Virtue

Audit Report

MOVEBIT

✉

contact@bitslab.xyz

🐦

https://twitter.com/movebit_

Thu Jul 10 2025

Virtue Audit Report

1 Executive Summary

1.1 Project Information

Description	The project is a lending collateral system.
Type	DeFi
Auditors	MoveBit
Timeline	Tue Jun 24 2025 - Mon Jun 30 2025
Languages	Move
Platform	IOTA
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/Virtue-CDP/move-contracts
Commits	823e067fa86d8e902dd35bdcca0e50c21e1d144d 6968bf4d4eced4c81783d9be05fbfdd09354ca10 2199684b4b9b7647060cb9cfb18bc236f6ccc788

1.2 Files in Scope

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

ID	File	SHA-1 Hash
MOV	virtue_usd/Move.toml	61669a93cd90904d3634e336639772ecc78f1db0
MRE	virtue_usd/sources/module_request.move	43c1290909d9644230e63455eaa4991f2b17f697
LSU	virtue_usd/sources/limited_supply.move	528faba24b80e6170b8d0277b0e0d33d03620115
ADM	virtue_usd/sources/admin.move	9da1f4f1a7e76d3b01a1eb0bcfe976983ca47e73
VUS	virtue_usd/sources/vusd.move	81b7665fddad861fbc5bff110944c933d10da934
MOV2	virtue_cdp/Move.toml	3cdfaea1cb4da0cf2761d228c86c095e571f6ef8
EVE	virtue_cdp/sources/events.move	804c84f80ce8bd9f880d64069e8814ac33ef0b5e
WIT	virtue_cdp/sources/witness.move	80c9608a069e8eddc62b567ea9e249decc2bf365
MEM	virtue_cdp/sources/memo.move	b9f7b95373806388f50194446a6956ef9aae8f64
RES	virtue_cdp/sources/response.move	76c8be105238623bb2b1a8979fda1733f0b8c42d

VAU	virtue_cdp/sources/vault.move	75d5dde83937e3fb16ba8e0d0cd9f2e5ad6a79f0
VER	virtue_cdp/sources/version.move	7332a9616feb126a1e608d11cfc1e753a52ec798
REQ	virtue_cdp/sources/request.move	87353ca413d0999d94e227aa544169ec26c3b93e
MOV3	virtue_oracle/Move.toml	e4ca52d8a5f1f4aeaccf02718e5e88e5e55e9ae5
COL	virtue_oracle/sources/collector.move	b68638925584b6c60c44008d8b936de9d01130ec
LIS	virtue_oracle/sources/listing.move	950c315b12cec39a799c7d39affa2b06cb6bed83
RES1	virtue_oracle/sources/result.move	288f6399d2a5324708865fdd28fe1de091906d0d
AGG	virtue_oracle/sources/aggregater.move	5f9f5300b2bccd56c22ea13c03e7271a06a2be0b
MOV4	virtue_framework/Move.toml	cc3dcc11e7e9cfacb526153c18919e9cc6659659
ACC	virtue_framework/sources/account.move	34e4f68efe264ce823f17088d76af717dc405d22
FLO	virtue_framework/sources/float.move	9c2eacd48d6aaa0e12b25a5106d7fa8d2203e054
DOU	virtue_framework/sources/double.move	899efe9b3a670379c7e8d7fc1c299805a4abc412

LTA	virtue_framework/sources/linked_table.move	21a1e1b632189315b5e8ab0d8b6b89828f782258
VUS	virtue_usd/sources/vusd.move	98137dbec64fcde3ef34a620f4953a6f9f24713c
EVE	virtue_cdp/sources/events.move	ae49df76b99c9ceaa6d0fff11af6ef3802ba7451
RES	virtue_cdp/sources/response.move	cfc2240d6b84d4f4c9e4e9be7466333d09a263f9
VAU	virtue_cdp/sources/vault.move	a7c9a09e1b50de909f17885686edd632e245d1a6
REQ	virtue_cdp/sources/request.move	02eb86a625f6c8e5ad35a81248fbc84891c0d964
COL	virtue_oracle/sources/collector.move	1000ae3ec4eb9a6fe3caea14573953467a0e0a6e
LIS	virtue_oracle/sources/listing.move	05fa78936a8e24539adf3a90797f68feb57718c1
AGG	virtue_oracle/sources/aggregater.move	149523671e76f2df000a8f014966bb4a8f185f09
MOV3	virtue_framework/Move.toml	2249d7c16a694489ad2442cfd6993ef1a7bb4c41
ACC	virtue_framework/sources/account.move	ce3ab8b59694c1baf8150d7fd8f11a57304eeae3
LTA	virtue_framework/sources/linked_table.move	a8949c09f8db56d8353ed6cb4d9dc4bc1a4cef23

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	10	7	3
Informational	2	2	0
Minor	4	3	1
Medium	3	1	2
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(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 `Virtue` to identify any potential issues and vulnerabilities in the source code of the `Virtue` 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
ACC-1	Missing Check Length for `alias`	Minor	Fixed
ACC-2	Typo	Informational	Fixed
AGG-1	Lack of Version Control	Medium	Acknowledged
AGG-2	Redundant Code	Informational	Fixed
VAU-1	Potential Liquidation Risk	Major	Fixed
VAU-2	Centralization Risk	Medium	Acknowledged
VAU-3	Divide Before Multiplying	Minor	Fixed
VAU-4	Collateral May be Locked	Minor	Acknowledged
VAU-5	Missing Check for Parameters	Minor	Fixed
VUS-1	Missing `claimable_map` Remove Mechanism	Medium	Fixed

3 Participant Process

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

- Admin can create a new vault through the `create()` function.
- Admin can set the VUSD supply limit through the `set_supply_limit()` function.
- Admin can update the liquidation rule type through the `set_liquidation_rule()` function.
- Admin can add a witness type to request checklist through the `add_request_check()` function.
- Admin can remove a witness type from request checklist through the `remove_request_check()` function.
- Admin can add a witness type to response checklist through the `add_response_check()` function.
- Admin can remove a witness type from response checklist through the `remove_response_check()` function.
- Admin can set beneficiary address through the `set_beneficiary()` function.
- Admin can set supply limit for module `M` through the `set_supply_limit<M>()` function.
- Admin can add version for module `M` through the `add_version<M>()` function.
- Admin can remove version for module `M` through the `remove_version<M>()` function.
- Admin can remove module configuration through the `remove_module<M>()` function.
- Admin can create a new aggregator for a coin type through the `virtue_oracle::aggregater::new()` function.
- Admin can create and share a new aggregator through the `create()` function.
- Admin can set rule weights through the `set_rule_weight()` function.

- Admin can modify weight thresholds through the `set_weight_threshold()` function.
- Admin can register new coin types through the `register()` function.
- Admin can add a new request rule through the `add_rule()` function.
- Admin can remove an existing request rule through the `remove_rule()` function.
- Admin can mint VUSD tokens through the `mint<M>()` function.
- Admin can burn VUSD tokens through the `burn<M>()` function.

User

- User can modify their position through the `update_position()` function.
- User can liquidate undercollateralized positions through the `liquidate()` function.
- User can destroy response objects after processing through the `destroy_response()` function.
- User can trigger price aggregation through the `aggregate()` function.
- User can create a debt request through the `debtor_request()` function.
- User can add a witness to a request through the `add_witness()` function.
- User can create a donation request to repay someone else's debt through the `donor_request()` function.
- User can create a new account through the `virtue_framework::account::new()` function.
- User can generate an account request from transaction context through the `request()` function.
- User can generate an account request from an existing account through the `request_with_account()` function.
- User can receive objects into an account through the `receive()` function.

Beneficiary

- Beneficiary can claim collected tokens of type `T` through the `claim<T, M>()` function.

4 Findings

ACC-1 Missing Check Length for `alias`

Severity: Minor

Status: Fixed

Code Location:

virtue_framework/sources/account.move#39

Descriptions:

The `virtue_framework::account::new()` function does not check the length of the `alias` parameter passed in.

Suggestion:

It is recommended to limit the length of the `alias` parameter passed in.

Resolution:

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

ACC-2 Typo

Severity: Informational

Status: Fixed

Code Location:

virtue_framework/sources/account.move#48

Descriptions:

The parameter `accout` should be `account`.

Suggestion:

It is recommended to change the parameter `accout` to `account`.

Resolution:

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

AGG-1 Lack of Version Control

Severity: Medium

Status: Acknowledged

Code Location:

virtue_oracle/sources/aggregater.move#138

Descriptions:

Some functions lack version control, such as `aggregate()`.

If those are missing, users might call the deprecated functions.

Suggestion:

It is suggested to add the version control logic in those modules.

AGG-2 Redundant Code

Severity: Informational

Status: Fixed

Code Location:

virtue_oracle/sources/aggregater.move#147

Descriptions:

The check for `total_weight == 0` is redundant, because `total_weight < self.weight_threshold()` covers this case.

Suggestion:

It is recommended to delete the redundant code.

Resolution:

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

VAU-1 Potential Liquidation Risk

Severity: Major

Status: Fixed

Code Location:

virtue_cdp/sources/vault.move

Descriptions:

When the contract is liquidating and updating positions, it does not check whether the position is healthy, which will lead to the following problems: If the collateral price plummets, causing the user's position to be unhealthy, the liquidator can generate a liquidation UpdateRequest at this time. If the user repays and deposits collateral at this time, making the position healthy again, the liquidator can still use the generated liquidation UpdateRequest to liquidate the position that has become healthy.

Suggestion:

It is recommended to check whether the position is healthy when liquidating and updating the position. If the position is already healthy, there is no need to liquidate.

Resolution:

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

VAU-2 Centralization Risk

Severity: Medium

Status: Acknowledged

Code Location:

virtue_cdp/sources/vault.move#337,345

Descriptions:

Centralization risk was identified in the smart contract:

- Admin can use the `set_supply_limit()` function to set the VUSD supply limit corresponding to a certain collateral token to prevent users from repaying, borrowing, or withdrawing collateral assets.
- Admin can use the `remove_module<M>()` function to remove module configurations to prevent users from repaying, borrowing, or withdrawing collateral assets.

Suggestion:

It is recommended that measures be taken to reduce the risk of centralization, such as a multi-signature mechanism.

VAU-3 Divide Before Multiplying

Severity: Minor

Status: Fixed

Code Location:

virtue_cdp/sources/vault.move#545

Descriptions:

In the `interest_amount()` function, the amount is divided by `31_536_000_000` first, and then multiplied by `position.debt_amount`, which may cause precision loss and result in less actual interest.

Suggestion:

It is recommended to multiply by `position.debt_amount` first, and then divide by `31_536_000_000`.

Resolution:

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

VAU-4 Collateral May be Locked

Severity: Minor

Status: Acknowledged

Code Location:

virtue_cdp/sources/vault.move#224

Descriptions:

Since there is no other way to withdraw collateral in the contract, when there are active positions in the vault, if certain modules are deprecated or the VUSD supply limit corresponding to this type of collateral is set to 0, this type of collateral will be locked in the contract.

Suggestion:

It is recommended to ensure that there are no active positions that depend on the module before calling the `remove_module()` function, or implement another collateral extraction mechanism so that the contract can extract the collateral assets and reduce project losses.

VAU-5 Missing Check for Parameters

Severity: Minor

Status: Fixed

Code Location:

virtue_cdp/sources/vault.move#83;

virtue_usd/sources/limited_supply.move#23

Descriptions:

The `new()` function does not check whether the value of the numeric type parameter passed in is within a reasonable range.

Suggestion:

It is recommended to check whether the value of the numeric type parameter passed in is within a reasonable range.

Resolution:

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

VUS-1 Missing `claimable_map` Remove Mechanism

Severity: Medium

Status: Fixed

Code Location:

virtue_usd/sources/vusd.move#243

Descriptions:

The contract `claimable_map` has a balance mapping for adding token types, but does not provide a function to remove specific token type mappings. If certain token types or module types are no longer supported, there should be a corresponding removal mechanism that can be used to remove these data.

Suggestion:

It is recommended to add a mechanism to remove data in `claimable_map`.

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.

MOVEBIT VERIFIED
MOVEBIT