MMT Coin

Audit Report

contact@bitslab.xyz

https://twitter.com/movebit_

Wed Oct 15 2025

MMT Coin Audit Report

1 Executive Summary

1.1 Project Information

Description	MMT Coin is the official implementation of the MMT token, issued by Momentum Finance. MMT serves as the governance token of Momentum — the Move ecosystem's liquidity engine that empowers veMMT holders to drive protocol governance, liquidity allocation, and cross-chain ecosystem growth. The package on Sui mainnet is 0x35169bc93e1fddfcf3a82a9eae726d349689ed59e4b065369af8789fe59f8608.
Type	DEX
Auditors	Sprig, Leon@BitsLab
Timeline	Tue Oct 14 2025 - Wed Oct 15 2025
Languages	Move
Platform	Sui
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/mmt-finance/mmt-coin
Commits	474b64a30bdb6f829281cba76bf638aaac40d5cb b515e9c7608769842190e1ce1258b9b44e66630d

1.2 Files in Scope

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

ID	File	SHA-1 Hash
MMT	sources/mmt.move	663e90b7c36ab37f7eb5af7ab22756b126c5c5f0

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	2	1	1
Informational	2	1	1
Minor	0	0	0
Medium	0	0	0
Major	0	0	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 MMT Finance to identify any potential issues and vulnerabilities in the source code of the MMT Coin 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 2 issues of varying severity, listed below.

ID	Title	Severity	Status
MMT-1	Non-Frozen Coin Metadata	Informational	Acknowledged
MMT-2	Migration Function Mappings	Informational	Fixed

3 Participant Process

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

Admin

- The admin can deploy the token and obtain token management caps through `init()`.

The MMT team has transferred these caps to a MSafe multi-sig wallet 0xbe6faaf95e723873a5f4b34dc2ff51d25c739f618a39f53fd9a71b3d201c7aa8.
All processes will be managed through a multi-signature wallet to eliminate any single point of failure. This approach has been confirmed by the Momentum team.

4 Findings

MMT-1 Non-Frozen Coin Metadata

Severity: Informational

Status: Acknowledged

Code Location:

sources/mmt.move#29

Descriptions:

The coin metadata is not frozen. If the metadata is modified, it may cause the token to become unusable.

Suggestion:

It is recommended to use `public_freeze_object` instead of `public_share_object`.

Resolution:

The clients replied they might update the metadata in the future.

MMT-2 Migration Function Mappings

Severity: Informational

Status: Fixed

Code Location:

sources/mmt.move#18

Descriptions:

The `create_regulated_currency_v2()` is deprecated in the latest Sui framework. In the documentation of Sui, smart contracts should update logic that relies on the coin module to use the coin_registry module instead.

Suggestion:

It is recommended to use `coin_registry::new_currency_with_otw` instead of `coin::create_regulated_currency_v2`.

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.