SALUS SECURITY

AUG 2024

CODE SECURITY ASSESSMENT

LISTA DAO

Overview

Project Summary

- Name: Lista Dao - Emission
- Platform: BNB Smart Chain
- Language: Solidity
- Repository:
    - https://github.com/lista-dao/lista-dao-contracts
- Audit Range: See Appendix - 1

Project Dashboard

Application Summary

Name	Lista Dao - Emission
Version	v2
Type	Solidity
Dates	Aug 13 2024
Logs	Aug 09 2024; Aug 13 2024

Vulnerability Summary

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

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	Check of live in exit function will stop user's withdraw	Low	Centralization	Mitigated

2.2 Notable Findings

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

1. Check of live in exit function will stop user's withdraw

Severity: Low	Category: Centralization
Target: - contracts/Jar.sol	

Description

contracts/Jar.sol:L208-L229

```
function exit(uint256 wad) external update(msg.sender) nonReentrant {
    require(live == 1, "Jar/not-live");
    ...
}
```

In Jar contracts, the `exit()` function will revert if the `live` isn't 1. This means if the `auth` owner changes the `live` then the user can't call the `exit` to withdraw the user's funds.

If the owner's or admin's private key is compromised, an attacker can block the user's withdrawal.

Recommendation

It is recommended to remove the `live` check or use `_checkIsLive` in the exit function.

Status

This issue has been mitigated by the team. The team has stated that they will use a multisig wallet as the owner.

2.3 Informational Findings

No Informational issues are found.

Appendix

Appendix 1 - Files in Scope

This audit covered the following files in commit 8244238:

File	SHA-1 hash
contracts/Interaction.sol	430f6f3de99df24fc32317aaced583751d894ed8
contracts/Jar.sol	671ed7661e36724514528015c25dd0f0a4601cd4