SALUS SECURITY

NOV 2023

CODE SECURITY ASSESSMENT

P12

Overview

Project Summary

- Name: p12 - MUD template with p12 editor
- Version: commit 92e881e
- Platform: EVM-compatible chains
- Language: Solidity
- Repository: https://github.com/ProjectTwelve/mud-template-p12-editor
- Audit Range: See Appendix - 1

Project Dashboard

Application Summary

Name	p12 - MUD template with p12 editor
Version	v2
Type	Solidity
Dates	Nov 20 2023
Logs	Nov 2 2023; Nov 20 2023

Vulnerability Summary

Total High-Severity issues	0
Total Medium-Severity issues	0
Total Low-Severity issues	1
Total informational issues	3
Total	4

Contact

E-mail: support@salusec.io

Risk Level Description

High Risk	The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for clients' reputations or serious financial implications for clients and users.
Medium Risk	The issue puts a subset of users' sensitive information at risk, would be detrimental to the client's reputation if exploited, or is reasonably likely to lead to a moderate financial impact.
Low Risk	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low impact in view of the client's business circumstances.
Informational	The issue does not pose an immediate risk, but is relevant to security best practices or defense in depth.

Content

Introduction

1.1 About SALUS

At Salus Security, we are in the business of trust.

We are dedicated to tackling the toughest security challenges facing the industry today. By building foundational trust in technology and infrastructure through security, we help clients to lead their respective industries and unlock their full Web3 potential.

Our team of security experts employ industry-leading proof-of-concept (PoC) methodology for demonstrating smart contract vulnerabilities, coupled with advanced red teaming capabilities and a stereoscopic vulnerability detection service, to deliver comprehensive security assessments that allow clients to stay ahead of the curve.

In addition to smart contract audits and red teaming, our Rapid Detection Service for smart contracts aims to make security accessible to all. This high calibre, yet cost-efficient, security tool has been designed to support a wide range of business needs including investment due diligence, security and code quality assessments, and code optimisation.

We are reachable on Telegram (https://t.me/salusec), Twitter (https://twitter.com/salus_sec), or Email (support@salusec.io).

1.2 Audit Breakdown

The objective was to evaluate the repository for security-related issues, code quality, and adherence to specifications and best practices. Possible issues we looked for included (but are not limited to):

- Risky external calls
- Integer overflow/underflow
- Transaction-ordering dependence
- Timestamp dependence
- Access control
- Call stack limits and mishandled exceptions
- Number rounding errors
- Centralization of power
- Logical oversights and denial of service
- Business logic specification
- Code clones, functionality duplication

1.3 Disclaimer

Note that this security audit is not designed to replace functional tests required before any software release and does not give any warranties on finding all possible security issues with the given smart contract(s) or blockchain software, i.e., the evaluation result does not guarantee the nonexistence of any further findings of security issues.

Findings

2.1 Summary of Findings

ID	Title	Severity	Category	Status
1	Concerns about third-party dependency	Low	Dependencies	Acknowledged
2	Unused console library	Informational	Redundancy	Acknowledged
3	Unused custom error definition	Informational	Redundancy	Acknowledged
4	Upgrading pragma to at least 0.8.4	Informational	Configuration	Acknowledged

2.2 Notable Findings

Significant flaws that impact system confidentiality, integrity, or availability are listed below.

1. Concerns about third-party dependency

Severity: Low	Category: Dependencies
Target: - contracts/src/systems/IncrementSystem.sol	

Description

The current IncrementSystem contract relies on the MUD framework. The MUD framework is beyond the current scope and is treated as a trusted third-party dependency during this audit.

It should be noted that the MUD framework is still under development, and its security cannot be guaranteed. Any security vulnerabilities in the MUD framework could impact the current IncrementSystem template.

Recommendation

The team should monitor the security of critical third-party dependencies for the project.

Status

This issue has been acknowledged by the team.

2.3 Informational Findings

2. Unused console library

Severity: Informational	Category: Redundancy
Target: - contracts/src/systems/IncrementSystem.sol	

Description

In the actual deployment and application, there is no need to use the "console" library.

contracts/src/systems/IncrementSystem.sol:L4

```
import {console} from "forge-std/console.sol";
```

Recommendation

Consider removing the line that imports the "console" library to clean up the code and reduce unnecessary dependencies.

Status

This issue has been acknowledged by the team.

3. Unused custom error definition

Severity: Informational	Category: Redundancy

Target:
- contracts/src/systems/IncrementSystem.sol

Description

There is a custom error definition named MyCustomError, but it does not appear to be used anywhere in the contract's code or its associated functions.

contracts/src/systems/IncrementSystem.sol:L9

```
error MyCustomError();
```

Recommendation

Consider removing the custom error definition MyCustomError to eliminate code redundancy.

Status

This issue has been acknowledged by the team.

4. Upgrading pragma to at least 0.8.4

Severity: Informational	Category: Configuration
Target: - contracts/src/systems/IncrementSystem.sol	

Description

In the contract, the custom error MyCustomError is defined. The custom errors feature was introduced to Solidity in version 0.8.4. However, the contract specifies the pragma directive as pragma solidity >=0.8.0.

contracts/src/systems/IncrementSystem.sol:L2

```
pragma solidity >=0.8.0;
```

Recommendation

Consider upgrading the pragma directive to pragma solidity >=0.8.4 or a later version to ensure compatibility with the custom errors feature.

Status

This issue has been acknowledged by the team.

Appendix

Appendix 1 - Files in Scope

This audit covered the following files in commit 92e881e:

File	SHA-1 hash
contracts/src/systems/IncrementSystem.sol	ac56c66ce34d6dbff8f074abc2cf303a3b0cc96d