Aftermath Finance Liquid Staking Derivative

Audit Report

MOVEBIT

contact@movebit.xyz

https://twitter.com/movebit_

Tue Nov 21 2023

Aftermath Finance Liquid Staking Derivative Audit Report

1 Executive Summary

1.1 Project Information

Description	Aftermath is building an all-in-one platform for trading, investing, and earning yield that is fast, inexpensive, and fully transparent.
Type	Staking
Auditors	MoveBit
Timeline	Mon Oct 30 2023 – Tue Nov 21 2023
Languages	Move
Platform	Sui
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/AftermathFinance/liquid-staking-derivative
Commits	7bef7c7180625c3eab337ae17529b10ffab3202c eeadc66bfa5611145baf0ba28932e7cfad1e413f 3799894081a06aa8286a71010a618312e1c4d9d0

1.2 Files in Scope

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

ID	File	SHA-1 Hash
MOV	packages/lsd/Move.toml	f3eaed29d6227ba5d07d50cd6caaff38079213e1
SOR	packages/lsd/sources/utils/sort.move	7bd56e313d459b83b2905591df698478a879088a
CAL	packages/lsd/sources/utils/calculations.move	30f4b447602af6eec7618103991cb1a3f024496d
SSV	packages/lsd/sources/staked_sui_vault.move	c30c1e04a9d022d9b1350c7bde49354c7975873e
ACT	packages/lsd/sources/internal/actions.move	368b6737b138f7d20dc173778e856b16ac668b7e
EVE	packages/lsd/sources/internal/events.move	2198004ce64cfac5b18f523a1a0fd9f827f502fe
VAL	packages/lsd/sources/internal/validator.move	f9cffae6b2085d03340b0b4c529895cd20a2978c
SSVS	packages/lsd/sources/internal/staked_sui_vault_state.move	9204d1fa0f143994a4f6a7d414445888445d4273
STO	packages/lsd/sources/internal/storage.move	45bb5d2e84e34ee2a2ff7837a329f95e25344c96
REC	packages/lsd/sources/internal/receipt.move	fc80768550c2cb26d024c9eef28a434bc9819159
PLSIRM	packages/lsd/sources/internal/record.move	29cb2d0177fa4a44cc8f5f5282865f6767c53849

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	4	3	1
Informational	1	1	0
Minor	2	2	0
Medium	0	0	0
Major	1	0	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"** 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) 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 Aftermath Finance to identify any potential issues and vulnerabilities in the source code of the Liquid Staking Derivative 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 4 issues of varying severity, listed below.

ID	Title	Severity	Status
REC–1	Visibility of `burn` And `claim_specified_amount` May Change To Private	Minor	Fixed
SSV–1	Centralization Risk	Major	Acknowledged
SSV–2	Duplicated Error Codes	Minor	Fixed
SSV1–1	Code Optimization	Informational	Fixed

3 Participant Process

Here are the relevant actors with their respective abilities within the Liquid Staking Derivative Smart Contract:

Admin

- Admin can upgrade the `StakedSuiVault` object with the `migrate` function
- Admin can update the address of the dev wallet with the `update_dev_account` function
- Admin can update the max incentive reward with the `update_max_crank_incentive_reward` function
- Admin can update reference gas price with the `update_reference_gas_price` function
- Admin can update the minimum `Coin<SUI>` required with `update_min_staking_threshold` function
- Admin can update the minimum field requests with the `update_min_fields_requests_per_tx` function
- Admin can update the number of epochs with the `update_pool_rates_epoch_gap` function
- Admin can update the number of validators with the `update_unstaking_bunch_size` function
- Admin can update the total protocol fee with the `update_default_unstake_total_fee` function
- Admin can update allocation of the total protocol fee with the `update_default_unstake_fee_allocations` function
- Admin can update the discount taken on the `Treasury` 's allocation with the `update_default_unstake_referee_discount` function
- Admin can update the max unstake fee with the `update_atomic_unstake_max_fee` function
- Admin can update the min unstake fee with the `update_atomic_unstake_min_fee` function
- Admin can update allocation of the total protocol fee with the `update_atomic_unstake_fee_allocations` function

- Admin can update the unstake referee discount with the `update_atomic_unstake_referee_discount` function

- Admin can update the target value of the amount of liquidity held with the `update_atomic_unstake_sui_reserves_target_value` function

Validator

- Validator can create a new `UnverifiedValidatorOperationCap` with the `rotate_operation_cap` function

- Validator can update validator fee with the `rotate_operation_cap` function

Owner

- Owner can give the underlying `StakedSuiVaultStateV1` object the authority to mint and burn `Coin<AFSUI>` with the `authorize` function

- Owner can remove the authority with the `renoke_auth` function

User

- Users can change the epoch with the `epoch_was_changed` function

- Users can stake with `request_stake, request_stake_and_keep, request_stake_vec, request_stake_vec_and_keep, request_stake_staked_sui, request_stake_staked_sui_and_keep, request_stake_staked_sui_vec, request_stake_staked_sui_vec_and_keep` functions

- Users can unstake with `request_unstake, request_unstake_vec, request_unstake_atomic, request_unstake_atomic_and_keep, request_unstake_vec_atomic, request_unstake_vec_atomic_and_keep` functions

- Users can claim SUI with `claim_from_atomic_unstake_sui_reserves, claim_from_atomic_unstake_sui_reserves_and_keep` functions

4 Findings

REC-1 Visibility of `burn` And `claim_specified_amount` May Change To Private

Severity: Minor

Status: Fixed

Code Location:

packages/lsd/sources/internal/receipt.move#68-73;

packages/lsd/sources/internal/receipt.move#101-118

Descriptions:

In `receipt.move` both `burn` and `claim_specified_amount` function are declared as public friend function, howevevr, neither `lsd::staked_sui_vault_state` nor `lsd::staked_sui_vault` call any of them directly.

Suggestion:

It is suggested to change the visibility of these two functions from public friend to private.

SSV-1 Centralization Risk

Severity: Major

Status: Acknowledged

Code Location:

packages/lsd/sources/staked_sui_vault.move

Descriptions:

The `Admin` has the following privileges:

- Admin can upgrade the `StakedSuiVault` object with the `migrate` function
- Admin can update the address of the dev wallet with the `update_dev_account` function
- Admin can update the max incentive reward with the `update_max_crank_incentive_reward` function
- Admin can update reference gas price with the `update_reference_gas_price` function
- Admin can update the minimum `Coin<SUI>` required with `update_min_staking_threshold` function
- Admin can update the minimum field requests with the `update_min_fields_requests_per_tx` function
- Admin can update the number of epochs with the `update_pool_rates_epoch_gap` function
- Admin can update the number of validators with the `update_unstaking_bunch_size` function
- Admin can update the total protocol fee with the `update_default_unstake_total_fee` function
- Admin can update allocation of the total protocol fee with the `update_default_unstake_fee_allocations` function
- Admin can update the discount taken on the `Treasury` 's allocation with the `update_default_unstake_referee_discount` function

- Admin can update the max unstake fee with the `update_atomic_unstake_max_fee` function
- Admin can update the min unstake fee with the `update_atomic_unstake_min_fee` function
- Admin can update allocation of the total protocol fee with the `update_atomic_unstake_fee_allocations` function
- Admin can update the unstake referee discount with the `update_atomic_unstake_referee_discount` function
- Admin can update the target value of the amount of liquidity held with the `update_atomic_unstake_sui_reserves_target_value` function

Suggestion:

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

Resolution:

It is acknowledged by the dev team that `AdminCap` can only perform very limited functions, and multisig, community goverence will be used in the future to further prevent the centralization.

SSV-2 Duplicated Error Codes

Severity: Minor

Status: Fixed

Code Location:

packages/lsd/sources/staked_sui_vault.move#40-46

Descriptions:

In the `staked_sui_vault.move`, both error codes `EVersionIncompatibility,` `EDeprecated` are set to 0.

Though they serve similar functionalities, the former one is used when "The admin calls `migrate` on an outdated package." and the latter is used when "One tries to call deprecated function."

Thus, This could potentially lead to confusion when trying to distinguish between these two types of errors based on their codes.

Suggestion:

It is suggested to assign unique values to each error code to avoid this issue.

Resolution:

is is fixed by the dev team.

SSV1-1 Code Optimization

Severity: Informational

Status: Fixed

Code Location:

packages/lsd/sources/internal/staked_sui_vault_state.move#295

Descriptions:

The `staked_sui_vault_state::create` function is used to create the `StakedSuiVaultStateV1` entity and initialize it to its default state. Restricting each of these to be less than `FIXED_ONE` is unnecessary because of the following `assert` statement.

```
assert!(
    default_unstake_treasury_allocation
        + default_unstake_dev_wallet_allocation
        + default_unstake_crank_incentive_allocation
            == FIXED_ONE,
    EInvalidPercentage
)
```

Similarly for `atomic_unstake_treasury_allocation` related variables. Also since `max_atomic_unstake_fee` is constrained to be less than `FIXED_ONE`, it is not necessary to restrict `min_atomic_unstake_fee` to be less than `FIXED_ONE` again.

Suggestion:

It is recommended to remove unnecessary code and increase the readability of the code.

Resolution:

The client followed the suggestion and fixed this 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.