vibrantX

Audit Report

MOVEBIT

contact@movebit.xyz

https://twitter.com/movebit_

Mon Jan 08 2024

vibrantX Audit Report

1 Executive Summary

1.1 Project Information

Description	Aggregator on Aptos
Type	DeFi
Auditors	MoveBit
Timeline	Tue Jan 02 2024 - Mon Jan 08 2024
Languages	Move
Platform	Aptos
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/vibrantX-finance/contracts
Commits	bf3f4138563f594b0d6ebda6d5535fb687d66b04 825eba54796482642d6dd18078be9a77f2b6ee33

1.2 Files in Scope

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

ID	File	SHA-1 Hash
MOV5	move-examples/vibrantx/Move.toml	748b1d3eb421fa8a5cb6bba721d10d6147096e8a
HEL	move-examples/vibrantx/sources/helpers.move	df7b1644df95dae1097c6817a101f0ff7d43cf98
TLS	move-examples/vibrantx/sources/liquid_staking/thala_lsd.move	f4cc0cbe67fb534b6e5d55c64c76a5075e4acfbc
AMN	move-examples/vibrantx/sources/liquid_staking/amnis.move	b790a45dbfd3da968e37e5292061dc5af37ba7d4
LEV	move-examples/vibrantx/sources/liquid_staking/lsd_events.move	f42209b14561c3a6cfe34de3a92382864e7cb126
VPM	move-examples/vibrantx/sources/vibrantx_package_manager.move	153bc1390a0f7422bd6dde28ffda119a57a821cf
ARI	move-examples/vibrantx/sources/lending/aries.move	e15a045c3395838b4e37ac82482a4ee2c29465bd
LEV1	move-examples/vibrantx/sources/lending/lending_events.move	a77718f73187d8334ee396aa86e2044de6b923f0
ADE	move-examples/vibrantx/sources/multi/amnis_dex.move	c76804c44e7e4e14963c9bacafaa03607eb9c30b
PAN	move-examples/vibrantx/sources/dexes/pancakeswap.move	3218202508576bd5d8765be5771388e57c8027b0
DEV	move-examples/vibrantx/sources/dexes/dex_events.move	d7f547fe2d851eb759efeaefc94faabba77f262d

LIQ	move-examples/vibrantx/sources/dexes/liquidswap.move	80e8458122b6784897321d7829178c2d469af52c
THA	move-examples/vibrantx/sources/dexes/thalaswap.move	85eed8dcc16dbfe4b3bf2b58c884751ee00b7dc7
SUS1	move-examples/vibrantx/sources/dexes/sushiswap.move	5a41c1e78774b4c0917c4ced278ee8b0e895d46f

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	6	6	0
Informational	1	1	0
Minor	1	1	0
Medium	1	1	0
Major	3	3	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 vibrantX to identify any potential issues and vulnerabilities in the source code of the vibrantX 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
LIQ-1	Function Does't Return	Major	Fixed
LIQ-2	Fixed Slippage	Medium	Fixed
THA-1	Wrong Type Parameter	Major	Fixed
THA-2	Unused Constant	Informational	Fixed
VPM-1	Unused `friend` Functions	Major	Fixed
VPM-2	Lack of Events Emit	Minor	Fixed

3 Participant Process

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

Admin

- The admin can publish new modules or upgrade existing modules in this package by calling the `upgrade()` function.

User

- Users can invoke the `vibrantx` module to interact with dexes such as `liquidswap`, `pancake`, `sushiswap`, `thalaswap`, etc. for `claim_rewards`, `remove_liquidity`, `add_liquidity`, `lend`, `stake`, etc.

4 Findings

LIQ-1 Function Does't Return

Severity: Major

Status: Fixed

Code Location:

move-examples/vibrantx/sources/dexes/liquidswap.move#142-218

Descriptions:

When the token is not sorted before, the function will be recalled, and the execution of the previous function is not terminated, which will cause the code to be executed twice.

Suggestion:

It is recommended that a return statement be added for functions that do not have a return value.

Resolution:

The client followed the suggestion and fixed this issue.

LIQ-2 Fixed Slippage

Severity: Medium

Status: Fixed

Code Location:

move-examples/vibrantx/sources/dexes/liquidswap.move#142;

move-examples/vibrantx/sources/dexes/pancakeswap.move#54;

move-examples/vibrantx/sources/dexes/sushiswap.move#54;

move-examples/vibrantx/sources/dexes/thalaswap.move#91

Descriptions:

Slippage protects users from losing tokens in some paris, but the fixed slippage settings can also lead to failed trades with high price volatility.

Suggestion:

It is recommended to use an additional parameter so that the user can set the value of the slippage.

Resolution:

The client followed the suggestion and fixed this issue.

THA-1 Wrong Type Parameter

Severity: Major

Status: Fixed

Code Location:

move-examples/vibrantx/sources/dexes/thalaswap.move#24-31

Descriptions:

The type parameter `Token4` received by the `get_weighted_reserves` function is not passed to the `weighted_pool::pool_balances_and_weights` function, and there is a duplicate of the type parameter Token, so make sure this is by design.

Suggestion:

It is recommended to ensure that this is as designed.

Resolution:

The client followed the suggestion and fixed this issue.

THA-2 Unused Constant

Severity: Informational

Status: Fixed

Code Location:

move-examples/vibrantx/sources/dexes/thalaswap.move#20

Descriptions:

There are unused constants in the entire module.

```
const DEFAULT_REMOVE_LIQUIDITY_SLIPPAGE: u64 = 100;
```

Suggestion:

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

Resolution:

The client followed the suggestion and fixed this issue.

VPM-1 Unused `friend` Functions

Severity: Major

Status: Fixed

Code Location:

move-examples/vibrantx/sources/vibrantx_package_manager.move#36

Descriptions:

The `add_address` function is not used in this module and the `vibrantx_package_manager` module does not set the `friend` module so the `add_address` function can't be called by anyone, thus causing the module function to be disabled.

Suggestion:

It is recommended to add a function interface so that the function can be used correctly.

Resolution:

The client followed the suggestion and fixed this issue.

VPM-2 Lack of Events Emit

Severity: Minor

Status: Fixed

Code Location:

move-examples/vibrantx/sources/vibrantx_package_manager.move#36,49

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 sensitive functions.

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.