CredShields
Audit Report

Oct 18th, 2022 • CONFIDENTIAL

Description

This document details the process and result of the security audit performed by CredShields Technologies PTE. LTD. on behalf of AssetMantle between Sept 10th, 2022, and Sept 20th, 2022, and a retest was performed on 18th October 2022.

Author

Shashank (Co-founder, CredShields)

shashank@CredShields.com

Reviewers

Aditya Dixit (Research Team Lead)

aditya@CredShields.com

Prepared for

AssetMantle

Table of Contents

1. Executive Summary

AssetMantle engaged CredShields to perform a Web application and External Network audit from September 10th, 2022, to September 20th, 2022. During this timeframe, Fifteen (15) vulnerabilities were identified. **A retest was performed on 18th October 2022.**

During the audit, four (4) vulnerabilities were found that had a severity rating of either High or Critical. These vulnerabilities represent the greatest immediate risk to "AssetMantle" and should be prioritized for remediation. **As of 18th October 2022, all High and Critical vulnerabilities have been addressed.**

The table below shows the in-scope assets and a breakdown of findings by severity per asset. *Section 2.3* contains more information on how severity is calculated.

	Critical	High	Medium	Low	Info	Σ
AssetMantle Web + External Networks	1	3	5	6	0	15
	1	**3**	**5**	**6**	**0**	**15**

Table: Findings Overview

The security audit was conducted by the CredShields team to focus on identifying vulnerabilities in AssetMantle Web Applications and External Network scope during the testing window while abiding by the policies set forth by AssetMantle.
Maintaining a healthy security posture requires constant review and refinement of existing security processes. Running a CredShields audit allows AssetMantle's internal security team and development team to uncover specific vulnerabilities and better understand the current security threat landscape.

Reviewing the remaining resolved reports for a root cause analysis can further educate AssetMantle's internal development and security teams and allow manual or automated procedures to be put in place to eliminate entire classes of vulnerabilities in the future. This proactive approach helps contribute to future-proofing the security posture of AssetMantle's assets.

2. Methodology

AssetMantle engaged CredShields to perform a security audit of AssetMantle's Web Applications and external network. The following sections cover how the engagement was put together and executed.

2.1 Preparation phase

CredShields reviewed all the documents, such as the API and product documentation, to understand the features, functionalities, and access control of the AssetMantle. The team reviewed all the functions and prepared maps and graphs to understand the logic flow and access control better.

A testing window from September 10th, 2022, to September 20th, 2022, was agreed upon during the preparation phase.

2.1.1 Scope

During the preparation phase, the following scope for the engagement was agreed-upon:

ASSETS IN SCOPE
Asset 1:

AssetMantle Web Applications **- *.assetmantle.one** **Asset 2:** **External networks**

Table: Asset(s) in Scope

2.1.2 Audit Goals

Audit procedures at CreditShields involve both automated (in-house) tools and manual analysis. However, the majority of audit methods require a manual review of the application's source code.
The testing was done in accordance with the standards of the OWASP, along with an extended self-developed checklist based on industry standards and best practices for blockchain RPC security. The team focused heavily on the core concept behind all the functionalities, along with preparing test and edge cases. This included understanding the business logic and how it could have been exploited.

During the audit, the focus was on verifying the codebase's security, resilience, and compliance. The audit activities can be categorized into the following:

- **Security** - The identification of security issues in each application and its interaction with other projects.
- **Sound Architecture** - Analyzing this system's architecture through the lens of established best practices and general software best practices.

2.2 Retesting phase

AssetMantle is actively partnering with CredShields to validate the remediations implemented towards the discovered vulnerabilities. The first round of retesting was performed on 18th October 2022.

2.3 Vulnerability Classification and Severity

Discovering vulnerabilities is important, but estimating the associated risk to the business is just as important.

To adhere to industry guidelines, CredShields follows OWASP's Risk Rating Methodology. This is calculated using two factors - **Likelihood** and **Impact**. Each of these parameters can take three values - **Low**, **Medium**, and **High.**

These depend upon multiple factors such as Threat agents, Vulnerability factors (Ease of discovery and exploitation, etc.), and Technical and Business Impacts. The likelihood and the impact estimate are put together to calculate the overall severity of the risk.

CredShields also define an **Informational** severity level for vulnerabilities that do not align with any of the severity categories and usually have the lowest risk involved.

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Overall, the categories can be defined as described below -

1. **Informational**

   We believe in the importance of technical excellence and pay a great deal of attention to its details. Our coding guidelines, practices, and standards help ensure that the software we audit is stable and reliable.

   Informational vulnerabilities should not be a cause for alarm but rather a chance to improve the quality of the codebase by emphasizing readability and good practices.

   They do not represent a direct risk to the product but suggest improvements and best practices that can not be categorized under any other severity categories.

   Code maintainers should use their own judgment as to whether to address such issues.

2. **Low**

   Vulnerabilities in this category represent a low risk to the product and the organization. The risk is either relatively small and may or may not be exploited on a recurring basis, or a risk that the client indicates is not significant, given the client's business circumstances.

3. **Medium**

   Medium severity issues are those that are usually introduced due to weak or erroneous logic in the code.

These issues may lead to exfiltration or modification of some of the private information belonging to the end-user, and exploitation would be detrimental to the client's reputation under certain unexpected circumstances or conditions. These conditions are outside the control of the adversary.

These issues should eventually be fixed under a certain timeframe and remediation cycle.

4. High

High severity vulnerabilities represent a greater risk to the product and the organization. These vulnerabilities may lead to a limited loss of confidentiality, integrity, and availability for some of the end users.

They may or may not require external conditions to be met, or these conditions may be manipulated by the attacker, but the complexity of exploitation will be higher.

These vulnerabilities, when exploited, will impact the client's reputation negatively.

They should be fixed immediately.

5. Critical

Critical issues are directly exploitable bugs or security vulnerabilities. These issues do not require any external conditions to be met.

The issue puts the vast majority of, or large numbers of, users' sensitive information at risk of modification or compromise.

The client's reputation will suffer a severe blow, or there will be serious financial repercussions.

2.4 CredShields staff

The following individual at CredShields managed this engagement and produced this report:

- **Shashank, Co-founder CredShields**
    - shashank@CredShields.com

Please feel free to contact this individual with any questions or concerns you have around the engagement or this document.

3. Findings

This chapter contains the results of the security assessment. Findings are sorted by their severity and grouped by the asset and CWE classification if applicable. Each asset section will include a summary. The table in the executive summary contains the total number of identified security vulnerabilities per asset per risk indication.

3.1 Findings Overview

3.1.1 Vulnerability Summary

During the security assessment, a total of fifteen (15) security vulnerabilities were identified in the assets.

VULNERABILITY TITLE	SEVERITY	CWE | Vulnerability Type
XMLRPC Login Bruteforce	Medium	CWE-307: Improper Restriction of Excessive Authentication Attempts
WordPress User Enumeration	Low	CWE-799: Improper Control of Interaction Frequency

TLS 1.2 Supports Weak Cipher Suites	Low	CWE-799: Improper Control of Interaction Frequency
Unauthenticated Docker Registry	Critical	Sensitive Information Disclosure
Multiple Weak password policy	Medium	CWE-521: Weak Password Requirements
Password Hash is stored in LocalStorage permanently	Medium	Security misconfiguration
Unencrypted Communication	Medium	CWE-319: Cleartext Transmission
Publicly Exposed Pinata Admin API Credentials	High	Sensitive Information Disclosure
Misconfigured Content-Security-Policy	Low	Security Misconfiguration
Session Cookie missing Secure Attribute	Low	Security Misconfiguration
Mnemonic Stored on the Server	High	Security Misconfiguration
CSRF to Add and Remove NFTs from Wishlist	Medium	CWE-352: Cross-Site Request Forgery
0 Price for NFT causes Denial of Service	High	Denial of Service (application level)
MKDocs Vulnerable Version	Low	CWE-79: Improper Neutralization of Input During Web Page Generation
Outdated Javascript Library (Moment.js)	Low	Using Components with known Vulnerabilities

3.1.2 Findings Summary

The Credshields security team found multiple security issues on the web application platform. The team found that the application was not clearing local storage after the user chose to log out from the application. The team also noticed that the main application could be DOSed by setting the selling price to zero. The application was also vulnerable to CSRF attacks due to the lack of anti-CSRF tokens. The team looked for old software versions, and few instances were found. Addressing these issues will enhance the overall security of the web application and external networks.

4. Remediation Status

AssetMantle is actively partnering with CredShields from this engagement to validate the discovered vulnerabilities' remediations. The table shows the remediation status of each finding.

VULNERABILITY TITLE	SEVERITY	REMEDIATION STATUS
XMLRPC Login Bruteforce	Medium	Pending Fix
WordPress User Enumeration	Low	Pending Fix
TLS 1.2 Supports Weak Cipher Suites	Low	**Won't Fix**
Unauthenticated Docker Registry	Critical	**Not Applicable**
Multiple Weak password policy	Medium	Pending Fix
Password Hash is stored in LocalStorage permanently	Medium	**Won't Fix**
Unencrypted Communication	Medium	Pending Fix
Publicly Exposed Pinata Admin API Credentials	High	Pending Fix
Misconfigured Content-Security-Policy	Low	**Won't Fix**
Session Cookie missing Secure Attribute	Low	**Won't Fix**
Mnemonic Stored on the Server	High	Not Applicable

CSRF to Add and Remove NFTs from Wishlist	Medium	**Fixed [18/10/2022]**
0 Price for NFT causes Denial of Service	High	Pending Fix
MKDocs Vulnerable Version	Low	**Won’t Fix**
Outdated Javascript Library (Moment.js)	Low	**Won’t Fix**

Table: Summary of findings and status of remediation

5. Bug Reports

Bug ID#1 [Pending Fix]

XMLRPC Login Bruteforce

Vulnerability Type
CWE-307: Improper Restriction of Excessive Authentication Attempts

Severity
Medium

Description
XML-RPC uses XML encoding over HTTP to provide a remote procedure call protocol. It's commonly used to execute various functions in a WordPress instance for APIs and other automated tasks. MySQL query is performed, so it could be used by attackers to cause a DoS.
wp.getUserBlogs, **wp.getCategories**, or **metaWeblog.getUsersBlogs** are some of the methods that can be used to brute-force credentials. **system.multicall** method can be used to amplify this attack.

Vulnerable URL
- https://blog.assetmantle.one/xmlrpc.php

PoC
1. Send a POST request to the "/xmlrpc.php" as shown below with the credentials to guess in the request body.
2. This can be abused further to result in an amplification attack by using system.multicall method to guess multiple credentials at a time.

```
POST /xmlrpc.php HTTP/2
Host: blog.assetmantle.one
```

```
Cookie: _color_system_schema=default
Sec-Ch-Ua: "Chromium";v="92", " Not A;Brand";v="99", "Google Chrome";v="92"
Sec-Ch-Ua-Mobile: ?0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/92.0.4515.159 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image
/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Content-Length: 164
<methodCall>
<methodName>wp.getUsersBlogs</methodName>
<params>
<param><value>admin</value></param>
<param><value>pass</value></param>
</params>
</methodCall>
```

Impacts

Attackers can exploit this vulnerability to bruteforce the credentials for the admin account and other users by using a wordlist of usernames and passwords and sending simultaneous requests.

Remediation

The file xmlrpc.php should be blocked for external access but it should be noted that this breaks some plugins.

Another way to mitigate this attack is by disabling the ability to call the system.multicall method in your WordPress installation by editing your **functions.php** file. Adding the function **mmx_remove_xmlrpc_methods()** will alleviate the problem, like so:

```
function mmx_remove_xmlrpc_methods( $methods ) {
unset( $methods['system.multicall'] );
return $methods;
}
add_filter( 'xmlrpc_methods', 'mmx_remove_xmlrpc_methods');
```

Reference

https://blog.cloudflare.com/a-look-at-the-new-wordpress-brute-force-amplification-attack

Retest

The team is planning to deploy a separate instance of WordPress and hence this will be fixed later.

Bug ID#2 [Pending Fix]

WordPress User Enumeration

Vulnerability Type

CWE-799: Improper Control of Interaction Frequency

Severity

Low

Description

User Enumeration is an attack where an attacker thoroughly scans a web application to discover the login names of the web application. These discovered usernames could then be used along with the password brute force vulnerability to guess their passwords.

Vulnerable URL:

- https://blog.assetmantle.one/wp-json/wp/v2/users
- https://blog.assetmantle.one/wp-login.php

PoC

1. Visit the URLs shown above to reveal the username for the existing user account on the WordPress blog.
2. It's also possible to guess usernames on the admin login page when a wrong username is entered.

Impacts

By knowing the existing usernames on the platform, it becomes really easy for an attacker to brute-force their passwords and take over admin accounts.

Remediation

You can reduce the attack surface and make user enumeration harder by following the below
steps:
1. Disable the WordPress REST API if you are not using it,

2. Disable WordPress XML-RPC if you are not using it,
3. Configure your webserver to block requests to /?author=<number>,
4. Don't expose /wp-admin and /wp-login.php directly to the public Internet.
It is recommended to install a plugin called WP Hardening to prevent user enumeration, among other common WordPress vulnerabilities and misconfigurations. - https://wordpress.org/plugins/wp-security-hardening/

Retest

The team is planning to deploy a separate instance of WordPress, and hence this will be fixed later.

Bug ID#3 [Won't Fix]

TLS 1.2 Supports Weak Cipher Suites

Vulnerability Type
CWE-799: Improper Control of Interaction Frequency

Severity
Low

Description
The software stores or transmits sensitive data using an encryption scheme that is theoretically sound but is not strong enough for the level of protection required.

A weak encryption scheme can be subjected to brute force attacks that have a reasonable chance of succeeding using current attack methods and resources.

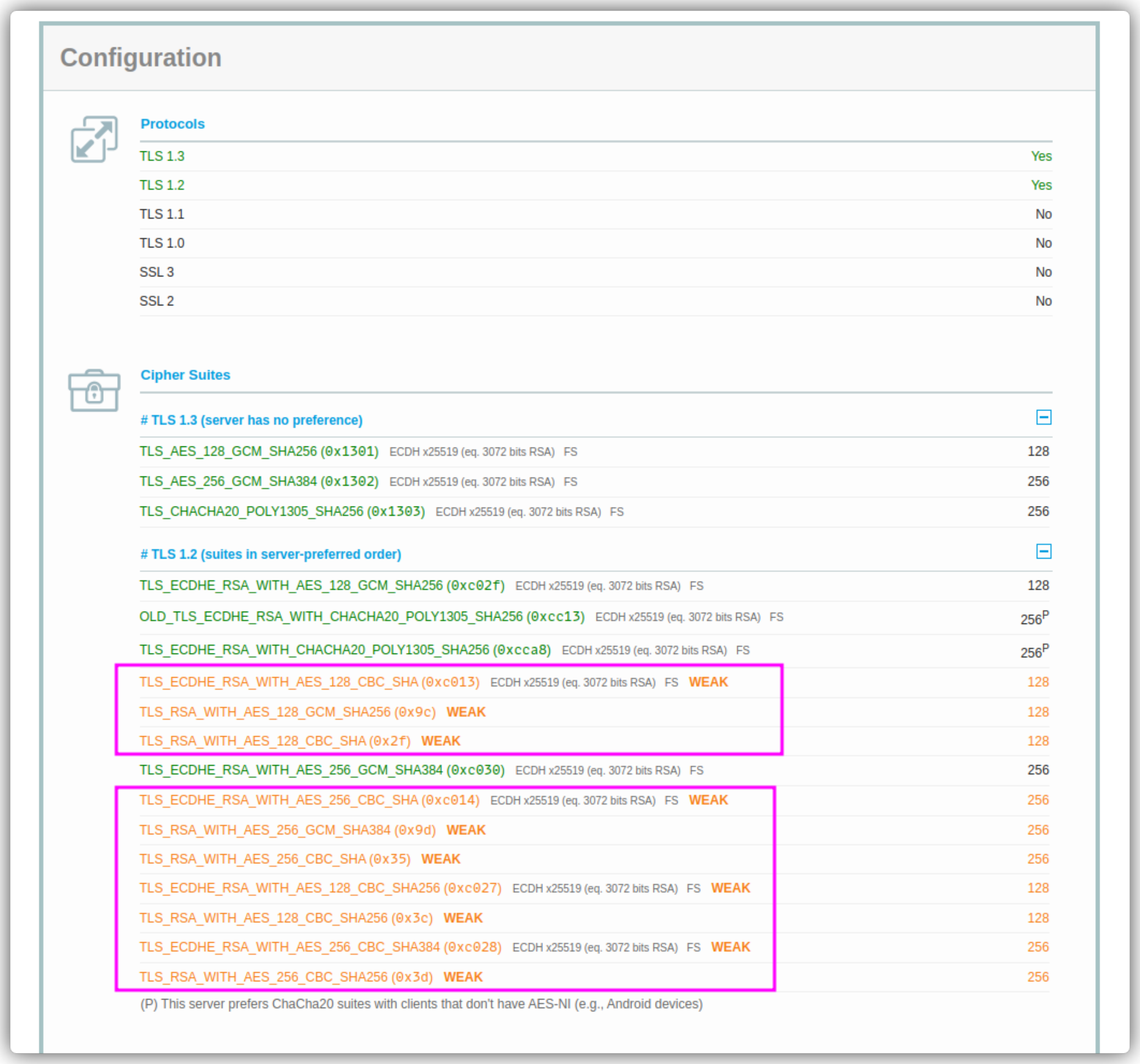
TLS 1.2 implementations by the server assetmantle.one was found to be supporting weak CBC and other cipher suites. These cipher suites offer additional security over Electronic Codebook (ECB) mode but have the potential to leak information if used improperly.

Vulnerable URL

- assetmantle.one

PoC

1. Scan the server using SSLLabs at https://www.ssllabs.com/ssltest/analyze.html?d=assetmantle.one&s=104.18.18.111&hideResults=on



Configuration

Protocols

TLS 1.3	Yes
TLS 1.2	Yes
TLS 1.1	No
TLS 1.0	No
SSL 3	No
SSL 2	No

Cipher Suites

# TLS 1.3 (server has no preference)	
TLS_AES_128_GCM_SHA256 (0x1301) ECDH x25519 (eq. 3072 bits RSA) FS	128
TLS_AES_256_GCM_SHA384 (0x1302) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_CHACHA20_POLY1305_SHA256 (0x1303) ECDH x25519 (eq. 3072 bits RSA) FS	256

# TLS 1.2 (suites in server-preferred order)	
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) ECDH x25519 (eq. 3072 bits RSA) FS	128
OLD_TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc13) ECDH x25519 (eq. 3072 bits RSA) FS	256P
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8) ECDH x25519 (eq. 3072 bits RSA) FS	256P
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) ECDH x25519 (eq. 3072 bits RSA) FS **WEAK**	128
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c) **WEAK**	128
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f) **WEAK**	128
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) ECDH x25519 (eq. 3072 bits RSA) FS **WEAK**	256
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d) **WEAK**	256
TLS_RSA_WITH_AES_256_CBC_SHA (0x35) **WEAK**	256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027) ECDH x25519 (eq. 3072 bits RSA) FS **WEAK**	128
TLS_RSA_WITH_AES_128_CBC_SHA256 (0x3c) **WEAK**	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028) ECDH x25519 (eq. 3072 bits RSA) FS **WEAK**	256
TLS_RSA_WITH_AES_256_CBC_SHA256 (0x3d) **WEAK**	256

(P) This server prefers ChaCha20 suites with clients that don't have AES-NI (e.g., Android devices)

Impacts

An attacker may be able to exploit this issue to conduct man-in-the-middle attacks and decrypt and tamper with the communications between the affected service and clients.

Remediation

As a best practice, consider supporting only cipher suites that are known to be secure. Disable any cipher suites that use encryption with less than 128-bit key lengths or utilize RC4 algorithms. Enabled TLS cipher suites must be ranked as MEDIUM strength by the

current version of OpenSSL at a minimum. However, HIGH is ideal. Ensure that the cipher suites are ordered from strongest to weakest.
Here's a reference - https://www.acunetix.com/blog/articles/tls-ssl-cipher-hardening/

Retest
The team is using Cloudflare SSL, and hence it is working as intended and considered overall safe.

Bug ID#4 [Not Applicable]

Unauthenticated Docker Registry

Vulnerability Type

Sensitive Information Disclosure

Severity

Critical

Description

The server at https://containers.assetmantle.one/ is using a publicly exposed docker registry v2.0 without authentication. This is hosting three repositories -

1. `assetmantle/half-life`
2. `assetmantle/mantlenode`
3. `library/node`

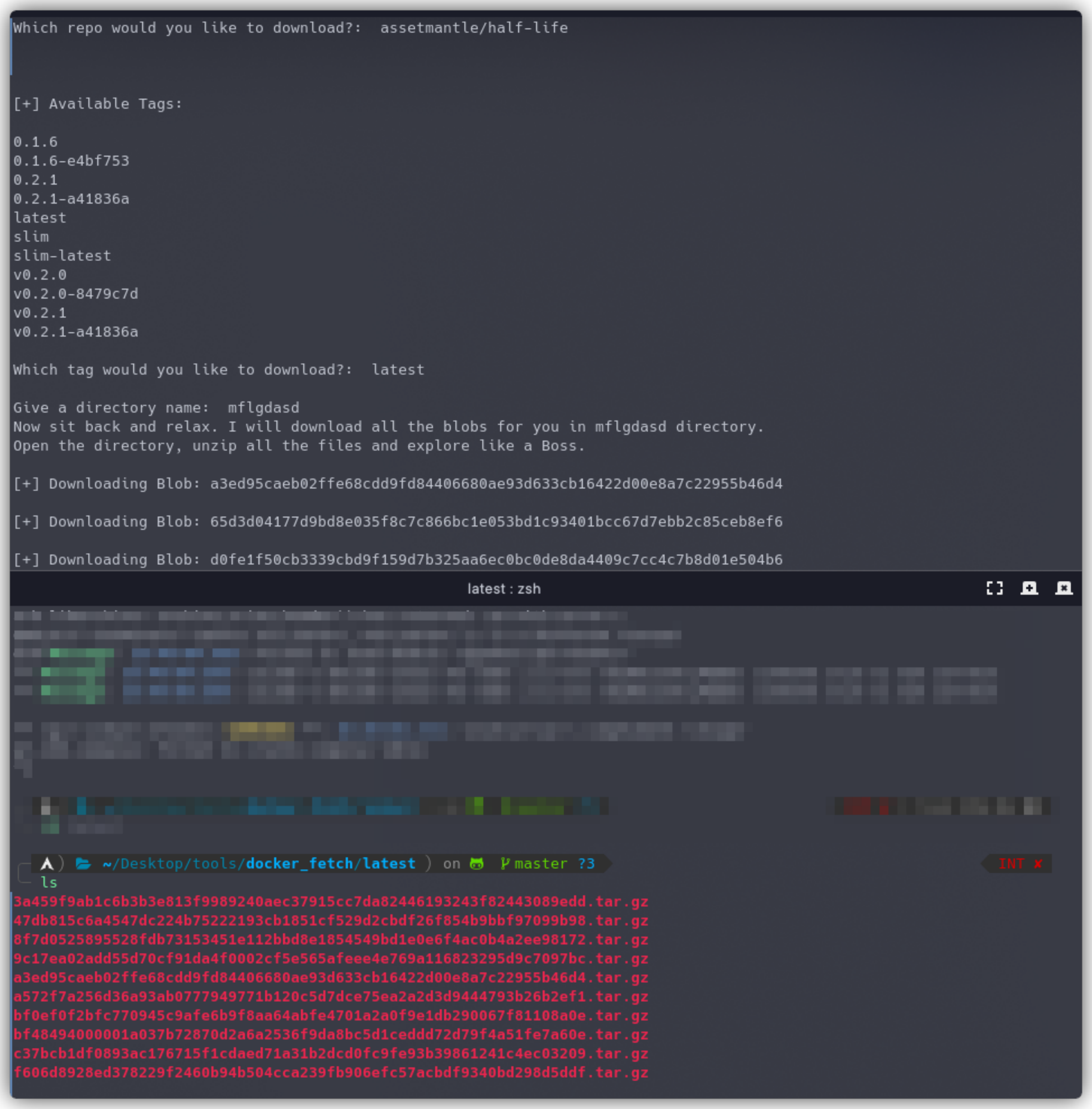
These repositories contain entire Linux filesystems hosting internal cosmos tools and assetmantle node-related data.

Vulnerable URL

- https://containers.assetmantle.one/

PoC

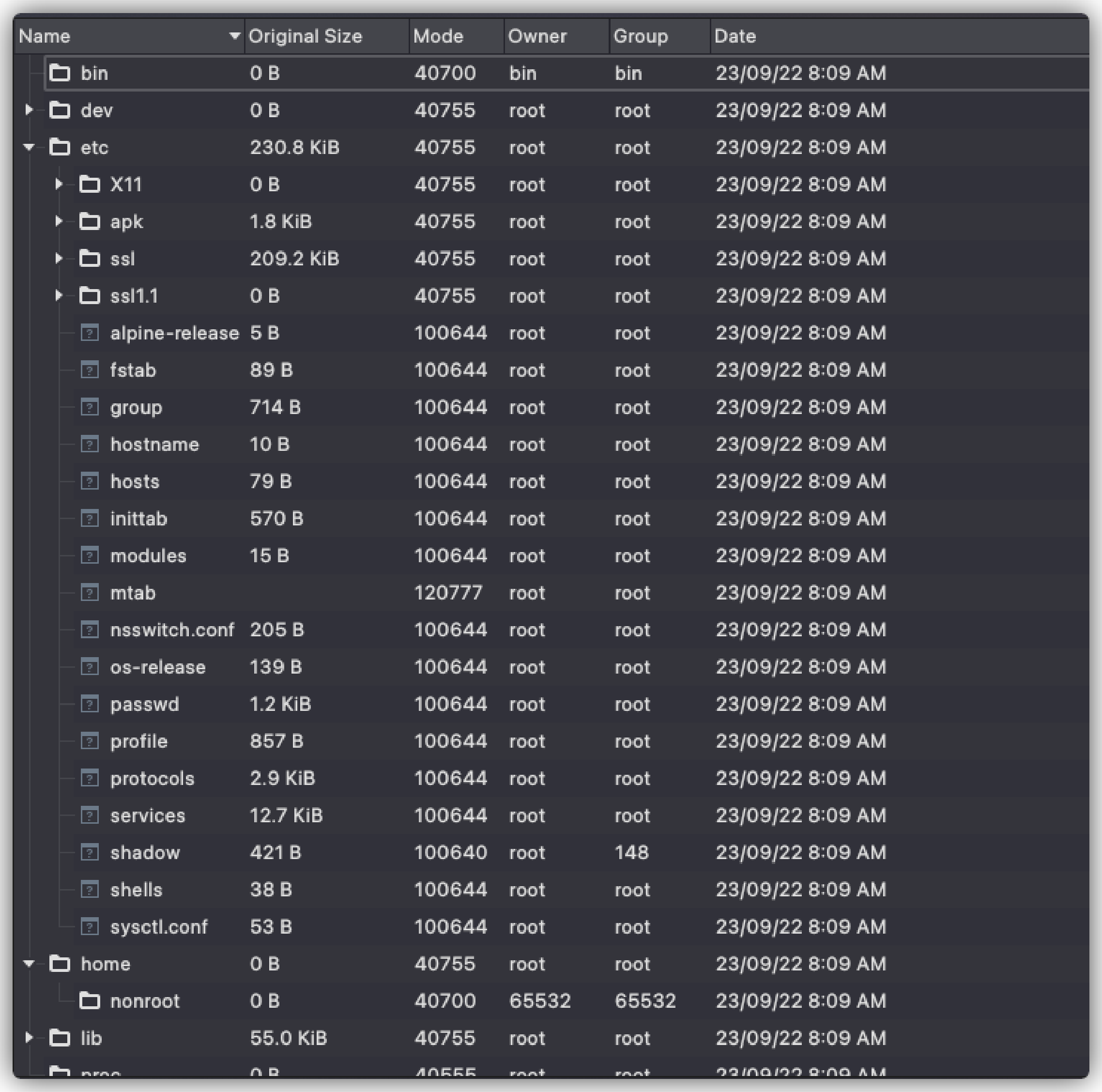
1. Scan the server using docker_fetch.
2. Note that the registry exposes three repositories mentioned above.
3. Enter the name of the repository to download and save it to a folder.
4. The downloaded data can be seen below:



```
Which repo would you like to download?:  assetmantle/half-life



[+] Available Tags:

0.1.6
0.1.6-e4bf753
0.2.1
0.2.1-a41836a
latest
slim
slim-latest
v0.2.0
v0.2.0-8479c7d
v0.2.1
v0.2.1-a41836a

Which tag would you like to download?:  latest

Give a directory name:  mflgdasd
Now sit back and relax. I will download all the blobs for you in mflgdasd directory.
Open the directory, unzip all the files and explore like a Boss.

[+] Downloading Blob: a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4

[+] Downloading Blob: 65d3d04177d9bd8e035f8c7c866bc1e053bd1c93401bcc67d7ebb2c85ceb8ef6

[+] Downloading Blob: d0fe1f50cb3339cbd9f159d7b325aa6ec0bc0de8da4409c7cc4c7b8d01e504b6
```

latest : zsh

```
[redacted]

 ) ~/Desktop/tools/docker_fetch/latest ) on  master ?3                                  INT ✘
  ls
3a459f9ab1c6b3b3e813f9989240aec37915cc7da82446193243f82443089edd.tar.gz
47db815c6a4547dc224b75222193cb1851cf529d2cbdf26f854b9bbf97099b98.tar.gz
8f7d0525895528fdb73153451e112bbd8e1854549bd1e0e6f4ac0b4a2ee98172.tar.gz
9c17ea02add55d70cf91da4f0002cf5e565afeee4e769a116823295d9c7097bc.tar.gz
a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4.tar.gz
a572f7a256d36a93ab0777949771b120c5d7dce75ea2a2d3d9444793b26b2ef1.tar.gz
bf0ef0f2bfc770945c9afe6b9f8aa64abfe4701a2a0f9e1db290067f81108a0e.tar.gz
bf48494000001a037b72870d2a6a2536f9da8bc5d1ceddd72d79f4a51fe7a60e.tar.gz
c37bcb1df0893ac176715f1cdaed71a31b2dcd0fc9fe93b39861241c4ec03209.tar.gz
f606d8928ed378229f2460b94b504cca239fb906efc57acbdf9340bd298d5ddf.tar.gz
```



Name	Original Size	Mode	Owner	Group	Date
bin	0 B	40700	bin	bin	23/09/22 8:09 AM
dev	0 B	40755	root	root	23/09/22 8:09 AM
etc	230.8 KiB	40755	root	root	23/09/22 8:09 AM
X11	0 B	40755	root	root	23/09/22 8:09 AM
apk	1.8 KiB	40755	root	root	23/09/22 8:09 AM
ssl	209.2 KiB	40755	root	root	23/09/22 8:09 AM
ssl1.1	0 B	40755	root	root	23/09/22 8:09 AM
alpine-release	5 B	100644	root	root	23/09/22 8:09 AM
fstab	89 B	100644	root	root	23/09/22 8:09 AM
group	714 B	100644	root	root	23/09/22 8:09 AM
hostname	10 B	100644	root	root	23/09/22 8:09 AM
hosts	79 B	100644	root	root	23/09/22 8:09 AM
inittab	570 B	100644	root	root	23/09/22 8:09 AM
modules	15 B	100644	root	root	23/09/22 8:09 AM
mtab		120777	root	root	23/09/22 8:09 AM
nsswitch.conf	205 B	100644	root	root	23/09/22 8:09 AM
os-release	139 B	100644	root	root	23/09/22 8:09 AM
passwd	1.2 KiB	100644	root	root	23/09/22 8:09 AM
profile	857 B	100644	root	root	23/09/22 8:09 AM
protocols	2.9 KiB	100644	root	root	23/09/22 8:09 AM
services	12.7 KiB	100644	root	root	23/09/22 8:09 AM
shadow	421 B	100640	root	148	23/09/22 8:09 AM
shells	38 B	100644	root	root	23/09/22 8:09 AM
sysctl.conf	53 B	100644	root	root	23/09/22 8:09 AM
home	0 B	40755	root	root	23/09/22 8:09 AM
nonroot	0 B	40700	65532	65532	23/09/22 8:09 AM
lib	55.0 KiB	40755	root	root	23/09/22 8:09 AM
proc	0 B	40555	root	root	23/09/22 8:09 AM

Impacts

These repositories are exposing internal organizational data and toolings including the whole docker repositories and their versions.

Remediation

It is recommended to implement authentication on the endpoint and not expose the registry publicly unless required.

Retest

The team informed us that this is working as intended, as the docker files are public on the docker registry. Hence this was not a valid find.

Bug ID#5 [Pending Fix]

Multiple Weak password policy

Vulnerability Type
Weak Password Requirements [CWE-521]

Severity
Medium

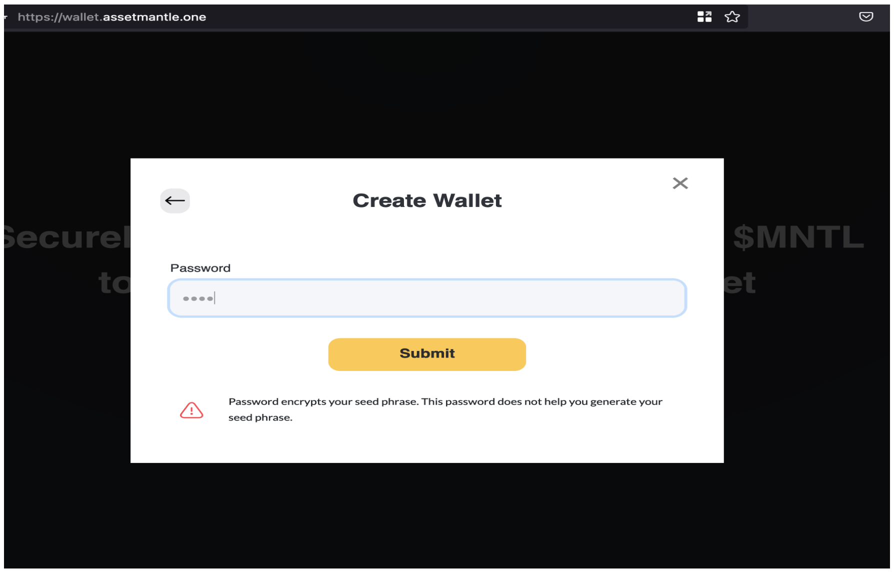
Description
The application does not require that users should have strong passwords, which makes it easier for attackers to compromise user accounts.
An authentication mechanism is only as strong as its credentials. For this reason, it is important to require users to have strong passwords. Lack of password complexity significantly reduces the search space when trying to guess a user's password, making brute-force attacks easier.
https://cwe.mitre.org/data/definitions/521.html

Vulnerable URL

- https://wallet.assetmantle.one/

PoC

1. Go to https://wallet.assetmantle.one/ and click on “Create Wallet”
2. While the generation of Keystore, we will be asked to enter a password.
3. We will notice we can keep weak passwords like “1234”


https://wallet.assetmantle.one
Create Wallet
Password
Submit
Password encrypts your seed phrase. This password does not help you generate your seed phrase.

Impacts
An attacker can guess weak passwords using dictionary attacks.

Remediation
Implement a strong password policy
1. Allow all characters to be used for passwords to avoid shortening the keyspace for brute-force guessing.
2. Do not impose character restrictions such as "must have at least X number of specific character types" in the password. This will shorten the keyspace for brute-force guessing.
3. Disallow short password lengths. 8 characters are generally considered a good minimum password length.
4. Allow for a large maximum password length.
5. Do not advertise the maximum password length, as this will shorten the key space for brute-force guessing.

Retest
The team informed us that this will be fixed in the next release.

Bug ID#6 [Won't Fix]

Password Hash is stored in LocalStorage permanently

Vulnerability Type
Security misconfiguration

Severity
Medium

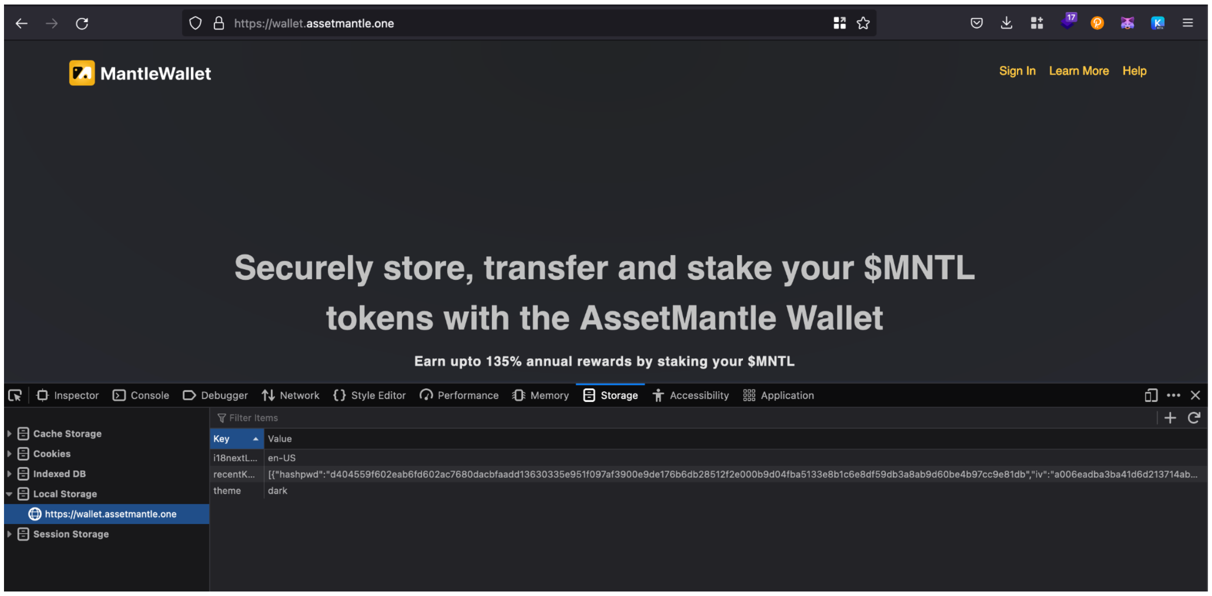
Description
The application is using LocalStorage to store the session information such as hashed passwords and addresses.
When the user logs out of the application or their session expire, the storage should be cleared, but this is not happening. The hashed password can be found in the local storage even after the user logs out or the browser is restarted.

Vulnerable URL

- https://wallet.assetmantle.one/

PoC

1. Go to https://wallet.assetmantle.one/ and create an account.
2. Sign in to the account using your KeyStore file and then log out.
3. Now, even if we close the tab or close the browser and restart, we will still notice the hashed password in the LocalStorage of the browser.



Key	Value
i18nextL...	en-US
recentK...	[{"hashpwd":"d404559f602eab6fd602ac7680dacbfaadd13630335e951f097af3900e9de176b6db28512f2e000b9d04fba5133e8b1c6e8df59db3a8ab9d60be4b97cc9e81db","iv":"a006eadba3ba41d6d213714ab...
theme	dark

Impacts

A user using a shared PC or if a browser exploit is public, an attacker can extract the hash passwords of the user as it stays there permanently.

Remediation

Clear browser LocalStorage after the user logs out from the wallet application.

Retest

The team informed us that this is working as intended.

Bug ID#7 [Pending Fix]

Unencrypted Communication

Vulnerability Type
Cleartext Transmission - CWE-319

Severity
Medium

Description
The application allows users to connect to it over unencrypted connections. An attacker suitably positioned to view a legitimate user's network traffic could record and monitor their interactions with the application and obtain any information the user supplies. Furthermore, an attacker able to modify traffic could use the application as a platform for attacks against its users and third-party websites. Unencrypted connections have been exploited by ISPs and governments to track users and to inject adverts and malicious JavaScript. Due to these concerns, web browser vendors are planning to visually flag unencrypted connections as hazardous.

Vulnerable URLs

- http://grpc.assetmantle.one/

PoC

1. Visit the URLs shown above and note that the website can be accessed using HTTP and is not using TLS protection.

Impacts
Allowing HTTP connections can lead attackers to intercept data over the network by placing themselves strategically in between the victim’s system and executing a Man in the Middle Attack. This can expose all the traffic, including authentication sessions and cookies to the attackers.

Remediation

Applications should use transport-level encryption (SSL/TLS) to protect all communications passing between the client and the server. The Strict-Transport-Security HTTP header should be used to ensure that clients refuse to access the server over an insecure connection.

Retest

This is not on priority as there is negligible impact. The team will add a forced HTTPS redirection later.

Bug ID#8 [Pending Fix]

Publicly Exposed Pinata Admin API Credentials

Vulnerability Type

Sensitive Information Disclosure

Severity

High

Description

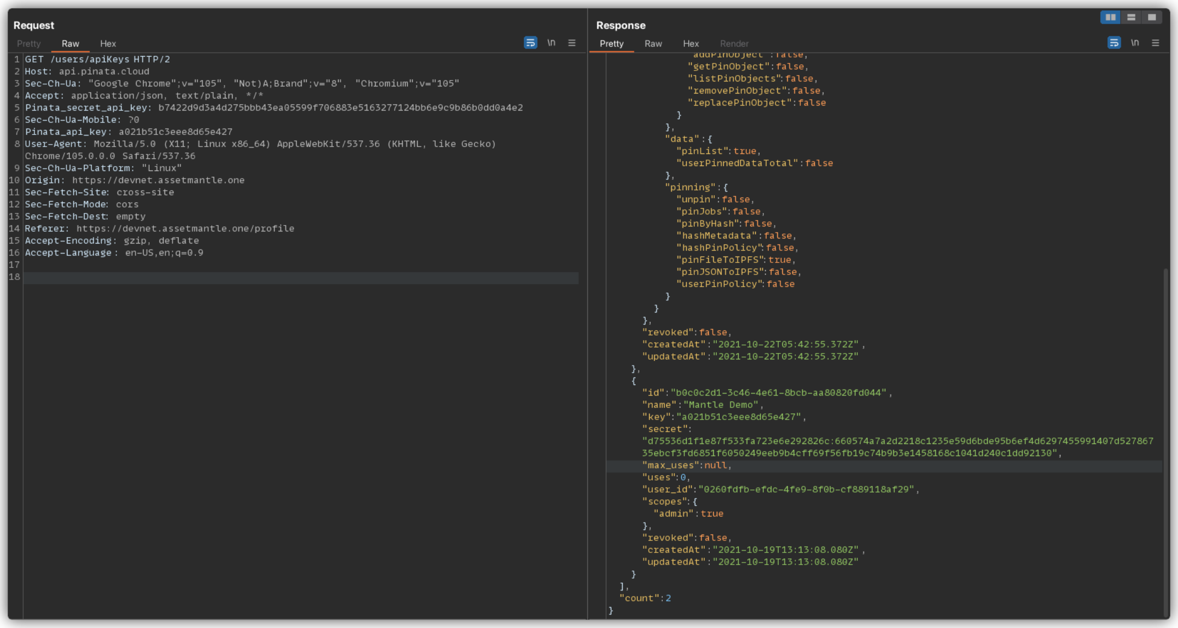
The devnet at https://devnet.assetmantle.one/profile is using Pinata API which can be seen in the ongoing requests. This is using the public and private API keys to interact with their API but these keys have admin privileges which can be seen when querying the API.

Vulnerable URLs

- https://api.pinata.cloud/data/testAuthentication
- **Pinata_secret_api_key**: b7422d9d3a4d275bbb43ea05599f706883e5163277124bb6e9c9b86b0dd0a4e2
- **Pinata_api_key**: a021b51c3eee8d65e427

PoC

1. Use the keys mentioned above to send a request to the endpoint -
2. Note that the API keys have admin privileges and are publicly available for everyone using the devnet.



```
GET /users/apiKeys HTTP/2
Host: api.pinata.cloud
Sec-Ch-Ua: "Google Chrome";v="105", "Not)A;Brand";v="8", "Chromium";v="105"
Accept: application/json, text/plain, */*
Pinata_secret_api_key: b7422d9d3a4d275bbb43ea05599f706883e5163277124bb6e9c9b86b0dd0a4e2
Sec-Ch-Ua-Mobile: ?0
Pinata_api_key: a021b51c3eee8d65e427
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/105.0.0.0 Safari/537.36
Sec-Ch-Ua-Platform: "Linux"
Origin: https://devnet.assetmantle.one
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://devnet.assetmantle.one/profile
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```

```
              "addPinObject":false,
              "getPinObject":false,
              "listPinObjects":false,
              "removePinObject":false,
              "replacePinObject":false
            }
          },
          "data":{
            "pinList":true,
            "userPinnedDataTotal":false
          },
          "pinning":{
            "unpin":false,
            "pinJobs":false,
            "pinByHash":false,
            "hashMetadata":false,
            "hashPinPolicy":false,
            "pinFileToIPFS":true,
            "pinJSONToIPFS":false,
            "userPinPolicy":false
          }
        }
      },
      "revoked":false,
      "createdAt":"2021-10-22T05:42:55.372Z",
      "updatedAt":"2021-10-22T05:42:55.372Z"
    },
    {
      "id":"b0c0c2d1-3c46-4e61-8bcb-aa80820fd044",
      "name":"Mantle Demo",
      "key":"a021b51c3eee8d65e427",
      "secret":
      "d75536d1f1e87f533fa723e6e292826c:660574a7a2d2218c1235e59d6bde95b6ef4d6297455991407d527867
      35ebcf3fd6851f6050249eeb9b4cff69f56fb19c74b9b3e1458168c1041d240c1dd92130",
      "max_uses":null,
      "uses":0,
      "user_id":"0260fdfb-efdc-4fe9-8f0b-cf889118af29",
      "scopes":{
        "admin":true
      },
      "revoked":false,
      "createdAt":"2021-10-19T13:13:08.080Z",
      "updatedAt":"2021-10-19T13:13:08.080Z"
    }
  ],
  "count":2
}
```

```
GET /users/apiKeys HTTP/2
Host: api.pinata.cloud
Sec-Ch-Ua: "Google Chrome";v="105", "Not)A;Brand";v="8", "Chromium";v="105"
Accept: application/json, text/plain, */*
Pinata_secret_api_key:
b7422d9d3a4d275bbb43ea05599f706883e5163277124bb6e9c9b86b0dd0a4e2
Sec-Ch-Ua-Mobile: ?0
Pinata_api_key: a021b51c3eee8d65e427
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/105.0.0.0 Safari/537.36
Sec-Ch-Ua-Platform: "Linux"
Origin: https://devnet.assetmantle.one
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://devnet.assetmantle.one/profile
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```

Impacts
Publicly disclosing API keys might have devastating impacts on the data handled by the Pinata. Since the keys have admin privileges, it allows attackers to execute authenticated administrative actions and manage the data on Pinata.

Remediation
Revoke the existing admin and other API keys. It is recommended to not pass the credentials publicly or use an internal API so the Pinata endpoints are not exposed to end users.
If this is extremely required, generate keys with restrictive roles and use them instead of administrative keys.

Retest
After discussion with the team we released this is how Pinata works and hence the team will be moving out from Pinata later.

Bug ID#9 [Won't Fix]

Misconfigured Content-Security-Policy

Vulnerability Type
Security Misconfiguration

Severity
Low

Description
Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross-Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft, to site defacement, to malware distribution.

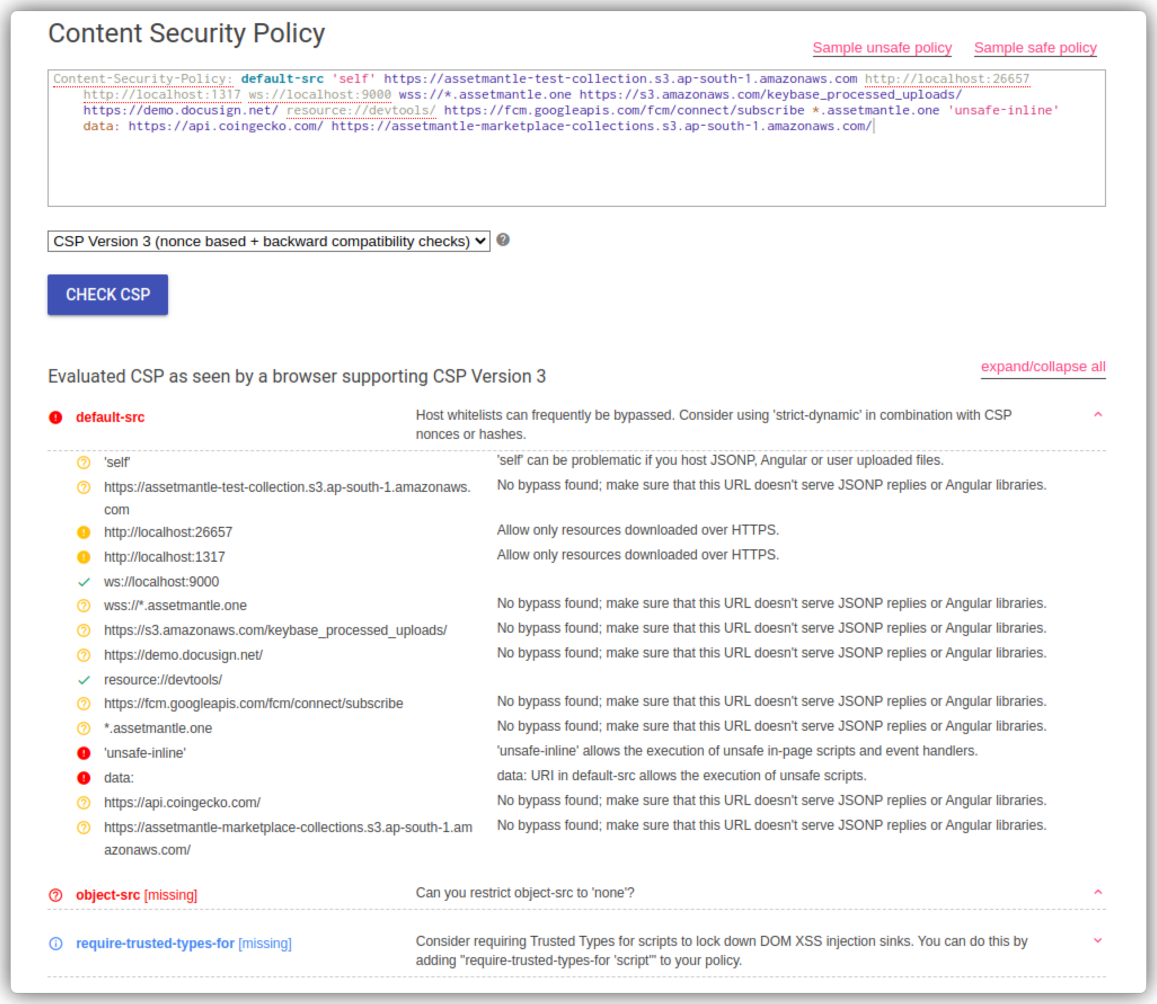
The CSP is found to be misconfigured in the application and allows the **unsafe-inline** and **data** attribute in **default-src** which can be used to execute scripts and render the CSP useless in case an attack like XSS is found.

Vulnerable URLs

- https://marketplace.assetmantle.one/*

PoC

1. Copy the CSP and validate it on Google's CSP Evaluator and observe the misconfigurations shown below



Content Security Policy

Sample unsafe policy Sample safe policy

```
Content-Security-Policy: default-src 'self' https://assetmantle-test-collection.s3.ap-south-1.amazonaws.com http://localhost:26657
    http://localhost:1317 ws://localhost:9000 wss://*.assetmantle.one https://s3.amazonaws.com/keybase_processed_uploads/
    https://demo.docusign.net/ resource://devtools/ https://fcm.googleapis.com/fcm/connect/subscribe *.assetmantle.one 'unsafe-inline'
    data: https://api.coingecko.com/ https://assetmantle-marketplace-collections.s3.ap-south-1.amazonaws.com/
```

CSP Version 3 (nonce based + backward compatibility checks)

CHECK CSP

Evaluated CSP as seen by a browser supporting CSP Version 3

expand/collapse all

●	default-src	Host whitelists can frequently be bypassed. Consider using 'strict-dynamic' in combination with CSP nonces or hashes.
?	'self'	'self' can be problematic if you host JSONP, Angular or user uploaded files.
?	https://assetmantle-test-collection.s3.ap-south-1.amazonaws.com	No bypass found; make sure that this URL doesn't serve JSONP replies or Angular libraries.
!	http://localhost:26657	Allow only resources downloaded over HTTPS.
!	http://localhost:1317	Allow only resources downloaded over HTTPS.
✓	ws://localhost:9000	
?	wss://*.assetmantle.one	No bypass found; make sure that this URL doesn't serve JSONP replies or Angular libraries.
?	https://s3.amazonaws.com/keybase_processed_uploads/	No bypass found; make sure that this URL doesn't serve JSONP replies or Angular libraries.
?	https://demo.docusign.net/	No bypass found; make sure that this URL doesn't serve JSONP replies or Angular libraries.
✓	resource://devtools/	
?	https://fcm.googleapis.com/fcm/connect/subscribe	No bypass found; make sure that this URL doesn't serve JSONP replies or Angular libraries.
?	*.assetmantle.one	No bypass found; make sure that this URL doesn't serve JSONP replies or Angular libraries.
●	'unsafe-inline'	'unsafe-inline' allows the execution of unsafe in-page scripts and event handlers.
●	data:	data: URI in default-src allows the execution of unsafe scripts.
?	https://api.coingecko.com/	No bypass found; make sure that this URL doesn't serve JSONP replies or Angular libraries.
?	https://assetmantle-marketplace-collections.s3.ap-south-1.amazonaws.com/	No bypass found; make sure that this URL doesn't serve JSONP replies or Angular libraries.
?	object-src [missing]	Can you restrict object-src to 'none'?
ⓘ	require-trusted-types-for [missing]	Consider requiring Trusted Types for scripts to lock down DOM XSS injection sinks. You can do this by adding "require-trusted-types-for 'script'" to your policy.

```
HTTP/2 200 OK
Date: Fri, 16 Sep 2022 06:23:31 GMT
Content-Length: 0
X-Frame-Options: sameorigin
X-Xss-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'self'
https://assetmantle-test-collection.s3.ap-south-1.amazonaws.com http://localhost:26657
http://localhost:1317 ws://localhost:9000 wss://*.assetmantle.one
https://s3.amazonaws.com/keybase_processed_uploads/ https://demo.docusign.net/
resource://devtools/ https://fcm.googleapis.com/fcm/connect/subscribe
```

```
*.assetmantle.one 'unsafe-inline' data: https://api.coingecko.com/
https://assetmantle-marketplace-collections.s3.ap-south-1.amazonaws.com/
X-Permitted-Cross-Domain-Policies: master-only
Cf-Cache-Status: DYNAMIC
Set-Cookie:
__cf_bm=Vx9Yxb2Dsy9GPotPjkRLXD84Tc0VTT2KDWPz1_H8LOI-1663309411-0-AaNqTzpW
sOJzAy7WIVLHF3hwcjxpwZVj8C9RGlh9DmNysnb0TbwjPZLn8Vff28cRpG7LHoNWDeKFD9C
EY54mP+8=; path=/; expires=Fri, 16-Sep-22 06:53:31 GMT; domain=.assetmantle.one;
HttpOnly; Secure; SameSite=None
Set-Cookie:
_cfuvid=MKkYgPxB5OBQqzWOA3ckhC897S4VMqXkYhyyNXIUGyg-1663309411090-0-6048
00000; path=/; domain=.assetmantle.one; HttpOnly; Secure; SameSite=None
Server: cloudflare
Cf-Ray: 74b7838a7cf39a93-NAG
```

Impacts

The **default-src** allows **unsafe-inline** and **data:** and the **object-src** is missing. These misconfigurations can be abused to execute Javascript payloads and exploit vulnerabilities like XSS.

Remediation

Do not use **unsafe-inline** and **data**. Restrict **object-src** to **none**. Use nonce along with the CSP.

Retest

This is more of a best practice than a security issue, and hence the issue won't be addressed.

Bug ID#10 [Won’t Fix]

Session Cookie missing Secure Attribute

Vulnerability Type

Security Misconfiguration

Severity

Low

Description

Cookies are often a key attack vector for malicious users (typically targeting other users), and the application should always take due diligence to protect cookies. This section looks at how an application can take the necessary precautions when assigning cookies and how to test that these attributes have been correctly configured.

The session cookie was missing a Secure attribute. If the Secure flag is set on a cookie, then browsers will not submit the cookie in any requests that use an unencrypted HTTP connection, thereby preventing the cookie from being trivially intercepted by an attacker monitoring network traffic.
If the Secure flag is not set, then the cookie will be transmitted in clear text if the user visits any HTTP URLs within the cookie's scope.

An attacker may be able to induce this event by feeding a user suitable links, either directly or via another website. Even if the domain that issued the cookie does not host any content that is accessed over HTTP, an attacker may be able to use links of the form http://example.com:443/ to perform the same attack.

Vulnerable URLs

- https://marketplace.assetmantle.one/*

PoC

1. Log in to the application, open the developer tools and go to the Storage tab to look at the cookies.
2. Observe that the session cookie **PLAY_SESSION** lacks the **Secure** attribute.

Impacts
To exploit this vulnerability, the attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection, such as public Wi-Fi or a corporate or home network that is shared with a compromised computer.
Common defenses, such as switched networks, are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack.
Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.

Remediation
The Secure flag should be set on all cookies that are used for transmitting sensitive data when accessing content over HTTPS. If cookies are used to transmit session tokens, then areas of the application that are accessed over HTTPS should employ their own session handling mechanism, and the session tokens used should never be transmitted over unencrypted communications.

Retest
This is more of a best practice than a security issue, and hence the issue won’t be addressed.

Bug ID#11 [Not Applicable]

Mnemonic Stored on the Server

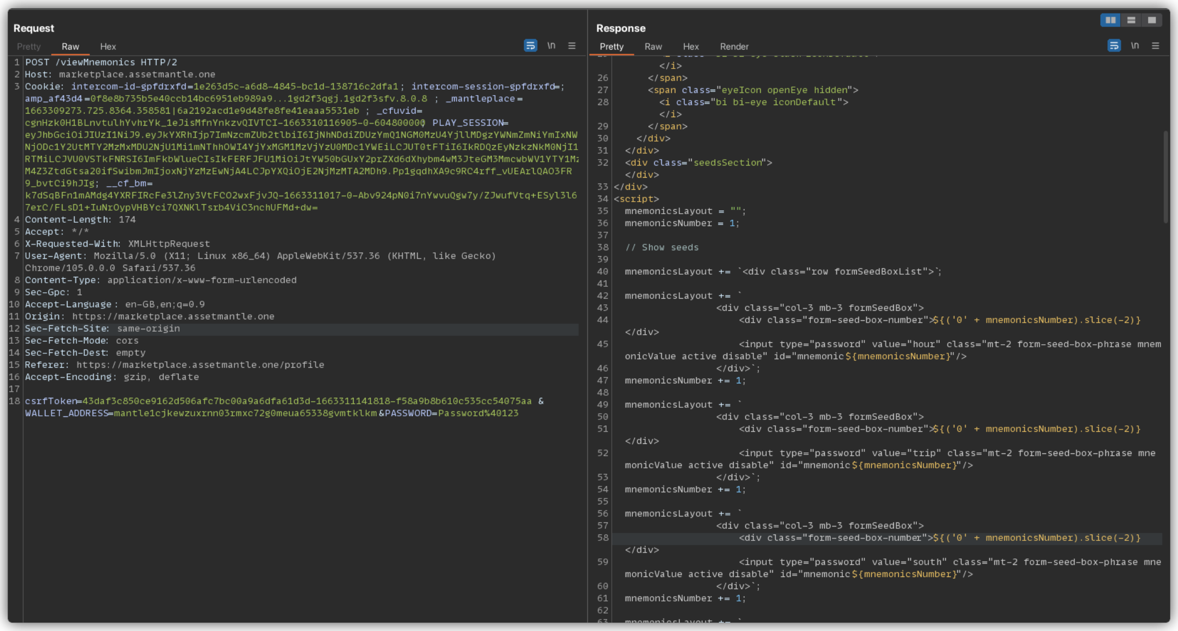
Vulnerability Type
Security Misconfiguration

Severity
High

Description
The application was found to be storing the mnemonic seed on the server side. This is not a good security practice.

Vulnerable URLs
- https://marketplace.assetmantle.one/*

PoC
1. Log in to the application and go to Profile and click on "View Seed Phrase".
2. Enter your password, and the application will make a request to "/viewMnemonics".
3. It can be seen that it is retrieving the data from the server.



```
POST /viewMnemonics HTTP/2
Host: marketplace.assetmantle.one
Cookie: intercom-id-gpfdrxfd=1e263d5c-a6d8-4845-bc1d-138716c2dfa1; intercom-session-gpfdrxfd=;
amp_af43d4=0f8e8b735b5e40ccb14bc6951eb989a9...1gd2f3qgj.1gd2f3sfv.8.0.8 ; _mantleplace=
1663309273.725.8364.358581|6a2192acd1e9d48fe8fe41eaaa5531eb ; _cfuvid=
cgnHzk0H1BLnvtulhYvhrYk_1eJisMfnYnkzvQIVTCI-1663310116905-0-604800000; PLAY_SESSION=
eyJhbGciOiJIUzI1NiJ9.eyJkYXRhIjp7ImNzcmZUb2tlbiI6IjNhNDdiZDUzYmQ1NGM0MzU4YjllMDgzYWNmZmNiYmIxNW
NjODc1Y2UtMTY2MzMxMDU2NjU1Mi1mNThhOWI4YjYxMGM1MzVjYzU0MDc1YWEiLCJUT0tFTiI6IkRDQzEyNzkzNkM0NjI1
RTMiLCJVU0VSTkFNRSI6ImFkbWlueCIsIkFERFJFU1MiOiJtYW50bGUxY2prZXd6dXhybm4wM3JteGM3MmcwbWV1YTY1Mz
M4Z3ZtdGtsa20ifSwibmJmIjoxNjYzMzEwNjA4LCJpYXQiOjE2NjMzMTA2MDh9.Pp1gqdhXA9c9RC4rff_vUEArlQAO3FR
9_bvtCi9hJIg; __cf_bm=
k7dSqBFn1mAMdg4YXRFIRcFe3lZny3VtFCO2wxFjvJQ-1663311017-0-Abv924pN0i7nYwvuQgw7y/ZJwufVtq+ESyl3l6
7erC/FLsD1+IuNrOypVHBYci7QXNKlTsrb4ViC3nchUFMd+dw=
Content-Length: 174
Accept: */*
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/105.0.0.0 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Sec-Gpc: 1
Accept-Language: en-GB,en;q=0.9
Origin: https://marketplace.assetmantle.one
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://marketplace.assetmantle.one/profile
Accept-Encoding: gzip, deflate

csrfToken=43daf3c850ce9162d506afc7bc00a9a6dfa61d3d-1663311141818-f58a9b8b610c535cc54075aa &
WALLET_ADDRESS=mantle1cjkewzuxrnn03rmxc72g0meua65338gvmtklkm&PASSWORD=Password%40123
```

```
        </i>
      </span>
      <span class="eyeIcon openEye hidden">
        <i class="bi bi-eye iconDefault">
        </i>
      </span>
    </div>
  </div>
  <div class="seedsSection">
  </div>
</div>
<script>
  mnemonicsLayout = "";
  mnemonicsNumber = 1;

  // Show seeds

  mnemonicsLayout += `<div class="row formSeedBoxList">`;

  mnemonicsLayout += `
                <div class="col-3 mb-3 formSeedBox">
                    <div class="form-seed-box-number">${('0' + mnemonicsNumber).slice(-2)}
  </div>
                    <input type="password" value="hour" class="mt-2 form-seed-box-phrase mnem
  onicValue active disable" id="mnemonic${mnemonicsNumber}"/>
                </div>`;
  mnemonicsNumber += 1;

  mnemonicsLayout += `
                <div class="col-3 mb-3 formSeedBox">
                    <div class="form-seed-box-number">${('0' + mnemonicsNumber).slice(-2)}
  </div>
                    <input type="password" value="trip" class="mt-2 form-seed-box-phrase mne
  monicValue active disable" id="mnemonic${mnemonicsNumber}"/>
                </div>`;
  mnemonicsNumber += 1;

  mnemonicsLayout += `
                <div class="col-3 mb-3 formSeedBox">
                    <div class="form-seed-box-number">${('0' + mnemonicsNumber).slice(-2)}
  </div>
                    <input type="password" value="south" class="mt-2 form-seed-box-phrase mne
  monicValue active disable" id="mnemonic${mnemonicsNumber}"/>
                </div>`;
  mnemonicsNumber += 1;

  mnemonicsLayout += `
```

```
POST /viewMnemonics HTTP/2
Host: marketplace.assetmantle.one
Cookie: intercom-id-gpfdrxfd=1e263d5c-a6d8-4845-bc1d-138716c2dfa1;
intercom-session-gpfdrxfd=;
amp_af43d4=0f8e8b735b5e40ccb14bc6951eb989a9...1gd2f3qgj.1gd2f3sfv.8.0.8;
_mantleplace=1663309273.725.8364.358581|6a2192acd1e9d48fe8fe41eaaa5531eb;
_cfuvid=cgnHzk0H1BLnvtulhYvhrYk_1eJisMfnYnkzvQIVTCI-1663310116905-0-604800000;
PLAY_SESSION=eyJhbGciOiJIUzI1NiJ9.eyJkYXRhIjp7ImNzcmZUb2tlbiI6IjNhNDdiZDUzYmQ1
NGM0MzU4YjllMDgzYWNmZmNiYmIxNWNjODc1Y2UtMTY2MzMxMDU2NjU1Mi1mNThhO
WI4YjYxMGM1MzVjYzU0MDc1YWEiLCJUT0tFTiI6IkRDQzEyNzkzNkM0NjI1RTMiLCJVU0VSTk
FNRSI6ImFkbWlueCIsIkFERFJFU1MiOiJtYW50bGUxY2prZXd6dXhybm4wM3JteGM3Mmcwb
WV1YTY1MzM4Z3ZtdGtsa20ifSwibmJmIjoxNjYzMzEwNjA4LCJpYXQiOjE2NjMzMTA2MDh9.
Pp1gqdhXA9c9RC4rff_vUEArlQAO3FR9_bvtCi9hJIg;
__cf_bm=k7dSqBFn1mAMdg4YXRFIRcFe3lZny3VtFCO2wxFjvJQ-1663311017-0-Abv924pN0i
7nYwvuQgw7y/ZJwufVtq+ESyl3l67erC/FLsD1+IuNrOypVHBYci7QXNKlTsrb4ViC3nchUFMd+
dw=
Content-Length: 174
Accept: */*
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/105.0.0.0 Safari/537.36
```

```
Content-Type: application/x-www-form-urlencoded
Sec-Gpc: 1
Accept-Language: en-GB,en;q=0.9
Origin: https://marketplace.assetmantle.one
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://marketplace.assetmantle.one/profile
Accept-Encoding: gzip, deflate

csrfToken=43daf3c850ce9162d506afc7bc00a9a6dfa61d3d-1663311141818-f58a9b8b610
c535cc54075aa&WALLET_ADDRESS=mantle1cjkewzuxrnn03rmxc72g0meua65338gvmtklk
m&PASSWORD=Password%40123
```

Impacts

If the server gets compromised, all the users will lose their data, NFT’s and wallets.

Remediation

It is recommended to not store the mnemonic seed on the server. Users should have complete control over their wallets and its seed.

Retest

The team informed that this is working as intended.

Bug ID#12 [Fixed]

CSRF to Add and Remove NFTs from Wishlist

Vulnerability Type
CWE-352: Cross-Site Request Forgery

Severity
Medium

Description
Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated. CSRF attacks specifically target state-changing requests, not theft of data, since the attacker has no way to see the response to the forged request. With a little help of social engineering (such as sending a link via email or chat), an attacker may trick the users of a web application into executing actions of the attacker's choosing.

The endpoints to add and remove NFT from wish were found to be affected and lacked CSRF validations and tokens.

Vulnerable URLs
- https://marketplace.assetmantle.one/deleteFromWishList?nftId=2bd6dd8c0c571f525ce848e490bc6f22070d81001cb91078f2fdd2c5c15a2679.png
- https://marketplace.assetmantle.one/addToWishList?nftId=2bd6dd8c0c571f525ce848e490bc6f22070d81001cb91078f2fdd2c5c15a2679.png

PoC
1. Send the above requests to an authenticated victim and note that the NFT ID's mentioned in the URL will be added or removed from the wishlist without validation.

Impacts
This vulnerability allows attackers to send crafted URLs to victims to add or delete NFTs from their wishlists without their confirmation.

Remediation

The following principles should be followed to defend against CSRF:

- Check if your framework has built-in CSRF protection and use it
- If the framework does not have built-in CSRF protection, add CSRF tokens to all state-changing requests (requests that cause actions on the site) and validate them on the backend
- For stateful software use the synchronizer token pattern
- For stateless software use double submit cookies
- Implement at least one mitigation from Defense in Depth Mitigations section
- Consider SameSite Cookie Attribute for session cookies but be careful to NOT set a cookie specifically for a domain as that would introduce a security vulnerability that all subdomains of that domain share the cookie. This is particularly an issue when a subdomain has a CNAME to domains not in your control.
- Consider implementing user interaction-based protection for highly sensitive operations
- Consider the use of custom request headers
- Consider verifying the origin with standard headers
- Remember that any Cross-Site Scripting (XSS) can be used to defeat all CSRF mitigation techniques!
- See the OWASP XSS Prevention Cheat Sheet for detailed guidance on how to prevent XSS flaws.
- Do not use GET requests for state-changing operations.
- If for any reason you do it, protect those resources against CSRF.

All these points are elaborated in the OWASP Cross-Site Request Forgery Prevention Cheat Sheet

Retest

Anti-CSRF token has been added to mitigate the vulnerability.

Bug ID#13 [Pending Fix]

0 Price for NFT causes Denial of Service

Vulnerability Type

Denial of Service (application level)

Severity

High

Description

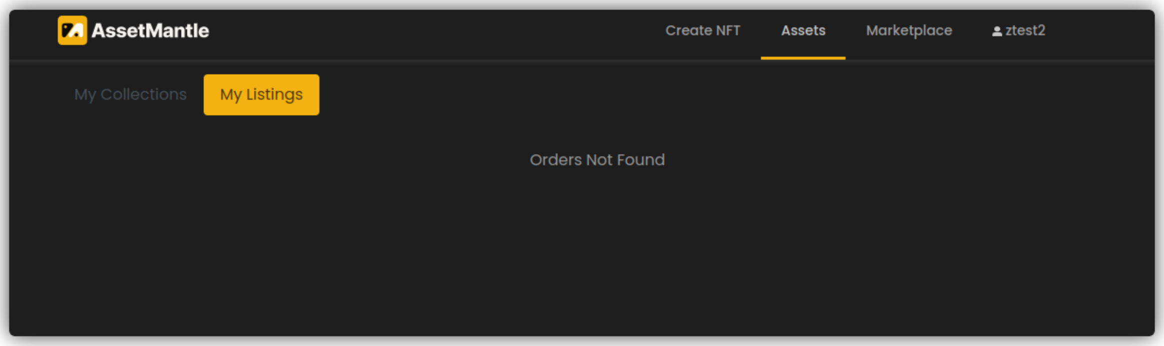
The application is not handling 0 price for selling the NFT properly and causes a denial of service if a user tries to sell their NFT for 0 tokens.
The "My Listings" page becomes unresponsive and does not show any of the orders.

Vulnerable URLs

- https://devnet.assetmantle.one/assets

PoC:

1. Go to the Marketplace and create an NFT.
2. Go to "My Collections" and sell the NFT for 0 amount.
3. After the order is created, observe that the "My Listings" page won't show any of the orders.



Orders Not Found

Impacts

This vulnerability causes a denial of service for the Listings page and the user's orders won't show up there.

Remediation
Implement an input validation on the NFT amount or handle the errors properly so that the listing page still works if the user enters 0 for the amount.

Retest
This will be fixed in the next release.

Bug ID#14 [Won't Fix]

MKDocs Vulnerable Version

Vulnerability Type
CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

Severity
Low

Description
The documentation portal is using MKDocs 1.2.3 which is vulnerable to a Cross-SIte Scripting (XSS) in the search results parameters.
Since the malicious input is controlled from the administrator's side, the chances of exploitation is really low.

Vulnerable URLs

- https://docs.assetmantle.one/

PoC

1. View the source at https://docs.assetmantle.one/ and observe the version number is 1.2.3.
2. The corresponsind commit for the XSS fix can be found here - https://github.com/mkdocs/mkdocs/commit/5cf196361bb0f8364f667ed98888ffa064982efa

Impacts
XSS vulnerabilities can be used to exploit users and steal their sessions and cookies. It may be used to impersonate them and execute malicious javascript codes in their browsers. In this case, it is near to impossible since the malicious input is controlled by the admins.

References
https://security.snyk.io/vuln/SNYK-PYTHON-MKDOCS-2438396

Remediation
Update MKDocs to its latest version.

Retest
At this stage it was determined that the bug was not exploitable and hence this won’t be fixed.

Bug ID#15 [Won't Fix]

Outdated Javascript Library (Moment.js)

Vulnerability Type
Using Components with known Vulnerabilities

Severity
Low

Description
The application was found to be using a vulnerable and outdated javascript component which was affected by multiple publicly known exploited and CVEs.
Moment.js 2.18.1 was found implemented on the Marketplace application which was outdated.

Vulnerable URLs

- https://marketplace.assetmantle.one/assets/javascripts/library/moment.min.js

PoC

1. Go to the URL shown above and note the outdated version number.

Impacts
This particular version of moment.js is affected by the following vulnerabilities:

- Regular Expression Denial of Service (ReDoS) **CVE-2017-18214**
- This vulnerability impacts npm (server) users of moment.js, especially if user-provided locale string, eg fr is directly used to switch moment locale. **CVE-2022-24785**
- Regular Expression Denial of Service (ReDoS), Affecting moment package, versions >=2.18.0 <2.29.4 **CVE-2022-31129**

References
https://security.snyk.io/package/npm/moment/2.18.1

Remediation
Update the library to its latest version.

Retest
At this stage it was determined that the bug was not exploitable and hence this won’t be fixed.

6. Appendix 1

6.1 Disclosure:

The Reports neither endorse nor condemn any specific project or team, nor do they guarantee the security of any specific project. The contents of this report do not, and should not be interpreted as having any bearing on, the economics of tokens, token sales, or any other goods, services, or assets.

Emerging technologies such as Blockchain, Smart Contracts, and Solidity carry a high level of technical risk and uncertainty. There is no warranty or representation made by this report to any Third Party in regards to the quality of code, the business model or the proprietors of any such business model, or the legal compliance of any business.

In no way should a third party use these reports to make any decisions about buying or selling a token, product, service, or any other asset. It should be noted that this report is not investment advice, is not intended to be relied on as investment advice, and has no endorsement of this project or team. It does not serve as a guarantee of the project's absolute security.

CredShields Audit team owes no duty to any third party by virtue of publishing these Reports.