Volo Smart Contract

Audit Report

contact@bitslab.xyz

https://twitter.com/movebit_

Thu Sep 07 2023

Volo Smart Contract Audit Report

1 Executive Summary

1.1 Project Information

Description	Volo is a liquid staking solution that allows users to maintain liquidity while participating in staking
Type	DeFi
Auditors	MoveBit
Timeline	Tue Aug 08 2023 - Tue Aug 22 2023
Languages	Move
Platform	Sui
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/Sui-Volo/volo-liquid-staking-contracts
Commits	d088758139f34f27a2acf65cdc3e1f89dfcd6596

1.2 Files in Scope

The following are the SHA1 hashes of the original reviewed files.

ID	File	SHA-1 Hash
MOV	liquid_staking/Move.toml	663a37b750de9856ae19d8861f702a1ca836a7ac
NPO	liquid_staking/sources/native_pool.move	e716fccee5e831b8c0432fbce1d45420c5513606
MAT	liquid_staking/sources/math.move	001a8bedea7a00a81a277993160b704b956c8041
VSE	liquid_staking/sources/validator_set.move	08b2ac325abdef61c322846698f33b06a7f870b5
OWN	liquid_staking/sources/ownership.move	83dc76c6a5cfd3150a9b3267e3b95e45781a8a14
UTI	liquid_staking/sources/unstake_ticket.move	b2e323bcde03459d4ff320821ee208feb0bbb3c2
CER	liquid_staking/sources/cert.move	a31a9be25b2943e65b5eaf6c9cf9f2b937608f08

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	9	8	1
Informational	0	0	0
Minor	3	3	0
Medium	1	1	0
Major	5	4	1
Critical	0	0	0

1.4 MoveBit Audit Breakdown

MoveBit aims to assess repositories for security-related issues, code quality, and compliance with specifications and best practices. Possible issues our team looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Integer overflow/underflow by bit operations
- Number of rounding errors
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting
- Unchecked CALL Return Values
- The flow of capability
- Witness Type

1.5 Methodology

The security team adopted the **"Testing and Automated Analysis"**, **"Code Review"** and **"Formal Verification"** strategy to perform a complete security test on the code in a way that is closest to the real attack. The main entrance and scope of security testing are stated in the conventions in the "Audit Objective", which can expand to contexts beyond the scope according to the actual testing needs. The main types of this security audit include:

(1) Testing and Automated Analysis

Items to check: state consistency / failure rollback / unit testing / value overflows / parameter verification / unhandled errors / boundary checking / coding specifications.

(2) Code Review

The code scope is illustrated in section 1.2.

(3) Formal Verification

Perform formal verification for key functions with the Move Prover.

(4) Audit Process

- Carry out relevant security tests on the testnet or the mainnet;
- If there are any questions during the audit process, communicate with the code owner in time. The code owners should actively cooperate (this might include providing the latest stable source code, relevant deployment scripts or methods, transaction signature scripts, exchange docking schemes, etc.);
- The necessary information during the audit process will be well documented for both the audit team and the code owner in a timely manner.

2 Summary

This report has been commissioned by Volo to identify any potential issues and vulnerabilities in the source code of the Volo smart contract, as well as any contract dependencies that were not part of an officially recognized library. In this audit, we have utilized various techniques, including manual code review and static analysis, to identify potential vulnerabilities and security issues.

During the audit, we identified 9 issues of varying severity, listed below.

ID	Title	Severity	Status
NPO-1	Unused Constants	Minor	Fixed
NPO-2	Incorrect Function Logic	Major	Fixed
NPO-3	Centralization Risk	Major	Acknowledged
NPO-4	Lack of Validation for Version	Minor	Fixed
NPO-5	Inconsistent Handling of `reward_fee`	Major	Fixed
UTI-1	Redundant Function Calls in `wrap_unstake_ticket`	Minor	Fixed
VSE-1	Inconsistent Deduction Logic in `remove_stakes` Function	Major	Fixed
VSE-2	Incorrect Sort Function Implementation	Major	Fixed
VSE-3	Risk of Self-Dos	Medium	Fixed

3 Participant Process

Here are the relevant actors with their respective abilities within the Volo Smart Contract :

Admin

- Admin can update the minimum allowed Sui number of stakes through `change_min_stake`.
- Admin can update the threshold of `unstake_fee` through `change_unstake_fee_threshold`.
- Admin can withdraw fee through `collect_fee`.
- Admin can pause/unpause `NativePool` through `set_pause`.
- Admin can update rewards to upgrade the ratio of tokens with requirements through `update_rewards`.
- Admin can update `base_unstake_fee` through `change_base_unstake_fee`.
- Admin can update validators and their priorities in the validator set through `update_validators`.

User

- Users can stake `Sui` through `stake`.
- Users can unstake `voloSUI` to get some `UnstakeTicket` or `Sui` through `unstake`.
- Users can burn `UnstakeTicket` to get `Sui` through `burn_ticket`.
- Users can use `voloSUI` to mint `UnstakeTicket` through `mint_ticket`.

4 Findings

NPO-1 Unused Constants

Severity: Minor

Status: Fixed

Code Location:

liquid_staking/sources/native_pool.move#28,42

Descriptions:

There are two unused constants `EPOCH_DURATION` and `E_LIMIT_TOO_BIG` in the `native_pool` module.

Suggestion:

It is recommended to remove unused constants.

Resolution:

The client has followed our suggestion and fixed the issue.

NPO-2 Incorrect Function Logic

Severity: Major

Status: Fixed

Code Location:

liquid_staking/sources/native_pool.move#357

Descriptions:

When both branch judgments in the `sub_total_staked_unsafe` function are else, `last_total_staked` is subtracted twice in L390. Such as: The first user after the contract is deployed to `stake` and `unstake` at the same `epoch`. the `last_total_staked` will subtract twice. It will affect the value of `get_ratio` in L601, the ratio will become bigger and then affect the calculation of rewards.

Suggestion:

It is recommended to modify the function logic of the `sub_total_staked_unsafe` function.

Resolution:

The client has followed our suggestion and fixed the issue.

NPO-3 Centralization Risk

Severity: Major

Status: Acknowledged

Code Location:

liquid_staking/sources/native_pool.move#292

Descriptions:

There are some risks of centralization in the contract, the admin can set the `total_rewards` of the `NativePool`, which will result in a change in the rate calculation of the contract.

Suggestion:

It is recommended to take some measures to mitigate centralization risk.

Resolution:

The client replied that it's not possible to calculate rewards on-chain. We plan to upgrade contracts when SUI implements all the needed methods.

NPO-4 Lack of Validation for Version

Severity: Minor

Status: Fixed

Code Location:

liquid_staking/sources/native_pool.move#272

Descriptions:

The function `update_validators` does not check the version.

Suggestion:

It is recommended to check the version in this function before interaction with the pool to interact with the package version of the pool must be less than the package version.

Resolution:

The client has followed our suggestion and fixed the issue.

NPO-5 Inconsistent Handling of `reward_fee`

Severity: Major

Status: Fixed

Code Location:

liquid_staking/sources/native_pool.move#581

Descriptions:

In the `update_rewards` function, the `total_rewards` set by the `set_rewards_unsafe` function includes `reward_fee`, but in line 581 of the `unstake_amount_from_validators` function `sub_rewards_unsafe(self, rewards - reward_fee)` subtracted `reward_fee`.

Suggestion:

In the `unstake_amount_from_validators` function, modify the `sub_rewards_unsafe` call on line 581 to not subtract the `reward_fee`.

Resolution:

The client has followed our suggestion and fixed the issue.

UTI-1 Redundant Function Calls in `wrap_unstake_ticket`

Severity: Minor

Status: Fixed

Code Location:

liquid_staking/sources/unstake_ticket.move#124-126

Descriptions:

It has been observed that redundant function calls are made to retrieve parameters already available from the initialized `UnstakeTicket` struct. The functions `get_value`, `get_unlock_epoch`, and `get_unstake_fee` are called despite these values being accessible directly from the struct's initialization parameters.

Suggestion:

It is recommended to refactor the `wrap_unstake_ticket` function to directly utilize the parameters used to initialize the `UnstakeTicket` struct for writing events.

Resolution:

The client has followed our suggestion and fixed the issue.

VSE-1 Inconsistent Deduction Logic in `remove_stakes` Function

Severity: Major

Status: Fixed

Code Location:

liquid_staking/sources/validator_set.move#188

Descriptions:

In the `remove_stakes` function, when the condition of L184 is not satisfied, the logic of L188-L191 will be executed. The value of `requested_amount` should be changed to `requested_amount - principal_value`. If the value of `requested_amount` is not updated, the actual amount withdrawn will be greater than `requested_amount`.

Suggestion:

It is recommended to modify `requested_amount` to `requested_amount - principal_value` at L188-L191 in `remove_stakes` function.

Resolution:

The client has followed our suggestion and fixed the issue.

VSE-2 Incorrect Sort Function Implementation

Severity: Major

Status: Fixed

Code Location:

liquid_staking/sources/validator_set.move#87

Descriptions:

There is a problem with the sorting logic of the `sort_validators` function, the result of the function is not sorted according to the size of `vldr_prior`.

Suggestion:

It is recommended to modify the sorting logic of the `sort_validators` function.

Resolution:

The client has followed our suggestion and fixed the issue.

VSE-3 Risk of Self-Dos

Severity: Medium

Status: Fixed

Code Location:

liquid_staking/sources/validator_set.move#199

Descriptions:

In line 199 of the `remove_stakes` function, it is necessary to deal with the situation that `staked_sui_mut_ref - requested_amount` is less than `1 Sui`, otherwise it will cause self-dos.

Suggestion:

It is recommended to modify the function logic to ensure that in line 199 `staked_sui_mut_ref - requested_amount` is greater than 1 Sui.

Resolution:

The client has followed our suggestion and fixed the issue.

Appendix 1

Issue Level

- **Informational** issues are often recommendations to improve the style of the code or to optimize code that does not affect the overall functionality.
- **Minor** issues are general suggestions relevant to best practices and readability. They don't post any direct risk. Developers are encouraged to fix them.
- **Medium** issues are non-exploitable problems and not security vulnerabilities. They should be fixed unless there is a specific reason not to.
- **Major** issues are security vulnerabilities. They put a portion of users' sensitive information at risk, and often are not directly exploitable. All major issues should be fixed.
- **Critical** issues are directly exploitable security vulnerabilities. They put users' sensitive information at risk. All critical issues should be fixed.

Issue Status

- **Fixed:** The issue has been resolved.
- **Partially Fixed:** The issue has been partially resolved.
- **Acknowledged:** The issue has been acknowledged by the code owner, and the code owner confirms it's as designed, and decides to keep it.

Appendix 2

Disclaimer

This report is based on the scope of materials and documents provided, with a limited review at the time provided. Results may not be complete and do not include all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your own risk. A report does not imply an endorsement of any particular project or team, nor does it guarantee its security. These reports should not be relied upon in any way by any third party, including for the purpose of making any decision to buy or sell products, services, or any other assets. TO THE FULLEST EXTENT PERMITTED BY LAW, WE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, IN CONNECTION WITH THIS REPORT, ITS CONTENT, RELATED SERVICES AND PRODUCTS, AND YOUR USE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NOT INFRINGEMENT.