MOVEBIT
Securing the Move Ecosystem

MAVEN
by MSafe

Maven Smart Contract **Audit Report**

https://twitter.com/movebit_

contact@movebit.xyz

Maven Smart Contract Audit Report

1 Executive Summary

1.1 Project Information

Description	A Multi-Signature wallet on Sui.
Type	Multi-Signature wallet
Auditors	MoveBit
Timeline	Apr 13, 2023 – May 4, 2023
Languages	Move
Platform	Sui
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/Momentum-Safe/MavenCore
Commits	6fa0a179ee21f61bccc4f89331c6bb817999fdb5 2320f971e9d0d2839a039fc42ddcfa97d508447a ff98ad151f2c5761afd1eeed222a91af3de5f780 9c921b8d30cf1f70c7fd3134e2af412192718387

1.2 Files in Scope

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

ID	Files	SHA-1 Hash
AOR	move/sources/operations/admin_operation.move	a9363c35e5edc7b1835ec7856c54092571a09b94
COR	move/sources/operations/coin_operation.move	77e59f3e4077b04d70139483c9e7caa2ecc393b1
OPR	move/sources/operations/object_operation.move	01e3775d7fd1d9e7c1bed6aa776e899d47f72dce
RPR	move/sources/operations/recovery_operation.move	7a3730014f953adfec6d35a40c46aa0f56a5a577
AWL	move/sources/allowlist.move	7b53f5e0b0a03eae0bb52582ed0ddacfc147187f
MID	move/sources/id.move	a8fac108e132fb2ea74503bb104a2abc45ed1b5e
MAV	move/sources/maven.move	946cd1c16568680aee7769ea67e8f22bba4add1c
OCT	move/sources/order_context.move	4797eb77bd1c43fecec95432a55d0c4af695b4f9
OTL	move/sources/order_timelock.move	b873ada34749a217fa9d895e87f3ca28326205a6
ORD	move/sources/order.move	bba65b29e22290cd429f7685985df9a12502213b
PAU	move/sources/pause.move	c78c9bb2386399456bd22f70f05f64e616282493
PMI	move/sources/permission.move	8d95536302f54cafbf5bb839ddc4d49c0dbd4fcc
RFE	move/sources/reference.move	86bdc877cb9db688ba5cec96ec516b83ae779bef

ROL	move/sources/role.move	23c186974e88c55434c18dc7abdc47a434c38cfd
SDL	move/sources/spending_limit.move	70cc38bc41721eb33294dfa3d27424b042e4ecf8
TLK	move/sources/time_lock.move	32062633381d23c7e6cb7f644f64a7bc7e997daf
UTL	move/sources/utils.move	16926faedac9bb568fd646cb8acfc963adf19497
VUT	move/sources/vault.move	5b17cf4d62f2d741f0f0d9b5ebb1fd3b74d2b11a

1.3 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

1.4 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 **Momentum-Safe** to identify any potential issues and vulnerabilities in the source code of the **Maven** 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 **7** issues of varying severity, listed below.

ID	Title	Severity	Status
MAV-01	Lack of Validation for `name` and `uri` Parameters in `execute_meta_info` Function	Minor	Fixed
MAV-02	Ambiguity Issue with Proposal Approval and Rejection	Informational	Acknowledged
MAV-03	DoS Attack Caused by Failed Transfer Operations on the Same Object	Medium	Fixed
MAV-04	Duplicated `Seq` Contexts Created in `execute_admin_operation` Function	Minor	Fixed
ORD-01	Denial of Service from Privileged User where Permission Has Single Signer Settings	Medium	Fixed
PMI-01	`Time Lock` overflow issue in default `Maven` struct	Medium	Fixed
SDL-01	The unit test is throwing an error when executed	Minor	Fixed

3 Participant Process

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

Admin

- Admin can update paused state through `set_paused`.
- Admin can change owner through `change_owner`.
- Admin can upgrade the contract version.
- Admin can globally lock to prevent critical errors.

Maven Admin

- Maven admin can create a maven.
- Maven admin can edit signer.
- Maven admin can edit role.
- Maven admin can edit permission.
- Maven admin can edit the permission table.
- Maven admin can edit the allowlist.
- Maven admin can edit spending limits.

Maven User

- Maven users can propose new normal/time-locked operations.
- Maven users can vote in favor or against a normal/time-locked operation.
- Maven users can execute an approved normal/time-locked operation.
- Maven users can skip a rejected normal/time-locked operation.
- Maven users can deposit coins or objects in the vault.
- Maven users can consume coins through a spending limit.
- Maven users can transfer assets between Maven wallets.

4 Findings

MAV-01 Lack of Validation for `name` and `uri` Parameters in `execute_meta_info` Function

Severity: Minor

Status: Fixed

Code Location: move/sources/maven.move, L181–L188.

Descriptions: When modifying the `name` and `uri` fields of the Maven struct, it is necessary to validate the `name` and `uri` parameters. The `execute_meta_info` function modifies the `name` and `uri` fields but does not perform validation.

Suggestion: To address this issue, we recommend adding validation for the `name` and `uri` parameters in the `execute_meta_info` function to align with the behavior of the `update_meta_info` function.

Resolution: The client followed our suggestion and fixed this issue.

MAV-02 Ambiguity Issue with Proposal Approval and Rejection

Severity: Informational

Status: Acknowledged

Code Location: move/sources/maven.move, L501.

Descriptions: If the threshold for permission is set too low, it is possible for a proposal to be both approved and rejected, resulting in an ambiguous execution result.

Suggestion: Make the state of the proposal more clear.

MAV-03 DoS Attack Caused by Failed Transfer Operations on the Same Object

Severity: Medium

Status: Fixed

Code Location: move/sources/permission.move, L779.

Descriptions: Performing multiple transfer operations on the same object in one proposal, the first transfer operation will succeed, and the later transfer operations for this object will fail when `execute_object_operation` is executed, causing the subsequent proposal execution process to be blocked. This causes a DoS attack, the way to resolve this is to revote this proposal to be rejected and skipped.

Suggestion: Add a check for duplicated object IDs in `execute_object_operation` & `execute_object_to_maven_vault_operation`, and ignore the redundant IDs.

MAV-04 Duplicated `Seq` Contexts Created in `execute_admin_operation` Function

Severity: Minor

Status: Fixed
Code Location: move/sources/maven.move, L649-L653.
Descriptions: In the `execute_admin_operation` function, two identical `Seq` contexts are created by `order_context::new_seq_context(maven_id)`.
Suggestion: In the `execute_admin_operation` function, duplicated `Seq` contexts should be avoided. This can be solved by deleting one `Seq` context.

ORD-01 Denial of Service from Privileged User where Permission Has Single Signer Settings

Severity: Medium

Status: Fixed
Descriptions: A proposal exists where the proposer and approver are in the same roles, and the role only has one signer with the authority to vote on the proposal. If the role chooses not to execute or reject the proposal, this will cause the execution queue to be blocked, resulting in a denial of service (DoS) attack.
Suggestion: We suggest that you conduct research and assess the impact of the issue, and determine whether appropriate measures need to be taken to prevent any problems from occurring.

Resolution: The client followed our suggestion and fixed this issue.

PMI-01 `Time Lock` Overflow Issue in Default `Maven` Struct

Severity: Medium

Status: Fixed
Code Location: move/sources/time_lock.move, L17-L18.
Descriptions: If no other operations are created through the pro `operation`, then the time lock for recovery in the default Maven structure is set to `MAX_U64`. If a proposal is initiated at this time and is approved, calling `start_permission_recovery` → `order_timelock::sta`

`rt_timelock_order` → `time_lock::new` will cause an overflow and crash, rendering the operation unable to execute. The only solution is to cancel the permission recovery.

Suggestion: Modify the relevant functions to ensure that the time lock for recovery in the default Maven struct is set to a value that will not cause an overflow. This will prevent the operation from crashing and allow it to execute as intended.

Resolution: The client followed our suggestion and fixed this issue.

SDL-01 The Unit Test is Throwing an Error When Executed

Severity: Minor

Status: Fixed
Code Location: move/sources/spending_limit.move, L302.
Descriptions: The `clock::create_for_testing` function in SUI has been modified to return a `Clock` object instead of a `shared` object. This change has caused the tests to fail when running with the latest version (0.32) of the Sui client.

Suggestion: Call `clock::share_for_testing` to use the `Clock` object returned by the `clock::create_for_testing` function as a shared object.

Resolution: The client followed our 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.
- **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

Securing the Move Ecosystem

https://twitter.com/movebit_

contact@movebit.xyz