Penetration Testing Report

TNTECH CPTC Tryout 2026

Prepared By: Team 20

Date: September 13, 2026

Version: 1.0

Statement of Confidentiality

The contents of this document have been developed by Team 20. The contents of this document are considered to be proprietary and business confidential information. This information is to be used only in the performance of its intended use. This document may not be released to another vendor, business partner or contractor without prior written consent from Team 20. Additionally, no portion of this document may be communicated, reproduced, copied or distributed without the prior consent of Team 20.

The contents of this document do not constitute legal advice. Team 20's offer of services that relate to compliance, litigation or other legal interests are not intended as legal counsel and should not be taken as such.

Table of Contents

Contact Information

Obsidian Contacts	
Contact	**Title**
Denise Calder	Chief Executive Officer
Kent Ortega	Chief Technical Officer
Hank Mercer	Chief Information Security Officer
Rhett Flores	Summer Intern

Assessor Contact		
Assessor Name	**Title**	**Assessor Contact Email**
Team 20		

Executive Overview

Executive Summary

Obsidian Rift (“Obsidian” herein) contracted Team 20 to perform a Network Penetration Test of Obsidian’s internal network to identify security weaknesses, determine the impact to Obsidian, document all findings in a clear and repeatable manner, and provide remediation recommendations.

During the penetration test against Obsidian, Team 20 identified 12 findings that threaten the confidentiality, integrity, and availability of Obsidian's information systems. The findings were categorized by severity level, with 6 of the findings being assigned a critical-risk rating, 3 high-risk, and 2 medium-risk findings. There was also 1 informational finding related to enhancing security monitoring capabilities within the internal network.

To provide a comprehensive view of the organization’s security posture, this report highlights both existing security strengths and opportunities for improvement. It also includes a remediation roadmap to prioritize key initiatives and guide the organization’s continued efforts to strengthen its overall security posture.

Obsidian should create a remediation plan based on the Remediation Summary section of this report, addressing all high findings as soon as possible according to the needs of the business. Obsidian should also consider performing periodic vulnerability assessments.

Key Security Strengths

- The containerization topology creates segmentation and isolation between services, as well as a central management service to easily implement security controls.

Key Areas of Improvement

- The primary area for improvement is the prevalence of unauthenticated services that allow users to view, modify, and manage sensitive information and critical infrastructure without appropriate access controls. Implementing proper authentication and authorization mechanisms would significantly reduce the risk of unauthorized access and improve the organization’s overall security posture.
- Another area for improvement is the organization’s overall approach to password security and management. Strengthening employee training around the importance of strong, unique passwords and effective password management practices would significantly reduce security risks and improve the organization’s overall security posture.

Summary of Findings

During the course of testing, Team 20 uncovered a total of 12 findings that pose a material risk to Obsidian's information systems. Team 20 also identified 1 informational findings that, if addressed, could further strengthen Obsidian's overall security posture. Informational findings are observations for areas of improvement by the organization and do not represent security vulnerabilities on their own. The below chart provides a summary of the findings by severity level.

In the course of this penetration test **6 Critical**, **3 High**, **2 Medium** and **1 Info** vulnerabilities were identified:

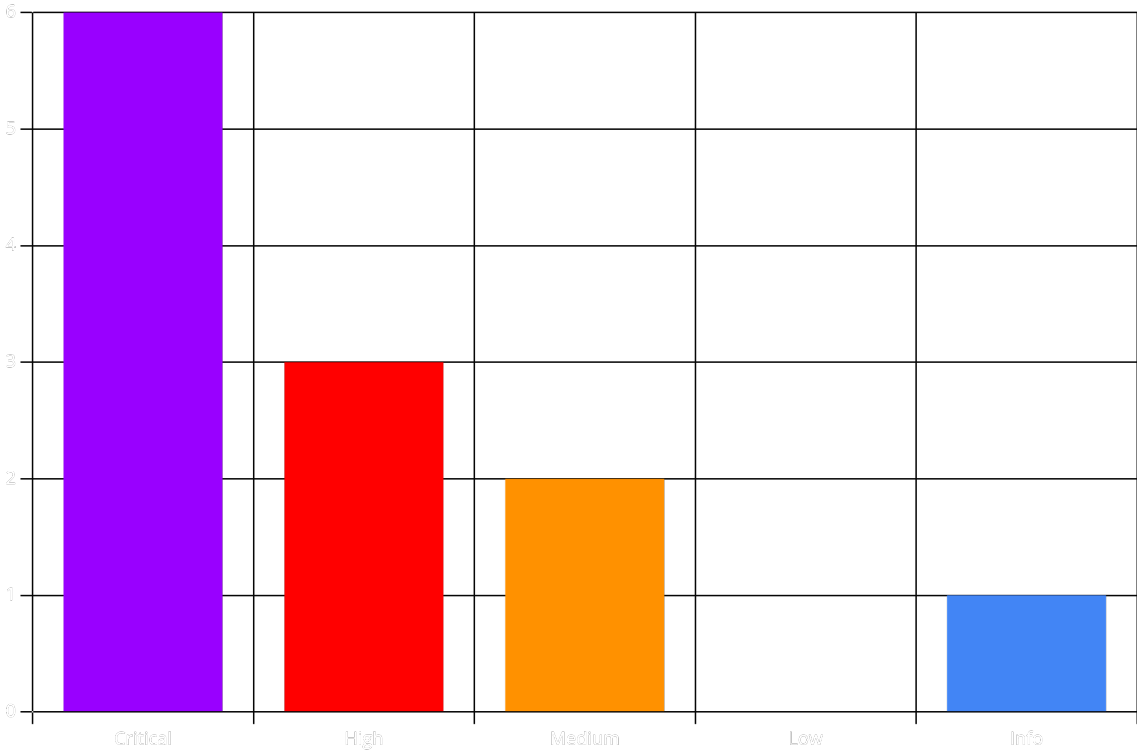


Figure 1 - Distribution of identified vulnerabilities

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings Details section of this report.

#	Severity Level	Finding Name	Page
	9.9 (Critical)	Credential Reuse	
	9.9 (Critical)	Credentials Exposed to Kubernetes Service Account	
	9.9 (Critical)	Exposed Kubernetes Service Account	
	9.9 (Critical)	Unessessary Kubernetes Service Account Permissions	
	9.8 (Critical)	Insecure Critical Service Credentials	
	9.8 (Critical)	Unauthenticated Developer API	
	8.6 (High)	Unauthenticated Record Access	
	8.2 (High)	Unauthenticated NFS Share With Improper Permissions	

#	Severity Level	Finding Name	Page
	7.5 (High)	NFS Share Containing Credentials	
	6.8 (Medium)	Authenticated XSS	
	6.3 (Medium)	Weak Password Policy	
	0.0 (Info)	SSH Password Authentication	

Remediation Summary

As a result of this assessment there are several opportunities for Obsidian to strengthen its internal network security. Remediation efforts are prioritized below starting with those that will likely take the least amount of time and effort to complete. Obsidian should ensure that all remediation steps and mitigating controls are carefully planned and tested to prevent any service disruptions or loss of data.

Immediate

Unauthenticated Developer API

- Credential Reuse Credentials Exposed to Kubernetes Service Account - Rotate exposed credentials
- Unnecessary Kubernetes Service Account Permissions - Remove unnecessary permissions
- NFS Share Containing Credentials - Remove and rotate SSH credentials from the NFS share
- Insecure Critical Service Credentials - Rotate the OpenPLC Credentials
- Unauthenticated Record Access - Temporarily restrict or disable access while authentication can be orchestrated due to the critical security implications
- Unauthenticated Developer API - Remove write access to the NFS network share

1 - 2 Months

- SSH Password Authentication - Migrate SSH from password to key authentication
- Exposed Kubernetes Service Account - Remove and rotate service account token from containers that don't require Kubernetes API access
- Unauthenticated NFS Share With Improper Permissions - Restrict NFS share access to trusted hosts
- Authenticated XXS - Add content escaping and filtering to the News field on the admin panel of the WordPress website.

6 Months - 1 Year

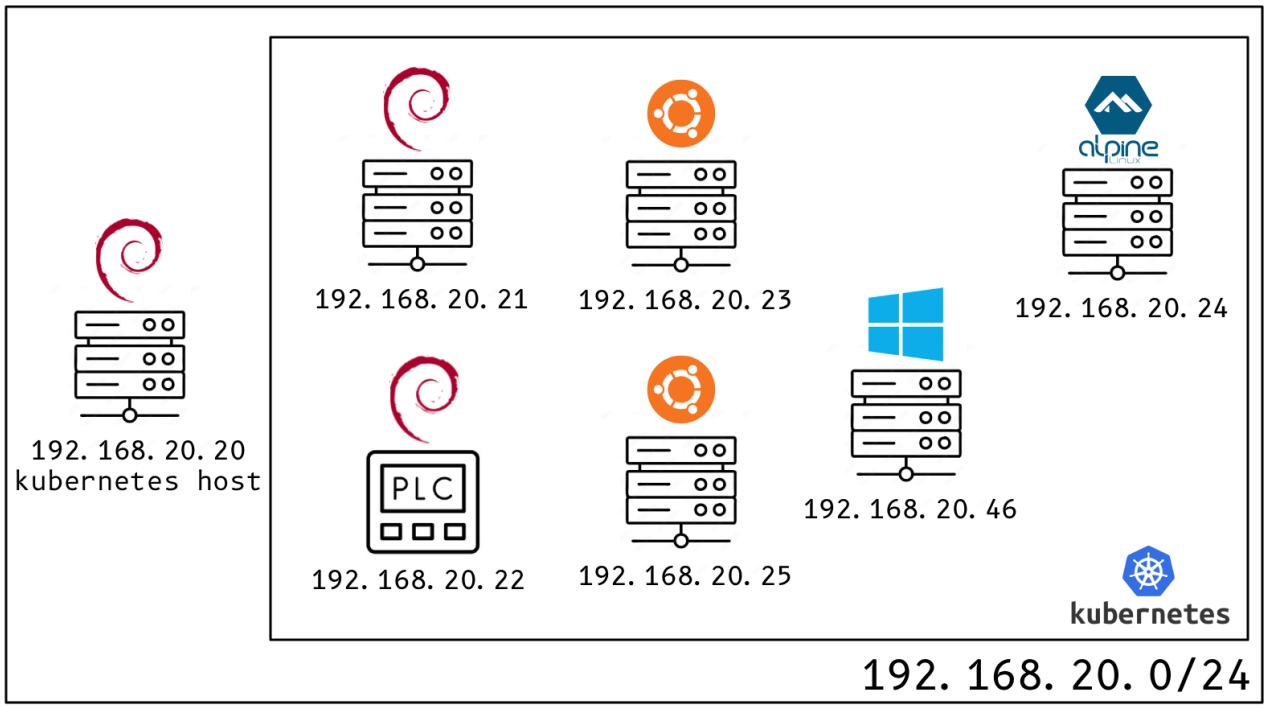
- Weak Password Policy - Perform ongoing internal network vulnerability assessments and password audits
- Educate systems and network administrators and developers on security hardening best practices

Engagement Outline

Methodology

Team 20 performed testing under a “Black Box” approach from September 12, 2026, to September 13, 2026 without credentials or any advance knowledge of Obsidian’s externally facing environment with the goal of identifying unknown weaknesses. Testing was performed from a non-evasive standpoint with the goal of uncovering as many misconfigurations and vulnerabilities as possible. Testing was performed remotely from Team 20's assessment labs. Each weakness identified was documented and manually investigated to determine exploitation possibilities and escalation potential. Team 20 sought to demonstrate the full impact of every vulnerability, up to and including internal domain compromise.

Network Diagram



Scope

The scope of this assessment was one internal network range and any cloud infrastructure discovered during the penetration test.

In Scope Assets

Host/URL/IP Address	Description
192.168.20.0/24	Obsidian internal network

Technical Findings

Critical Findings

Credential Reuse

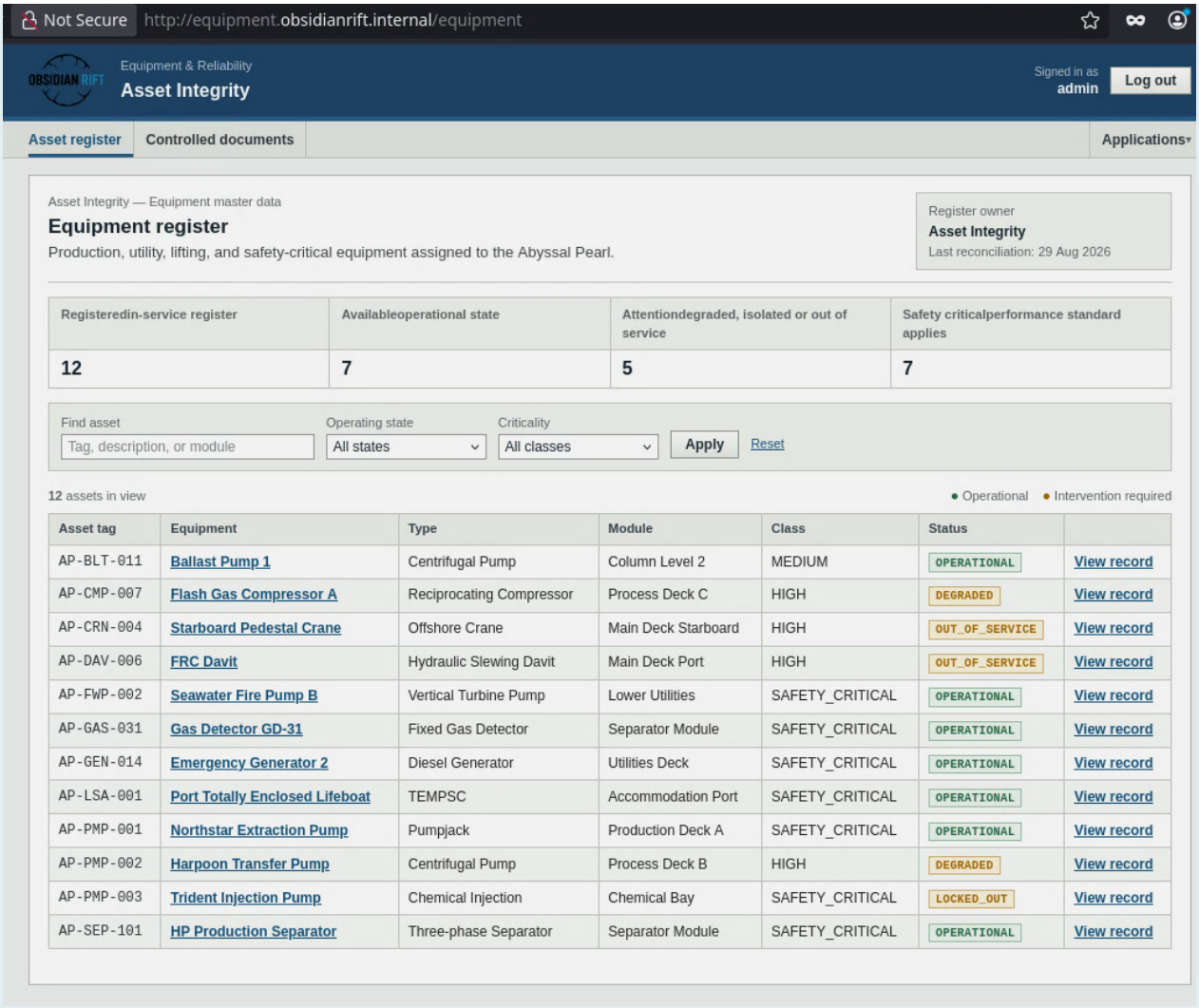
CVSS Score	**9.9**
CWE	CWE-521 - Weak Password Requirements
CVSS String	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
Affected Scope	• 192.168.20.22 • 192.168.20.23 • 192.168.20.24 • 192.168.20.25
Likelihood	The likelihood of exploitation is high, as any one of the services could be compromised and result in access to every service.
Technical Impact	Authentication can be bypassed on every service after the compromise of a user on one host.
Business Impact	The reuse of credentials increases the amount and likelihood of sensitive data exposure in the event of a compromise.
Overview	n/a
Summary	Many credentials are reused across many services, allowing for easier initial compromise and catastrophic data exposure when one account is compromised.

Finding Evidence

Example of credential resuse with admin across boxes and services:

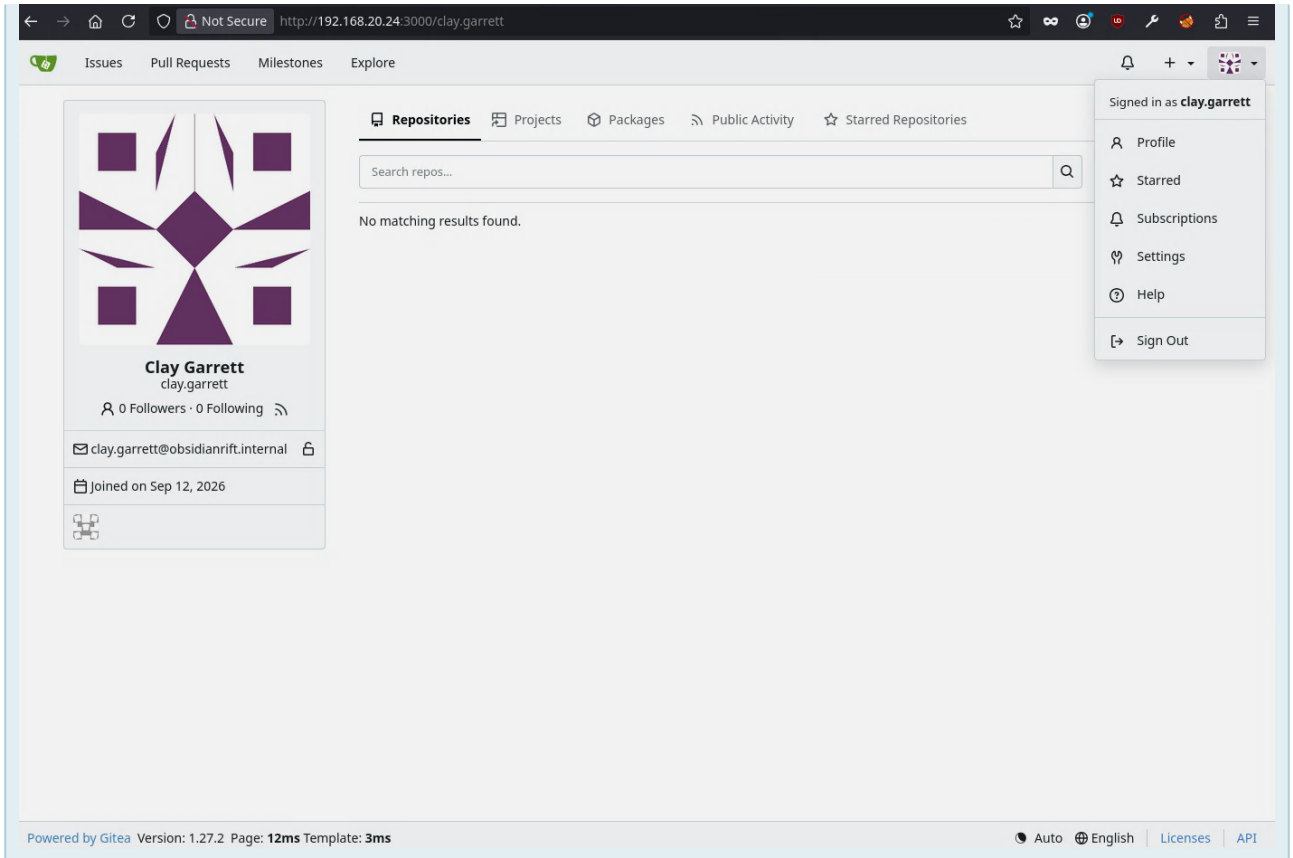
```
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[~]
└──╼ $sshpass -p '[redacted]' ssh admin@192.168.20.22 id
uid=1000(admin) gid=1000(admin) groups=1000(admin)
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[~]
└──╼ $sshpass -p '[redacted]' ssh admin@192.168.20.23 id
uid=2001(admin) gid=2001(admin) groups=2001(admin)
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[~]
└──╼ $sshpass -p '[redacted]' ssh admin@192.168.20.24 id
┌─[✗]─[user@CPTC-2026-Tryout-team20-jumpbox]─[~]
└──╼ $sshpass -p '[redacted]' ssh admin@192.168.20.25 id
uid=10002(admin) gid=10002(admin) groups=10002(admin)
```

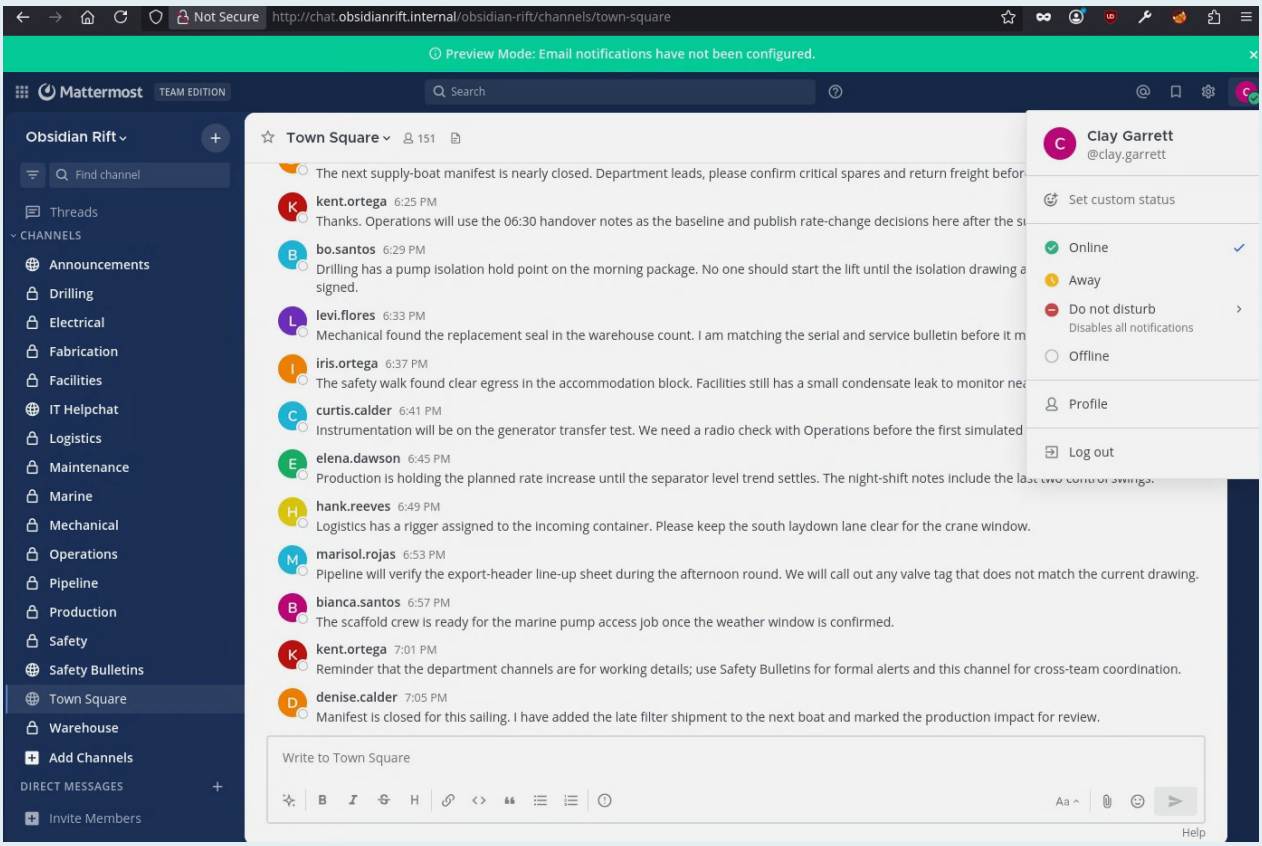
(same password)

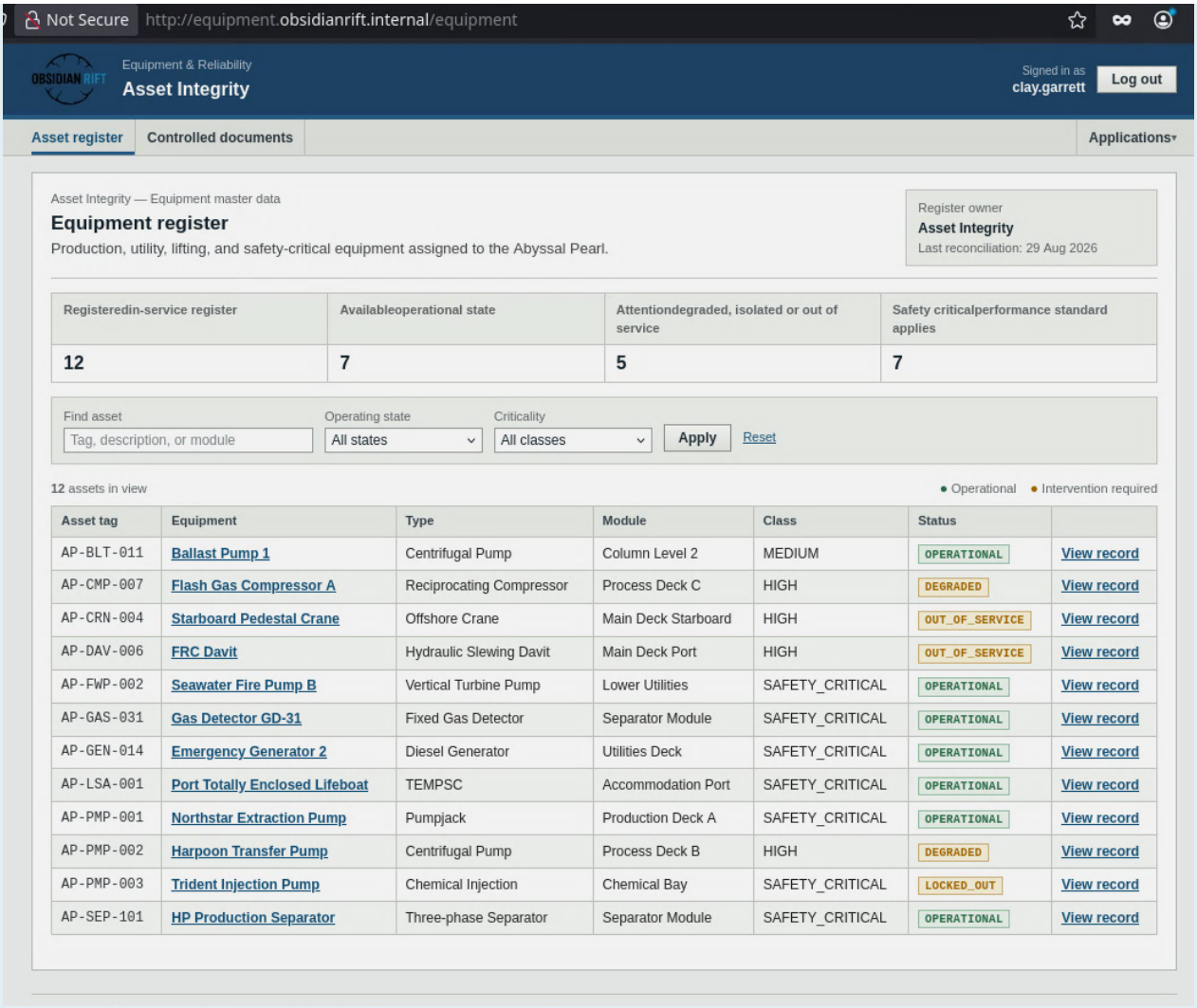


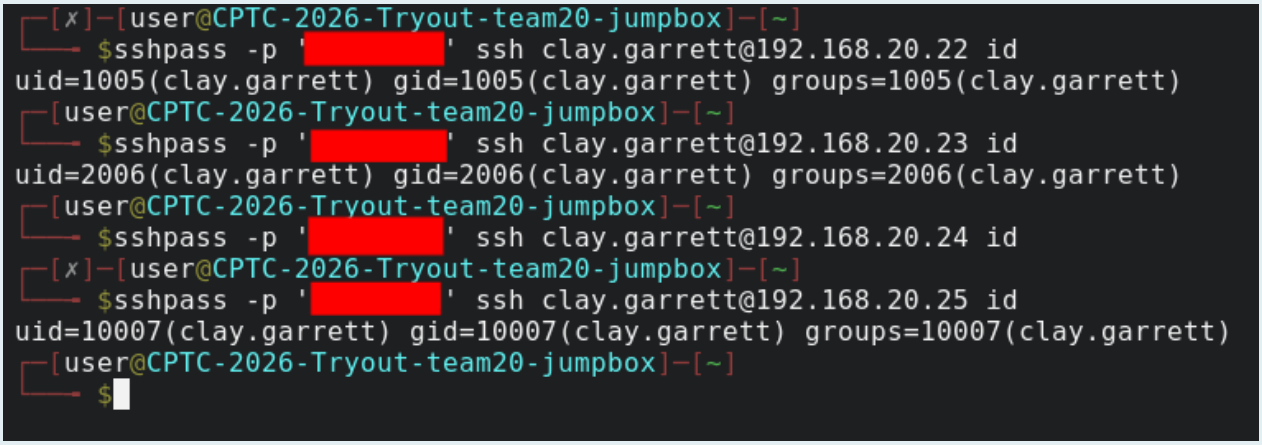
Other repeated credentials that provide access to all of linux boxes and some additional services:

```
[
  {
    "name": "Rig Operations Admin",
    "email": "operations@obsidianrift.internal",
    "password": "[redacted]",
    "is_admin": true
  },
  {
    "name": "Wyatt Dawson",
    "email": "wyatt.dawson@obsidianrift.internal",
    "password": "[redacted]",
    "is_admin": true
  },
  {
    "name": "Clay Garrett",
    "email": "clay.garrett@obsidianrift.internal",
    "password": "[redacted]",
    "is_admin": true
  },
  {
    "name": "Offshore Contractor",
    "email": "contractor@obsidianrift.internal",
    "password": "[redacted]",
    "is_admin": false
  }
]
```


Not Secure
http://192.168.20.24:3000/clay.garrett
Issues
Pull Requests
Milestones
Explore
Signed in as clay.garrett
Profile
Starred
Subscriptions
Settings
Help
Sign Out
Repositories
Projects
Packages
Public Activity
Starred Repositories
Search repos...
No matching results found.
Clay Garrett
clay.garrett
0 Followers · 0 Following
clay.garrett@obsidianrift.internal
Joined on Sep 12, 2026
Powered by Gitea Version: 1.27.2 Page: 12ms Template: 3ms
Auto
English
Licenses
API


Not Secure
http://chat.obsidianrift.internal/obsidian-rift/channels/town-square
Preview Mode: Email notifications have not been configured.
Mattermost
TEAM EDITION
Search
Obsidian Rift
Find channel
Threads
CHANNELS
Announcements
Drilling
Electrical
Fabrication
Facilities
IT Helpchat
Logistics
Maintenance
Marine
Mechanical
Operations
Pipeline
Production
Safety
Safety Bulletins
Town Square
Warehouse
Add Channels
DIRECT MESSAGES
Invite Members
Town Square
151
kent.ortega 6:25 PM
Thanks. Operations will use the 06:30 handover notes as the baseline and publish rate-change decisions here after the
bo.santos 6:29 PM
Drilling has a pump isolation hold point on the morning package. No one should start the lift until the isolation drawing
signed.
levi.flores 6:33 PM
Mechanical found the replacement seal in the warehouse count. I am matching the serial and service bulletin before it
iris.ortega 6:37 PM
The safety walk found clear egress in the accommodation block. Facilities still has a small condensate leak to monitor
curtis.calder 6:41 PM
Instrumentation will be on the generator transfer test. We need a radio check with Operations before the first simulated
elena.dawson 6:45 PM
Production is holding the planned rate increase until the separator level trend settles. The night-shift notes include the
hank.reeves 6:49 PM
Logistics has a rigger assigned to the incoming container. Please keep the south laydown lane clear for the crane window.
marisol.rojas 6:53 PM
Pipeline will verify the export-header line-up sheet during the afternoon round. We will call out any valve tag that does not match the current drawing.
bianca.santos 6:57 PM
The scaffold crew is ready for the marine pump access job once the weather window is confirmed.
kent.ortega 7:01 PM
Reminder that the department channels are for working details; use Safety Bulletins for formal alerts and this channel for cross-team coordination.
denise.calder 7:05 PM
Manifest is closed for this sailing. I have added the late filter shipment to the next boat and marked the production impact for review.
Write to Town Square
Help
Clay Garrett
@clay.garrett
Set custom status
Online
Away
Do not disturb
Disables all notifications
Offline
Profile
Log out





Recommendation

- Provide training to employees to raise awareness of the importance of using unique passwords for different accounts.
- All passwords should be changed to unique passwords throughout the organization.

Credentials Exposed to Kubernetes Service Account

CVSS Score	**9.9**
CWE	CWE-312 - Cleartext Storage of Sensitive Information
CVSS String	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
Affected Scope	• 192.168.20.22 • 192.168.20.23 • 192.168.20.24 • 192.168.20.25
Likelihood	This exploit relies on the compromise of a box at the user level, the discovery of the Kubernetes user account, and the discovery of the secrets. However, each individual step is not high in complexity, leading to a significant likelihood of exploitation.
Technical Impact	The credentials revealed by the secret would allow an attacker to gain access to all of the available services on the internal network.
Business Impact	The exploitation of this vulnerability could cause the leak of all of the company's stored PII and information accessible by internal services at any authentication level.
Overview	Kubernetes is a service that manages a cluster of connected servers. Compromise of any one node can cause the whole network to be compromised if the cluster isn't configured correctly.
Summary	The Kubernetes service account has access to secrets containing a multitude of credentials.

Finding Evidence

Secret exposed through service token:

Exposed reused credentials:

Recommendation

- Plaintext credentials should be removed from the Kubernetes secrets.
- All exposed credentials should be rotated.

Exposed Kubernetes Service Account

CVSS Score	**9.9**
CWE	CWE-522 - Insufficiently Protected Credentials
CVSS String	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
Affected Scope	• 192.168.20.21 • 192.168.20.22 • 192.168.20.23 • 192.168.20.24 • 192.168.20.25
Likelihood	The exploit requires the attack to first gain user-level access to any of the machines. There is significant risk in the wide availability of this exploit, as any service compromise compromises the cluster.
Technical Impact	Any compromised container allows an attacker to have some level of access to the cluster.
Business Impact	The ability to reach deeper into the cluster increases the risk to sensitive data located on other machines.
Overview	Kubernetes is a service that manages a cluster of connected servers. Compromise of any one node can cause the whole network to be compromised if the cluster isn't configured correctly.
Summary	The service account token is mounted into every container. This allows an attacker to use any compromised host as a way to attack the cluster using the Kubernetes API, depending on the permissions of the service account.

Finding Evidence

Recommendation

- If access to the Kubernetes API is required, access to the service credentials should be restricted to a more privileged user.
- If the API is not being used by a container, then the volume containing the token should not be mounted at all.

Unessessary Kubernetes Service Account Permissions

CVSS Score	**9.9**
CWE	CWE-269 - Improper Privilege Management
CVSS String	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
Affected Scope	• 192.168.20.22 • 192.168.20.23 • 192.168.20.24 • 192.168.20.25
Likelihood	This exploit relies on the compromise of a box at the user level, the discovery of the Kubernetes user account, and the discovery of the execute permission. However, each individual step is not high in complexity, leading to a significant likelihood of exploitation.
Technical Impact	The ability to run arbitrary commands on containers allows for privilege escalation to the root user on box 192.168.20.22. This also allows for RCE on the other boxes.
Business Impact	The privilege escalation potential of the execute permission creates a risk of more data loss and exfiltration, as the attacker can control the entire box.
Overview	Kubernetes is a service that manages a cluster of connected servers. Compromise of any one node can cause the whole network to be compromised if the cluster isn't configured correctly.
Summary	The Kubernetes Service account accessible to every box has permission to run commands on pods.

Finding Evidence

Exploitation and rooting of host 192.168.20.22:

```
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[~]
└──╼ $TOKEN="$(cat kube_token)"
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[~]
└──╼ $APISERVER="https://192.168.20.21:6443"
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[~]
└──╼ $./kubectl --server="$APISERVER"   --certificate-authority=ca.crt   --token="$TOKEN"   -n obsidian-rift   exec -it abyssal-pearl-65d75794df-t84t5   -c portal -
- /bin/sh
# bash
root@abyssal-pearl-65d75794df-t84t5:/app# id
uid=0(root) gid=0(root) groups=0(root)
root@abyssal-pearl-65d75794df-t84t5:/app# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host proto kernel_lo
       valid_lft forever preferred_lft forever
2: eth0@if59: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UP group default qlen 1000
    link/ether 1e:5d:6a:47:f8:c1 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.42.0.23/24 brd 10.42.0.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::1c5d:6aff:fe47:f8c1/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
3: lan0@if2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1000
    link/ether 00:16:3e:b0:1d:dc brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.20.22/24 brd 192.168.20.255 scope global lan0
       valid_lft forever preferred_lft forever
    inet6 fd42:de52:efc9:460a:16:3e00:9b0:1ddc/64 scope global dynamic mngtmpaddr proto kernel_ra
       valid_lft forever preferred_lft forever
    inet6 fe80::16:3e00:9b0:1ddc/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
root@abyssal-pearl-65d75794df-t84t5:/app#
```

Permission list:

```
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[~]
└──╼ $kubectl   --server="https://192.168.20.20:6443"   --certificate-authority=ca.crt   --token="$TOKEN"   auth can-i --list -n obsidian-rift
Resources                                       Non-Resource URLs                      Resource Names   Verbs
pods/exec                                       []                                     []               [create]
selfsubjectreviews.authentication.k8s.io        []                                     []               [create]
selfsubjectaccessreviews.authorization.k8s.io   []                                     []               [create]
selfsubjectrulesreviews.authorization.k8s.io    []                                     []               [create]
configmaps                                      []                                     []               [get list watch]
pods/log                                        []                                     []               [get list watch]
pods                                            []                                     []               [get list watch]
secrets                                         []                                     []               [get list watch]
                                                [/.well-known/openid-configuration/]   []               [get]
                                                [/.well-known/openid-configuration]    []               [get]
                                                [/api/*]                               []               [get]
                                                [/api]                                 []               [get]
                                                [/apis/*]                              []               [get]
                                                [/apis]                                []               [get]
                                                [/healthz]                             []               [get]
                                                [/healthz]                             []               [get]
                                                [/livez]                               []               [get]
                                                [/livez]                               []               [get]
                                                [/openapi/*]                           []               [get]
                                                [/openapi]                             []               [get]
                                                [/openid/v1/jwks/]                     []               [get]
                                                [/openid/v1/jwks]                      []               [get]
                                                [/readyz]                              []               [get]
                                                [/readyz]                              []               [get]
                                                [/version/]                            []               [get]
                                                [/version/]                            []               [get]
                                                [/version]                             []               [get]
                                                [/version]                             []               [get]
```

Recommendation

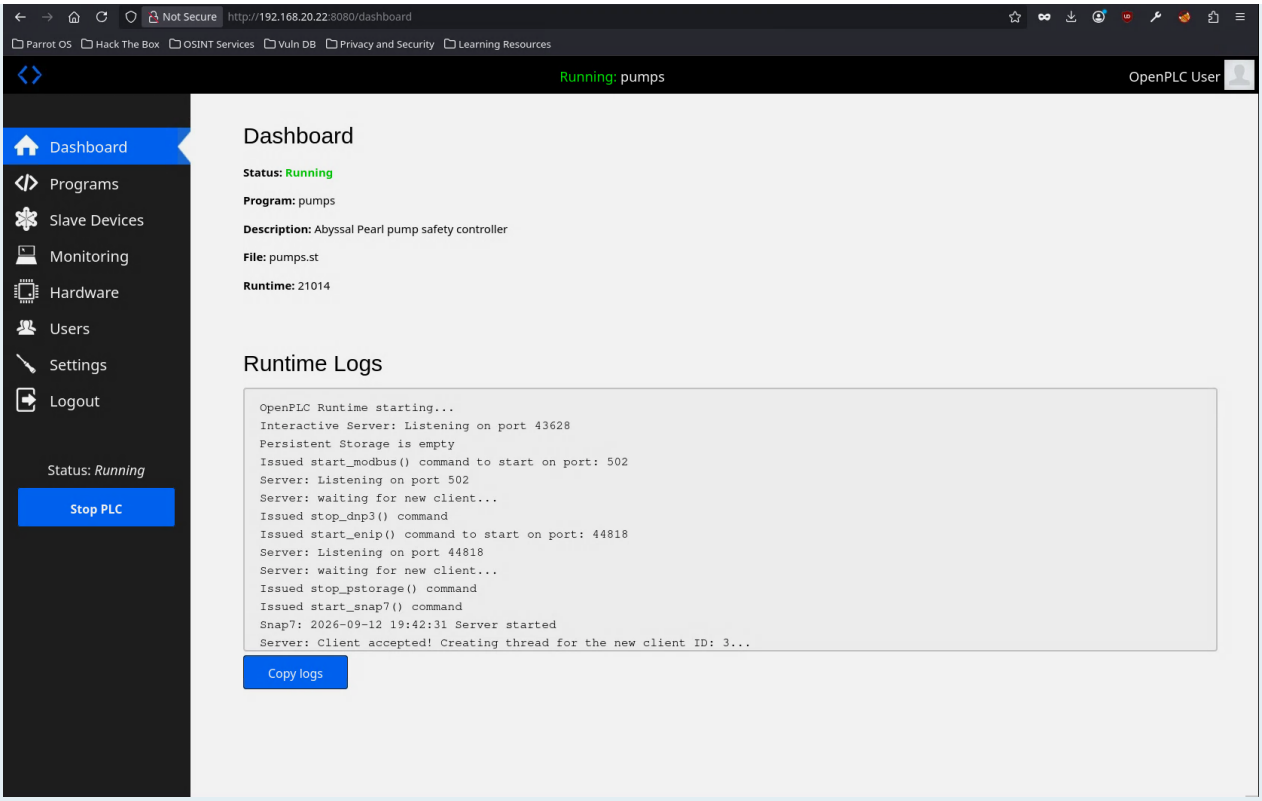
- The permission `pods/exec` should be removed from the Service User

Insecure Critical Service Credentials

CVSS Score	**9.8**
CWE	CWE-1391 - Use of Weak Credentials
CVSS String	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/CR:H/IR:H/AR:H
Affected Scope	192.168.20.22
Likelihood	The OpenPLC server can be enumerated through a simple port scan of the network and has very simple login information. This makes exploitation likely.
Technical Impact	Access to this service would allow an attacker to have full control of the physical infrastructure of the company.
Business Impact	Unauthorized access to this service could destroy the company by inhibiting operations and causing harm to employees.
Overview	OpenPLC is a web server that manages Programmable Logic Controllers (PLCs). PLCs control physical systems like valves and pumps, making them incredibly important to secure.
Summary	The OpenPLC service's credentials are extremely insecure and guessable, as well as being accessible through a low-complexity chain of attacks. An attacker with access to this service could put employees in danger and destroy availability by halting all operations.

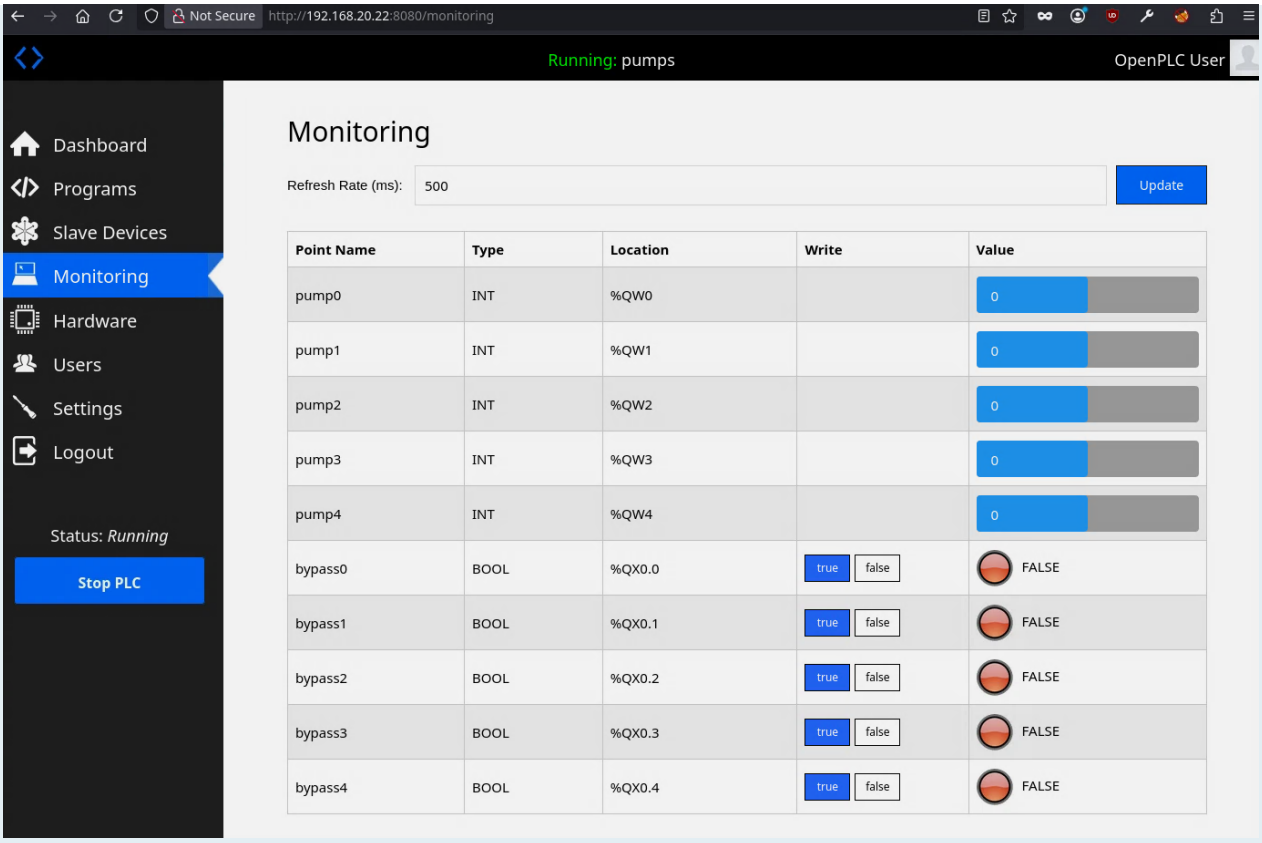
Finding Evidence

Login with weak credentials:



Credential access via Kubernetes service account:

Ability to disable bypass valves:



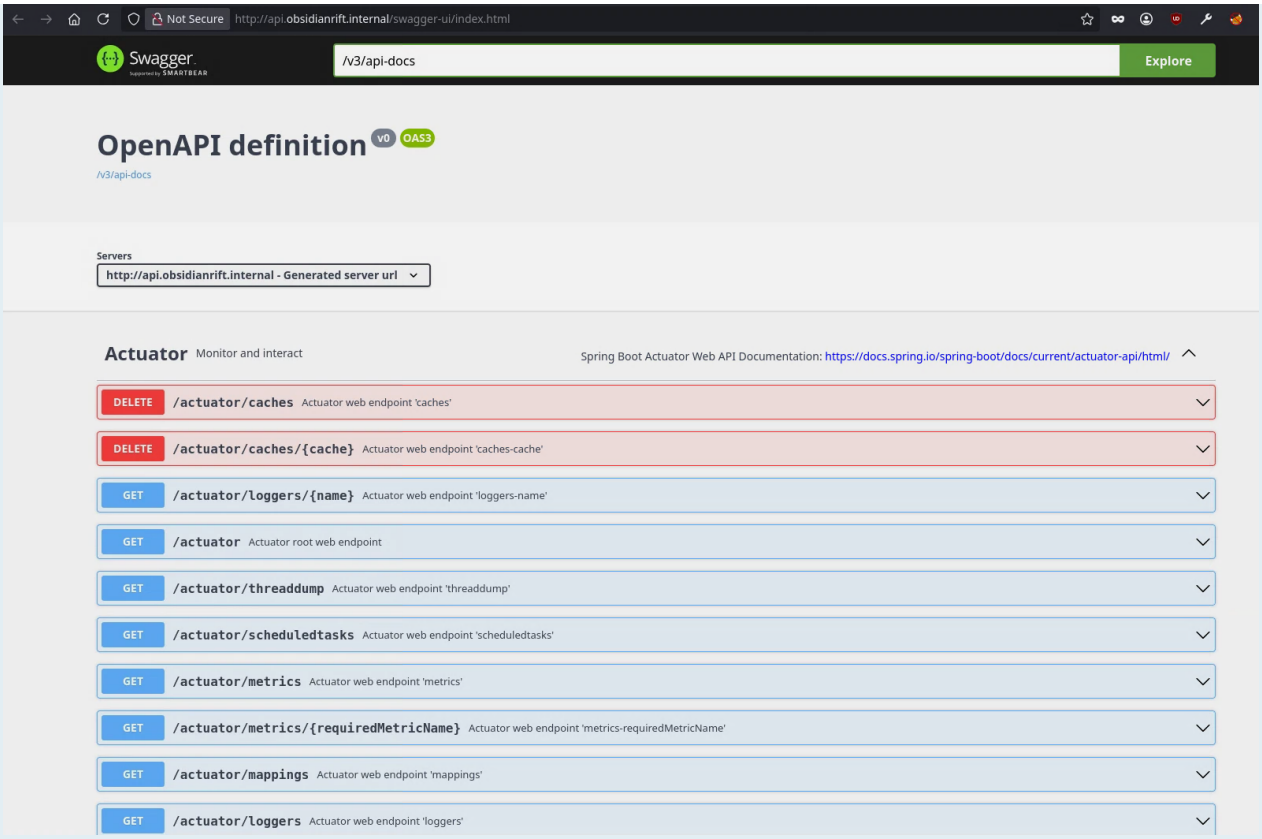
Recommendation

- Change the credentials for the OpenPLC server

Unauthenticated Developer API

CVSS Score	**9.8**
CWE	CWE-862 - Missing Authorization
CVSS String	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Affected Scope	192.168.20.25 - api.obsidianrift.internal
Likelihood	This security vulnerability has a high chance of exploitation due to the simple discovery of the api endpoint and the accompanying documentation, which reveals everything an attacker could do with unauthenticated access.
Technical Impact	An unauthenticated user can write to profiles, access a multitude of database and user credentials, and cause shutdowns.
Business Impact	Access to this API could damage infrastructure, reveal personally identifiable information about any of the employees, and leak business information. An attacker could issue commands that result in harm to employees, irreparably damaging the company.
Overview	A web application programming interface, or API, allows web requests to interact with a server in a multitude of ways. For example, APIs can be used to fetch data from a server, request that the server change or delete data, or trigger an arbitrary event.
Summary	The API endpoint allows unauthenticated access to sensitive information about critical infrastructure, exposes credentials, allows modification of data, and provides limited control over OT infrastructure.

Finding Evidence

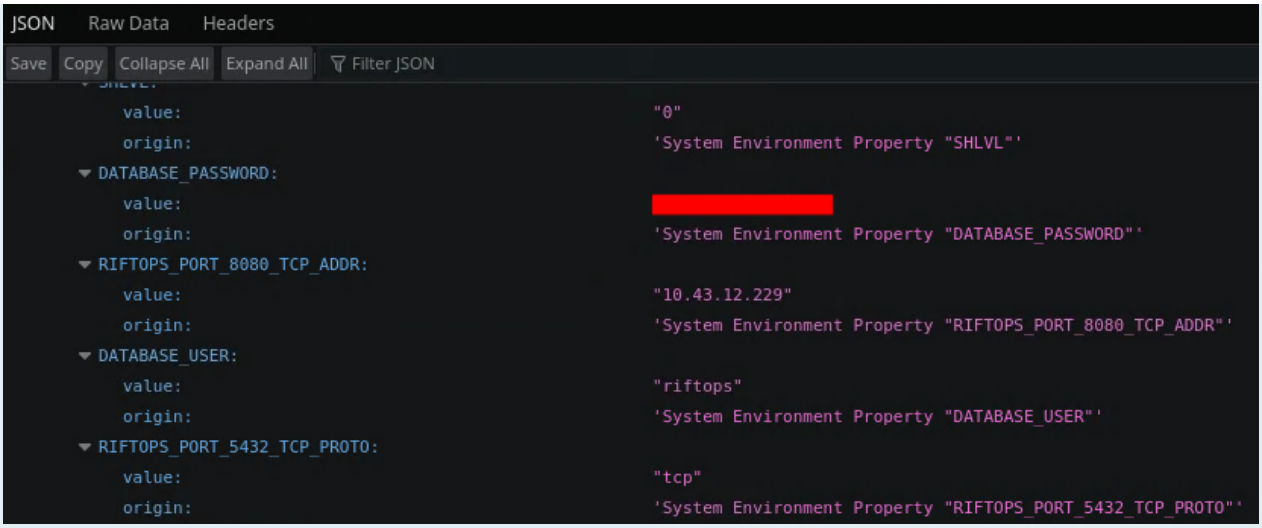


`http://api.obsidianrift.internal/api/equipment/1`

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
-<Equipment>
   <id>1</id>
   <assetTag>AP-PMP-001</assetTag>
   <name>Northstar Extraction Pump</name>
   <type>Pumpjack</type>
   <location>Production Deck A</location>
   <serialNumber>OR-NP-88214</serialNumber>
   <status>OPERATIONAL</status>
   <criticality>SAFETY_CRITICAL</criticality>
  -<description>
      Primary extraction pump for the north production manifold.
   </description>
   <manualPath>manuals/northstar-service-manual.txt</manualPath>
</Equipment>
```

`http://api.obsidianrift.internal/actuator/env`



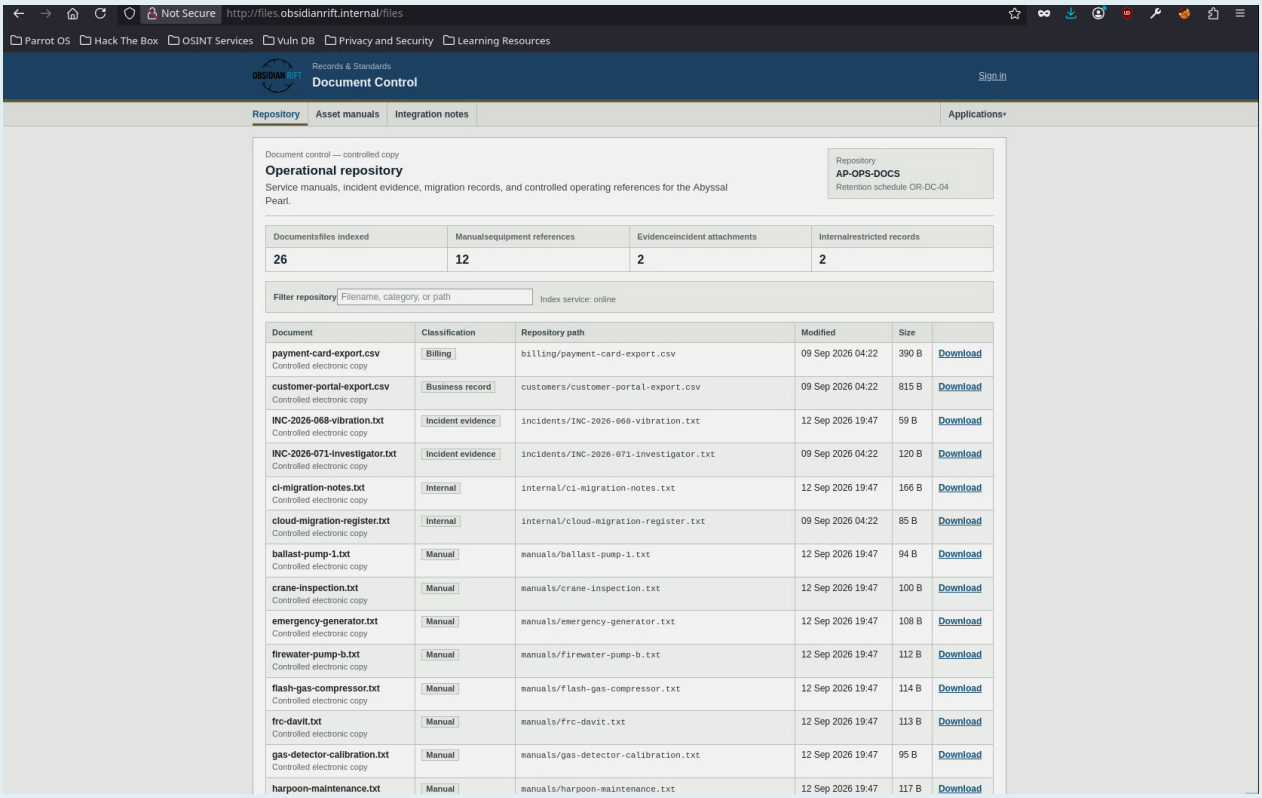
Recommendation

Both the API and the documentation should have authentication to prevent the very sensitive credentials and operations from being accessed by an unauthorized user.

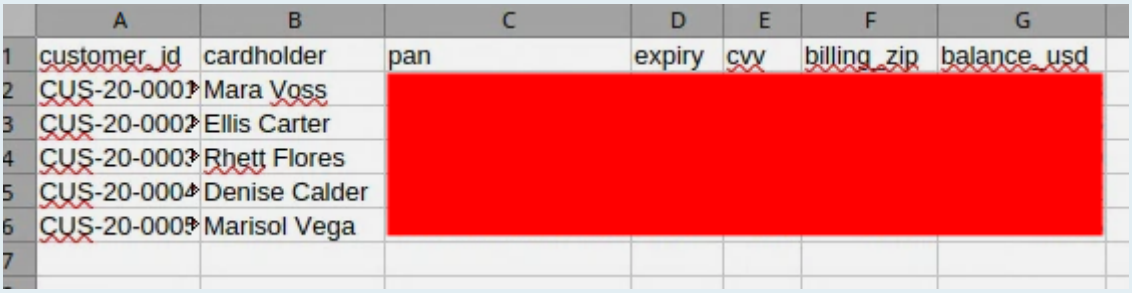
High Findings

Unauthenticated Record Access	
CVSS Score	**8.6**
CWE	CWE-862 - Missing Authorization
CVSS String	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:N/A:N
Affected Scope	192.168.20.25
Likelihood	The discovery and exploitation of this service is very likely due to the easy enumeration and lack of any security.
Technical Impact	The exposed credentials in the exposed files allow an attacker to gain more access to internal systems.
Business Impact	PII exposure can severely damage a company's reputation.
Overview	The file server allows storage of arbitrary documents.
Summary	The file server is exposed without any authentication. This allows an attacker to gain access to sensitive files containing PII, payment information, and credentials.

Finding Evidence

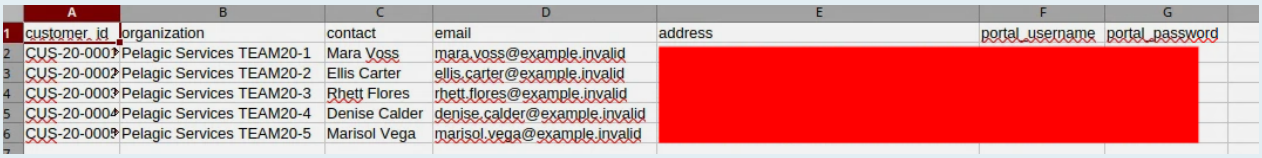


payment-card-export.cvs



	A	B	C	D	E	F	G
1	customer_id	cardholder	pan	expiry	cvv	billing_zip	balance_usd
2	CUS-20-0001	Mara Voss	[redacted]				
3	CUS-20-0002	Ellis Carter					
4	CUS-20-0003	Rhett Flores					
5	CUS-20-0004	Denise Calder					
6	CUS-20-0005	Marisol Vega					
7							

customer-portal-export.cvs



	A	B	C	D	E	F	G
1	customer_id	organization	contact	email	address	portal_username	portal_password
2	CUS-20-0001	Pelagic Services TEAM20-1	Mara Voss	mara.voss@example.invalid	[redacted]		
3	CUS-20-0002	Pelagic Services TEAM20-2	Ellis Carter	ellis.carter@example.invalid			
4	CUS-20-0003	Pelagic Services TEAM20-3	Rhett Flores	rhett.flores@example.invalid			
5	CUS-20-0004	Pelagic Services TEAM20-4	Denise Calder	denise.calder@example.invalid			
6	CUS-20-0005	Pelagic Services TEAM20-5	Marisol Vega	marisol.vega@example.invalid			
7							

Recommendation

- The domain `files.obsidianrift.internal` should be behind proper authentication.

Unauthenticated NFS Share With Improper Permissions

CVSS Score	**8.2**
CWE	CWE-732 - Incorrect Permission Assignment for Critical Resource
CVSS String	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:H/A:N
Affected Scope	192.168.20.23
Likelihood	The NFS share is highly visible on any network scan and can be easily enumerated and accessed without any other information, making exploitation very likely.
Technical Impact	Any files in the share can be accessed without authentication, including sensitive technical data.
Business Impact	Procedures can be silently modified and cause catastrophic failure in emergency and routine situations.
Overview	Network File Share or NFS allows a server to share files and directories over a network.
Summary	The NFS share is exposed to all users on the network, allowing an attacker to gain access and write any file. The share contains procedures, including emergency protocols, that an attacker could modify.

Finding Evidence

Enumeration of the network share:

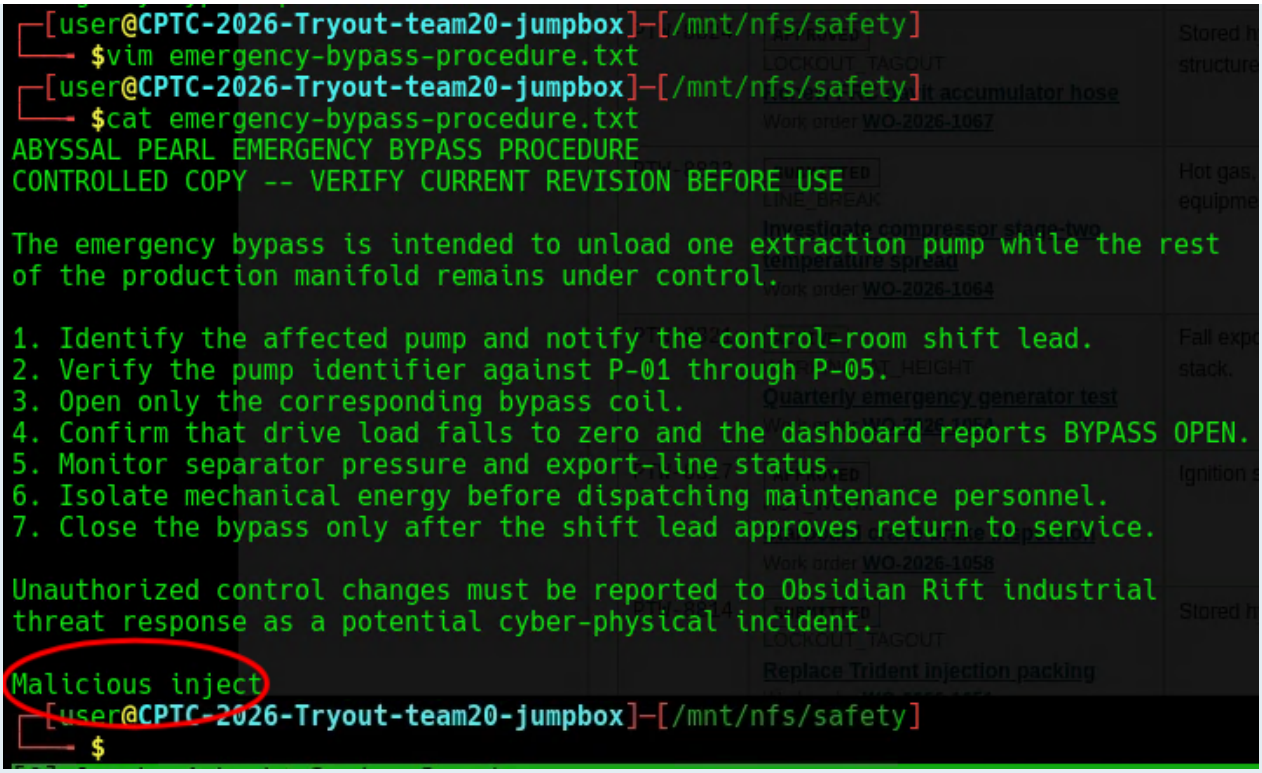
```
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[~/notes]
└──╼ $showmount -e 192.168.20.23
Export list for 192.168.20.23:
/srv/nfs/share *
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[~/notes]
└──╼ $sudo mount 192.168.20.23:/srv/nfs/share /mnt/nfs
sudo: unable to resolve host CPTC-2026-Tryout-team20-jumpbox: Temporary failure in name resolution
Created symlink '/run/systemd/system/remote-fs.target.wants/rpc-statd.service' → '/usr/lib/systemd/system/rpc-statd.service'.
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[~/notes]
└──╼ $cd /mnt/nfs
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[/mnt/nfs]
└──╼ $ls
maintenance  operations  safety  vendors
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[/mnt/nfs]
└──╼ $ls ./*
./maintenance:
pump-maintenance.csv

./operations:
night-shift-handover.txt  production-summary-Q3.txt

./safety:
emergency-bypass-procedure.txt

./vendors:
legacy-archive.conf  satellite-support.txt
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[/mnt/nfs]
└──╼ $
```

Modification of a vital procedure:



Recommendation

- Restrict access to the share to specific trusted hosts.
- Remove write permissions from the share.

NFS Share Containing Credentials

CVSS Score	**7.5**
CWE	CWE-522 - Insufficiently Protected Credentials
CVSS String	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Affected Scope	192.168.20.23
Likelihood	The NFS share is highly visible on any network scan and can be easily enumerated and accessed without any other information, making exploitation very likely.
Technical Impact	Access to this share exposes credentials that can be used to gain a foothold in multiple hosts and provide access to multiple services.
Business Impact	This share gives any unauthenticated user access to the company's internal procedures. The exposed credentials can be used to expose more of the company's information due to credential reuse.
Overview	Network File Share or NFS allows a server to share files and directories over a network.
Summary	Plaintext SSH credentials are stored on an NFS share.

Finding Evidence

Enumeration of the network share:

```
┌──[user@CPTC-2026-Tryout-team20-jumpbox]─[~/notes]
└─→ $showmount -e 192.168.20.23
Export list for 192.168.20.23:
/srv/nfs/share *
┌──[user@CPTC-2026-Tryout-team20-jumpbox]─[~/notes]
└─→ $sudo mount 192.168.20.23:/srv/nfs/share /mnt/nfs
sudo: unable to resolve host CPTC-2026-Tryout-team20-jumpbox: Temporary failure in name resolution
Created symlink '/run/systemd/system/remote-fs.target.wants/rpc-statd.service' → '/usr/lib/systemd/system/rpc-statd.service'.
┌──[user@CPTC-2026-Tryout-team20-jumpbox]─[~/notes]
└─→ $cd /mnt/nfs
┌──[user@CPTC-2026-Tryout-team20-jumpbox]─[/mnt/nfs]
└─→ $ls
maintenance  operations  safety  vendors
┌──[user@CPTC-2026-Tryout-team20-jumpbox]─[/mnt/nfs]
└─→ $ls ./*
./maintenance:
pump-maintenance.csv

./operations:
night-shift-handover.txt  production-summary-Q3.txt

./safety:
emergency-bypass-procedure.txt

./vendors:
legacy-archive.conf  satellite-support.txt
┌──[user@CPTC-2026-Tryout-team20-jumpbox]─[/mnt/nfs]
└─→ $
```

A configuration file exposes SSH credentials that are valid for the host.

```
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[/mnt/nfs]
└──╼ $cat vendors/legacy-archive.conf
# Obsidian Rift legacy archive agent
# Archive transfer settings provisioned by Offshore IT.

archive_host=files.obsidianrift.internal
archive_path=/srv/nfs/share
ssh_username=rhett.flores
ssh_password=[redacted]
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[/mnt/nfs]
└──╼ $
[0] 0:vim- 1:bash* 2:bash  3:bash  4:bash  5:bash
```

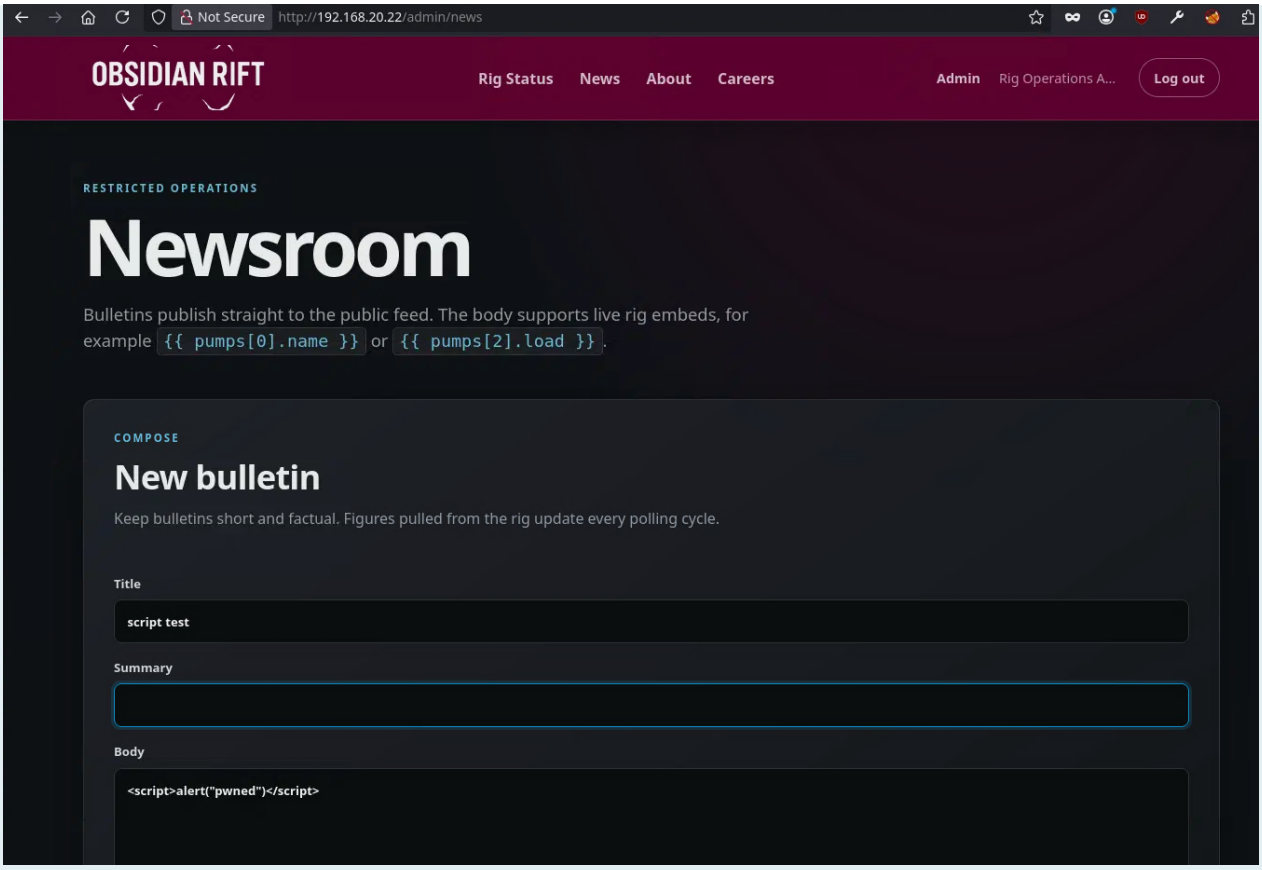
```
┌─[✗]─[user@CPTC-2026-Tryout-team20-jumpbox]─[/mnt/nfs]
└──╼ $ssh rhett.flores@192.168.20.23
rhett.flores@192.168.20.23's password:
rhett.flores@debian-2-565b5bd869-qhd5b:~$ id
uid=2004(rhett.flores) gid=2004(rhett.flores) groups=2004(rhett.flores)
rhett.flores@debian-2-565b5bd869-qhd5b:~$
```

Recommendation

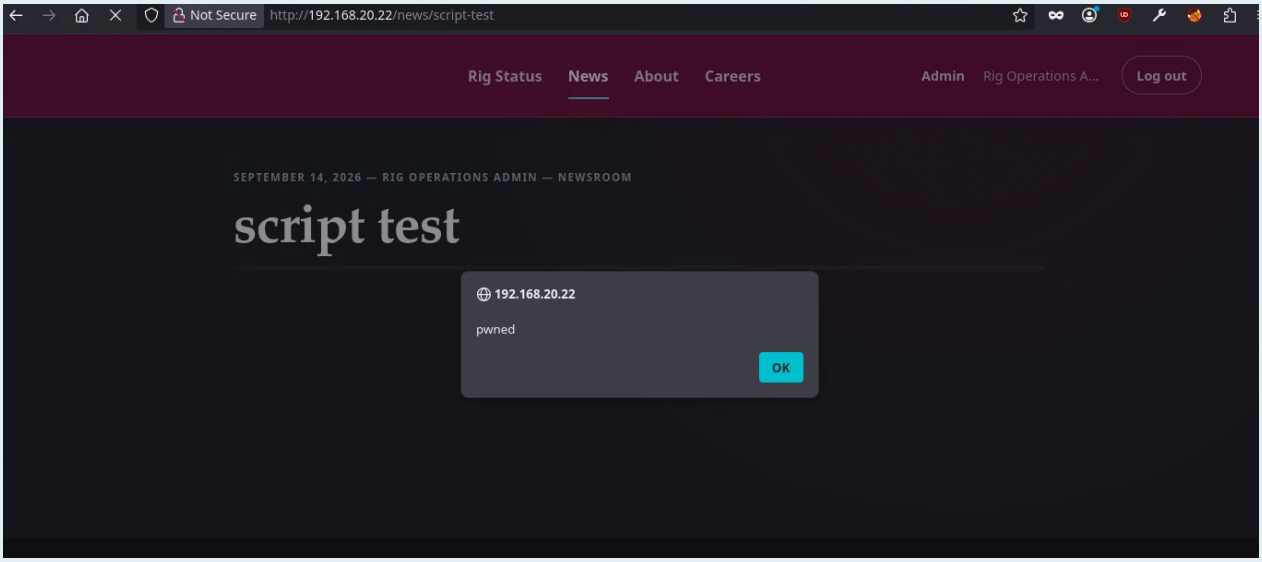
- Remove credentials from within the share.
- Restrict access to the share to specific trusted hosts.
  - Hosts can still be compromised; minimize the effect of a system compromise by storing credentials securely.
- Rotate exposed credentials

Medium Findings

Authenticated XSS	
CVSS Score	**6.8**
CWE	CWE-116 - Improper Encoding or Escaping of Output
CVSS String	CVSS:3.1/AV:N/AC:L/PR:H/UI:R/S:U/C:H/I:H/A:H
Affected Scope	192.168.20.22
Likelihood	This vulnerability is unlikely due to the prerequisite of a compromised admin account. However, if admin credentials are discovered by an attacker or there is an insider threat, exploiting the vulnerability is simple.
Technical Impact	A bad actor could compromise the systems of users who visit the page.
Business Impact	The reputation of the company could be damaged if a bad actor uses the website to distribute malware to users who visit the website.
Overview	The Newsroom news publishing service allows administrators to publish news stories directly to the main page of the ObsidianRift website.
Summary	A lack of input sanitization allows a compromised administrator account to post malicious code that will execute on client devices when the news story is opened.
Finding Evidence	
Injection of XXS payload	



Detonation of XXS payload



Recommendation

- Input sanitization or escaping should be applied to all user input from any user, even if they are authenticated.

Weak Password Policy

CVSS Score	**6.3**
CWE	-
CVSS String	CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:C/C:H/I:N/A:N
Likelihood	Passwords are used extensively throughout the organization and will be the first, and often last, line of defense for important data and services. Weak passwords will very likely be exploited.
Technical Impact	The lack of password policies makes user compromise more likely.
Business Impact	The increase in user compromise causes a higher risk to the confidentiality and integrity of data and the availability of services.
Overview	n/a
Summary	The passwords throughout the organization do not have secure minimum requirements. If obtained, many of the password hashes can be cracked easily. The passwords for some important services can be easily guessed.

Finding Evidence

Example of password cracking:

```
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[~]
└──╼ $john --wordlist=/usr/share/seclists/Passwords/Common-Credentials/10k-most-common.txt admin_hash
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
[redacted]          (?)
1g 0:00:00:00 DONE (2026-09-13 11:35) 1.020g/s 73.46p/s 73.46c/s 73.46C/s batman..sexy
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[~]
└──╼ $
```

Recommendation

- Enforce password policies organization-wide.
- Provide password strength training to all employees.

Info Findings

SSH Password Authentication	
CVSS Score	**0.0**
CWE	-
CVSS String	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:N
Affected Scope	• 192.168.20.22 • 192.168.20.23 • 192.168.20.24 • 192.168.20.25
Likelihood	The compromise of a password-protected server is much more likely than that of a key-protected server, as the user must be in possession of the private key file.
Technical Impact	Switching to SSH key authentication from password authentication greatly increases the security of a server.
Business Impact	Securing access to the servers makes the data therein much safer.
Overview	SSH allows secure remote access to the terminal of a machine.
Summary	SSH password authentication is not necessarily insecure, but it allows for insecure passwords. Ideally, SSH should be secured by cryptographic key authentication.

Finding Evidence

This credential reuse would be more difficult with key authentication if the private keys are secured correctly:

```
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[~]
└──╼ $sshpass -p [redacted] ssh admin@192.168.20.22 id
uid=1000(admin) gid=1000(admin) groups=1000(admin)
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[~]
└──╼ $sshpass -p [redacted] ssh admin@192.168.20.23 id
uid=2001(admin) gid=2001(admin) groups=2001(admin)
┌─[user@CPTC-2026-Tryout-team20-jumpbox]─[~]
└──╼ $sshpass -p [redacted] ssh admin@192.168.20.24 id
┌─[✗]─[user@CPTC-2026-Tryout-team20-jumpbox]─[~]
└──╼ $sshpass -p [redacted] ssh admin@192.168.20.25 id
uid=10002(admin) gid=10002(admin) groups=10002(admin)
```

Recommendation

- Switch servers to Key Authentication.

Appendix

Finding Severities

Each finding has been assigned a severity rating of critical, high, medium, low or info. The rating is based off of an assessment of the priority with which each finding should be viewed and the potential impact each has on the confidentiality, integrity, and availability of Obsidian's data.

Rating	CVSS Score Range
Critical	9.0 – 10.0
High	7.0 – 8.9
Medium	4.0 – 6.9
Low	0.1 – 3.9
Info	0.0

Network Inventory

IP Address	Port	Service	OS
192.168.20.20	5355, 6433, 10250	Kubernetes Node	Debian 12
192.168.20.21	25, 110, 5355, 6443, 10250	Mail Server, Kubernetes Node	Debian 12
192.168.20.22	22, 80, 102, 502, 8080, 8443, 44818	Programmable Logic Controller	Debian 13
192.168.20.23	22, 80, 111, 199, 2049, 8065, 8443, 20048, 32765, 36531	Mattermost Server, Fileshare Server	Ubuntu 24.04 LTS
192.168.20.24	22, 2222, 2375, 3000, 5000, 5001, 44683	Docker and Gitea	Alpine v3.24
192.168.20.25	22, 53, 80, 5432, 8080	DNS, DB, API, and records	Ubuntu 22.04 LTS
192.168.20.46	Outside of scope	Outside of scope	Windows

Subdomain Discovery

URL	Description	Discovery Method
chat.obsidianrift.internal	Mattermost Chat Webserver	Port Scan
files.obsidianrift.internal	Document Management	Port Scan
api.obsidianrift.internal	Internal Management API	Redirect On file.obsidanrift.internal
incidentreport.obsidianrift.internal	Internal Management API	Redirect On file.obsidanrift.internal
admin.obsidianrift.internal	Wordpress Management	Gitea Instance

Compromised Users

Username	Type	Method	Notes
rhett.flores	linux user	NFS File	
operations@obsidianrift.internal	Mattermost Admin Account & linux user	Kubernetes Data	Can see all messages and manage the internals chat service
wyatt.dawson	linux user	Kubernetes Data	
operations	linux user	Kubernetes Data	
clay.garrett	linux user	Kubernetes Data	Gitea User
contractor	linux user	Kubernetes Data	
admin	linux user	Password hash in compromised database	Web admin
root@192.168.20.22	linux user	Kubernetes Privileged Pod	

Indictors of Compromise

Host	Description	Sha256 Hash
192.168.20.22-23	linpeas.sh	416fae4633e2b137318ac28d2a9484affff5af01bb81c20e89f9339ab0171afa