SALUS SECURITY

OCT 2024

CODE SECURITY ASSESSMENT

AVALON FINANCE

Overview

Project Summary

- Name: Avalon Finance - USDA
- Platform: EVM-compatible chains
- Language: Solidity
- Repository:
  - https://github.com/avalonfinancexyz/USDa-oft/
- Audit Range: See Appendix - 1

Project Dashboard

Application Summary

Name	Avalon Finance - USDA
Version	v2
Type	Solidity
Dates	Oct 18 2024
Logs	Sep 24 2024; Oct 18 2024

Vulnerability Summary

Total High-Severity issues	0
Total Medium-Severity issues	1
Total Low-Severity issues	1
Total informational issues	0
Total	2

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	Centralization risk	Medium	Centralization	Acknowledged
2	Third-party dependencies	Low	Dependency	Acknowledged

2.2 Notable Findings

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

1. Centralization risk	
Severity: Medium	Category: Centralization
Target: - contracts/USDa.sol	

Description

The `BitULocking` contract exists for the owner's account. The owner can add any address as a `mintVault` or `burnVault`. While `mintVault` can mint any number of tokens for any address, `burnVault` can destroy tokens for any address.

If the private key of the owner's address is compromised, an attacker can add his address as `mintVault` to mint a large number of tokens for himself and destroy other users' tokens at will.

If the privileged account is a regular EOA account, this could be a concern and pose a risk to other users.

Recommendation

We recommend transferring privileged accounts to multi-sig accounts with timelock governors for enhanced security. This ensures that no single person has full control over the accounts and that any changes must be authorized by multiple parties.

Status

This issue has been acknowledged by the team.

2. Third-party dependencies	
Severity: Low	Category: Dependency
Target: - contracts/USDa.sol	

Description

The `` `AUSD` `` contract relies on the `` `OFTV2` `` contract to enable the basic functionality of the token. The current audit treats third-party entities as black boxes and assumes they are working correctly. However, in reality, third parties could be compromised, resulting in the disruption of token functionalities.

Recommendation

We understand that the business logic requires interaction with the third parties. We encourage the team to regularly monitor the statuses of third parties to reduce the impacts when they are not functioning properly.

Status

This issue has been acknowledged by the team.

2.3 Informational Findings

No Informational Findings had been found.

Appendix

Appendix 1 - Files in Scope

This audit covered the following files in commit 3933493:

File	SHA-1 hash
contracts/USDa.sol	91dfeef0c76ac514b62ba4f8ce2c035870c62aca
contracts/USDaOFTAdapter.sol	3face2d87a3bd902c55ae74f08bcf5b11f673b84