SALUS SECURITY

MAY 2024

CODE SECURITY ASSESSMENT

TEAHOUSE FINANCE

Overview

Project Summary

- Name: Teahouse Finance - Incremental Audit
- Platform: EVM-compatible chains
- Language: Solidity
- Repository:
    - https://github.com/TeahouseFinance/TeaVaultV3Pair
- Audit Range: See Appendix - 1

Project Dashboard

Application Summary

Name	Teahouse Finance - Incremental Audit
Version	v2
Type	Solidity
Dates	May 30 2024
Logs	May 28 2024; May 30 2024

Vulnerability Summary

Total High-Severity issues	0
Total Medium-Severity issues	0
Total Low-Severity issues	1
Total informational issues	1
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 **4**
1.1 About SALUS 4
1.2 Audit Breakdown 4
1.3 Disclaimer 4
Findings **5**
2.1 Summary of Findings 5
2.2 Notable Findings 6
1. Missing events for functions that change critical state 6
2.3 Informational Findings 7
2. Missing zero address checks 7
Appendix **8**
Appendix 1 - Files in Scope 8

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	Missing events for functions that change critical state	Low	Logging	Resolved
2	Missing zero address checks	Informational	Data Validation	Resolved

2.2 Notable Findings

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

1. Missing events for functions that change critical state

Severity: Medium	Category: Logging
Target: - contracts/TeaVaultV3Pair.sol	

Description

Events allow capturing the changed parameters so that off-chain tools/interfaces can register such changes that allow users to evaluate them. Missing events do not promote transparency and if such changes immediately affect users' perception of fairness or trustworthiness, they could exit the protocol causing a reduction in protocol users.

In the TeaVaultV3Pair.sol, events are lacking in the following privileged setter functions.
- setRewardClaimer()
- claimAndForwardReward()

Recommendation

It is recommended to emit events for critical state changes.

Status

The team has resolved this issue in commit 417ebd8.

2.3 Informational Findings

2. Missing zero address checks	
Severity: Informational	Category: Data Validation
Target: - contracts/TeaVaultV3Pair.sol	

Description

It is considered a security best practice to verify addresses against the zero address during initialization or setting. However, this precautionary step is absent for address variables in the following functions.

contracts/TeaVaultV3Pair.sol:L906-L908

```
function setRewardClaimer(address _rewardClaimer) external onlyOwner {
    rewardClaimer = _rewardClaimer;
}
```

contracts/TeaVaultV3Pair.sol:L910-L934

```
function claimAndForwardReward(INileGauge _gauge, address _to) external
onlyRewardClaimer nonReentrant {
    ...
    rewardToken.safeTransfer(_to, balance - fee);
    ...
}
```

Recommendation

Consider adding zero address checks for address variables.

Status

The team has resolved this issue in commit 417ebd8.

Appendix

Appendix 1 - Files in Scope

This audit covered the following files in commit d91dcc6:

File	SHA-1 hash
TeaVaultV3Pair.sol	2e450a7add22025f83ba53c3b3a4618781ddba50
TeaVaultV3PairHelper.sol	deaa13c117ade873f1ea72b353d8e854a955b0db

And we audited the commit 2ecf324 that introduced new features to this repository.

File	SHA-1 hash
TeaVaultV3Pair.sol	58d34daa1eda7473d1145c8e42e82ff52af8221d
TeaVaultV3PairHelper.sol	4e0f0f1101a0943689c458cc74f956107de32652