Next-Generation Power, Electric, & Water

Penetration Test Report

January 10th, 2021

[redacted]

Table of Contents

Executive Summary

On January 8-9 2021, our team conducted a penetration test for Next-Generation Power, Electric & Water (NJPW) in order to identify security vulnerabilities. Our penetration test revealed the ability to access dam control systems and servers by exploiting several vulnerabilities. These vulnerabilities were exploited from within the internal network of NGPEW. This report thoroughly details our methods, findings, and suggestions for resolving the identified security issues at NGPEW. We've also included a comparison with our first assessment in Appendix A.

Key Observations

High Priority Issues

- **Sensitive Controls Disclosure.** The dam monitoring panel did not require authentication to access. Any attacker that manages to breach the perimeter will be able to view the dam status and control the dam flow, power, and settings. Due to the critical nature of the infrastructure, any misuse could have disastrous consequences for NEW and its customers.
- **Unauthorized Access.** From within the company network, one piece of software is running with elevated privileges and one server is running outdated software with numerous known vulnerabilities. These can be leveraged to allow an attacker more access or to allow them to run code where those operations should be forbidden.

Recommendations

- **Restrict Privileges as Much as Possible.** Accounts and assets should only have access to the resources and permissions they need to complete the necessary tasks. Excess privileges can be leveraged by attackers to run malicious programs or access other, more private information and assets.
- **Strengthen Authentication.** While changing and strengthening passwords and implementing an account lockout policy have made exploiting authentication significantly more difficult, many of NGPEW's assets qualify as critical infrastructure, which means that any compromise is an unacceptable risk. Implementing multi-factor authentication on these assets and restricting access to only necessary personnel would add another layer of protection to these systems.

Scope

The scope for this penetration test encompassed the computers on the 10.0.1.0/24, 10.0.5.0/24, and 10.0.10.0/24 subnets within the NEW internal network. Tests were performed from positions outside the scope subnets and from within the 10.0.1.0/24 subnet.

Positive Security Controls

NGPEW took several measures to increase their security posture independent of our test.

- **Network Segmentation.** We could not establish network connectivity to two of the subnets from our attacker machines. From inside the network, we observed that related assets were encompassed by their own firewalls. This prevents an attacker in one part of the network from accessing more sensitive data in another part of the network and makes it more difficult to assess the networks and their assets.
- **No accessible credentials.** While evaluating publicly available information about NGPEW and its employees, we did not uncover any functional credentials which we could use to impersonate a user and gain a foothold in the network. While moving through the network, we did not find any usable and/or exposed credentials.
- **Account lockout.** We tested logins by trying multiple passwords with usernames obtained during our previous engagement. These efforts were prevented by the account lockout policy, which disabled the Windows accounts we tried to access with this method after 10 failed attempts within a short period. Similarly, Rocket.Chat had a limit on the number of login attempts in a short amount of time. These policies make guessing a password through trial-and-error impractical or impossible, which prevents attackers from using this method to gain access to the network.

Severity Rankings

During our assessment, we identified security weaknesses in applications and systems, and have provided recommendations to address these weaknesses. Based on industry-standard practices, the following ratings have been assigned to each observation based upon the risk they pose to Next-Generation Power, Electric & Water systems and applications.

Critical

- Vulnerability presents an immediate threat to business systems or data, or could significantly impact business operations.

High

- Vulnerability could allow significant access to business systems or data, or could strongly impact business operations.

Medium

- Vulnerability could impact business systems or data or could reduce business performance.

Low

- Vulnerability should not impact business operations but should be addressed to meet the expected operation.

Informational

- There is little to no risk associated with the observation or vulnerability and is provided as context for other observations.

Technical Findings

Critical

Unauthenticated VNC Access to Dam Monitoring System

Threat Level: Critical

Affected Hosts:

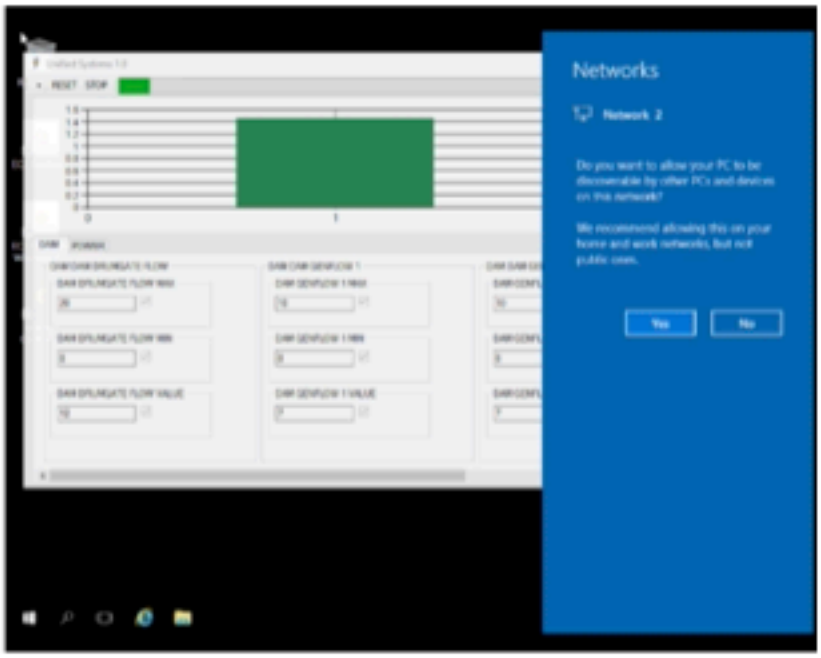
- 10.0.5.50

Description:

We were able to gain access to the machine running the dam monitoring system without authentication.

Exploitation Details:

From the Nmap scan, we discovered an open port with Virtual Network Computing (VNC) running on the server. One of our consultants could log into the host without providing credentials and view/edit the "Unified Systems 1.0" dam control panel. We did not go further on this host to prevent affecting the system in any way.



Potential Business Impact:

If an attacker accessed this system, it could have disastrous consequences for both the company and its customers, as the attacker would have the ability to turn the power off, slow down the flow of the dam, and change other settings. The company's main services are water and power. This system could give an attacker access to its controls, resulting in financial losses and decreased trust from the communities served.

Remediation:

NJPW should require authentication to VNC systems and utilize complex and strong passwords, specifically on the "Unified Systems 1.0" damn control panel. They should also implement logging on this system and back up all system settings and information elsewhere.

High

Remote Code Execution on Outdated Web Server

Threat Level: High

Affected Host:

- 10.0.5.152

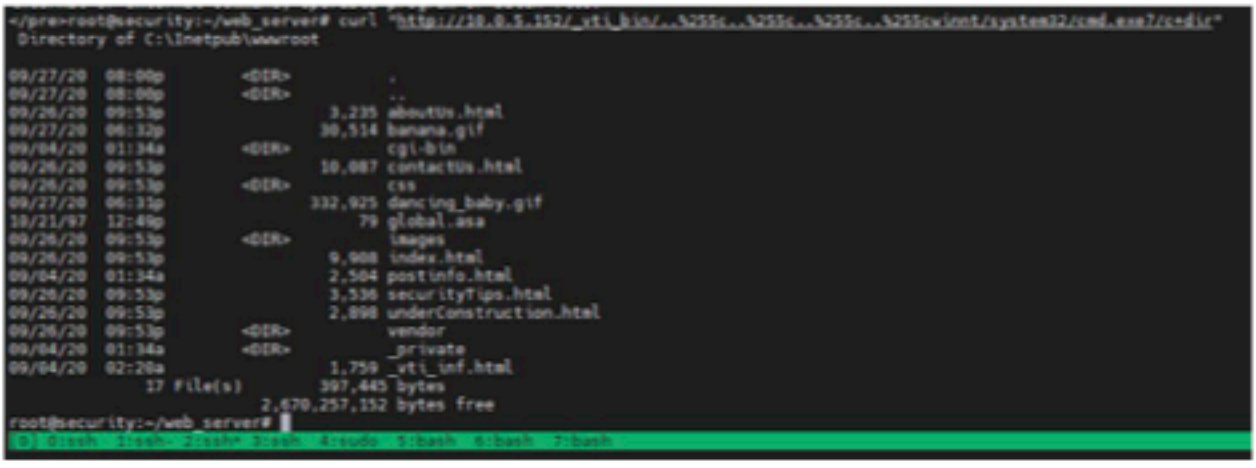
Description:

The IIS web server running on 10.0.5.152 is "End of Life" and is no longer maintained or supported by the vendor. This version has numerous public vulnerabilities, one of which was used to gain remote code execution on the server (CVE-2000-0884).

Exploitation Details:

The ngpew.com website is hosted on 10.0.5.152 and is running an outdated version of Microsoft IIS (version 4.0) which is no longer supported by the vendor.

This outdated web server is vulnerable to many exploits, IIS Unicode Directory Traversal being one of them. This was exploited by one of our consultants as shown.



Potential Business Impact:

An attacker can use this publicly-known vulnerability to gain access to the web server to modify the website and serve malicious content or gain further access to the internal corporate network.

Remediation:

Update the web server and/or the operating system to the newest version to ensure that the system has support for the duration of use.

References:

https://www.cvedetails.com/cve/CVE-2000-0884/

Remote Code Execution on Billing and Invoice System

Threat Level: High

Affected Host:

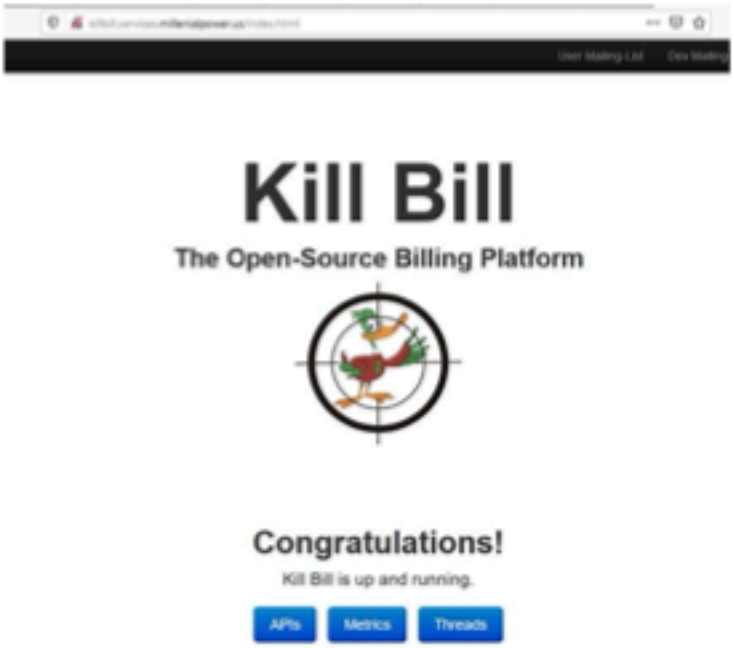
- 10.0.5.75

Description:

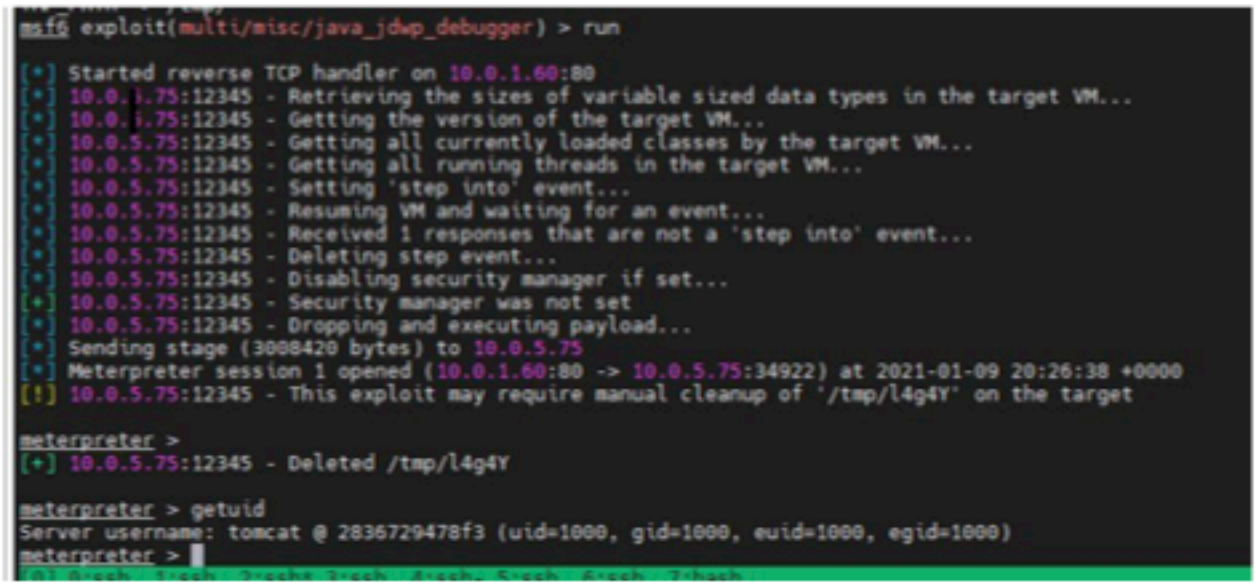
The Kill Bill software used by Next-Generation Power, Electric & Water had a Java Debugging Port publicly exposed. We used this to get code execution on the container which runs the Kill Bill App.

Exploitation Details:

While scanning the network, we came across a website that served the Kill Bill web app which is a billing and invoicing platform. On port scanning, we found that the host had Java Debug Wire Protocol (JDWP) service exposed on port 12345 and Java Remote Management Interface (RMI) exposed on port 8000.



Using a publicly-available exploit with Metasploit, we obtained a shell on the remote machine. Upon investigating, we find that our shell is inside a Docker container instead of the main host. However, our account has administrator access privileges through sudo, so we can get root privileges inside the Docker container.



Potential Business Impact:

With root privileges on the container, an attacker can modify the billing website to insert malicious content. The attacker can also access personal information if the site is used to store NEW customer information.

Remediation:

Firewall debug ports such as 12345 (JDWP) and 8000 (Java RMI) so that they are not accessible publicly. Service accounts such as tomcat should not have administrator privileges.

MySQL Database for Billing System Accessible Externally.

Threat Level: High

Affected Host:

- 10.0.5.75

Description:

The MySQL Database on the billing host was reachable publicly and didn't have any host restrictions. Using credentials from the shell we got earlier, we were able to access the data on the server for the Kill Bill service and the Admin Panel..

Exploitation Details:

While going through the files on the Kill Bill Docker container, we found the configuration files which had the password for MySQL database used by the service.

```
root@2836729478f3:/# cat ./var/lib/killbill/killbill.properties
cat ./var/lib/killbill/killbill.properties
org.killbill.billing.osgi.bundles.jruby.conf.dir=/var/lib/killbill/config
org.killbill.billing.osgi.dao.password=
org.killbill.billing.osgi.dao.url=jdbc:mysql://localhost:3306/killbill
org.killbill.billing.osgi.dao.user=root
org.killbill.catalog.uri=SpyCarAdvanced.xml
org.killbill.dao.password=
org.killbill.dao.url=jdbc:mysql://localhost:3306/killbill
org.killbill.dao.user=root
org.killbill.osgi.bundle.install.dir=/var/lib/killbill/bundles
org.killbill.server.baseUrl=http://localhost:8080
org.killbill.billing.plugin.kpm.kpmPath=/opt/kpm-0.9.0-linux-x86_64/kpm
org.killbill.billing.plugin.kpm.bundlesPath=/var/lib/killbill/bundles
```

As there was no host restriction on the MySQL database, we could connect to it from our host and get access to the data.

```
root@security:~# mysql -u root -h 10.0.5.75 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 78
Server version: 5.5.5-10.3.14-MariaDB-1:10.3.14+maria~bionic mariadb.org binary distribution

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
[0] 0:ssh  1:ssh*  2:ssh-  3:ssh  4:ssh  5:ssh  6:ssh  7:bash
```

There were two databases on the host, one for the billing service and the another for the admin interface for the service. At that point, we found no personal information in the database.

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| kaui               |
| killbill           |
| mysql              |
| performance_schema |
+--------------------+
5 rows in set (0.00 sec)
```

```
Database changed
mysql> SHOW TABLES;
+----------------------------------------+
| Tables_in_killbill                     |
+----------------------------------------+
| account_email_history                  |
| account_emails                         |
| account_history                        |
| accounts                               |
| adyen_hpp_requests                     |
| adyen_notifications                    |
| adyen_payment_methods                  |
| adyen_responses                        |
| analytics_account_fields               |
| analytics_account_tags                 |
| analytics_account_transitions          |
| analytics_accounts                     |
| analytics_bundle_fields                |
| analytics_bundle_tags                  |
| analytics_bundles                      |
| analytics_currency_conversion          |
```

Since we had access to the MySQL database as root, we could read and write files on the host machine as the mysql user.

```
mysql> SELECT 'testing' INTO DUMPFILE '/tmp/test';
Query OK, 1 row affected (0.00 sec)

mysql> SELECT LOAD_FILE('/tmp/test');
+-------------------------------------------------+
| LOAD_FILE('/tmp/test')                          |
+-------------------------------------------------+
| 0x74657374696E67                                |
+-------------------------------------------------+
1 row in set (0.00 sec)

mysql>
```

Output

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
mysql:x:999:999::/home/mysql:/bin/sh
```

Potential Business Impact:

With access to the database, an attacker could obtain customer information and credentials if this service is used by employees/customers in the future. Having file read/write permissions, a motivated attacker could get code execution on the database host as well.

Remediation:

Configure MariaDB so that only connections from necessary hosts are allowed. Preferably configure it so that it is not exposed publicly. Create separate database users for services, so that compromise of one service does not lead to complete database access. Set the secure_file_priv option in MariaDB so that the database cannot have complete access to the file system.

References:

https://mariadb.com/kb/en/server-system-variables/#secure_file_priv

VNC Service Running as Administrator on Dam Control Panel

Threat Level: High

Affected Host:

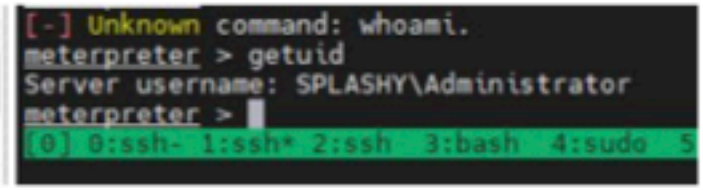
- 10.0.5.50

Description:

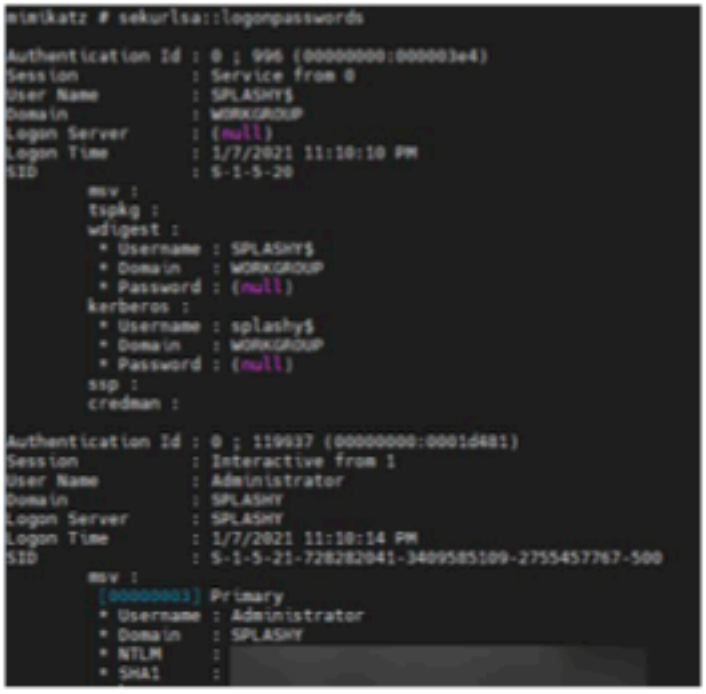
The open VNC service on 10.0.5.50 is running as the local administrator user.

Exploitation Details:

Running the command "getuid" confirms the user is the Administrator.



One of our consultants was able to extract the hashed password for the Administrator account on the VNC service using mimikatz.



Potential Business Impact:

An attacker can gain administrative access to the VNC server just by connecting. The attacker can then access other services running on the machine like the dam control panel.

Remediation:

Restrict access to only allow administrators with a strong password and multi-factor authentication.

Medium

Plaintext Password on Windows Machine

Threat Level: Medium

Affected Hosts:

- 10.0.5.150
- C:\Windows\Temp\UserScript.ps1

Description:

The PowerShell script "UserScript.ps1" contains an Administrator password stored in plaintext.

Exploitation Details:

```
([adsi]"WinNT://$env:computername/Administrator").SetPassword('[redacted]')

netsh advfirewall firewall add rule name="WinRM-HTTP" dir=in localport=5985 protocol=TCP action=allow
netsh advfirewall firewall add rule name="WinRM-HTTP" dir=in localport=5986 protocol=TCP action=allow
Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled false



winrm quickconfig -q
winrm set winrm/config/winrs '@{MaxMemoryPerShellMB="1024"}'
winrm set winrm/config '@{MaxTimeoutms="1800000"}'
winrm set winrm/config/service '@{AllowUnencrypted="true"}'
winrm set winrm/config/service/Auth '@{Basic="true"}'
winrm set winrm/config/client/auth '@{Basic="true"}'

Start-Service WinRM
Set-Service WinRM -StartupType Automatic

Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled false
netdom renamecomputer "$env:COMPUTERNAME" /Newname "splashy" /Force /Reboot 0


C:\Windows\Temp>
```

Potential Business Impact:

An attacker can easily obtain these credentials once accessing the machine and use them to access other systems or services.

Remediation:

Remove the sensitive data from the file and store it in an encrypted format.

Login Over Cleartext Protocol

Threat Level: Medium

Affected Hosts:

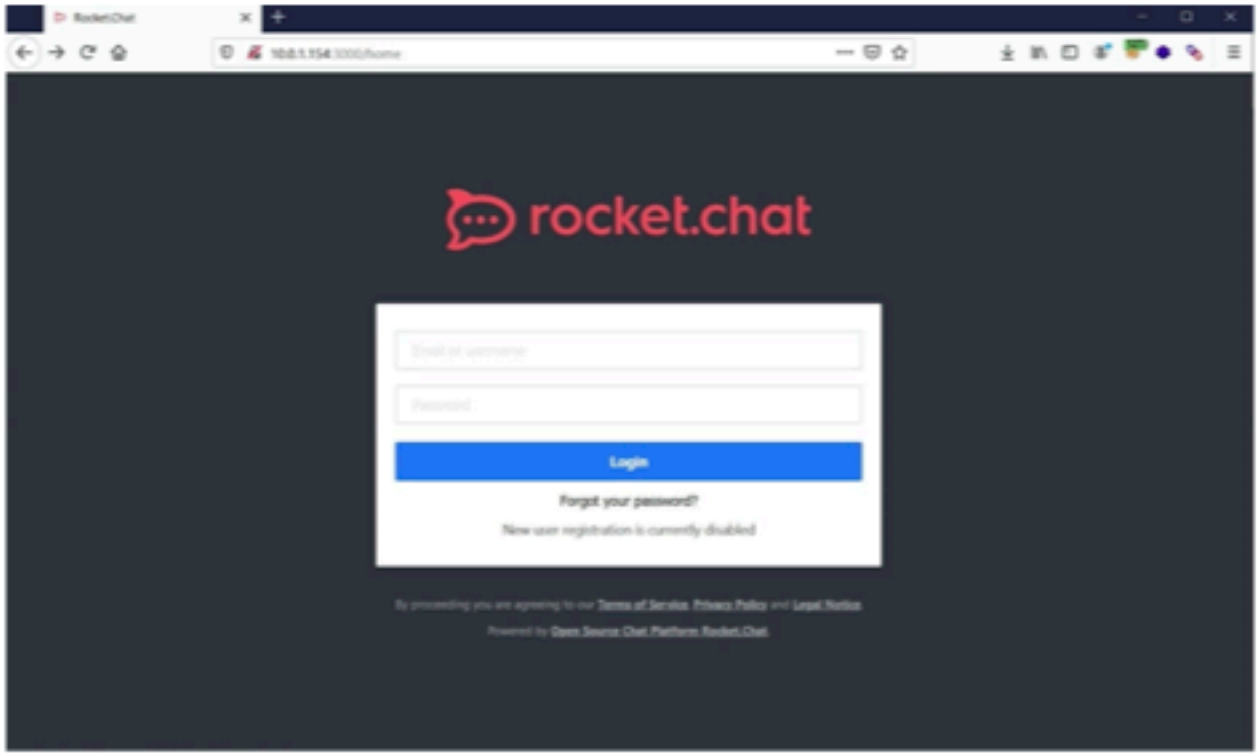
- http://10.0.1.154:3000/home
- http://10.0.1.154:3000/admin/info

Description:

This Rocket.Chat application allows for logins over cleartext communication (HTTP).

Exploitation Details:

Browse to http://10.0.1.154:3000 and observe the login over HTTP.



Viewing the request in Burp Suite shows the password is hashed in the request.

{"username":"casacy"},"password":{"digest":"e100fbce000c04ec40037a80a891fb2f05aeedc23f856a2d3c0b1580625d755e",

Potential Business Impact:

Any login credentials or sensitive data not protected by a protocol that encrypts internet traffic (TLS/SSL) are sent over the network in cleartext and an attacker can utilize a packet sniffing software to obtain this data.

Remediation:

The application should use transport-level encryption (SSL or TLS) to protect all communications passing between the client and the server. Attempts to login over HTTP should be redirected to HTTPS.

Low

Dam Monitor Information Disclosure

Threat Level: Low

Affected Host:

- http://10.0.10.15

Description:

The host shows JSON data from the monitor for the dam.

Exploitation Details:

Use curl or browse to http://10.0.10.15

Potential Business Impact:

This exposes monitoring information of the dam systems.

Remediation:

Require authentication or an API key to view the monitoring data.

Informational

Enticement Information

Threat Level: Informational

Affected Hosts:

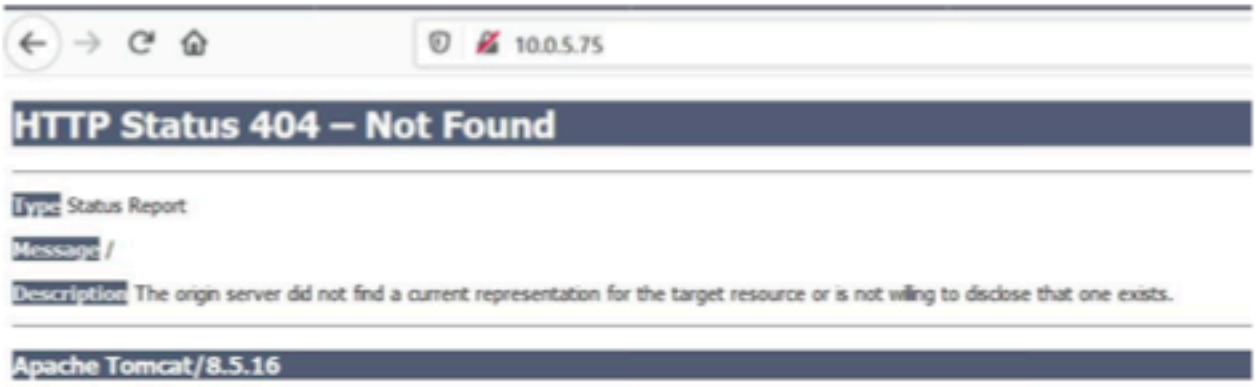
- http://10.0.5.75
- http://10.0.5.153
- http://10.0.10.15

Description:

The affected URLs display server names and versions.

Exploitation Details:

Browse to http://10.0.5.75 to view the server running Apache Tomcat version 8.5.16.



Potential Business Impact:

Enticement information can be used to further understand how the application works and allow an attacker to attempt targeted attacks based on the information disclosed.

Remediation:

Remove all references to server names and versions. Handle all errors by displaying a generic error message to the user.

Detailed Narrative

Prior to the start of the penetration test, we conducted initial reconnaissance by searching for open-source intelligence on the company domain (ngpew.com) and information about NGPEW employees. We came across a public GitHub repository with some company documents in the Git history. We used these to create username lists to test for accessible logins.

Upon starting the test, we attempted to nmap scan all of the subnets in scope and accessed the 10.0.1.0/24 subnet, but could not access the 10.0.5.0/24 and 10.0.10.0/24 subnets. We found that NGPEW implemented network segmentation to control access to their essential systems.

We scanned each device on the initially-accessible subnet, 10.0.1.0/24, to attempt to find a foothold for subsequent tests. We noted that NGPEW implemented many of the recommendations we had included in our report from the previous quarter's engagement. Specifically, new users had to be added to the Rocket.Chat system by an administrator and the login used a delay mechanism to prevent brute-force discovery of passwords.

On January 9th 13:07, we received credentials to the security.corp.millenialpower.us (10.0.1.60), through which we could access the 10.0.5.0/24 and 10.0.10.0/24 subnets. On 10.0.5.50, we found an unauthenticated VNC service which gave us administrative access to the dam monitoring system. Due to the sensitive nature of the dam monitoring system, we only performed a surface-level review of the system in the interest of protecting existing business operations.

We also noted an outdated web server running on one of the systems, on which we used a known exploit to remotely execute code.

Finally, we found an exposed Java Debugging Port on 10.0.5.75, on which we could gain Remote Code Execution. This host was running the Billing and Invoicing System. We accessed the database for these systems with credentials from the configuration files for the software.

Conclusion

NGPEW's security posture has improved significantly since our last test, as demonstrated by the reduced number and severity of our findings. All vulnerabilities can only be exploited from within the NGPEW network, and we could not access the network from outside. At this point, repairing the vulnerabilities will be for the purpose of preventing damage and privilege escalation should an attacker access the network.

Appendices

A. Findings Remediated Since Last Test

Finding	Remediation Status
Exposed Credentials Leading to Domain Compromise	**Remediated**
Domain Administrator Credentials Leaked in Rocket.Chat	**Remediated**
Plaintext Credentials on Employee Workstations	**Not Tested**
Remote Code Execution on Mantis Bug Tracker	**Remediated**
Remote Code Execution on Outdated Web Server	**Not Remediated**
VNC Access to Dam Monitoring System	**Not Remediated**
Open MongoDB	**Remediated**
Publicly Exposed Employee Github Credentials	**Remediated**
Weak Password for Redis Database	**Not Tested**
Data Exposure on Internal Ticketing System	**Remediated**
Easily Guessable Passwords	**Remediated**
Exposed MODBUS Devices	**Remediated**
PLC Information Disclosure	**Remediated**

B. Tools Used

- Metasploit
- BurpSuite Community Edition
- Dirb
- Dirbuster
- Nmap
- Curl
- Fping
- Mimikatz
- Kerbrute
- Impacket
- FFUF
- SMBClient
- RPCClient
- Enum4linux
- ldapsearch