Mosaic

Audit Report

contact@bitslab.xyz

https://twitter.com/movebit_

Mon Nov 18 2024

Mosaic Audit Report

1 Executive Summary

1.1 Project Information

Description	Mosaic is a DEX aggregator and DeFi hub.
Type	DeFi
Auditors	MoveBit
Timeline	Mon Oct 28 2024 - Mon Nov 18 2024
Languages	Move
Platform	Others
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/kitelabs-io/mosaic-sc https://github.com/kitelabs-io/mosaic-amm
Commits	https://github.com/kitelabs-io/mosaic-sc: 3eb8fb2db01badcd7f1d510e66f6b6b706dfb48b https://github.com/kitelabs-io/mosaic-amm: 16bfbfd5f404a8dcf68f861d83feb6229367e1ed 647d1c5a31242d507f25b07fa48eabdbbd82f109 90f69c234bafc63a03dc558eeb8530b438aad531.

1.2 Files in Scope

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

ID	File	SHA-1 Hash
MOV6	mosaic/Move.toml	02182d05e99bd2475a09692ad27553a5df7cb7f0
ROU2	mosaic/sources/router.move	13727f27fd4505533c3cec977ada35a9e77be1b2
2RA	mosaic/sources/executor/2_razorswap.move	ed61e4ff735561287ff88914bf451ee0b5f4e443
1AN	mosaic/sources/executor/1_animeswap.move	f532bf394b7d9c21c34ad13b1433496936ee1c6c
7MA	mosaic/sources/executor/7_mosaic_amm.move	c4d63a9f742f2f884a867d6f3e13690a1480d9fe
6ST	mosaic/sources/executor/6_stakedmove.move	08fa5b9bffb1cbee7d9485a7baedad0455008f5e
4MS	mosaic/sources/executor/4_meridian_stable.move	d4e8e9e2a590706cdfbb4126f2f6d87afe3c4b86
5MW	mosaic/sources/executor/5_meridian_weighted.move	a13595a8a3c2778992e128f211650d64d6bcad71
3LI	mosaic/sources/executor/3_liquidswap.move	6fbc4573036a82fffffbcd8884540c5237802c86
MOV	Move.toml	c782a7e1fba4f07b4dce0ff89d9c15557b23c876
MAT	sources/libs/math.move	2200e728f6d13a8a0306036bc58ee52765e1d79c

FHE	sources/libs/fa_helper.move	4efc9cc047963b038626dfdec895a82b5ea39b05
SCU	sources/libs/stable_curve.move	6ddb17fe15c09f9fb7f22d04be593170293e8431
MSP	sources/libs/math.spec.move	523ec30972e4a2d980c29487755e983dc370bf6b
F64	sources/libs/fp64.move	f88ef5dc9bb39959b22aa7a3ad8aa9a712416a8b
GCO	sources/swap/global_config.move	ce358578fb204df592c0f002ea91a9635dbbfd98
EME	sources/swap/emergency.move	f6108c7c986bc24e94d3e65fe6be716792662129
CCO	sources/swap/coin_converter.move	dcb4791a38a5272e1c66776f0124115a30daf6d3
TRE	sources/swap/treasury.move	afa8a88af450501c1aab665779787b64bbaadac8
ROU	sources/swap/router.move	468cf7f8077c6a9e43261c66616aa68033999c37
SCR	sources/swap/scripts.move	7e5f9e37883b3af6f4b27a3a1cad2c2edcf31e20
LPO	sources/swap/liquidity_pool.move	d1bfe44192acd11fb730ccedb278cf0d00b746db

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	6	5	1
Informational	2	2	0
Minor	2	1	1
Medium	2	2	0
Major	0	0	0
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(Optional)

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 Mosaic to identify any potential issues and vulnerabilities in the source code of the Mosaic 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 6 issues of varying severity, listed below.

ID	Title	Severity	Status
GCO-1	Lack of Events Emit	Minor	Acknowledged
LPO-1	Precision Loss in `split_fee_to_protocol`	Medium	Fixed
LPO-2	Unused Constants	Informational	Fixed
ROU-1	Unchecked Parameters	Medium	Fixed
ROU-2	Test Code in the Contract	Informational	Fixed
ROU1-1	Unused Constants	Minor	Fixed

3 Participant Process

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

- Admin can pause all operations through `pause()` .
- Admin can resume all operations through `resume()` .
- Admin can disable condition forever through `disable_forever()` .
- Admin can set the protocol admin account through `set_protocol_admin()` .
- Admin can set emergency admin account through `set_emergency_admin()` .
- Admin can set the fee admin account through `set_fee_admin()` .
- Admin can set a new default fee through `set_default_fee()` .
- Admin can set the default protocol fee through `set_default_protocol_fee()` .
- Admin can set fee for the specific pool through `set_fee()` .
- Admin can set protocol fee for the specific pool through `set_protocol_fee()` .
- Admin can withdraw tokens from treasury to the `protocol_admin` account through `withdraw_both_assets()` \ `withdraw_both_coins<X, Y>()` \ `withdraw_one_coin<CoinType>()` .

User

- Users can create a new liquidity pool for `X` / `Y` pair through `create_pool_both_coins<X, Y>()` \ `create_pool_one_coin<CoinX>()` \ `create_pool_both_assets()` .
- Users can add new liquidity into the pool `X` / `Y` and get liquidity token `LP` through `add_liquidity_both_coins<X, Y>()` \ `add_liquidity_one_coin<X>()` \ `add_liquidity_both_assets()` .
- Users can remove (burn) liquidity tokens `LP` from account, get `X` and `Y` coins back through `remove_liquidity_both_coins<X, Y>()` \ `remove_liquidity_one_coin<X>()` \ `remove_liquidity_both_assets()` .
- Users can swap exact amount of coin `X` for coin `Y` through `swap_exact_coin_for_coin<X, Y>()` \ `swap_coin_for_exact_coin<X, Y>()` \ `swap_exact_coin_for_asset<CoinIn>()` \ `swap_coin_for_exact_asset<CoinIn>()` \

`swap_exact_asset_for_coin<CoinOut>()` \ `swap_asset_for_exact_coin<CoinOut>()` \ `swap_exact_asset_for_asset()` \ `swap_asset_for_exact_asset()`.

4 Findings

GCO-1 Lack of Events Emit

Severity: Minor

Status: Acknowledged

Code Location:

sources/swap/global_config.move

Descriptions:

The contract lacks appropriate events for `set_emergency_admin()` and `set_fee_admin()` .These are the key functions.The lack of event records for the above functions may cause inconvenience in the subsequent tracking of NFT issuance and contract status changes.

Suggestion:

It is recommended to emit events for the functions.

LPO-1 Precision Loss in `split_fee_to_protocol`

Severity: Medium

Status: Fixed

Code Location:

sources/swap/liquidity_pool.move#656

Descriptions:

In the `split_fee_to_protocol` function, when calculating protocol fees, the variable `protocol_fee_multiplier` (adjusted by the precision `PROTOCOL_FEE_SCALE`) is multiplied by the token amount deposited into the pool and then divided by the precision `FEE_SCALE`. This calculation method, which involves multiplication followed by division, may lead to precision loss.

Suggestion:

It is recommended to modify the calculation to adjust the sequence or increase intermediate precision.

Resolution:

The customer took our advice and fixed the issue.Fee accuracy has been improved.

LPO-2 Unused Constants

Severity: Informational

Status: Fixed

Code Location:

sources/swap/liquidity_pool.move#57;

sources/swap/router.move#36,40

Descriptions:

There are unused constants in the contract.

```
const ERR_POOL_DOES_NOT_EXIST: u64 = 108;

const ERR_UNREACHABLE: u64 = 207;

const ERR_WRONG_COIN_ORDER: u64 = 209;
```

Suggestion:

It is recommended to remove unused constants if there's no further design.

ROU-1 Unchecked Parameters

Severity: Medium

Status: Fixed

Code Location:

sources/swap/router.move#100-250

Descriptions:

In the `router` contract, both the `addLiquidity` and `removeLiquidity` functions include an address parameter for the user, which is ultimately included in the event emission. However, there is no validation for this parameter, meaning users can pass any address when using these functions, potentially causing misleading events within the protocol.

Suggestion:

It is recommended to replace the address parameter with the signer type to ensure the authenticity of the user address. Within the function, use `signer::address_of()` to obtain the caller's actual address, preventing users from passing arbitrary addresses in events and causing confusion.

ROU-2 Test Code in the Contract

Severity: Informational

Status: Fixed

Code Location:

sources/swap/router.move#678

Descriptions:

There is a test code in the `calc_optimal_coin_values` function;

```
public fun calc_optimal_coin_values(
    pool: Object<LiquidityPool>,
    token_x: Object<Metadata>,
    token_y: Object<Metadata>,
    x_desired: u64,
    y_desired: u64,
    x_min: u64,
    y_min: u64
  ): (u64, u64) {
            ...
            assert!(x_returned <= x_desired, ERR_OVERLIMIT_X);
            std::debug::print(&x_returned);
            assert!(x_returned >= x_min, ERR_INSUFFICIENT_X_AMOUNT);
            return (x_returned, y_desired)
        }
    }
  }
```

Suggestion:

It is recommended to remove the test code `debug::print` in the `calc_optimal_coin_values` function.

ROU1-1 Unused Constants

Severity: Minor

Status: Fixed

Code Location:

mosaic/sources/router.move#80

Descriptions:

The constant `E_NOT_IMPLEMENTED` is never used.

Suggestion:

If this constant will not be used in subsequent designs, consider deleting it.

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.