**Incident Response Report for Premium House Lights Inc.**

**(Capstone Project)**

**James Kuzhilaparambil**

Contents

[Executive Summary 3](#_heading=h.gjdgxs)

[Incident Timeline 3](#_heading=h.30j0zll)

[Technical Analysis 3](#_heading=h.1fob9te)

[Incident Response 4](#_heading=h.3znysh7)

[Post-Incident Recommendations 4](#_heading=h.2et92p0)

[Conclusion 5](#_heading=h.tyjcwt)

[Appendix 5](#_heading=h.3dy6vkm)

[References 17](#_heading=h.1t3h5sf)

### Executive Summary

On 19/Feb/2022, Premium House Lights Inc. experienced a significant cybersecurity incident characterized by unauthorized access to its database and exfiltration of sensitive customer data. The attacker exploited vulnerabilities in the web server and database systems, ultimately gaining privileged access and transferring data offsite. This report outlines the incident timeline, provides a technical analysis of the attack, and offers recommendations for immediate remediation and long-term security enhancements.

Incident Timeline (See Appendix for details with screenshots)

1. Reconnaissance Phase: Attacker used a legitimate website crawler and conducted an automated scan, identifying potential vulnerabilities on the server.

2. Initial Access: Attacker uploaded `shell.php` via the `uploads` directory, enabling remote command execution.

3. System Enumeration: Attacker executed commands to gather system information, discovering network configurations and services running on the server.

4. Database Access: Attacker connected to the MySQL database, conducted searches, and managed to extract sensitive data after manipulating database tables.

5. Data Exfiltration: Backup of sensitive data was created and securely copied to an external server, followed by the deletion of evidence from the compromised system.

6. Exit: The attacker exited the system, covering their tracks.

### Technical Analysis

#### Attack Origin and Impact

The attacker utilized web server vulnerabilities to upload malicious files and gain unauthorized access to sensitive data. Evidence from server logs and database access records indicates the attacker successfully exfiltrated customer information, which could severely compromise customer trust and legal compliance.

#### Insight into How Systems Were Accessed

- The attacker first used automated reconnaissance tools to identify vulnerabilities in the web server.

- Successful exploitation was achieved by uploading a PHP web shell, allowing the attacker to execute arbitrary commands.

- Access to the MySQL database was gained through brute force attacks on credentials.

#### Outline of Weaknesses

1. \*\*Unsecured Upload Functionality\*\*: The presence of an `uploads` directory without sufficient validation allowed the attacker to upload malicious files.

2. \*\*Weak Password Policies\*\*: The attacker successfully brute-forced access due to inadequate password strength.

3. \*\*Lack of Network Segmentation\*\*: All critical servers were on the same VLAN, increasing risk exposure.

### Incident Response

#### Recommended Steps to Contain and Remediate the Incident

1. Immediate Containment:

- Isolate affected systems from the network to prevent further data exfiltration.

- Change passwords and revoke access for all accounts suspected to be compromised.

2. Remediation Steps:

- Conduct a full forensic investigation to assess the extent of the breach.

- Remove the uploaded web shell and any other unauthorized files.

- Patch vulnerabilities identified during the incident investigation.

- Implement stronger access controls and monitoring mechanisms.

### Post-Incident Recommendations

#### Future Protection Measures

1. Enhance Security Policies:

- Implement strict input validation and file type restrictions for the uploads directory.

- Adopt multifactor authentication (MFA) for database access.

2. Network Segmentation:

- Create separate VLANs for web, database, and file servers to limit lateral movement by potential attackers.

3. Regular Security Audits:

- Conduct regular vulnerability assessments and penetration testing to identify and remediate weaknesses proactively.

Rationale and Return on Investment

Implementing these recommendations aligns with the NIST Cybersecurity Framework, particularly the Identify, Protect, Detect, Respond, and Recover functions. By enhancing security posture, Premium House Lights can protect sensitive customer data, maintain customer trust, and mitigate potential legal ramifications of data breaches. The investment in security infrastructure and training will ultimately reduce the likelihood of future incidents, ensuring business continuity and protecting the company's reputation.

### Conclusion

This report serves as a comprehensive guide to understanding the incident and the necessary steps to enhance security. It is crucial for Premium House Lights to act swiftly to address vulnerabilities and bolster defenses against potential threats.

### Appendix

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Database access log

The following tables, columns, rows, user info and values from company database tables matches that to extortion email snippet of stolen data.

Webserver access log

A screenshot of a computer

Description automatically generated

Threat attacker using a legitimate website crawler that checks for technical issues to find vulnerable websites.

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Automated scan with many requests per second done as a reconnaissance.

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

Page relocated 301 code

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

Attacker checking for upload.php and get code 200. They check for uploads/randomfile1 and uploads/frand2 getting 404. Then switch to just uploads/ to get 200. 15 seconds later they send a new get request to uploads but instead of Mozilla browser, it is curl as user agent. Curl is a command line utility to retrieve and run retrieved file remotely. 9 seconds later they instead of requesting they posted to web server on uploads page a file called shell.php to be sent specifically to curl. Shell.php initiates a shell connection to allow remote command execution. Attacker can reverse shell to understand the configuration from the web server (134.122.33.221).

A computer screen shot of a computer

Description automatically generated

TCP stream for Packet user data including user agent, server info, content length, file creator, actual input in python.

A screen shot of a computer

Description automatically generated

Nmap port scans on different open and closed ports followed by port 80 (http) and 443 (https) until port scan for 136.243.111.17 for the first GET request that was 200 ok.

A screenshot of a computer

Description automatically generated

Attacker using shell interface, account name ([www.data](http://www.data)), python command to change command prompt to include the account, server name & current working directory followed by list command to see shell.php file they uploaded, grep search to see what packages installed on this machine specifically looking for nmap and check for addresses on the interface and network configuration to discover two network cards eth0 with public web server 134.122.33.221and eth1(internal private address).

A screenshot of a computer

Description automatically generated

Hacker tried stealth nmap scan but denied for lack of root user privileges. Then did regular nmap scan on above ip addresses to discover open ssh, telnet and http ports.

A computer screen with white text

Description automatically generated

Then used telnet to machine to discover server has ubuntu operating system and it is a database.

A screenshot of a computer

Description automatically generated

Attacker tried various login attempts until successful in last one.

A screenshot of a computer

Description automatically generated

They found Linux license version installed , current system timestamp setup, interface configuration and connections, number of pending updates, restart pending. Also, they ran a netstat command to know all the current connections to this device which includes port 22 (SSH) and port 23 from a web server telnet connection (10.10.1.2) that threat attacker is using.

A screenshot of a computer

Description automatically generated

The database access log shows attacker connecting to mysql database and searching through database contents with queries with little success.

A white background with black and red lines

Description automatically generatedThereafter attacker tries same on ph1 database with success, later resetting time zone to confuse the analysts, locks the database tables while session manipulation and read/search, extracts sensitive data and unlocks the table to restore to normal system condition.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

The attacker gained root user access to the system through exposed vulnerabilities found by reconnaissance (netstat) and gained access to database. Later attacker made a database backup from ph1 to ph1.db with root user password and got access to first 50 entries there. After running list command securely copied from ph1.db to [fierce@178.62.228.28:/tmp/ph1.db](about:blank) indicating data exfiltration. Later the attacker deleted ph1.db on the system to cover tracks and exited the system.

### References

*Security and privacy controls for information systems and organizations*. (2020). https://doi.org/10.6028/nist.sp.800-53r5

Wilson, M., & Hash, J. (2003). *Building an information technology security awareness and training program*. https://doi.org/10.6028/nist.sp.800-50

*MITRE ATT&CK®*. (n.d.). https://attack.mitre.org/

*CIS controls*. (n.d.). CIS. https://www.cisecurity.org/controls

Chakravarty, S. (2024, September 26). *CIS Controls v8 Released | SANS Institute*. https://www.sans.org/blog/cis-controls-v8/

*Wireshark • Learn*. (n.d.). Wireshark. https://www.wireshark.org/learn

*NetworkMiner - the NSM and network Forensics analysis tool* . (n.d.). Netresec. https://www.netresec.com/?page=NetworkMiner

Sitechecker. (n.d.). *Website Checker for Agencies & Multi-Product Companies ᐈ*. https://sitechecker.pro/

Flozz. (n.d.). *GitHub - flozz/p0wny-shell: Single-file PHP shell*. GitHub. https://github.com/flozz/p0wny-shell

*Pastebin.com - #1 paste tool since 2002!* (n.d.). Pastebin. <https://pastebin.com/>

MySQL. (2022). *MySQL Query log*.

SiteCheckerBotCrawler, & MSIE, 6.0. (2022). *Web Server Access Log*.