TBook Token

Audit Report

MOVEBIT

contact@bitslab.xyz

https://twitter.com/movebit_

Wed Dec 24 2025

TBook Token Audit Report

1 Executive Summary

1.1 Project Information

Description	TBook Token is a token on the Sui blockchain
Type	DeFi
Auditors	Alex,Bear Two
Timeline	Tue Dec 23 2025 - Wed Dec 24 2025
Languages	Move
Platform	Sui
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/tbook-dev/tbook-token-sui/
Commits	2dfe159e8d5c66ea946cdaa25db33565e4b176a8 c5bb74da8d6d8ad1b776e938cb950d8099795373

1.2 Files in Scope

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

ID	File	SHA-1 Hash
TTO	sources/tbook_token.move	3271246852345e55b503974a224f65d09767d4e2

1.3 Issue Statistic

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

ID	Title	Severity	Status
TTO-1	Potential Arithmetic Overflow in Supply Cap Check	Minor	Fixed

3 Participant Process

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

Admin

- `mint` - Mint new `BOOK` tokens and send them to the specified address.
- `burn` - Destroy a specified number of `BOOK` tokens and reduce the total supply.

4 Findings

TTO-1 Potential Arithmetic Overflow in Supply Cap Check

Severity: Minor

Status: Fixed

Code Location:

sources/tbook_token.move#38

Descriptions:

The `mint` function enforces a total supply cap using the following check: `assert!(coin::total_supply(treasury_cap) + amount <= TOTAL_SUPPLY, 1);` .
This validation relies on performing an addition (`total_supply + amount`) before the comparison. If `coin::total_supply(treasury_cap) + amount` overflows `u64` , the transaction will abort at a lower-level arithmetic overflow, rather than failing with the intended application-level error code `1` .
As a result, the abort reason becomes inconsistent and bypasses the explicit supply cap check, making error handling less predictable and harder to reason about for integrators and off-chain tooling.

Suggestion:

It is recommended to modify it to `amount < TOTAL_SUPPLY - coin::total_supply(treasury_cap)` .

Resolution:

The team adopted our advice and fixed this issue by modifying it to `amount < TOTAL_SUPPLY - coin::total_supply(treasury_cap)` , which can be found at c5bb74da8d6d8ad1b776e938cb950d8099795373.

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