Superposition

Audit Report

MOVEBIT

contact@movebit.xyz

https://twitter.com/movebit_

Tue Mar 26 2024

Superposition Audit Report

1 Executive Summary

1.1 Project Information

Description	Superposition is a lending platform where users can deposit collaterals and lend assets with a dynamic collateralization rate.
Type	Lending
Auditors	MoveBit
Timeline	Mon Mar 04 2024 - Tue Mar 26 2024
Languages	Move
Platform	Aptos
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/concordia-fi/system/
Commits	77a2587f6d7fb7952aa6604d15b7a5bf190af966 5dfd5c47e1ff7b25d5bb7aac7d06c99fb8718423 285d00c3a5081fb19cb0581145affe6e5189b172 b9b2464b351014ec8a2da6c70063514ad342b61c

1.2 Files in Scope

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

ID	File	SHA-1 Hash
REP	move/superp/sources/repay.move	6116f01f2fa4f1eb32d783300555c6f92bc36aa4
EAD	move/superp/sources/entry_admin.move	8c436876ff8e81b63f74efd2ca31af8b261b31d4
EPU	move/superp/sources/entry_public.move	eaca8d64ab86b793d00be65d4ca45c24349aaadb
BRO	move/superp/sources/broker.move	e2e2cde3a64a2c58878a1013be011c3dc0153971
BSP1	move/superp/sources/borrow.spec.move	4012cd11a9c480102010517a39b10dcb57bab5c7
EDE	move/superp/sources/entry_deprecated.move	09c7ba6e65dc2f1ad40701dff9d4aa58d752e6fc
POR	move/superp/sources/portfolio.move	bb3c5e0511f497e14f9a5fb8e0e11d2136203c72
CON	move/superp/sources/config.move	be299d6fe995714a5424261a0b281960aa6ef10b
RED	move/superp/sources/redeem.move	c2ae605c86edf4721478d045bac10107600113ce
BOR	move/superp/sources/borrow.move	b9caac4b560d5157b4b2720998f29f5216a0ebc8
TEN	move/superp/sources/tenant.move	5972a41149c2e8c170a0f36648601f93f3ce26e2

LIQ	move/superp/sources/liquidate.move	cadf6359e587d70d44b3365a500d28f2a1dc6205
TSP3	move/superp/sources/tenant.spec.move	df75b89f630c9d5a118fc4efa0b2ab14fe48c726
INT	move/superp/sources/interest.move	35664adce99bad758a4c37d7fa96c4cc01aaa149
LSP	move/superp/sources/lend.spec.move	b21975d84f8dd749df07e57c00012dada3a8933b
LEN	move/superp/sources/lend.move	767862f9c6e6a15ee94e6dbe0690062a17deb922
BSP2	move/superp/sources/broker.spec.move	5425ec2e0f93cb7ecc422f96aa4f75f475969207
AAP	move/hocket/sources/admin_api.move	efa53ee7a88e6cabb51b7157c8d66558fe57151b
HOC	move/hocket/sources/hocket.move	33ff7f3ea89a27d384cd4c7dad017c731e8e79ec
ACO1	move/hocket/sources/admin_config.move	e81e87781424f4847391159f15a59f07c477292b
PST	move/hocket/sources/packet_state.move	fd084bf114948ebb35f9be27418520f4c276d196
SST2	move/hocket/sources/signer_state.move	4ea02380a617f72162f82493e86d51d503a96bb7
SSP2	move/concbox/sources/set.spec.move	44a3a45fedbff104afac06964cb7ca1caa8cbcaf
LIT1	move/concbox/sources/lit.move	57ba02d13acbb704bab25e2ae3586e58234fe3fa

LSP1	move/concbox/sources/lit.spec.move	aacd9083cbc097d48c8efbe24294c9e052c89f86
MAT	move/concbox/sources/math.move	914a6a86a8e12fd447aadba708cb13d6de762bc5
MSP	move/concbox/sources/math.spec.move	2bec0fc804345ccb9ec166fbbeae5225ebaf8ee3
RA	move/concbox/sources/ra.move	d376ecccc61d4c75763e964f3136ab6f0a969690
SET	move/concbox/sources/set.move	fb8975a9d5e427911b06b907bdbf8c1a2034c1a4
VAU	move/concbox/sources/vault.move	1a25195a394879990ebb166f3d5e7b0fcb539b63
VSP3	move/concbox/sources/vault.spec.move	432982f332d65bc70f28a21c9a5c8ec05d187aaa
BCU	move/concbox/sources/binary_cursor.move	37abe847a6542bdecbb1b9d5120b54915f8915c6

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	10	10	0
Informational	0	0	0
Minor	5	5	0
Medium	2	2	0
Major	1	1	0
Critical	2	2	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 Concordia to identify any potential issues and vulnerabilities in the source code of the Superpostion 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
BCU-1	Deserialization Should Explicitly Check `Data` Length	Minor	Fixed
BRO-1	`update` Function Will Reset The Broker	Critical	Fixed
BRO-2	`set_interest_updated` Will Set Arbitrary Timestamp	Critical	Fixed
BRO-3	Set Functions Lack of Access Control	Major	Fixed
BRO-4	`view_broker` Should Not Return False Bool Directly	Medium	Fixed
BRO-5	Rounding Errors Handling Is Not Best Practice	Medium	Fixed
HOC-1	`verify_platform` Uses A Magic Number	Minor	Fixed
POR-1	`simple_map::create` Is Deprecated	Minor	Fixed
RED-1	Vault Amount Checks Are Not Implemented In Every Operation	Minor	Fixed

TEN-1	Fee Rates Should Be Hard Capped Under 100%	Minor	Fixed

3 Participant Process

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

Admin

Administrative Functions

- `admin` can **propose a new admin** through `propose_new_admin`.
- `proposed admin` can **accept the admin role** through `accept_admin_role`.

Broker Settings

- `admin` can **set a broker's interest rate** for a specific token type through `set_broker_interest_rate_v2`.
- `admin` can **set broker's maximum borrowing amount** for a specific token type through `set_broker_max_borrow`.
- `admin` can **set broker's maximum deposit amount** for a specific token type through `set_broker_max_deposit`.
- `admin` can **pause or resume broker activities** (such as trading, lending, borrowing, redeeming, repaying) for a specific token type through functions like `set_broker_pause`.

Tenant Settings

- `admin` can **pause or resume tenant operations** through `set_tenant_pause`.
- `admin` can **set various fees and rates for tenants**, such as:
  - `set_tenant_liquidation_fee_address`
  - `set_tenant_liquidation_fee_rate`
  - `set_tenant_interest_fee_address`
  - `set_tenant_interest_rate`
  - `set_tenant_stability_fee_address`
  - `set_tenant_stability_fee_rate`

Liquidation

- `admin` can **apply an approved liquidation** through `liquidate`.

User

Lending Functions

- `user` can **lend liquidity and collateralize the deposit notes** through `lend_v2` (`lend` is deprecated).

Borrowing Functions

- `user` can **borrow liquidity and record loan notes** through `borrow_v2` (`borrow` is deprecated).

Redeeming Functions

- `user` can **burn deposit notes and redeem liquidity** (amount to redeem is in deposit note units) through `redeem_v2` (`redeem` is deprecated).

Repayment Functions

- `user` can **add liquidity back to the pool and burn loan notes** (amount is in notes to burn) through `repay_v2` (`repay` is deprecated).

4 Findings

BCU-1 Deserialization Should Explicitly Check `Data` Length

Severity: Minor

Status: Fixed

Code Location:

move/concbox/sources/binary_cursor.move#47-81

Descriptions:

In the deserialization methods: `read_u8`, `read_u16`, `read_u32`, `read_u64`, `read_u128`, and `read_u256...` , they do not check the length of bytes before consuming the data.

If the input's length is not enough, it will cause the function to panic.

Suggestion:

It is suggested to add the length check before consuming the data, for example:

```
public fun read_u8(cur: &mut Cursor): u8 {
    let length = vector::length(&cur.data);
    assert!( length >= 1, E_BYTES_LEN_OUT_OF_INDEX);
    let val = vector::trim_reverse(&mut cur.data, length - 1);
    from_bcs::to_u8(val)
}
```

Resolution:

It is fixed by the client via adding the explicit checks.

BRO-1 `update` Function Will Reset The Broker

Severity: Critical

Status: Fixed

Code Location:

move/superp/sources/broker.move#234-253

Descriptions:

In the `broker.move`, `update` function may be used to update some parameters of a broker.

But currently, the update function will reset everything to 0, which destroys the broker. If any user deposits to this broker, then they will lose all their funds.

Suggestion:

It is suggested to either remove this function or modify it to align the design.

Resolution:

The client has fixed this issue by removing `update` function.

BRO-2 `set_interest_updated` Will Set Arbitrary Timestamp

Severity: Critical

Status: Fixed

Code Location:

move/superp/sources/broker.move#501-504

Descriptions:

`set_interest_updated` is a public function that can be called by anyone to set the `broker.interest_updated_at` to any `timestamp` they want.

There are two problems mixed here.

Firstly, this is a privileged function without access control.

Secondly, `interest_updated_at` should only be changed when interest is accrued. If it's allowed to change the `interest_updated_at` to an arbitrary timestamp, then the admin may accrue an infinite amount of interest and thus drain out the pool.

Suggestion:

It is suggested to remove this function.

Resolution:

The client has fixed this issue by removing `set_interest_updated` function.

BRO-3 Set Functions Lack of Access Control

Severity: Major

Status: Fixed

Code Location:

move/superp/sources/broker.move#309-332;

move/superp/sources/entry_admin.move

Descriptions:

In both `broker.move` and `entry_admin.move` smart contracts, there are some set functions that could change the important variants such as `set_tenant_pause` , `set_tenant_liquidation_fee_address` ...

However, there is no access control for those functions, allowing anyone to set arbitrary numbers, take the profits of interest rates, etc.

Suggestion:

It is recommended to add access control to the privileged functions.

Resolution:

The client has fixed this issue by adding the access control.

BRO-4 `view_broker` Should Not Return False Bool Directly

Severity: Medium

Status: Fixed

Code Location:

move/superp/sources/broker.move#109-113

Descriptions:

In `broker.move`, the `view_broker` is a function that reads the current state of the broker. However, instead of reading the bool values from the broker, several values directly return false:

```
is_lend_paused: false,
is_borrow_paused: false,
is_redeem_paused: false,
is_repay_paused: false,
is_paused: false,
```

This will send wrong values for not only this view function but also other functions that call it, for example: `borrow_with_ticket`, `lend_with_ticket`, etc.

Suggestion:

It is recommended to read the values from the broker instead of returning false bool directly.

Resolution:

The client has fixed this issue by reading values from broker.

BRO-5 Rounding Errors Handling Is Not Best Practice

Severity: Medium

Status: Fixed

Code Location:

move/superp/sources/broker.move#411;

move/superp/sources/broker.move#352

Descriptions:

In both `borrow` and `repay` functions, a rounding handling increases `amount` by one. However, this brutal force method is not a good practice since it may add one extra layer to the rounding (if it is already rounded up).

Suggestion:

It is suggested to use a more subtle rounding method.

Resolution:

The client has fixed this issue by addressing the rounding properly.

HOC-1 `verify_platform` Uses A Magic Number

Severity: Minor

Status: Fixed

Code Location:

move/hocket/sources/hocket.move#151

Descriptions:

In the `verify_platform` function, it checks that if the `p.body.platform == 1` which means that if it's on Aptos.

However, using a magic number is confusing and not easy to modify in the future.

Suggestion:

It is suggested to improve the code's readability and facilitate refactoring by defining a constant for every magic number, giving it a clear and self-explanatory name.

Resolution:

The client has fixed this issue by replacing the magic number with a const `APTOS_PLATFORM_ID`.

POR-1 `simple_map::create` Is Deprecated

Severity: Minor

Status: Fixed

Code Location:

move/superp/sources/portfolio.move#40,54

Descriptions:

in many smart contract, `simple_map::create` is used, but it's a deprecated function, and `simple_map::new` should be used instead

Suggestion:

It is recommended to use `new` function to create the map.

Resolution:

The client has fixed this issue by implementing the `simple_map::new` function.

RED-1 Vault Amount Checks Are Not Implemented In Every Operation

Severity: Minor

Status: Fixed

Code Location:

move/superp/sources/redeem.move#92-137;

move/superp/sources/repay.move#94-140

Descriptions:

In `super postion`, the user can perform different operations such as `borrow, redeem, repay`. And each operation will affect the vault amount differently.

However, some operations (repay, redeem) don't have vault checks before and after the operation. And the pre-vault amount calculations are not the same for the other.

This would lower the security of the vault and cause misalignment in design.

Suggestion:

It is recommended to apply the same vault checks to every operation.

Resolution:

The client has fixed this issue by applying the same vault checks.

TEN-1 Fee Rates Should Be Hard Capped Under 100%

Severity: Minor

Status: Fixed

Code Location:

move/superp/sources/tenant.move#127-150

Descriptions:

In `tenant.move`, `set_liquidation_fee_rate`, `set_interest_rate_fee_rate`, `set_stability_fee_rate` can set arbitrary fee rates without limitation.

If any of the fee rate is over 100%, then it'd become the total loss of the funds.

Suggestion:

It is recommended to hard cap the fee rate under 100%.

Resolution:

The client has fixed this issue by hard-capping all the fee rate setting function.

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.