Earnium

Audit Report

contact@bitslab.xyz

https://twitter.com/movebit_

Fri Jul 25 2025

Earnium Audit Report

1 Executive Summary

1.1 Project Information

Description	Earnium is an AMM DEX built on the Aptos blockchain. User can trade, add/remove LP on Aptos using Earnium.
Type	DeFi
Auditors	MoveBit
Timeline	Thu Jul 10 2025 - Mon Jul 14 2025
Languages	Move
Platform	Aptos
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/earnium-aptos/core
Commits	52fbef97b8ae0f3b36d5e0a65f117ffe17e2425b eae795498856bf92b8157b19b777817fe17c0ffa 136e3486aec24e083a09e3c92a3e434a9b467407 3d0aab23ab663e68d8af0a90da7415340899a455

1.2 Files in Scope

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

ID	File	SHA-1 Hash
MOV	src/swap_v2/Move.toml	4f137228ef71f2ad4513e57df4cb829bce8cf2e9
ROU	src/swap_v2/sources/router.move	67c548772ff7ef42d6a3a0e9bbc1214f54fc1f3c
LPO	src/swap_v2/sources/liquidity_pool.move	2a367d7bcad939f463d7699af28ecbaeb6f369a9
GST	src/swap_v2/sources/global_state.move	cbcbef0052208679312ab9323f0232500628c948
LSP	src/swap_v2/sources/liquidity_stake_pool.move	f720eeca6d9e91033b3822e4f53e1e65201a8a69
FMA	src/swap_v2/sources/fees_manager.move	c070578faa7d1330c86cd1611ea3cfb82ea9e306

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	7	7	0
Informational	3	3	0
Minor	3	3	0
Medium	1	1	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 Earnium to identify any potential issues and vulnerabilities in the source code of the Earnium 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 7 issues of varying severity, listed below.

ID	Title	Severity	Status
FMA-1	Numerical Operations With Different Precision	Medium	Fixed
FMA-2	Lack of MAXIUM_LOCK_TIME Validation	Minor	Fixed
FMA-3	Unused Data	Informational	Fixed
FMA-4	Lack of Event Emit	Informational	Fixed
LPO-1	Missing Check for Balance	Minor	Fixed
LSP-1	Incorrect Event Parameter	Informational	Fixed
ROU-1	Lack of Array Length Check	Minor	Fixed

3 Participant Process

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

Admin

- Admin can add incentives with custom finish time through the `only_operator_add_incentives()` function.
- Admin can set the operator address through the `set_operator()` function.
- Admin can set a new pending admin address through the `set_admin()` function.
- Admin can set default fees configuration through the `set_default_config_fees()` function.
- Admin can set fees for a specific pool through the `set_pool_fees()` function.
- Admin can set stable swap fee in basis points through the `set_stable_fee()` function.
- Admin can set volatile swap fee in basis points through the `set_volatile_fee()` function.
- Admin can set swap fee and schedule duration for a pool through the `set_pool_swap_fee()` function.
- Admin can set max fee schedule duration through the `set_max_fee_schedule_duration()` function.
- Admin can set whitelist status for reward tokens in a pool through the `set_whitelist_rw_token()` function.
- Admin can pause/unpause reward distribution for a token in a pool through the `pause_reward_token()` function.

Pending Admin

- Pending admin can accept the admin role through the `accept_admin()` function.

Pauser

- Pauser can set a new pauser address through the `set_pauser()` function.
- Pauser can pause/unpause the contract through the `set_pause()` function.

Pending Pauser

- Pending Pauser can accept the pauser role through the `accept_pauser()` function.

User

- User can create a liquidity pool through the `create_pool()` function.
- User can create a liquidity pool with fee schedule through the `create_pool_with_fee_schedule()` function.
- User can create a liquidity pool with a coin type through the `create_pool_coin()` function.
- User can create a liquidity pool with both coin types through the `create_pool_both_coins()` function.
- User can swap tokens through the `swap_entry()` function.
- User can swap tokens through a route through the `swap_route_entry()` function.
- User can swap coins for assets through the `swap_route_entry_from_coin()` function.
- User can swap coins for assets directly through the `swap_coin_for_asset_entry()` function.
- User can add liquidity through the `add_liquidity_entry()` function.
- User can add liquidity and stake through the `add_liquidity_and_stake_entry()` function.
- User can add liquidity with a coin type through the `add_liquidity_coin_entry()` function.
- User can add liquidity and stake with a coin type through the `add_liquidity_and_stake_coin_entry()` function.
- User can add liquidity with both coin types through the `add_liquidity_both_coins_entry()` function.
- User can add liquidity and stake with both coin types through the `add_liquidity_and_stake_both_coins_entry()` function.
- User can remove liquidity through the `remove_liquidity_entry()` function.

- User can add incentives through the `add_incentives()` function.
- User can stake LP tokens with unlock time through the `stake()` function.
- User can unstake LP tokens after unlock time through the `unstake()` function.
- User can claim accumulated rewards through the `claim_rewards()` function.
- User can transfer LP tokens through the `transfer()` function.
- User can stake LP tokens through the `stake()` function.
- User can stake and lock LP tokens for a specific duration through the `stake_and_lock()` function.
- User can unstake LP tokens through the `unstake()` function.
- User can claim rewards from multiple pools and NFTs through the `claim_all_rewards()` function.
- User can create a locked position by burning LP tokens through the `create_lock()` function.
- User can claim rewards from a specific liquidity pool through the `claim_rewards()` function.
- User can claim rewards from a specific liquidity position NFT through the `nft_claim_rewards()` function.

4 Findings

FMA-1 Numerical Operations With Different Precision

Severity: Medium

Status: Fixed

Code Location:

src/swap_v2/sources/fees_manager.move#482

Descriptions:

In the add_reward function, the remain_reward loses precision after being divided by `ACCUM_REWARD_SCALE`, and then is calculated with generate_reward which retains precision, resulting in incorrect rewards.

```
let amount = coin_wrapper::exact_deposit(reward_info.reward_store, reward_asset) as u128;
        let generate_reward = (amount * ACCUM_REWARD_SCALE) as u256;
        let acc_token_per_share = &mut reward_info.acc_token_per_share;
        let remain_reward = (generate_reward % total_supply) / (ACCUM_REWARD_SCALE as
u256);
        if (remain_reward > 0) {
            let rm = dispatchable_fungible_asset::withdraw(
                &object::generate_signer_for_extending(&pool.extend_ref),
                reward_info.reward_store,
                remain_reward as u64
            );
            primary_fungible_store::deposit(treasury_address(), rm);
            generate_reward -= remain_reward;
```

Suggestion:

It is recommended to add precision handling during reward calculation.:

```
generate_reward -= remain_reward * ACCUM_REWARD_SCALE
```

Resolution:

This issue has been fixed. The client has adopted our suggestions.

FMA-2 Lack of MAXIUM_LOCK_TIME Validation

Severity: Minor

Status: Fixed

Code Location:

src/swap_v2/sources/fees_manager.move#326-367

Descriptions:

In the `stake()` function, there is no check for lock time longer than `MAXIUM_LOCK_TIME`.

Suggestion:

It is recommended to add a validation of `MAXIUM_LOCK_TIME`.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

FMA-3 Unused Data

Severity: Informational

Status: Fixed

Code Location:

src/swap_v2/sources/fees_manager.move#33;

src/swap_v2/sources/liquidity_pool.move#183;

src/swap_v2/sources/liquidity_stake_pool.move#49

Descriptions:

There are unused constants and event in the contract.

Suggestion:

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

Resolution:

This issue has been fixed. The client has adopted our suggestions.

FMA-4 Lack of Event Emit

Severity: Informational

Status: Fixed

Code Location:

src/swap_v2/sources/fees_manager.move#164,174,180;

src/swap_v2/sources/liquidity_pool.move#827-836,841,849,861;

src/swap_v2/sources/global_state.move#46,52-65

Descriptions:

Several functions in the contract lack event logging, which is essential for blockchain transparency, off-chain data tracking, and frontend integration. Event logs allow external systems to monitor contract activities without querying the blockchain state directly.

- `set_default_config_fees()`
- `set_pool_fees()`
- `set_stable_fee()`
- `set_volatile_fee()`
- `set_operator()`
- `set_admin()`
- `accept_admin()`
- `set_pauser()`
- `accept_pauser()`
- `set_pause()`
- `set_pool_swap_fee()`
- `set_max_fee_schedule_duration()`

Suggestion:

It is recommended to add event emission for these operations.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

LPO-1 Missing Check for Balance

Severity: Minor

Status: Fixed

Code Location:

src/swap_v2/sources/liquidity_pool.move#760;

src/swap_v2/sources/liquidity_stake_pool.move#565

Descriptions:

The `burn()` function does not check whether the user balance is greater than the input `amount`.

The `update_pool_per_token()` function does not check whether the pool balance is greater than the generated reward.

If the balance is insufficient, subsequent operations may fail and waste gas resources.

Suggestion:

It is recommended to check whether the balance is sufficient before operation, and return immediately if it is insufficient to reduce resource consumption.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

LSP-1 Incorrect Event Parameter

Severity: Informational

Status: Fixed

Code Location:

src/swap_v2/sources/liquidity_stake_pool.move#387

Descriptions:

The `user` field in the event `LiquidityPositionBurned` is set to the configuration address `signer::address_of(&global_state::config_signer())` instead of the actual recipient `receiver`. The event log cannot correctly reflect the real user, which will cause errors in on-chain analysis.

Suggestion:

It is recommended to change the `user` field in the event `LiquidityPositionBurned` to the receiver `reciver`.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

ROU-1 Lack of Array Length Check

Severity: Minor

Status: Fixed

Code Location:

src/swap_v2/sources/router.move#137,151,164

Descriptions:

When iterating over the two array type parameters of the `swap_router` function, the lengths of the two arrays are not checked. If the lengths of the two arrays are different, an error will occur.

Suggestion:

It is recommended to add logic to determine whether the lengths of two arrays are equal.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

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.