MoveGPT

Audit Report

contact@bitslab.xyz

https://twitter.com/movebit_

Thu Apr 11 2024

MoveGPT Audit Report

1 Executive Summary

1.1 Project Information

Description	The First Al Launchpad built on the Aptos Chain, powering the Move Web3 Economy
Type	Launchpad
Auditors	MoveBit
Timeline	Sat Apr 06 2024 - Thu Apr 11 2024
Languages	Move
Platform	Aptos
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/ken-movegpt/movegpt-contract
Commits	4340dcd18e811aa8f152f89f037054cca902f1ef d9119649c70ff6d9bcc5c2f2495ef60022ee10cc

1.2 Files in Scope

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

ID	File	SHA-1 Hash
VES	sources/vesting.move	2078b7b5bc9c5777ca668b3b2958eaf4ee078414
BUY	sources/buy.move	199a2e0239530e46605a9b08636a8dcb7803e5e6
MTO	sources/movegpt_token.move	a9431d0c5fb38bf849381fe16a4f0cac340d8185
TCS	sources/tests/test_claim_sale.move	d4a0c9bcfbc6b878647b2b737375d34982201524
TVE	sources/tests/test_voting_escrow.move	a5383a3fa045f44a8c4be9d41aa9992477c47e2f
TBU	sources/tests/test_buy.move	f343c661ae8e3fc7130a99aeadc45121f5a4dd1c
TVE1	sources/tests/test_vesting.move	1dcebb4ca73a83315183c81985a076f04abbcf0f
THE	sources/tests/test_helper.move	9e49eb535567d94f98f85b84b544b5a8f3ea14b8
TTO	sources/tests/test_token.move	1a83791007d9a8a63c0e2e4ce90f866059ea5804
TAI	sources/tests/test_airdrop.move	2c933b0515d1c3edccfcaa406af158001e46a2e2
EPO	sources/epoch.move	dadf8510c62d6a089520746c097b2d04fc31ebc9

AIR	sources/airdrop.move	07f461837fb7f64e94379570957b7d9c257bc52c
VES1	sources/voting_escrow.move	8e8331016bddb9526eaf529b216bc19c338af90c
CSA	sources/claim_sale.move	80f8c4ee29d915879ee387d881231a93cdb2432a
PMA	sources/package_manager.move	01dc1a80aa995cc843a565fcdb45344c7d228b0a

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	10	10	0
Informational	0	0	0
Minor	3	3	0
Medium	6	6	0
Major	1	1	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

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 MoveGPT to identify any potential issues and vulnerabilities in the source code of the MoveGPT 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 10 issues of varying severity, listed below.

ID	Title	Severity	Status
BUY-1	`nonce` Always Be 0	Medium	Fixed
BUY-2	`Initialize` Function Lacks Privilege Control	Medium	Fixed
CSA-1	`refund_entry` Function Can Be Called Multiple Times	Major	Fixed
CSA-2	Some View Function Logic Errors	Medium	Fixed
CSA-3	`claim` Function Can Be Called Multiple Times	Medium	Fixed
CSA-4	`claim` Function May Cause DOS Problems	Medium	Fixed
CSA-5	Unnecessary Boolean Comparison	Minor	Fixed
CSA-6	Lack of Events Emit	Minor	Fixed
VES-1	Logic Error in Claim Function	Medium	Fixed
VES-2	Code Optimization	Minor	Fixed

3 Participant Process

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

Admin

- Admin can set vesting-related settings via the `set*` function.
- Admin can get lock coin by using the `claim_entry` function.
- Admin can add private or ido claimer by using the `add_private_claimers` or `add_ido_claimers` function.

User

- Users can get airdrop nft by using the `airdrop_entry` function.
- Users can refund their allocation by using the `refund_ido_entry` function.
- Users can claim their ido or private by using the `claim_private_entry` or `claim_ido_entry` .

Governance

- Governace can update the operator through `update_operator()` .
- Governace can update the governace through `update_governance()` .
- Governace can upgrade the module through `upgrade()` .

4 Findings

BUY-1 `nonce` Always Be 0

Severity: Medium

Status: Fixed

Code Location:

sources/buy.move#135

Descriptions:

The `nonce` string value added to the signature in the `buy` function is always 0 and there is no place to change it.

Suggestion:

It is recommended not to use hard-coded.

Resolution:

This issue has been fixed.

BUY-2 `Initialize` Function Lacks Privilege Control

Severity: Medium

Status: Fixed

Code Location:

sources/buy.move#53

Descriptions:

The `initialize` function can be called by any user and passed any parameter.

Suggestion:

It is recommended to add privilege control to the `initialize` function.

Resolution:

This issue has been fixed. The client added privilege control to the function.

CSA-1 `refund_entry` Function Can Be Called Multiple Times

Severity: Major

Status: Fixed

Code Location:

sources/claim_sale.move#247 346

Descriptions:

The `refund_entry` function did not update the user's status after the user was refunded resulting in the user being able to call `refund_entry` multiple times and reduce the value of `total_bought` at will. Also the `withdraw_round` function operator can be called multiple times.

Suggestion:

It is recommended to update the user's status after the call.

Resolution:

This issue has been fixed. The client modified the logic of the `refund_entry` function.

CSA-2 Some View Function Logic Errors

Severity: Medium

Status: Fixed

Code Location:

sources/claim_sale.move#101-106

Descriptions:

The view function to get information about `private_round` is still retrieved from the `ido_round` field.

Suggestion:

It is recommended that this be modified to the correct logic based on the design.

Resolution:

This issue has been fixed.

CSA-3 `claim` Function Can Be Called Multiple Times

Severity: Medium

Status: Fixed

Code Location:

sources/claim_sale.move#298

Descriptions:

`claim` related functions can be called multiple times by the user, it is recommended to confirm whether this design conforms to the design concept.

Suggestion:

It is recommended to determine the state of the user before calling it.

Resolution:

This issue has been fixed.

CSA-4 `claim` Function May Cause DOS Problems

Severity: Medium

Status: Fixed

Code Location:

sources/claim_sale.move#294

Descriptions:

In the `claim` function, when `lock_amount` is equal to round_config.balances, it will extract all the coins in `round_config.balances`, but by calculating the `lock_amount` may be less than round_config.balances, then the function will always fail when reaching the `else` branch to extract the `lock_amount` from balances.

Suggestion:

It suggests modifying the conditions to avoid dos problems.

Resolution:

This issue has been fixed. The client modified the logic for the `lock_amount` judgment.

CSA-5 Unnecessary Boolean Comparison

Severity: Minor

Status: Fixed

Code Location:

sources/claim_sale.move#303;

sources/buy.move#125

Descriptions:

There are statements in the contract that use Boolean variables to compare with Boolean values, such as `order_is_exist(order_id,buy_orders) == false`, and it is recommended to just use that field's value directly.

Suggestion:

It is recommended to fix them.

Resolution:

This issue has been fixed.

CSA-6 Lack of Events Emit

Severity: Minor

Status: Fixed

Code Location:

sources/claim_sale.move#155-230

Descriptions:

The smart contract lacks appropriate events for monitoring sensitive operations, which could make it difficult to track sensitive actions or detect potential issues.

Suggestion:

It is recommended to emit events for those important functions.

Resolution:

This issue has been fixed.

VES-1 Logic Error in Claim Function

Severity: Medium

Status: Fixed

Code Location:

sources/vesting.move#313

Descriptions:

The assertion function `current_time > vesting_config.start` causes the claim function to never reach the `if` branch of the `vesting_config.start > current_time` condition.

Suggestion:

It is recommended to ensure that the function conforms to the design.

Resolution:

This issue has been fixed. The client modified the logic of the `claim` function.

VES-2 Code Optimization

Severity: Minor

Status: Fixed

Code Location:

sources/vesting.move#142-229

Descriptions:

There is a lot of duplicate code in the `set*` function associated with setting global variables, such as `vesting_config.start = new_start_time` or `vesting_config.vesting_duration = new_duration_time`, and the same code can be extracted to make the code more readable.

Suggestion:

It is recommended to fix them.

Resolution:

This issue has been fixed.

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.