

October 10 2020

Prepared for NGPEW by Southeast Team 5

NGPEW Penetration Test Report

A comprehensive security assessment of NGPEW's network



Table of Contents

Introduction	3
Purpose	3
Scope	3
Methodology	3
Summary of Results	4
Major Assessment Targets	5
Bug Tracking Server	5
Dam Control Password	5
Domain Controller	7
Employee & Company Reconnaissance	8
Industrial Control Systems	9
NGPEW Website	9
Rocketchat Server	10
ThinVNC Windows Server	10
Finding Summary	12
Configuration	12
Credentials	13
Database	13
Industrial Control Systems	14
Miscellaneous	14
Web	15
Conclusion	16
Overall Risk Assessment	16
Compliance	16
Recommended Mitigations	17
Configuration	17
Credentials	18
Database	18
Industrial Control Systems	18
Miscellaneous	19
Web	19
General Recommendations	19
Acknowledgements	20
Appendix	21
Part A - Machines and Services Discovered	21
Part B - Employee & Company Reconnaissance	22
Part C - Artifacts Remaining on Machines	23

Introduction

Purpose

Team 5 was contracted by Next Generation Power, Electric, and Water (NGPEW) to conduct a security assessment of their various networks and systems. Our team was asked specifically to help with the following goals posed by NGPEW:

- > Identify gaps in facilities' digital security, security management, information sharing, protective measures, and dependencies related to preparedness, mitigation, response, resilience, and recovery.
- > Create facility protective and resilience measures indice that can be compared to similar facilities.
- > Track progress toward improving critical infrastructure security.
- > Develop a project plan for use of the Critical Infrastructure Engineering Enhancement Grant which we received.

During our penetration test, the primary goal was to assess the threats unique to an energy corporation, and consider how their exploitation could cause damage to the power system, equipment, and customers of the company. Our assessment began at 8:30am Eastern on Sat Oct 10th 2020 and concluded at 5:30pm Eastern on Sat Oct 10th 2020.

Scope

The scope of our assessment was set by NGPEW, and we were restricted to targets on the following subnets of NGPEW's network:

- > 10.0.1.0/24
- > 10.0.10.0/24

Methodology

As part of our assessment we were provided with virtual machines within NGPEW's network, provisioned through Virtual Desktop Infrastructure (VDI). Each member of our team had access to a Windows 10 and Kali Linux machine from which they could complete their assessment.

We first began our assessment by cataloguing the machines that were on the network, then moved into a more detailed penetration test on each machine. We looked for improper configurations, out of date software, sensitive credentials, and any other violations of security best practices. Additionally, we did some reconnaissance work to look for any company information being shared publicly.

Due to the limited time of our engagement we primarily focused on finding vulnerabilities that were easy for possible attackers to exploit, as these are the most likely targets to be compromised in the event of an attack. Only after finding these common exploits did we move on to things like reverse engineering proprietary software and looking for more intricate exploits.

Summary of Results

Throughout all of the different services and systems being hosted on NGPEW's network, we found multiple attack vectors. We were able to gain elevated access on numerous services, including but not limited to the internal bug tracking system, the internal employee instant messaging system, the network Domain Controller and the PLCs controlling the power equipment. We also found sensitive data being posted on public GitHub repositories, which could be used in a social engineering attack against NGPEW. Additionally, there were many other configuration issues and bad security practices found across the different machines and communication channels. We have explicitly detailed these vulnerabilities in our "Finding Summary" section, and have made recommendations for how to resolve them in the "Conclusion" section.

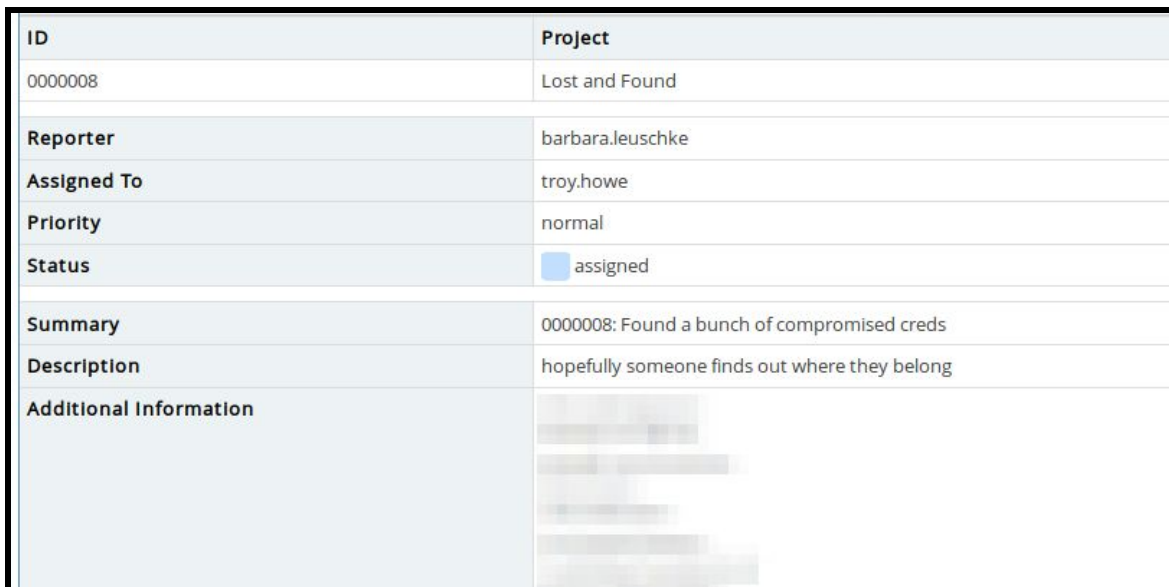
Major Assessment Targets

Bug Tracking Server

We found a bug tracking service, Mantis, running on the machine with IP address 10.0.1.153. This was found to be vulnerable to CVE-2017-7615, which allows unauthorized users to reset arbitrary passwords. After gaining administrator access, we found several internal reports that included sensitive data, including plaintext passwords and lists of administrator account usernames. It seemed like the administrator account's username was also the password of the account, but we were unable to verify whether this was true.

These exposures could give attackers a foothold in the network, allow them to view sensitive internal data and expose credentials to other critical services.

A new account named "Nathan Huckleberry" was created during our testing. This account should be deleted following the pentest. The administrator password was also reset, which was reported to the pentest point of contact. A technician should reset the administrator's password following the pentest.



ID	Project
0000008	Lost and Found
Reporter	barbara.leuschke
Assigned To	troy.howe
Priority	normal
Status	<input checked="" type="checkbox"/> assigned
Summary	0000008: Found a bunch of compromised creds
Description	hopefully someone finds out where they belong
Additional Information	[Blurred text]

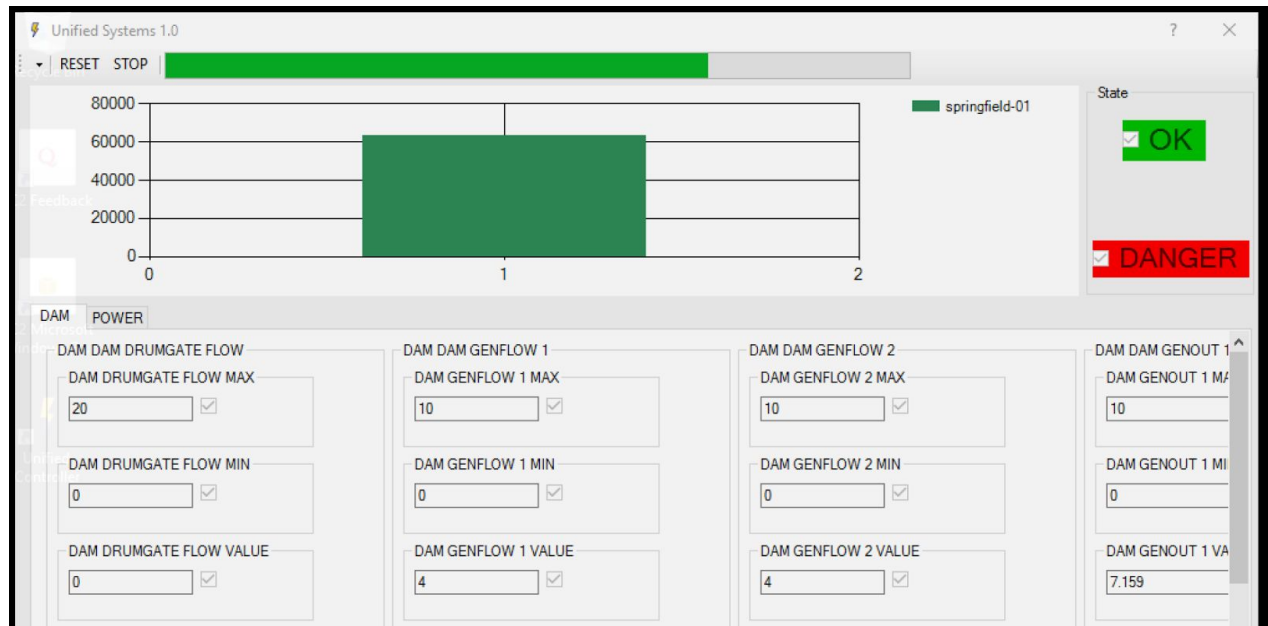
A screenshot of plaintext passwords posted in the Bug Tracking system.

Dam Control Password

While scanning 10.0.1.50, we discovered that it had a Virtual Network Computing (VNC) service running. We were able to connect to the machine with an unauthenticated VNC session. The VNC session was run with the administrator account, giving us administrator access to the machine. Additionally, we were able to access the dam control dashboard. Using this dashboard, we would have been able to change the operation of the dam and even stop operations of the dam. Dam failure

could lead to flooding, extreme monetary damages, and possible loss of life. We did not want to interfere with the dam's function, so out of an abundance of caution we did not test this capability.

Furthermore, we attempted to use our administrative access on the machine to pivot to other Windows machines. This proved unsuccessful, because there were no cached Kerberos tickets in memory and our machine did not have permissions to request new tickets from the domain controller.

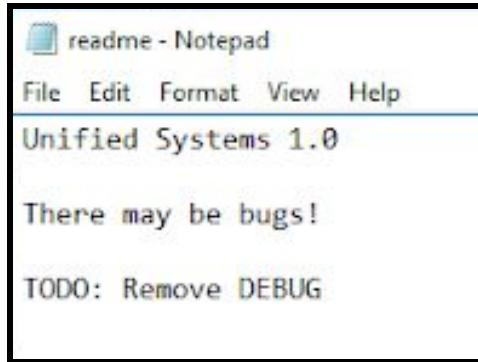


A screenshot of the dam control panel

Rather than testing the functionality of the dam controller manually and risking a potential disaster, we decided to use our domain administrator account to obtain and reverse-engineer the program running the controller, using a free tool called dotPeek. We found several security issues in the program, such as a hidden button labeled "<SECRET>" that triggered a "_debug" mode. This led to additional alert messages being deployed and the _debug flag being sent to an HTTP server along with the system state (this would presumably trigger some extra debugger functionality on the server). Seeing that the server was connected to a live dam control system, we made the decision not to test this behavior on the live server out of an abundance of caution.

```
private void SECRETToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Do you want to close enable debug?", "Activate Administrative Debug", MessageBoxButtons.YesNo) == DialogResult.Yes)
        SystemState.Set("_debug", (object) true);
    else
        SystemState.Set("_debug", (object) false);
}
```

A snippet of reverse engineered dam controller



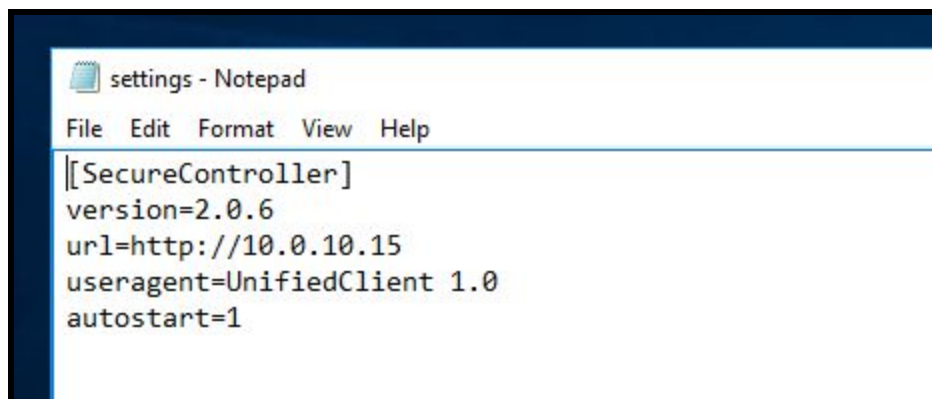
```
readme - Notepad
File Edit Format View Help
Unified Systems 1.0

There may be bugs!

TODO: Remove DEBUG
```

A snippet from the Readme of the dam controller application

By reverse-engineering the SendState() method, we found that the dams' state was sent over the network through an unencrypted HTTP POST endpoint to 10.0.10.15, which we found contained an unauthenticated, unencrypted GET endpoint to obtain the status of all dams in the system. We recommend adding authentication to this endpoint and moving to HTTPS in order to prevent information leakage.



```
settings - Notepad
File Edit Format View Help
[[SecureController]
version=2.0.6
url=http://10.0.10.15
useragent=UnifiedClient 1.0
autostart=1
```

A snippet of the config of the Dam Dashboard showing the unencrypted url

Domain Controller

Administrator credentials to the domain controller were posted in the Rocketchat main channel, giving us full access to the Windows machines on the domain. We noticed that sensitive credentials were posted in plaintext as "account descriptions". This included customer passwords, employee passwords, customer home addresses and employee home addresses. This gives attackers control over all employee accounts and leaks significant amounts of sensitive customer data.

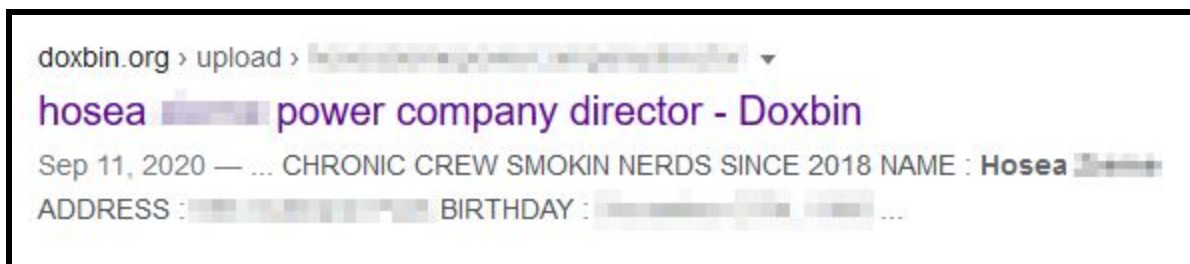
The implications of this data leak are significant. Users are prone to reuse passwords, and a dataset such as this one presents a treasure trove of potential passwords for an attacker to use on other associated accounts. In addition, the leakage of customer information presents a compliance issue (discussed further in [Compliance](#)).

Name	Type	Description
Ami [redacted]	User	[redacted]
Clifford [redacted]	User	[redacted]
Dominic [redacted]	User	[redacted]
Freddy [redacted]	User	[redacted]
Freddy C [redacted]	User	[redacted]
Gaylord [redacted]	User	[redacted]
Hilario [redacted]	User	[redacted]
Hosea [redacted]	User	[redacted]
Marcie [redacted]	User	[redacted]
Maxine [redacted]	User	[redacted]
Porfirio [redacted]	User	[redacted]
Ramiro [redacted]	User	[redacted]
Roosever [redacted]	User	[redacted]
Salome [redacted]	User	[redacted]
Thurman [redacted]	User	[redacted]
Tyler D [redacted]	User	[redacted]

A screenshot depicting the use of the description field to store credentials

Employee & Company Reconnaissance

Several avenues of information gathering were employed to enumerate employees of NGPEW and to locate any disclosed sensitive information. We were able to find employee and company accounts on LinkedIn, Twitter, and GitHub. While the employee accounts did not have any particularly sensitive information, the company GitHub account featured detailed equipment topology maps and employee organization charts, both of which could be leveraged in an attack against NGPEW’s hardware or employees. Additionally, a NGPEW employee was given access to the company GitHub organization claiming to be the Director of Technology, which they were not. This could indicate that the regulations for allowing new members into the GitHub organization are currently too lenient.



A screenshot showing publicly available information about a NGPEW employee

The exact information we found about NGPEW and its employees can be found in [Appendix B](#).

Industrial Control Systems

We found several PLC debug systems running on IP addresses 10.0.1.198-10.0.1.203. By simply connecting through a raw TCP connection on port 8080, we were able to access the debug interface of the systems. This allowed us to read the PLC chip's state, dump its firmware and configuration, change the configuration parameters, and enable development mode. The functionality offered by these PLCs allow an attacker to run any firmware they want on them. There is a wide range of attacks possible by changing the firmware, including a catastrophic failure of any and all machines controlled by the PLC. Since we have obtained full control over the PLCs through this debug shell, we have decided not to test these further. After some inquiry, we were told that these PLCs were related to dam operations, meaning that continued testing would be an extremely risky endeavor.

```
PLC DEBUG v0.1
[c] PLC-R-US 1994
=====
1> READ CPU REG
2> READ STATE DEBUG
3> DUMP FIRMWARE
4> DUMP CONFIG
5> CHANGE SAVED PARAM
6> ENABLE DEV MODE
7> PRINT DEBUG LOG
=====
CMD: .
```

A screenshot of the debug interface of the PLC

NGPEW Website

The machine at 10.0.1.152 hosted a public-facing NGPEW website. Because this is a system that customers interact with on a regular basis, we paid careful attention to this machine. During the penetration test, we found multiple ways for an attacker to either gain access to the machine or perform a denial of service attack on the server.

Firstly, the server allows unauthenticated TRACE, PUT, and DELETE HTTP requests. The PUT and DELETE requests allow an attacker to upload or delete any file on the server, respectively. TRACE requests send information back to the user that is normally kept secret on the server. This is normally used for debugging purposes, but it allows attackers to steal information such as cookies and credentials. Our angle of attack within this context was to upload a file known as a webshell, which would then give us access to the machine the server is hosted on. While we were not successful in gaining access in this manner, we believe that with more time we could have succeeded in our attack. We also believe that a malicious actor could achieve access on this machine as well, and we

recommend disabling TRACE, PUT, and DELETE requests in the server configurations. Throughout our testing and attack, several files were uploaded to the server that are potentially harmful. These files are listed in the [Appendix C](#) and we recommend removing them as soon as you can.

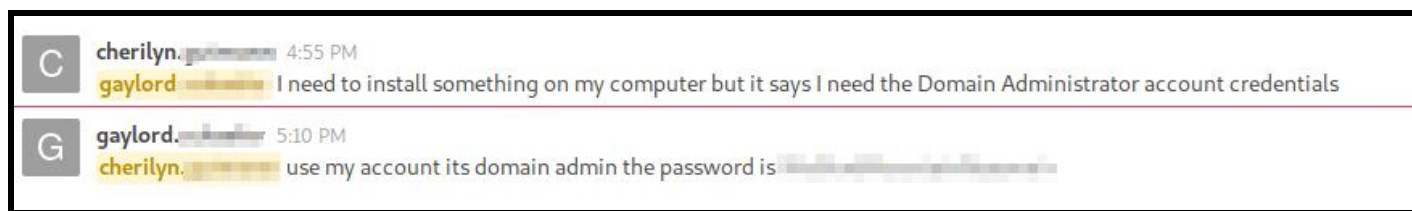
The server is also using an outdated version of IIS (version 4.0). Publicly available exploits exist for this version of IIS, and give attackers the ability to gain remote control of the machine and/or crash the web service. While we did not intentionally do so, this exploit is what led to our team taking down the website during the penetration test. We would like to sincerely apologize for the server's downtime and recommend that the IIS installation should be updated so that attackers cannot follow in our steps.

The website included a page listing examples of "secure passwords". If clients and employees use these passwords, this could present an easy attack vector for a malicious actor trying to login to user or employee accounts. These would most likely be the first guesses an attacker would make if they were trying to perform a credential stuffing attack.

Rocketchat Server

Rocketchat, an instant messaging system, was running on 10.0.1.154 on port 3000. Anyone with access to the service can create an account and view the main channel, where messages containing sensitive company data were frequently sent. For example, plaintext passwords were posted in this channel several times, including the password to the domain controller, which contains sensitive employee and client data as detailed above. These passwords allowed us to get administrative access to the Rocketchat service and domain controller administrator. We also noted that the admin account's password was the same as its username.

Two accounts named "nathan.huckleberry" and "test" were created on the RocketChat website. These accounts should be deleted following the pentest.



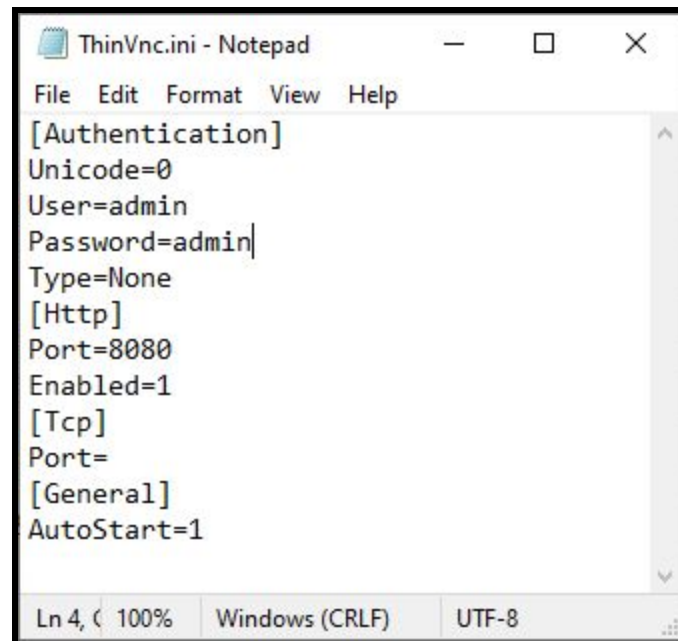
A screenshot showing the plaintext credentials posted to RocketChat

ThinVNC Windows Server

We found a ThinVNC HTTP proxy server running on port 8080 on 10.0.1.12. Most importantly, we found that the proxy did not require any authentication in order to access the main VNC functionality. Fortunately, the site seemed to be broken, with any connection to found VNC servers

failing with a black screen. We recommend adding authentication to this server or removing it altogether, since this can be used by an outsider to gain access to the internal network.

In addition, we found that the ThinVNC server was vulnerable to CVE-2019-17662, which allowed us to download arbitrary files stored on the server, such as the following configuration file. If authentication were enabled, this vulnerability would allow an outside attacker to bypass it by simply leaking the username and password. In addition, this configuration file shows several egregious misconfigurations, including the fact that the default username and password were chosen. As the ThinVNC developers have not updated the server to patch this vulnerability, we recommend replacing the server with an industry-standard equivalent such as [RealVNC](#).



```
ThinVnc.ini - Notepad
File Edit Format View Help
[Authentication]
Unicode=0
User=admin
Password=admin
Type=None
[Http]
Port=8080
Enabled=1
[Tcp]
Port=
[General]
AutoStart=1
Ln 4, 100% Windows (CRLF) UTF-8
```

A screenshot of the default credentials of the ThinVnc server

Finding Summary

Configuration

Finding ID	Issue and IP Address	Risk Level (low/med/high)	Description
CO-1	ThinVNC directory traversal 10.0.1.12	HIGH	The installed version of ThinVNC is vulnerable to a directory traversal attack through CVE-2019-17662, allowing arbitrary files such as configuration and system files to be leaked.
CO-2	Dam API not encrypted 10.0.10.50	HIGH	The API endpoint that keeps track of dam status is not encrypted or authenticated. The API should use HTTPS and have some form of user authentication.
CO-3	Admin password reset for Mantis Bug Tracker 10.0.1.153	HIGH	The installed version of Mantis Bug Tracker is outdated and vulnerable to CVE-2017-7615. This allows attackers to reset the password of any bug tracker accounts, including the administrator account.
CO-4	Unified Systems publicly accessible through VNC 10.0.1.50	HIGH	We found that the server hosted an unauthenticated VNC server, allowing access to a dam dashboard with several ominous actions such as "start," "stop," and "DANGER."
CO-5	ThinVNC lacks authentication 10.0.1.12	MEDIUM	The ThinVNC server is configured to require no authentication, allowing a user with HTTP access to bypass firewall restrictions to connect to a VNC server on the internal network.
CO-6	RocketChat unauthenticated access 10.0.1.154	MEDIUM	Anyone with access to the RocketChat webpage is able to register accounts and view the main company channel.
CO-7	Weak passwords for Active Directory accounts	MEDIUM	Many employee accounts had extremely weak passwords, often single words with no capitalization.

	10.0.1.100		
CO-8	10.0.1.150	MEDIUM	Unconfigured Apache server.

Credentials

Finding ID	Issue and IP Address	Risk Level (low/med/high)	Description
CR-1	Insecure password for RocketChat admin 10.0.1.154	HIGH	We found that the password for the RocketChat administrator account was the same as the username. This allowed easy access to the RocketChat admin panel.
CR-2	Mantis BugTracker password as account detail 10.0.1.153	HIGH	The Administrator password for Mantis BugTracker was listed as the real name of the account. We were unable to verify that this was actually the password.
CR-3	Domain accounts passwords in account details 10.0.1.100	HIGH	The password for each domain account was posted in plaintext on each account's description.
CR-4	Credentials posted in Mantis BugTracker 10.0.1.153	HIGH	Compromised credentials were posted in the bug tracker. These expose the company to a possible password reuse attack.
CR-5	Admin accounts listed in Mantis BugTracker 10.0.1.153	LOW	The BugTracker had a list of admin accounts. This allows attackers to easily bruteforce passwords.

Database

Finding ID	Issue and IP Address	Risk Level (low/med/high)	Description
DB-1	Customer data stored in Active Directory.	HIGH	Customer home addresses and unhashed plaintext passwords are stored in the active

	10.0.1.100		directory “account details”.
DB-2	Database in public subnet. 10.0.1.151	MEDIUM	The database is not located in a DMZ. This may allow easy exploitation and exposure of customer data.

Industrial Control Systems

Finding ID	Issue and IP Address	Risk Level (low/med/high)	Description
ICS-1	PLC debug ports publicly accessible 10.0.1.198-203	HIGH	The PLC devices kept port 8080 open as a debug port, allowing for unauthenticated access to the PLC firmware, configuration, as well as unauthenticated modification of the PLC device state.
ICS-2	PLCs are reachable from the corporate network. 10.0.1.198-203	HIGH	The PLC devices are directly accessible from the corporate network, opening these sensitive electronics controlling critical dam infrastructure to DoS attacks.

Miscellaneous

Finding ID	Issue and IP Address	Risk Level (low/med/high)	Description
M-1	Equipment topology map published publicly on GitHub	MED	A detailed topology map of the equipment controlled by NGPEW was published to a public GitHub repository. These details could be used by an attacker to determine which machines they should attack and how it would affect NGPEW’s clients.
M-2	Company organization chart published publicly on GitHub	MED	A complete organization chart of NGPEW’s employees was posted on a public GitHub repository. This information could be used in a social engineering attack.
M-3	Employee added to GitHub organization under false pretenses	LOW	An NGPEW employee requested to be added to the NGPEW GitHub organization claiming to be the Director of Technology,

which they were not. The comments implied that the employee's request was granted.

Web

Finding ID	Issue and IP Address	Risk Level (low/med/high)	Description
W-1	Server allows PUT, DELETE, and TRACE requests 10.0.1.152	HIGH	The server at 10.0.1.152 allows PUT, DELETE, and TRACE requests. This allows malicious actors to upload dangerous files, delete files on the web server, and view sensitive server information.
W-2	Server has outdated IIS version 10.0.1.152	HIGH	The server at 10.0.1.152 has IIS version 4.0 installed. Public exploits are freely available for this version of IIS, and allow an unauthenticated actor to run arbitrary code on the machine or crash the server.
W-3	Password examples in security page 10.0.1.152	MED	There is a page on the main NGPEW website at 10.0.1.152 describing proper security practices. It includes example passwords that we're worried may be reused. Several employees used these passwords for their corporate credentials.
W-4	HTTP does not redirect to HTTPS 10.0.1.152	MED	When navigating to the server at 10.0.1.152, it by default serves the insecure HTTP page. Any clients connecting to the page will risk their information being sent in plaintext over the Internet.
W-5	CSP not configured 10.0.1.152	MED	On the public-facing website, the server was not configured to offer a CSP directive. Setting a CSP directive prevents possible attacks such as XSS through a form submission.
W-6	HSTS not configured 10.0.1.152	LOW	The public-facing server is not configured to offer a Strict-Transport-Security header in order to prevent a user session from being intercepted and redirected to a malicious HTTP clone of the site by a malicious network.

Conclusion

Overall Risk Assessment

We conclude that the overall risk identified to NGPEW is **high**, as we have identified several attack vectors leading to issues from customer data leakage to potential loss of life. According to the Cyber Security and Infrastructure Agency's dam sector specific plan, "a dam failure could result in sudden downstream flooding that causes casualties; major destruction and property damage; and cascading disruptions to the Electricity, Transportation Systems, Communications, and Water Sectors, among others." A malicious entity could feasibly mount such attacks against NGPEW and we strongly recommend mitigation efforts to begin immediately.

Compliance

Note that this penetration test does not serve as a full audit of compliance under all applicable statutes and regulatory agencies. However, we would like to reveal some insight which our security audit has uncovered regarding compliance.

Since it processes customer payment cards directly, NGPEW is subject to regulations under the [Payment Card Industry Data Security Standard](#) (PCI DSS).

Customer passwords and home addresses are stored in plaintext in Active Directory. This is a direct violation of Requirements 7.1 and 8.2.1 of PCI DSS. Passwords should be hashed and salted using modern cryptographic schemes. According to Requirement 8.2.1, NGPEW must "use strong cryptography, to render all authentication credentials (such as passwords/phrases) unreadable during transmission and storage on all system components". According to Requirement 7.1, NGPEW must "limit access to ... cardholder data to only those individuals whose job requires such access."

Service default credentials are enabled on RocketChat. This is a violation of Requirement 2 of PCI DSS. From the official PCI Security Standards, NGPEW must "not use vendor-supplied defaults for system passwords and other security parameters."

Unnecessary services should be disabled, as stated in Requirement 2.2.5 of PCI DSS. There is an unconfigured Apache web server which directly violates this requirement. The requirement states that NGPEW must "Remove all unnecessary functionality, such as scripts, drivers, features, subsystems, file systems, and unnecessary web servers".

A password policy must be enforced on all services as stated in Requirement 8.2.3 of PCI DSS. NGPEW must "Require a minimum length of at least seven characters" and require that passwords "Contain both numeric and alphabetic characters". This is currently not the case in several NGPEW systems, including Active Directory and RocketChat.

Since it manages several dam infrastructures, NGPEW is also subject to the US National Institute of Standards and Technology's (NIST) [Framework for Improving Critical Infrastructure Cybersecurity](#). While this framework is voluntary, it nevertheless provides several useful guidelines for optimum security around dam management and surveillance systems.

For example, we found that several dam PLC devices were accessible from anywhere in the corporate network. This is a violation of the PR.AC-5 subcategory, which states: "network integrity is protected, incorporating network segregation where appropriate." In addition, the fact that the PLC debug ports were open to the network presented a violation of the PR.PT-3 subcategory, which states: "access to systems and assets is controlled, incorporating the principle of least functionality."

Recommended Mitigations

For each finding detailed in the "Finding Summary" section, we have outlined possible fixes to minimize or eliminate the associated security risk.

Configuration

Finding ID	Recommendation
CO-1	As there are no patches available for ThinVNC at this moment (10/10/2020), we recommend replacing the software altogether with an equivalent but far more secure alternative, such as TeamViewer or RealVNC .
CO-2	We recommend configuring the Werkzeug service to require a username and password to get and set dam statuses, in addition to adding an HTTPS requirement to avoid password leakage.
CO-3	Mantis Bug Tracker should be updated to the latest stable version and passwords should be rotated. At this time, all user accounts should be considered compromised.
CO-4	We recommend adding username and password authentication to the VNC login process, in addition to keeping all mission-critical software off the corporate network.
CO-5	We would recommend setting a username and password in ThinVNC's configuration files, but any such configuration could be easily leaked through CO-1. Instead, we recommend replacing the ThinVNC software as mentioned in our CO-1 recommendation.
CO-6	RocketChat account registration should be disabled. New accounts should be created directly by administrators. The current list of accounts should be validated to ensure no malicious accounts exist.
CO-7	Password policies should be enforced on all services to prevent weak passwords.

Passwords should also be rotated monthly.

CO-8 This service should be removed.

Credentials

Finding ID	Recommendation
------------	----------------

CR-1	The administrator password for RocketChat should be rotated and a password policy should be enforced.
------	---

CR-2	Passwords should never be posted in plaintext. The admin password should be immediately changed and the administrator's name should be changed to "Administrator".
------	---

CR-3	We recommend informing all affected users of the leakage and requiring them to reset their password immediately, in addition to changing the policies that led to the passwords being stored in plain text in the descriptions. Passwords should always be hashed and salted before being stored anywhere.
------	---

CR-4	Passwords should never be posted in plaintext. If credentials are compromised, their <i>usernames</i> should be reported to an administrator. Employees with access to bugtracker should be required to attend training reinforcing these policies. All passwords should be rotated.
------	---

CR-5	Employees should be urged not to post any administrator account names in publicly accessible locations. Posts including any account data should be private.
------	---

Database

Finding ID	Recommendation
------------	----------------

DB-1	Customer data should be stored in a secured database, separately from employee data. This secured database should be located in a DMZ and should not be easily accessible from inside the network.
------	--

DB-2	The database should be moved into a DMZ. Firewall rules should be created to only allow connections to the production database from the website.
------	--

Industrial Control Systems

Finding ID	Recommendation
------------	----------------

ICS-1	We recommend either disabling the debug functionality or placing the debug port behind a secure authentication service.
-------	---

ICS-2 Moving these PLCs off the corporate network and attaching them to a segmented lab network on their own non-internet-accessible VLAN would minimize their attack service.

Miscellaneous

Finding ID	Recommendation
------------	----------------

M-1	Make this repository private, or remove the files completely from GitHub.
-----	---

M-2	Make this repository private, or remove the files completely from GitHub.
-----	---

M-3	Implement a stricter and more thorough process for validating users attempting to join the GitHub organization
-----	--

Web

Finding ID	Recommendation
------------	----------------

W-1	Disable PUT, DELETE, and TRACE requests in the server's configuration.
-----	--

W-2	Update to the latest version of IIS as soon as possible.
-----	--

W-3	Instead of giving examples of strong passwords, it is pertinent to present a format for strong passwords. See General Recommendations for an example password format.
-----	---

W-4	Configure the server to use HTTPS instead of HTTP.
-----	--

W-5	Configure the server to give sufficiently strict CSP directives.
-----	--

W-6	Configure the server to give HSTS headers.
-----	--

General Recommendations

- Password Policies should be created and enforced.
 - Passwords must be at least 12 characters long.
 - Passwords must use numbers and special characters.
 - Passwords must not contain common dictionary words.
- A centralized service-monitoring service such as Splunk should be set up on the network.
- Employees should attend required training on the sensitivity of user data, account passwords and other internal data.
- Unused services should be turned off.

Acknowledgements

We would like to thank NGPEW for choosing our firm for this security audit, and for the speedy, clear communication throughout. We know you have many penetration specialists to choose from and we appreciate you choosing to work with us. We wish that our findings and recommendations will prove helpful and hope to work with NGPEW again.

Appendix

Part A - Machines and Services Discovered

10.0.1.0/24

IP	OS	Machine Purpose	Ports/Services
10.0.1.10	Windows	Workstation	135 - Windows RPC 139 - netbios-ssn 445 - microsoft-ds 3389 - Microsoft Terminal Services
10.0.1.11	Windows Server 2012	Workstation	135 - Windows RPC 139 - netbios-ssn 445 - microsoft-ds 3389 - Microsoft Terminal Services
10.0.1.12	Windows Server 2012	Workstation	135 - Windows RPC 139 - netbios-ssn 445 - microsoft-ds 3389 - Microsoft Terminal Services 8080 - ThinVNC proxy
10.0.1.13	Windows Server 2008/2012		80 - Windows IIS httpd 10.0 135 - Windows RPC 139 - netbios-ssn 445 - microsoft-ds 3389 - Microsoft Terminal Services
10.0.1.50	Windows Server 2002 R2	Dam Controller	135 - Windows RPC 139 - netbios-ssn 445 - microsoft-ds 3389 - Microsoft Terminal Services
10.0.1.100	Windows Server	Domain Controller	53 - DNS 88 - Kerberos 135 - Windows RPC 389 - LDAP 445 - microsoft-ds 464 - Kerberos 3268 - LDAP 3269 - Microsoft Terminal Services
10.0.1.150	Ubuntu	Unconfigured Web Server	22 - SSH 80 - HTTP

10.0.1.151	Ubuntu	MySQL Server	22 - SSH 3306 - MySQL
10.0.1.152	Windows	Main Website	80 - HTTP 135 - Windows RPC 443 - HTTPS 1027 - Windows RPC 1029 - Windows RPC
10.0.1.153	Ubuntu	Mantis Bug Tracker	22 - SSH 80 - HTTP
10.0.1.154	Ubuntu	RocketChat	22 - SSH 3000 - HTTP
10.0.1.198	SCADA	Dam Controls	8080 - PLC
10.0.1.199	SCADA	Dam Controls	8080 - PLC
10.0.1.200	SCADA	Dam Controls	8080 - PLC
10.0.1.201	SCADA	Dam Controls	8080 - PLC
10.0.1.202	SCADA	Dam Controls	8080 - PLC
10.0.1.203	SCADA	Dam Controls	8080 - PLC

10.0.10.0/24

IP	OS	Machine Purpose	Ports/Services
10.0.10.15		Dam API Server	80

Part B - Employee & Company Reconnaissance

The following accounts for employees were found on the public web. While these accounts themselves don't necessarily pose a security threat, employees must be careful not to post any sensitive company information on these platforms.

Name	Links
Grace Grantham	linkedin.com/in/grace-grantham-2a66001b6 twitter.com/gracegrantham14
Gaylord Schaefer	linkedin.com/in/gaylord-schaefer-4a18381b7 github.com/gaylord-schaefer

Tiny Glover	linkedin.com/in/tiny-glover-99550b1b6 github.com/tiny-glover
Barbara Leuschke	linkedin.com/in/barbara-leuschke-88a9651b7 twitter.com/hrleuschke
King Shields	linkedin.com/in/king-shields-34ba461b7 twitter.com/kingshields6
Hosea Zieme	github.com/HoseaZieme

We also found the following accounts for NGPEW on the public web. As mentioned above, potentially sensitive information was published on the GitHub account.

NGPEW	linkedin.com/company/next-generation-power-and-water/ github.com/Next-Generation-Power-and-Water
-------	--

The issue mentioned above regarding an employee being added to the GitHub organization under false pretenses is linked here: github.com/Next-Generation-Power-and-Water/home/issues/1

Part C - Artifacts Remaining on Machines

Software Installed on Dam Dashboard machine (10.0.1.50)

- > Firefox
- > Windump
- > Winpcap
- > Socks Proxy
- > Filezilla
- > WireShark

Files left on Dam Dashboard on C:\Users\Administrator\Downloads

- > Jaws.ps1
- > Sharphound.exe
- > Msf-backdoor.exe

Files left on Web Server (10.0.1.152) (File paths relative to server install location)

- > images/nmap.php
- > ../../../../nmap_test.txt
- > images/nmap_query_shell.html
- > images/nmap_jsp_shell.jsp
- > images/nmap_asp_shell.jsp
- > images/nmap_asp_shell.asp

- > images/nmap_aspx_shell.aspx

Mantis BugTracker (10.0.1.153)

- > Account named "Nathan Huckleberry" left on the bug tracker
- > Administrator password reset
 - > The new password was reported to the point of contact

RocketChat (10.0.1.154)

- > Account named "nathan.huckleberry" left on the chat server
- > Account named "test" left on the chat server