Incident Response Report
Extortion Email Incident Investigation
Premium House Lights

Name: Sumit Giri
Organization: Lighthouse Labs
Date: 02-12-2024

Table of Contents

1. **Executive Summary** 3
2. **Incident Timeline** 3
3. **Technical Analysis** 5
   - 3.1 **Attack Origin and Impact** 5
   - 3.2 **Insight into How Systems Were Accessed** 5
     - 3.2.1 **Initial Access** 5
     - 3.2.2 **Escalating to Automated Access Attempts** 6
     - 3.2.3 **Successful Exploit and Malicious Uploads** 6
     - 3.2.4 **Establishment of Reverse Shell and Privilege Escalation** 7
     - 3.2.5 **Network Reconnaissance and Scanning** 8
     - 3.2.6 **Interaction with Telnet Service** 8
     - 3.2.7 **MySQL Database Access** 9
     - 3.2.8 **Database Query Execution** 10
     - 3.2.9 **Database Dump** 11
     - 3.2.10 **File Examination and Exfiltration** 11
     - 3.2.11 **File Deletion and Termination** 11
   - 3.3 **Outline of Weaknesses that Allowed for This Incident to Occur** 12
     - 3.3.1 **Inadequate Web Application Security** 12
     - 3.3.2 **Lack of Privilege Management and Least Privilege Enforcement** 12
     - 3.3.3 **Weak Authentication Mechanisms** 13
     - 3.3.4 **Insecure Database Configuration** 13
     - 3.3.5 **Insufficient Network and Data Transfer Security** 13
     - 3.3.6 **Lack of Incident Detection and Response** 13
4. **Incident Response** 14
   - 4.1 **Preparation** 14
   - 4.2 **Detection and Analysis** 15
   - 4.3 **Containment, Eradication, and Recovery** 15
   - 4.4 **Post-Incident Review** 15
5. **Post-Incident Recommendations** 16
   - 5.1 **Protecting Against Future Attacks** 16
   - 5.2 **Recommended Adjustments to Security Policies** 16
   - 5.3 **Monitoring and Detection Enhancements** 17
   - 5.4 **Strengthening Vulnerability Management Practices** 17
6. **Appendix** 18
   - 6.1 **Additional Artifact Analysis** 18
7. **References** 20

1. Executive Summary

On **February 22, 2022**, Premium House Lights' Customer Support mailbox received an extortion email threatening the release of sensitive company data. The email, which prompted concerns about a possible breach, coincided with suspicious activity identified in system logs from **February 20, 2022**. The investigation into these anomalies revealed a targeted attack that successfully compromised the company's web server and database systems.

The attackers first conducted reconnaissance attempts on **February 20, 2022**, between **02:57:36 UTC** and **02:57:41 UTC**, scanning ports 443 and 80 for vulnerabilities. By **02:58:12 UTC**, the attackers escalated to more aggressive probing using an IP flagged as malicious, eventually deploying a web shell (shell.php) to gain remote access to the server at **02:59:04 UTC**. Over the course of the next few minutes, the attackers navigated through the system, escalating their privileges and identifying critical information.

By **03:00:18 UTC**, they had successfully logged into a low-privileged user account, discovered excessive MySQL privileges, and gained root-level access to the MySQL database. The attackers proceeded to extract sensitive customer data, including personal and purchase information, before creating a full backup of the database at **03:01:45 UTC**. This data was exfiltrated to a remote server by **03:02:26 UTC**, and the attackers then deleted the dumped file from the compromised server.

Although the attackers terminated their session at **03:02:44 UTC**, the exfiltration of customer data represents a significant risk, potentially impacting the company's reputation, customer trust, and compliance obligations.

This report outlines the actions taken in response to the incident, including securing the affected systems, analyzing logs for further signs of compromise, and restoring data from secure backups. It also provides long-term recommendations for improving system security and preventing future breaches, such as strengthening network defenses, enhancing employee training, and conducting regular security audits.

2. Incident Timeline

Time (UTC), 2022-02-20	Event
02:57:36 - 02:57:41	Initial reconnaissance attempts from IP `136.243.111.17` and `138.201.202.232` on port 443 (HTTPS) and port 80 (HTTP).
02:58:12	The attacker shifts to aggressive automated probing from IP `138.68.92.163` (flagged as malicious by VirusTotal).
02:58:40	The attacker finds the presence of the web shell (`shell.php`) in the /uploads directory using `HTTP GET` Request.
02:59:04	The attacker makes a `POST` request to `/uploads/shell.php` with a reverse shell payload, and the server responds by providing a web shell interface to execute commands remotely.
02:59:11	After gaining access via the web shell, the attacker runs the `whoami` command, confirming the user is `www-data`.

02:59:13	The attacker escalates the shell to a fully interactive one using `python -c 'import pty; pty.spawn("/bin/bash")'`.
02:59:16	The attacker lists files in the current directory using `ls -l`, revealing the contents of `/var/www/html/uploads`.
02:59:25	The attacker checks for the presence of nmap by running `dpkg -l | grep nmap`, confirming the installation of `nmap` and `nmap-common`.
02:59:29	Attacker runs ifconfig to discover network interfaces, revealing external IP (`134.122.33.221`) and internal IP (`10.10.1.2`).
02:59:37	Attacker initiates SYN scan using nmap to detect live hosts within internal network range `10.10.1.0/24` but lacks privileges for stealth scan.
02:59:45	Attacker performs Standard TCP Connect Scan, successfully identifying open ports on webserver (22/SSH, 80/HTTP) and database server (22/SSH, 23/Telnet).
02:59:55	Attacker connects to Telnet service on internal database server `10.10.1.3` (port 23).
02:59:56 - 03:00:11	Series of failed login attempts using usernames: `admin`, `administrator`, `phl`.
03:00:18	Attacker successfully logs in using `phl` account with password `phl123`.
03:00:27	Attacker runs `netstat -atunp` to identify active network connections, including on MySQL port 3306.
03:00:27	Attacker uses `sudo -l` to enumerate privileges for the `phl` user, revealing ability to execute MySQL-related commands as root without a password.
03:00:55	Attacker runs `sudo mysql -u root -p` to enter MySQL monitor interface.
03:00:58	Attacker runs show databases; revealing several databases, including `mysql`, `phl`, and `sys`.
03:01:02	Attacker switches to `mysql` system database to review user privileges and settings.
03:01:07	Attacker runs `show tables;` to identify key tables, including the `user` table.
03:01:10	Attacker retrieves data from `user` table, confirming existence of privileged users, including `root`.
03:01:13	Attacker switches to the `phl` database, containing customer data.
03:01:21	Attacker extracts sensitive customer data, including customer numbers, names, and purchase amounts.
03:01:31	Attacker initiates query: `SELECT * FROM customers LIMIT 5;`, extracting first five customer records.
03:01:45	Attacker performs database dump with `sudo mysqldump -u root -p phl > phl.db`, backing up the entire `phl` database.
03:01:49	Attacker uses file `phl.db` to examine file format, confirming it's a UTF-8 Unicode text file.
03:01:59	Attacker uses `head -50 phl.db` to review the first 50 lines of the dump, inspecting customer data.
03:02:17	Attacker uses `ls` command to list files in current directory, confirming presence of the `phl.db` dump file.
03:02:26	Attacker exfiltrates the data using SCP to transfer `phl.db` to remote server at IP `178.62.228.28`.

03:02:36	Attacker deletes the dumped file from the compromised server using `rm phl.db`.
03:02:38 - 03:02:44	Attacker terminates session, exiting MySQL at 03:02:38 UTC and server session at 03:02:41 UTC. The session is fully terminated at 03:02:44 UTC.

3. Technical Analysis

3.1 Attack Origin and Impact

The investigation has identified the origin of the attack as IP address 138.68.92.163, based in Germany, flagged as malicious by VirusTotal (VirusTotal, n.d.). This threat actor gained unauthorized access to Premium House Lights Inc.'s database and exfiltrated sensitive customer information. The stolen data, including customer details such as names, contact information, and payment history, was transferred to an external server controlled by the attacker. Subsequently, the company received a ransomware demand via email for 10 BTC, threatening to release this confidential data unless payment was made.

The analysis confirms that the data referenced in the ransom email matches the database structure and order exactly, which is not arranged alphanumerically. This alignment strongly indicates that the ransomware email sender is directly connected to the malicious attacker.

The impact of this breach is significant, given Premium House Lights Inc.'s reliance on customer trust and its high-end market positioning. Key implications include:

- **Financial Loss**: Potential ransom payment, customer compensation, and revenue loss due to reputational harm.
- **Reputational Damage**: Loss of customer trust may erode the company's loyal customer base and deter new clients.
- **Competitive Disadvantage**: Exposure of sensitive customer data could provide competitors with an edge or lead to customer migration.
- **Legal and Regulatory Risks**: The breach likely violates data protection regulations, such as the Personal Information Protection and Electronic Documents Act (PIPEDA), exposing the company to fines and legal action (Office of the Privacy Commissioner of Canada, 2021).

As an upscale brand, Premium House Lights Inc. must take swift and decisive action to address this breach, reinforce its security posture, and communicate transparently with its customers to minimize long-term consequences.

3.2 Insight into How Systems Were Accessed

3.2.1 Initial Access

The attack sequence commenced with an attempted reconnaissance phase (TA0043 | MITRE ATT&CK, n.d.) at **02:57:36 UTC**. The attacker initiated connection probes to the web server (IP: 10.10.1.2) and the database server (IP: 10.10.1.3) within the internal VLAN, using the external IP address **136.243.111.17**. The reconnaissance activity was observed from two

distinct IP addresses: **136.243.111.17** and **138.201.202.232** - both associated with the website crawling service **sitechecker.pro**, suggesting an automated scan or vulnerability assessment intended to identify weaknesses in the target system without raising alarms. Here is the associated screenshot of the webserver access log:

```
136.243.111.17 - - [19/Feb/2022:21:56:11 -0500] "GET / HTTP/1.1" 200 491 "-" "SiteCheckerBotCrawler/1.0 (+http://sitechecker.pro)"
138.201.202.232 - - [19/Feb/2022:21:56:13 -0500] "GET / HTTP/1.1" 200 491 "-" "SiteCheckerBotCrawler/1.0 (+http://sitechecker.pro)"
138.201.202.232 - - [19/Feb/2022:21:56:13 -0500] "GET / HTTP/1.1" 200 491 "-" "SiteCheckerBotCrawler/1.0 (+http://sitechecker.pro)"
138.201.202.232 - - [19/Feb/2022:21:56:13 -0500] "GET /?_escaped_fragment_= HTTP/1.1" 200 491 "-" "SiteCheckerBotCrawler/1.0 (+http://sitechecker.pro)"
138.201.202.232 - - [19/Feb/2022:21:56:13 -0500] "GET / HTTP/1.1" 200 491 "-" "SiteCheckerBotCrawler/1.0 (+http://sitechecker.pro)"
138.201.202.232 - - [19/Feb/2022:21:56:15 -0500] "GET / HTTP/1.1" 200 491 "-" "SiteCheckerBotCrawler/1.0 (+http://sitechecker.pro)"
138.201.202.232 - - [19/Feb/2022:21:56:17 -0500] "GET / HTTP/1.1" 200 491 "-" "SiteCheckerBotCrawler/1.0 (+http://sitechecker.pro)"
138.201.202.232 - - [19/Feb/2022:21:56:21 -0500] "GET / HTTP/1.1" 200 491 "-" "SiteCheckerBotCrawler/1.0 (+http://sitechecker.pro)"
136.243.111.17 - - [19/Feb/2022:21:57:37 -0500] "GET / HTTP/1.1" 200 491 "-" "SiteCheckerBotCrawler/1.0 (+http://sitechecker.pro)"
138.201.202.232 - - [19/Feb/2022:21:57:39 -0500] "GET / HTTP/1.1" 200 491 "-" "SiteCheckerBotCrawler/1.0 (+http://sitechecker.pro)"
138.201.202.232 - - [19/Feb/2022:21:57:40 -0500] "GET / HTTP/1.1" 200 491 "-" "SiteCheckerBotCrawler/1.0 (+http://sitechecker.pro)"
```

At **02:57:36 UTC**, the first connection attempt from IP **136.243.111.17** was made over HTTPS (port 443), but it was rejected by the server. However, the connection over HTTP (port 80) was successful. Simultaneously, IP **138.201.202.232** established a successful connection via a three-way handshake. These connections were brief and terminated at **02:57:41 UTC**, marking the end of the reconnaissance phase. Here is the associated Wireshark capture of the webserver:

124	2022-02-19 21:57:41.776921	138.201.202.232	134.122.33.221	TCP	68 39398 → 80 [FIN, ACK] Seq=361 Ack=492 Win=30336 Len=0 TSval=960935842 TSecr=3975401939
125	2022-02-19 21:57:41.777058	134.122.33.221	138.201.202.232	TCP	68 80 → 39398 [FIN, ACK] Seq=492 Ack=362 Win=64896 Len=0 TSval=3975403431 TSecr=960935842
126	2022-02-19 21:57:41.886522	138.201.202.232	134.122.33.221	TCP	68 39398 → 80 [ACK] Seq=362 Ack=493 Win=30336 Len=0 TSval=960935952 TSecr=3975403431

3.2.2 Escalating to Automated Access Attempts

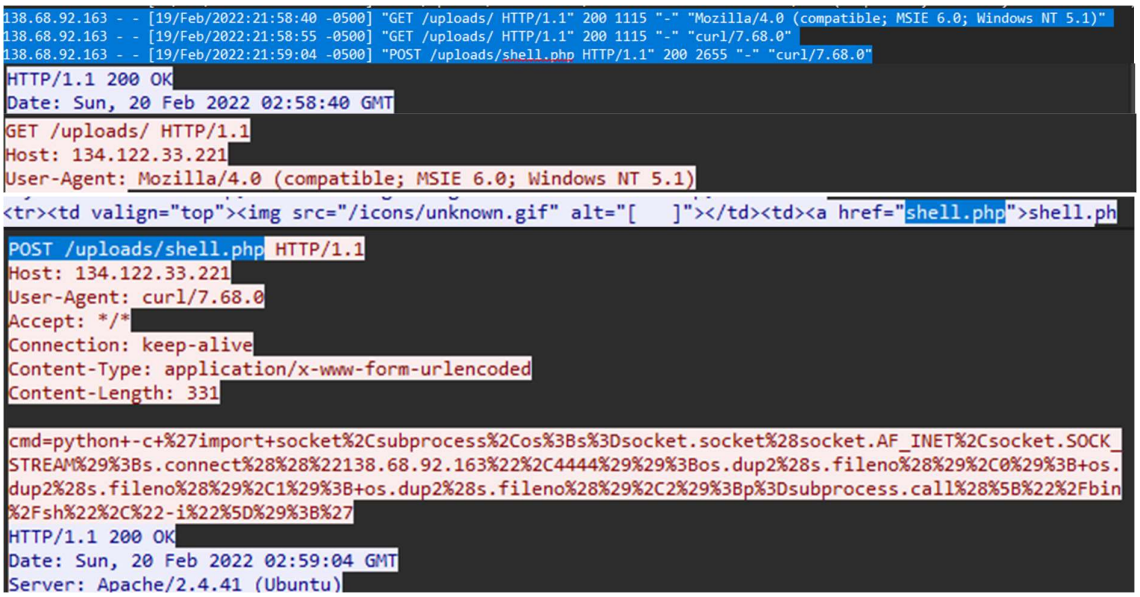
Following the reconnaissance phase, the attacker escalated their tactics to more aggressive probing (T1119 | MITRE ATT&CK, n.d.). At **02:58:32 UTC**, IP **138.68.92.163** initiated multiple high-frequency requests to the server, indicating an automated attack. Notably, **138.68.92.163** was flagged as malicious by a security vendor on VirusTotal, which strengthens the association of this IP address with the attack. Here are the screenshots from the server access logs and Wireshark captures:

```
138.68.92.163 - - [19/Feb/2022:21:58:22 -0500] "GET /randomfile1 HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:22 -0500] "GET /frand2 HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:22 -0500] "GET /index HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:22 -0500] "GET /archive HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:22 -0500] "GET /02 HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:22 -0500] "GET /register HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:22 -0500] "GET /en HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:22 -0500] "GET /forum HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:23 -0500] "GET /software HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
```

131	2022-02-19 21:58:12.322138	138.68.92.163	134.122.33.221	TCP	56 46086 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
132	2022-02-19 21:58:12.322185	134.122.33.221	138.68.92.163	TCP	56 80 → 46086 [RST] Seq=1 Win=0 Len=0
133	2022-02-19 21:58:12.322851	138.68.92.163	134.122.33.221	TCP	60 46086 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
134	2022-02-19 21:58:12.322861	134.122.33.221	138.68.92.163	TCP	56 443 → 46086 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
135	2022-02-19 21:58:12.558369	138.68.92.163	134.122.33.221	TCP	60 46342 → 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
136	2022-02-19 21:58:12.558369	138.68.92.163	134.122.33.221	TCP	60 46342 → 139 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
137	2022-02-19 21:58:12.558369	138.68.92.163	134.122.33.221	TCP	60 46342 → 587 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
138	2022-02-19 21:58:12.558369	138.68.92.163	134.122.33.221	TCP	60 46342 → 3389 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
139	2022-02-19 21:58:12.558369	138.68.92.163	134.122.33.221	TCP	60 46342 → 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

3.2.3 Successful Exploit and Malicious Uploads

At **02:58:40 UTC**, the attacker through a successful HTTP GET request finds the existence of the `shell.php`, which is commonly used for web shell deployment, within the /uploads/ directory successfully. At **02:59:04 UTC**, a POST request to /uploads/shell.php executed a reverse shell payload (TA1608 | MITRE ATT&CK, n.d.) using Python, designed to connect back to the attacker's system at IP **138.68.92.163** on port **4444**. The connection was successful, but no immediate output was captured, establishing the reverse shell. We have provided the evidence from the access log as well as the Wireshark captures corresponding to the above conclusion:



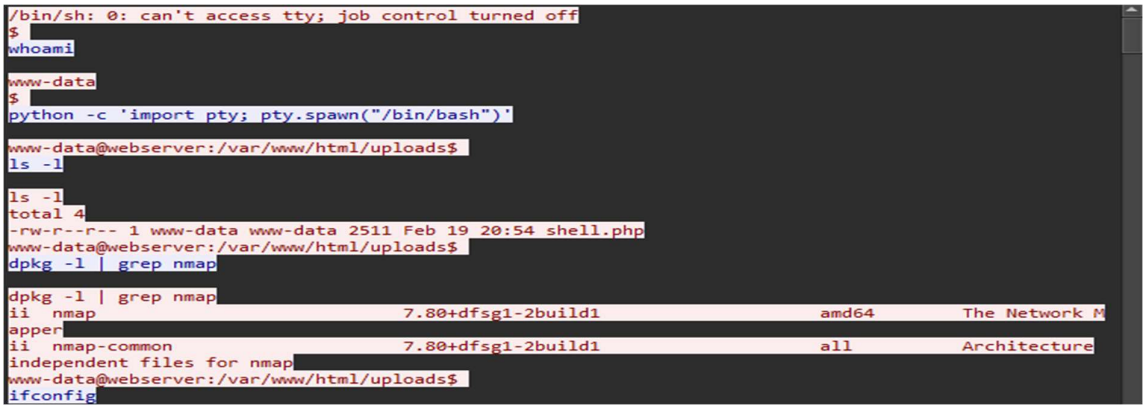
3.2.4 Establishment of Reverse Shell and Privilege Escalation

At **02:59:11 UTC**, the attacker executed the `whoami` command to identify the current user, revealing that they were operating as the `www-data user`, which is standard for web servers. At **02:59:13 UTC**, the attacker escalated their privilege (T1068 | MITRE ATT&CK, n.d.) to a fully interactive bash shell by running the command `pty.spawn("/bin/bash"),` enabling more control and the ability to execute commands with standard shell behavior.

The attacker continued reconnaissance activities, including:

- **02:59:16 UTC**: Listing contents of `/var/www/html/uploads` to confirm the presence of the uploaded `shell.php`.
- **02:59:25 UTC**: Checking the installed nmap version with `dpkg -l | grep nmap`, confirming version 7.80.
- **02:59:29 UTC**: Using `ifconfig` to reveal the server's network interfaces, discovering the external IP (134.122.33.221) and internal IP (10.10.1.2).

We found this evidence by applying the filter `tcp.stream eq 142` on the Wireshark capture of the webserver traffic. Here is the screenshot for the same:



```
ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 134.122.33.221  netmask 255.255.240.0  broadcast 134.122.47.255
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.1.2  netmask 255.255.255.0  broadcast 10.10.1.255
```

3.2.5 Network Reconnaissance and Scanning

At **02:59:37 UTC**, the attacker initiated a SYN scan using nmap to detect live hosts in the internal network range **10.10.1.0/24**, but was unable to perform the stealth scan due to insufficient privileges. Instead, the attacker performed a standard TCP Connect scan at **02:59:45 UTC**, successfully identifying open ports on both the web server (ports **22/SSH**, **80/HTTP**) and the database server (ports **22/SSH**, **23/Telnet**). This network reconnaissance (T1046 | MITRE ATT&CK, n.d.) revealed the web server (10.10.1.2) and database server (10.10.1.3) as the only active hosts. Here is the screenshot of the TCP stream of the webserver Wireshark capture:

```
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

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
```

3.2.6 Interaction with Telnet Service

At **02:59:55 UTC**, the attacker identified Telnet (port 23) on the internal database server (IP: 10.10.1.3) and successfully connected to the Telnet service which is a remote service (T1021 | MITRE ATT&CK, n.d.). Here is the screenshot of the TCP stream of the webserver Wireshark capture:

```
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:
```

A series of login attempts followed between **02:59:56 UTC** and **03:00:11 UTC**, resulting in several failed login attempts with usernames such as `admin`, `administrator`, and `phl`. At

03:00:18 UTC, the attacker successfully logged in using the **`phl`** account with the password **`phl123`**. Screenshot of Wireshark capture is given bellow:

```
Login incorrect
database login:
phl

phl
Password:
phl123


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

3.2.7 MySQL Database Access

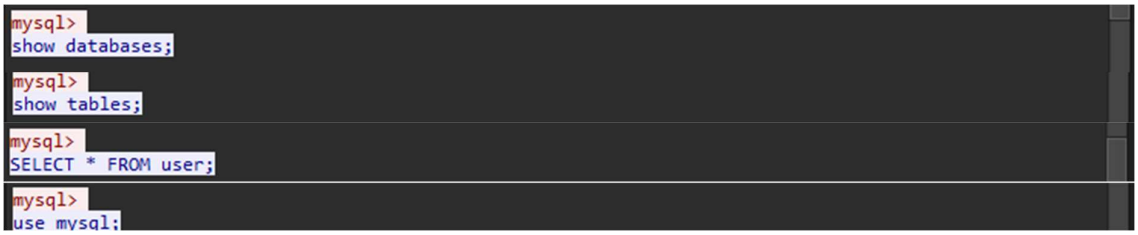
Upon gaining access to the phl account, the attacker proceeded with the discovery tactics (TA0007 | MITRE ATT&CK, n.d.):

- **03:00:27 UTC**: Ran `netstat -atunp` to identify active network connections, including those on ports **22 (SSH)**, **23 (Telnet)**, and **3306 (MySQL)**, indicating active connections with remote systems.
- **03:00:27 UTC**: Used the command `sudo -l` to enumerate the privileges for the `phl` user, revealing that `phl` had permission to execute MySQL-related commands (e.g., `mysql` and `mysqldump`) as root without a password.
- **03:00:55 UTC**: Successfully ran `sudo mysql -u root -p`, gaining access to the MySQL monitor interface.

```
phl@database:~$
netstat -atunp

netstat -atunp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:33060         0.0.0.0:*               LISTEN      -
tcp        0      0 147.182.157.9:22        142.112.199.247:42010   ESTABLISHED -
tcp        0      0 10.10.1.3:23            10.10.1.2:49522         ESTABLISHED -
tcp        0      0 10.10.1.3:23            10.10.1.2:43492         ESTABLISHED -
tcp        0      0 147.182.157.9:22        142.112.199.247:42024   ESTABLISHED -
tcp6       0      0 :::22                   :::*                    LISTEN      -
udp        0      0 127.0.0.53:53           0.0.0.0:*                           -
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.
```

The attacker continued to explore the database:

1. **03:00:58 UTC**: Executed `show databases;` to list the available databases, revealing `mysql`, `phl`, and `sys` databases.
2. **03:01:02 UTC**: Switched to the `mysql` system database to inspect MySQL user privileges and other global settings.
3. **03:01:07 UTC**: Ran `show tables;` to list the tables in the `mysql` database, identifying the `user` table which stores `user` privileges.
4. **03:01:10 UTC**: Retrieved data from the `user` table, confirming the existence of highly privileged users, including the `root` user with full database access.

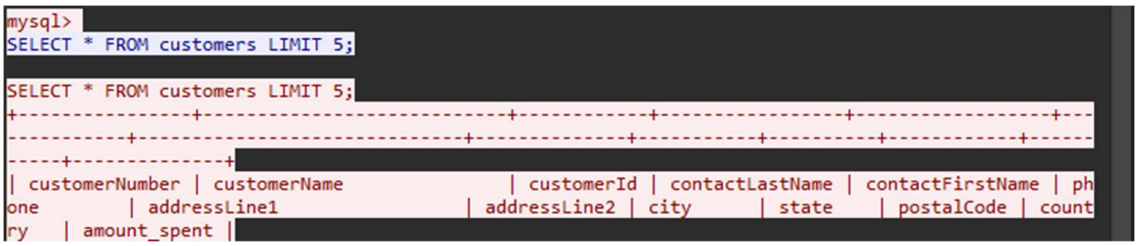


The attacker then focused on the phl database:

- **03:01:13 UTC**: Switched to the `phl` database, which contained a `customers` table.
- **03:01:21 UTC**: Extracted sensitive customer information from the `customers` table, including customer numbers, names, and purchase amounts.

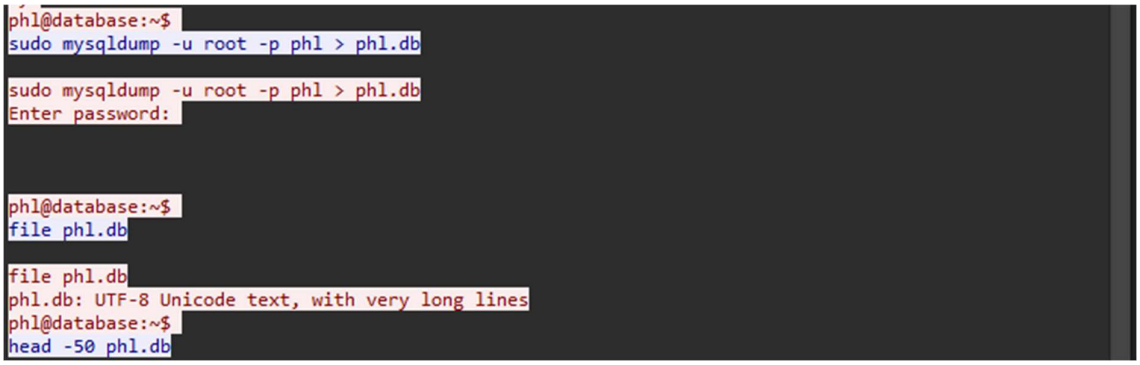
3.2.8 Database Query Execution

- **03:01:31 UTC**: Initiated a query using `SELECT * FROM customers LIMIT 5;`, retrieving the first five customer records, which included sensitive details such as customer identification numbers, names, contact details, and financial data.



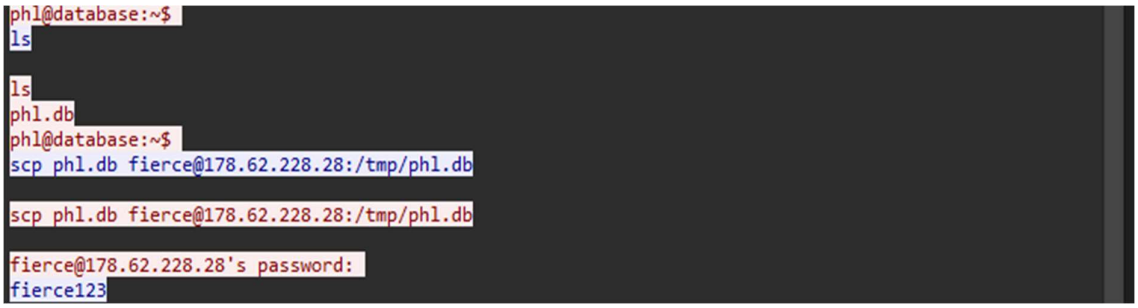
3.2.9 Database Dump

- **03:01:45 UTC**: Executed `sudo mysqldump -u root -p phl > phl.db` to back up the entire `phl` database, containing valuable customer data. The use of `sudo` indicated that the attacker had escalated privileges.
- **03:01:49 UTC**: Checked the file format using file `phl.db`, confirming that the dump was a "UTF-8 Unicode text" file.
- **03:01:59 UTC**: Reviewed the first 50 lines of the dump using `head -50 phl.db`, ensuring it contained valuable data.



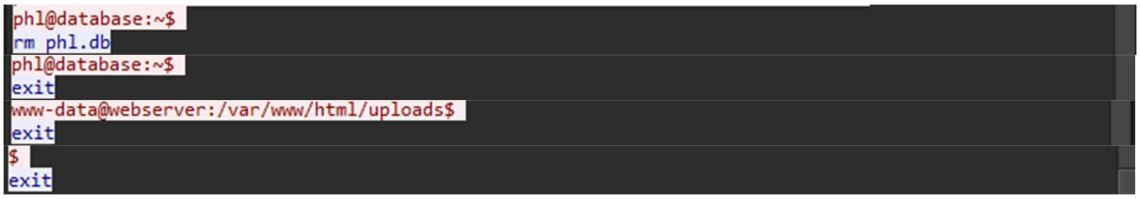
3.2.10 File Examination and Exfiltration

- **03:02:17 UTC**: Listed files in the current directory with ls to confirm the presence of the dumped file (`phl.db`).
- **03:02:26 UTC**: Exfiltrated the dumped file using SCP to a remote server (IP: **178.62.228.28**), logging into the server with credentials user: `fierce` and password `fierce123`. The file was stored in the `/tmp/` directory on the remote server.



3.2.11 File Deletion and Termination

- **03:02:36 UTC**: Deleted the dumped file from the compromised server using rm phl.db to cover up the breach.
- **03:02:38 UTC**: Exited the MySQL monitor.
- **03:02:41 UTC**: Terminated the server session.
- **03:02:44 UTC**: Completed the attack by fully terminating the session.



3.3 Outline of Weaknesses that Allowed for This Incident to Occur

Based on the incident timeline, several security weaknesses contributed to the successful data exfiltration and allowed the attacker to compromise the system. These weaknesses can be categorized as follows:

3.3.1 Inadequate Web Application Security

- **Web Shell Deployment**: The attacker identified and exploited the presence of a web shell (shell.php) in the /uploads directory. This vulnerability allowed the attacker to gain initial access to the server via a POST request that delivered a reverse shell payload. Web shells are often a result of inadequate security controls, such as improper input validation and file upload restrictions, which allow attackers to upload malicious files (T1505.003 | MITRE ATT&CK, n.d.).
- **Failure to Detect Reconnaissance**: The attacker conducted reconnaissance on ports 443 and 80 before exploiting the web shell. The absence of intrusion detection or prevention systems (IDS/IPS) to flag these reconnaissance attempts allowed the attacker to prepare for the attack undetected (M1031 | MITRE ATT&CK, n.d.).

3.3.2 Lack of Privilege Management and Least Privilege Enforcement

- **Weak Privilege Escalation and Excessive Privileges**: After gaining access through the web shell, the attacker was able to escalate the shell to a fully interactive one. Once inside, the attacker found that the phl user had sudo privileges to execute MySQL commands as the root user without requiring a password. This misconfiguration of user privileges allowed the attacker to escalate their access to the database and exfiltrate data. Adopting the principle of least privilege would have restricted the attacker's ability to perform such actions (CSF Tools, 2021a).
- **Improper User Access Controls**: The phl account had access to the MySQL root privileges, which should not have been granted. Proper user access control policies should enforce strict access rights and regularly audit user privileges to prevent unauthorized access (CWE-284: Improper Access Control, n.d.).

3.3.3 Weak Authentication Mechanisms

- **Unsecured Login Credentials**: The attacker successfully logged in to the Telnet service using the username phl and the password phl123. Using weak, easily guessable passwords, especially for privileged accounts, significantly reduces system security (T1552 | MITRE ATT&CK, n.d.).. Furthermore, Telnet, which transmits data in plaintext, should not have been used for remote access (Microsoft, n.d.). This allowed the attacker to easily capture login credentials during the attack. It is recommended to implement strong password policies and use more secure protocols such as SSH (Cisco Systems, Inc., n.d.).
- **Lack of Multi-Factor Authentication (MFA)**: There was no evidence of multi-factor authentication (MFA) implemented for remote or privileged access. MFA would have added an additional layer of security, significantly decreasing the likelihood of unauthorized access, even if passwords were compromised (CSF Tools, n.d.).

3.3.4 Insecure Database Configuration

- **Unencrypted Database Communication**: The attacker accessed sensitive customer data stored in the phl database without encountering any encryption, either for data at rest or in transit. This is a critical security flaw as sensitive information can be intercepted or exfiltrated if the database communication is not encrypted. Encrypting database connections and data storage is a standard industry practice to protect data confidentiality (Cloudflare, n.d.)
- **Lack of Database Activity Monitoring**: There were no apparent alerts or monitoring for the attacker's SQL queries, such as dumping the entire database. Without database activity monitoring, suspicious activities like large data exports can go unnoticed. Database activity monitoring (DAM) systems are essential for detecting unauthorized access and mitigating data exfiltration risks (Satori Cyber, 2022).

3.3.5 Insufficient Network and Data Transfer Security

- **Unrestricted Outbound File Transfers**: The attacker successfully exfiltrated the phl.db dump file via SCP to a remote server. The lack of outbound traffic filtering allowed the attacker to transfer the compromised data without interception. Implementing strict egress filtering policies that block unauthorized file transfers to external servers is a fundamental defense against data exfiltration (CSF Tools, 2021b).
- **Failure to Detect Data Exfiltration**: The data transfer was not flagged by any data loss prevention (DLP) systems, which should have monitored for large file transfers to untrusted locations. A well-configured DLP solution can prevent sensitive data from being exfiltrated, either through monitoring or blocking suspicious transfers (NIST CSRC, n.d.)

3.3.6 Lack of Incident Detection and Response

- **Delayed Response to Indicators of Compromise**: There was no sign of immediate detection or response to the attacker's actions, including file exfiltration and the removal of the dumped data. Real-time monitoring and automated incident response

mechanisms, such as alerting on abnormal file activities or network traffic, are critical to preventing or mitigating such attacks. An effective incident response plan should have been in place to handle these events swiftly (KPMG LLP, 2016)

- **Inadequate Session Termination and Cleanup**: While the attacker deleted the dumped database file after exfiltration, there was no indication of cleanup or monitoring for abnormal behavior during the session. Automated session termination or alerts when suspicious activity is detected could have minimized the damage.

4. Incident Response

This section outlines the measures taken to handle the security breach, using the NIST Cybersecurity Framework (CSF) as a guide. The incident response process is divided into four key phases: **Preparation**, **Detection and Analysis**, **Containment, Eradication, and Recovery**, and **Post-Incident Review**. Each identified vulnerability has been carefully analyzed, with corresponding actions implemented to mitigate risks and enhance security moving forward.

4.1 Preparation

Prior to the incident, the organization's cybersecurity measures were not robust enough to either prevent or detect the attack. Several vulnerabilities in the infrastructure stemmed from outdated configurations, ineffective security controls, and insufficient monitoring practices.

Key Weaknesses Identified:

Issue	Explanation
Unsafe File Upload Mechanism	A flawed upload function (upload.php) permitted files without appropriate validation, creating an entry point for malicious scripts.
Publicly Accessible Uploads Directory	Misconfigurations in the /uploads folder allowed public access and execution of uploaded files, potentially leading to remote exploitation.
Insecure Authentication Methods	The database server used weak and easily guessable credentials, with no restrictions on failed login attempts, exposing it to brute-force attacks.
Reliance on Vulnerable Protocols	Communication with the database server relied on Telnet, an insecure protocol that transmits data in plaintext, increasing exposure to interception.
Excessive User Privileges	A user account was granted unrestricted sudo access to sensitive database commands without requiring a password, increasing risks of privilege escalation.
Inadequate Network Segmentation	Lack of separation between the web server and internal resources allowed potential attackers to navigate laterally within the network.
Absence of Monitoring Tools	Failure to deploy monitoring solutions or maintain adequate logging made it difficult to detect unusual activities or track unauthorized access.
Unrestricted Outbound Traffic	No restrictions were placed on outbound connections, enabling attackers to exfiltrate data without triggering any alerts.
Outdated Security Policies	Existing policies failed to reflect modern threats or enforce secure practices, contributing to exploitable vulnerabilities.

No Incident Response Protocols	The organization lacked predefined response procedures, leading to delays in identifying and managing the breach effectively.

4.2 Detection and Analysis

The incident came to the organization's attention after the attacker sent a ransom email, threatening to expose stolen customer data. This external notification prompted an internal review to assess the breach. The security team conducted a thorough investigation to understand the scope of the attack and its impact. A step-by-step breakdown of the attack phases is available in Section 3.2.

4.3 Containment, Eradication, and Recovery

By the time the breach was detected, the attacker had already exited the system with exfiltrated data. Response efforts prioritized securing vulnerabilities to prevent further exploitation.

Key Actions Taken:

- **Containment:** While no immediate containment was required, entry points were secured to block similar attacks in the future.
- **Eradication:**
  - Reconfigured the upload.php endpoint with strict file type restrictions and input validation.
  - Identified and removed malicious scripts, such as shell.php.
- **Recovery:**
  - Verified the integrity of affected systems.
  - Strengthened overall system defenses against similar breaches.

Since no services were disrupted, recovery efforts focused primarily on security improvements rather than restoring operations.

4.4 Post-Incident Review

The breach underscored significant gaps in the organization’s security framework, including outdated practices, insufficient controls, and inadequate monitoring systems.

Planned Improvements:

- Update security policies to align with modern best practices.
- Introduce real-time monitoring tools and improve logging capabilities.
- Implement strict network segmentation to limit internal access.
- Launch organization-wide cybersecurity training to enhance employee awareness and readiness.

A detailed list of recommendations for long-term improvements is provided in the Post-Incident Recommendations section. These measures aim to build a more secure, resilient infrastructure and foster a culture of proactive cybersecurity.

5. Post-Incident Recommendations

This section outlines recommended actions for strengthening the organization's cybersecurity posture to prevent similar attacks in the future. Based on the findings from the incident investigation, these measures focus on enhancing security policies, improving detection and response capabilities, and addressing critical vulnerabilities.

5.1 Protecting Against Future Attacks

The company should implement a combination of technical and policy-driven adjustments to safeguard against future attacks. These proactive measures aim to address the vulnerabilities exploited in the recent breach and build a more resilient security framework.

- **Regular Penetration Testing**:
  Penetration testing should be conducted periodically to identify vulnerabilities before they can be exploited by attackers. This proactive approach will allow the company to address weaknesses in its infrastructure and applications in advance (PCI Security Standards Council, 2017).
- **Strengthen Authentication Mechanisms**:
  - Implement multi-factor authentication (MFA) across all critical systems (NIST, 2024)
  - Enforce strong password policies that mandate complex and unique passwords for all accounts (SANS Institute, 2022).
  - Apply account lockout policies to prevent brute-force attacks (OWASP, n.d.).
- **Improve Network Security**:
  - Deploy network segmentation with strict access control policies to prevent lateral movement within the network (NIST, 2022).
  - Implement robust firewalls and intrusion prevention systems (IPS) to detect and block malicious traffic.
  - Restrict unnecessary outbound traffic through egress filtering and data loss prevention (DLP) systems.
- **Secure Communication Protocols**:
  - Replace insecure protocols like Telnet with secure alternatives, such as SSH, for all remote access (Cisco Systems, Inc., n.d.).
  - Ensure encryption is used for all sensitive communications to prevent interception.

5.2 Recommended Adjustments to Security Policies

In addition to technical improvements, the organization must update and enforce its security policies to align with best practices and regulatory requirements. Regular updates to security policies will help maintain a strong defense against emerging threats.

- **Update and Enforce Security Policies**:
  - Review and revise existing security policies to address new threats and security trends.
  - Update password management, access control, and incident response policies to reflect best practices in the industry (SANS Institute, n.d.).

    - Conduct regular security policy reviews to ensure they remain effective and relevant (Bowen et al., 2006).
- **Incident Response Plan Development**:
    - Establish a formal incident response plan (IRP) that includes clearly defined roles, procedures, and communication strategies.
    - Ensure the plan is regularly tested and updated to reflect lessons learned from previous incidents.
- **Security Awareness Training**:
    - Implement ongoing security awareness programs for all employees to improve vigilance and preparedness for cyber threats.
    - Educate staff on the importance of safe password practices, phishing detection, and reporting suspicious activities (Canada, 2019).

5.3 Monitoring and Detection Enhancements

To ensure the company can detect and respond to threats in a timely manner, the following monitoring and detection strategies should be implemented:

- **Deploy Security Information and Event Management (SIEM) Tools**:
  SIEM systems will provide centralized logging and real-time analysis of security events across the network. This will allow the security team to detect suspicious activity and respond more quickly to incidents (NIST, n.d.).
- **Integrate Intrusion Detection and Response Systems**:
  Use Intrusion Detection Systems (IDS) and Endpoint Detection and Response (EDR) tools to monitor all endpoints for signs of compromise. These systems will provide real-time alerts and facilitate swift containment actions (Scarfone et al., 2007).
- **Enhance Logging and Monitoring Capabilities**:
  Ensure all critical systems are properly logged, with real-time monitoring in place to detect unauthorized access, data exfiltration, or anomalous behaviour.

5.4 Strengthening Vulnerability Management Practices

The following steps should be taken to address vulnerabilities identified in the incident investigation and to prevent similar security gaps from occurring in the future:

- **Secure File Upload Mechanisms**:
    - Implement strict input validation on file upload endpoints to prevent malicious file uploads (OWASP, n.d.).
    - Restrict file types to only necessary formats and disable script execution in upload directories.
    - Store uploaded files outside of the web root to prevent direct access.
- **Enhance Access Controls**:
    - Remove any passwordless sudo permissions for non-administrative users and ensure that all accounts follow the principle of least privilege.
    - Enforce multi-factor authentication for users with privileged access.
- **Regular System Updates and Patch Management**:
    - Ensure all systems, including servers, applications, and databases, are regularly updated with the latest security patches.

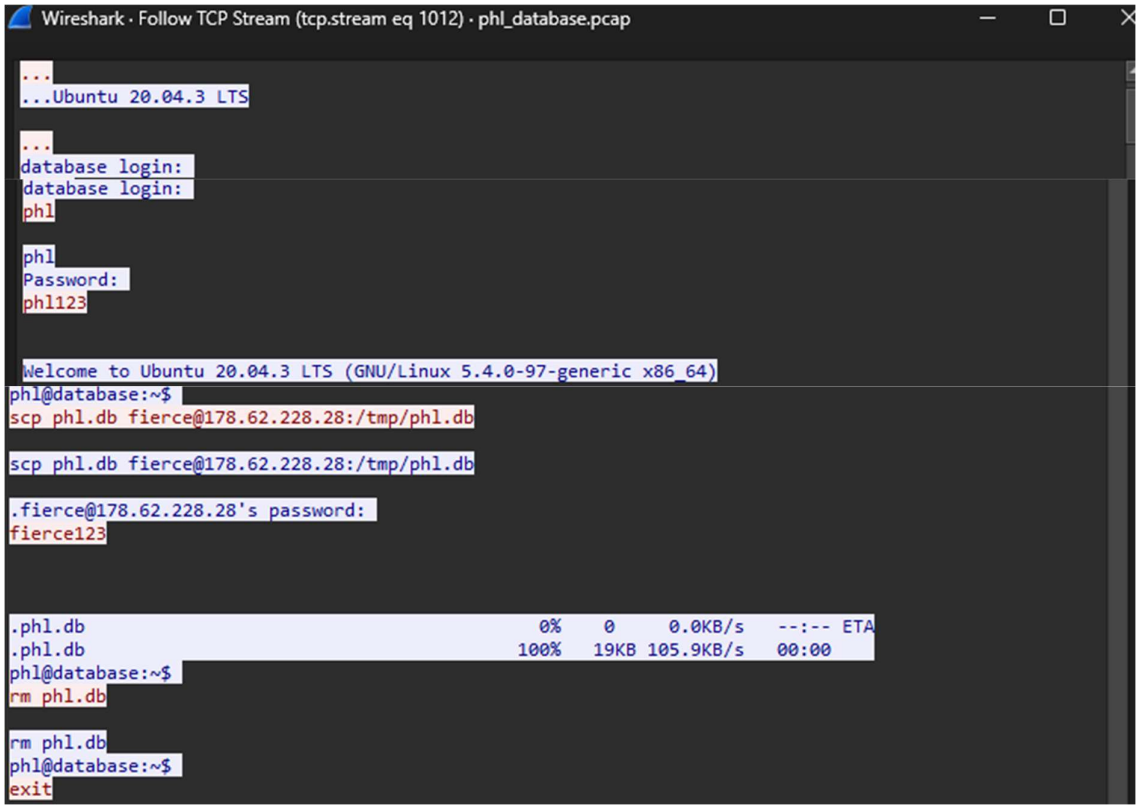
- Implement an automated patch management system to ensure that vulnerabilities are addressed promptly (Souppaya et al., 2022).

By implementing these recommendations, the company will strengthen its defenses against future attacks and create a more secure environment for both internal and external stakeholders. Regular evaluations and updates to the security framework, along with continuous monitoring, will ensure the organization remains resilient to emerging threats.

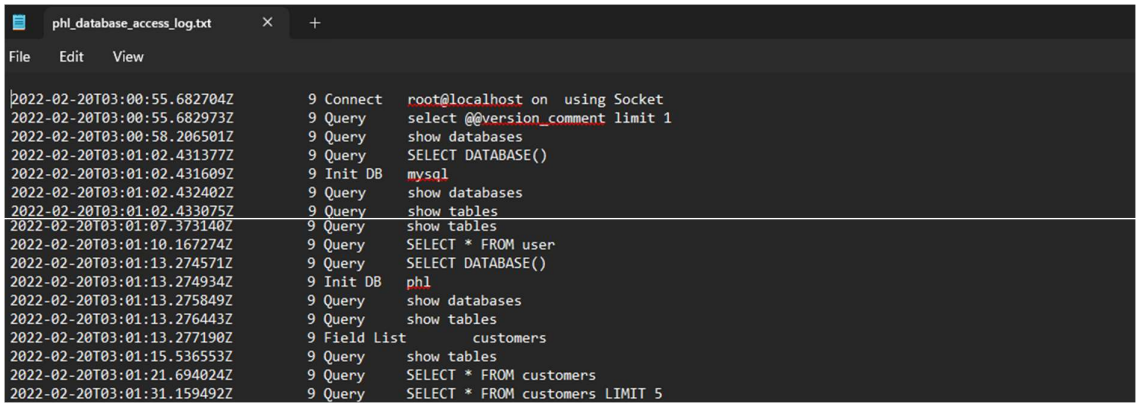
6. Appendix

6.1 Additional Artifact Analysis

- **Wireshark captures (phldatabase.pcap):** These are some of the captures from the database `pcap` file which corroborate the excat same timeline and actions of the threat actor as we have seen in the webserver `pcap` file:



- **Application Access Logs (phlaccesslog.txt):** We see the exact same activities on the database access log which confirms the outcome of our investigation that the database was accessed by the malicious actor.



- **Database Session Logs (phldatabaseshell.txt):** Here are more corroborative provided evidence of unauthorized access and commands executed within the database shell:

7. References

1. Bowen, P., Hash, J., Wilson, M., & National Institute of Standards and Technology. (2006). Information security handbook: A guide for managers (NIST Special Publication 800-100). National Institute of Standards and Technology. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-100.pdf
2. Canada Centre for Cyber Security. (2019, September 6). Provide employees with awareness training. Canadian Centre for Cyber Security. https://www.cyber.gc.ca/en/guidance/provide-employee-awareness-training
3. Cisco Systems, Inc. (n.d.). Configuring Secure Shell (SSH). In Cisco Secure Shell (SSH) (pp. 1–8). Retrieved December 2, 2024, from https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9400/software/release/16-9/configuration_guide/sec/b_169_sec_9400_cg/configuring_secure_shell__ssh_.pdf
4. Cloudflare. (n.d.). What is data at rest? Retrieved December 2, 2024, from https://www.cloudflare.com/learning/security/glossary/data-at-rest/
5. CSF Tools. (n.d.). 6.4: Require MFA for remote network access. Retrieved December 2, 2024, from https://csf.tools/reference/critical-security-controls/version-8/csc-6/csc-6-4/
6. CSF Tools. (2021a, March 5). AC-6: Least privilege. https://csf.tools/reference/nist-sp-800-53/r5/ac/ac-6/
7. CSF Tools. (2021b, March 5). Inbound and outbound communications traffic. CSF Tools - The Cybersecurity Framework for Humans. https://csf.tools/reference/nist-sp-800-53/r5/si/si-4/si-4-4/
8. Common Weakness Enumeration. (n.d.). CWE-284: Improper access control. https://cwe.mitre.org/data/definitions/284.html
9. KPMG LLP. (2016). 10 common cyber incident response mistakes. In Cyber insights for the federal government. https://assets.kpmg.com/content/dam/kpmg/pdf/2016/04/cyber-incident-response.pdf
10. Microsoft. (n.d.). MS09-042: Vulnerability in Telnet could allow remote code execution. Retrieved December 2, 2024, from https://support.microsoft.com/en-us/topic/ms09-042-vulnerability-in-telnet-could-allow-remote-code-execution-7d71e702-0539-73ab-dbbe-2ac5502c8420
11. MITRE ATT&CK®. (n.d.). Retrieved December 1, 2024, from https://attack.mitre.org/
12. NIST. (n.d.). Security information and event management (SIEM) tool. Retrieved December 2, 2024, from https://csrc.nist.gov/glossary/term/security_information_and_event_management_tool
13. NIST. (2024, March 12). Multi-factor authentication. https://www.nist.gov/itl/smallbusinesscyber/guidance-topic/multi-factor-authentication
14. NIST Computer Security Resource Center. (n.d.). Data loss prevention. Retrieved December 2, 2024, from https://csrc.nist.gov/glossary/term/data_loss_prevention
15. NIST Computer Security Resource Center. (2022, November 17). NIST publishes SP 800-215: Guide to a secure enterprise network landscape. https://csrc.nist.gov/News/2022/sp-800-215-secure-enterprise-network-landscape
16. Office of the Privacy Commissioner of Canada. (2021, December 8). The Personal Information Protection and Electronic Documents Act (PIPEDA). https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/

17. OWASP. (n.d.). File upload cheat sheet. https://cheatsheetseries.owasp.org/cheatsheets/File_Upload_Cheat_Sheet.html
18. OWASP Foundation. (n.d.). Blocking brute force attacks. Retrieved December 2, 2024, from https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks#:~:text=Locking%20Accounts,manually%20unlocked%20by%20an%20administrator
19. PCI Security Standards Council. (2017). Penetration testing guidance. https://listings.pcisecuritystandards.org/documents/Penetration-Testing-Guidance-v1_1.pdf
20. SANS Institute. (n.d.). Information security policy templates. Retrieved December 2, 2024, from https://www.sans.org/information-security-policy/
21. SANS Institute. (2022, May 5). Everything you need to know about password best practices for your organization. https://www.sans.org/blog/everything-you-need-to-know-about-passwords-for-your-organization/
22. Satori Cyber. (2022, August 14). Database activity monitoring (DAM). Satori. https://satoricyber.com/glossary/dam-database-activity-monitoring/
23. Scarfone, K., Mell, P., & National Institute of Standards and Technology. (2007). Guide to intrusion detection and prevention systems (IDPS) (NIST Special Publication 800-94). https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-94.pdf
24. Souppaya, M., Scarfone, K., & National Institute of Standards and Technology. (2022). Guide to enterprise patch management planning: Preventive maintenance for technology (NIST SP 800-40R4). https://doi.org/10.6028/nist.sp.800-40r4
25. VirusTotal. (n.d.). VirusTotal. Retrieved December 1, 2024, from https://www.virustotal.com/gui/home/search