SALUS SECURITY

FEB 2024

CODE SECURITY ASSESSMENT

BOUNCEBIT

Overview

Project Summary

- Name: BounceBit - Vault
- Platform: EVM-compatible Chains
- Language: Solidity
- Address:
    - BNB Smart Chain : 0x60cfC0c47767569865d979b4f2FF804aaF46083B
    - Ethereum: 0xe0D0Cf872C09968f1554730042Ee680354deC994
- Audit Scope: See Appendix - 1

Project Dashboard

Application Summary

Name	BounceBit - Vault
Version	v1
Type	Solidity
Date	Feb 09 2024
Logs	Feb 09 2024

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	Third-party dependencies	Low	Dependency	Pending
2	Missing two-step transfer ownership pattern	Informational	Business logic	Pending
3	Missing remove whitelist function	Informational	Business logic	Pending
4	Lack of indexed parameters in events	Informational	Business logic	Pending

2.2 Notable Findings

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

1. Third-party dependencies	
Severity: Low	Category: Dependency
Target: - BounceBitVault.sol	

Description

The BounceBitVault contract relies on a multisig wallet (the “to” address in the stake() function) to enable staking, reward distribution, and withdrawal. 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 stake functionality of BounceBitVault being unavailable.

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.

2.3 Informational Findings

2. Missing two-step transfer ownership pattern	
Severity: Informational	Category: Business logic
Target: - BounceBitVault.sol	

Description

The BounceBitVault contract inherits from the Ownable contract. This contract does not implement a two-step process for transferring ownership. Thus, ownership of the contract can easily be lost when making a mistake in transferring ownership.

Recommendation

Consider using the Ownable2Step contract from OpenZeppelin instead.

3. Missing remove whitelist function

Severity: Informational	Category: Business logic
Target: - BounceBitVault.sol	

Description

In the BounceBitVault contract, missing the ability to remove a whitelist. This results in the project side not being able to remove support for a particular token after adding it to the whitelist.

This means that when an exception occurs with a particular token, the project owner is unable to take contingency measures.

Recommendation

It is recommended to add removeWhiteList() function to remove support for a particular token.

4. Lack of indexed parameters in events

Severity: Informational	Category: Logging
Target: - BounceBitVault.sol	

Description

Event emission and properly indexed parameters assist off-chain observers watch, search, and filter on-chain activity.

BounceBitVault.sol:L17

```
event Staked(address token, address from, address to, uint256 amount);
```

In BounceBitVault.sol, the Staked event does not apply the indexed keyword to the token and from parameters.

Recommendation

Consider indexing applicable event parameters to support the searching and filtering abilities of offchain services.

Appendix

Appendix 1 - Files in Scope

This audit covered the contract from the address of

0x60cfC0c47767569865d979b4f2FF804aaF46083B in BNB Smart Chain;

0xe0D0Cf872C09968f1554730042Ee680354deC994 in Ethereum.