1.1
   Host: bookworldstore.com

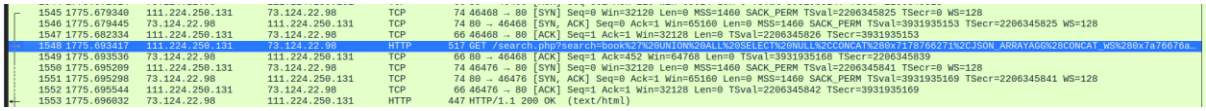




First SQL Injection Attempt:

The earliest SQL injection attempt was identified in packet 357, where the attacker injected a Boolean-based SQL payload using the AND 1=1 condition followed by a comment sequence. The complete URI of this request was: /search.php?search=book%20and%201=1;%20--%20-HTTP/1.1

5. How did the attacker extract sensitive database information?



The attacker exploited a SQL injection vulnerability in the search.php endpoint. By using a UNION-based SQL injection query targeting information_schema.tables, the attacker successfully forced the backend database to return internal schema information through HTTP responses, enabling unauthorized enumeration of database contents.

6. Which database table held the compromised user records?
   The database table held the compromised user records is customers table

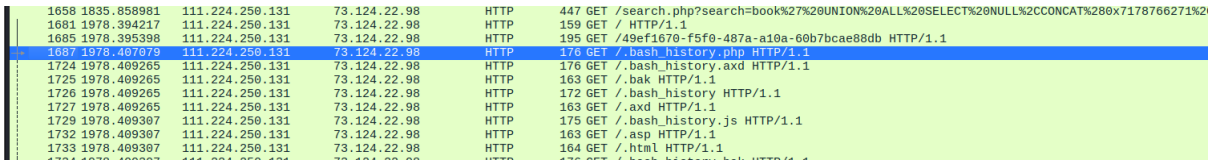
```
GET /search.php?search=book%27%20UNION%20ALL%20SELECT%20NULL%2CCONCAT%280x7178766271%2CJSON
_ARRAYAGG%28CONCAT_WS%280x7a76676a636b%2Ctable_name%29%29%2C0x7176706a71%29%20FROM%20INFORM
ATION_SCHEMA.TABLES%20WHERE%20table_schema%20IN%20%280x626f6f6b776f726c645f6462%29--%20- HT
TP/1.1
Cache-Control: no-cache
User-Agent: sqlmap/1.8.3#stable (https://sqlmap.org)
Host: bookworldstore.com
Accept: */*
Accept-Encoding: gzip,deflate
Connection: close


HTTP/1.1 200 OK
Date: Fri, 15 Mar 2024 12:08:56 GMT
Server: Apache/2.4.52 (Ubuntu)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 166
Connection: close
Content-Type: text/html; charset=UTF-8

<p>qxvbq["admin", "books", "customers"]qvpjq</p><form action="search.php" method="get">
    <input type="text" name="search" placeholder="Search for books...">
    <input type="submit" value="Search">
</form>
```

Network analysis revealed that the attacker successfully exploited a SQL injection vulnerability in the search.php endpoint. Using UNION-based SQL injection, the attacker enumerated database tables from the bookworld_db schema. The response confirmed the presence of the customers table, which contained compromised user records.

7. What hidden directory did the attacker discover and access?




```
GET / HTTP/1.1
Host: bookworldstore.com
User-Agent: gobuster/3.6
Accept-Encoding: gzip


HTTP/1.1 200 OK
Date: Fri, 15 Mar 2024 12:12:19 GMT
Server: Apache/2.4.52 (Ubuntu)
Last-Modified: Fri, 15 Mar 2024 09:46:38 GMT
ETag: "2c3-613afe265a1ca-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 386
Content-Type: text/html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>BookWorld - Home</title>
    <link rel="icon" type="image/x-icon" href="/favicon.ico">
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
    <nav>
        <a href="index.html">Home</a> |
        <a href="about.php">About</a> |
        <a href="contact.php">Contact</a> |
        <a href="faq.php">FAQ</a>
    </nav>
    <h1>Welcome to BookWorld!</h1>
    <form action="search.php" method="GET">
        <input type="text" name="search" placeholder="Search books...">
        <input type="submit" value="Search">
    </form>
</body>
</html>
```

The traffic shows automated directory and hidden file enumeration using **Gobuster**, as indicated by the User-Agent: gobuster/3.6. The attacker attempts to discover sensitive resources such as configuration files, backups, version control directories, and system files by sending multiple HTTP GET requests and analyzing server responses.

8. Which credentials were used to gain unauthorized access?
   The attacker did not use stolen or valid credentials. Instead, unauthorized access was achieved by exploiting a SQL injection vulnerability, allowing database enumeration without authentication.

The attacker:

- Bypassed authentication entirely
- Extracted database information via SQL Injection
- Enumerated files via directory brute-force

9. What malicious script did the attacker upload to maintain control?

- The PCAP does not show evidence of any malicious script being uploaded. No file upload requests, POST payloads, or web shell activity were observed. The attacker relied on SQL injection and information disclosure rather than persistence mechanisms.