MOVEBIT
Securing the Move Ecosystem

MoveFlow Smart Contract **Audit Report**

https://twitter.com/movebit_

contact@movebit.xyz

MoveFlow Smart Contract Audit Report

1 Executive Summary

1.1 Project Information

Description	A crypto asset streaming protocol.
Type	DeFi
Auditors	MoveBit
Timeline	Apr 13th, 2023 – Apr 27th, 2023
Languages	Move
Platform	Aptos
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/Move-Flow/moveflow
Commits	31242c3ce5e9a5007048f4bbf70b168faeb63a57 a4c8a16f3aef99756bedc648b2f43bfaa8a51670

1.2 Files in Scope

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

ID	Files	SHA-1 Hash

STM	sources/stream.move	6749d94aabbb8b62e15824eac840ede9da14dfe7

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	7	7	
Informational	2	2	
Minor	3	3	
Medium	1	1	
Major	1	1	
Critical			

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 sees 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 **MoveFlow** to identify any potential issues and vulnerabilities in the source code of the **MoveFlow** 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.

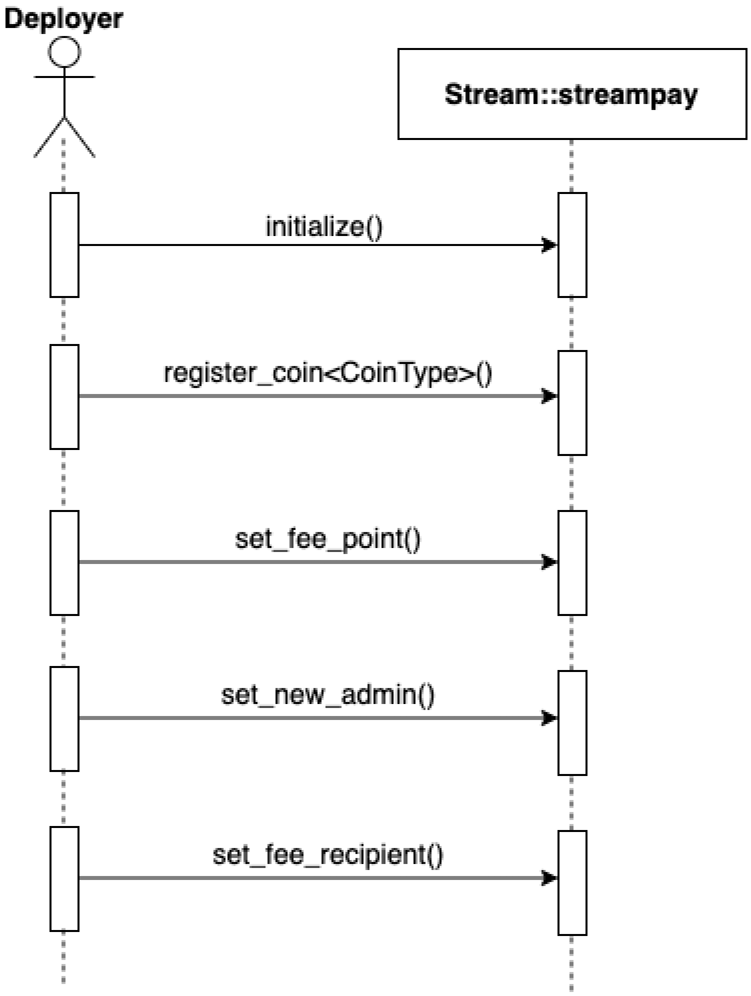
ID	Title	Severity	Status
STM-01	No Update to `stream.closed` in `close`	Major	Fixed
STM-02	No Update to `global.input_stream` in `set_new_recipient`	Medium	Fixed
STM-03	Unused Constant Detected: `STREAM_REJECT_EXTEND`	Minor	Fixed
STM-04	Parameter Not Verified	Minor	Fixed
STM-05	Incorrect Error Code Category Used	Minor	Fixed
STM-06	Inaccurate Comment	Informational	Fixed
STM-07	Typo	Informational	Fixed

3 Participant Process

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

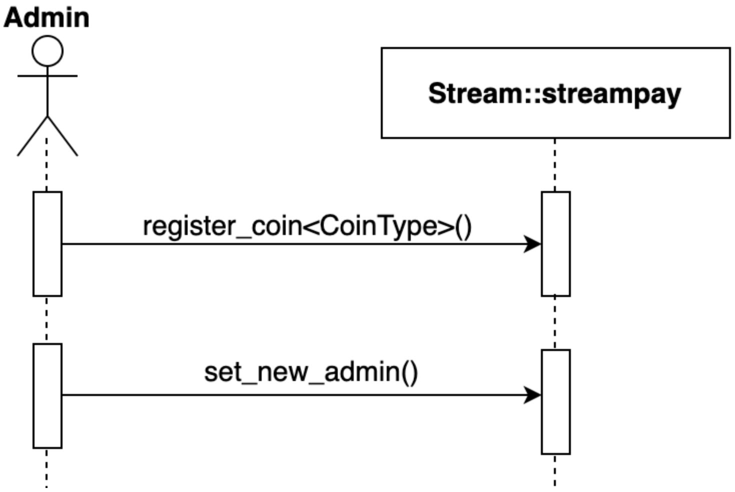
Deployer

- Deployer can initialize the `GlobalConfig` resource.
- Deployer can register a coin for the stream.
- Deployer can set the fee point.
- Deployer can set a new admin address.
- Deployer can set a new fee recipient address.



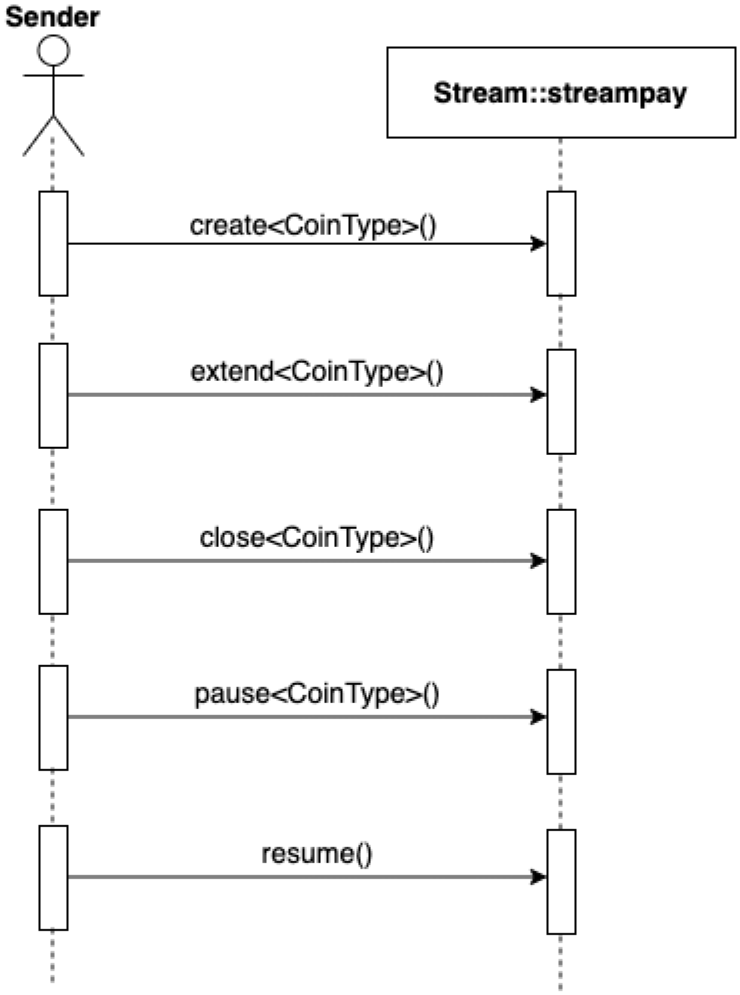
Admin

- Admin can register a coin for the stream.
- Admin can set a new admin address.



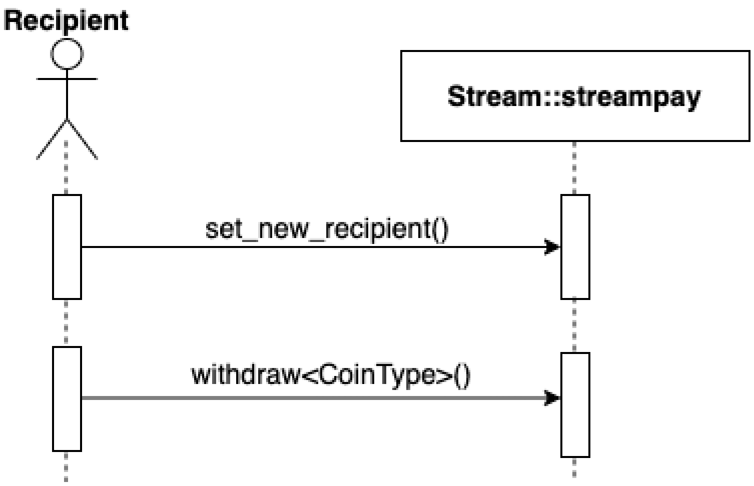
Sender

- Sender can create a stream.
- Sender can extend a stream created by itself.
- Sender can close a stream created by itself.
- Sender can pause a stream created by itself.
- Sender can resume a stream created by itself.



Recipient

- Recipient can withdraw from the stream.
- Recipient can set a new recipient for the steam.



4 Findings

STM-01 No Update to `stream.closed` in `close`

Severity: Major

Status: Fixed

Code Location: sources/Stream.move, L388.

Descriptions: In the `close` function, the state of the variable `stream.closed` is not updated after the program executes the withdrawal operation, the user can still extend and withdraw after steam is closed, but the deposited tokens can not be withdrawn because the amount to be withdrawn will be more than the amount in steam.

Suggestion: It is recommended to update the status of `stream.closed` at the end of the `close` function.

Resolution: The client followed our suggestion and fixed this issue in the commit `a4c8a16f3aef99756bedc648b2f43bfaa8a51670`.

STM-02 No Update to `global.input_stream` in `set_new_recipient`

Severity: Medium

Status: Fixed

Code Location: sources/Stream.move, L734.

Descriptions: In the `set_new_recipient` method, the `global.input_stream` is not

updated after changing the recipient, the front end will not be able to find the input stream corresponding to the new recipient.
Suggestion: It is recommended to update the global variable `global.input_stream` after changing the recipient.

Resolution: The client followed our suggestion and fixed this issue in the commit `a4c8a16f3aef99756bedc648b2f43bfaa8a51670` .

STM-03 Unused Constant Detected: STREAM_REJECT_EXTEND

Severity: Minor

Status: Fixed
Code Location: sources/Stream.move, L34.
Descriptions: The constant `STREAM_REJECT_EXTEND` is not used throughout the entire contract.
Suggestion: It is recommended to remove it to avoid increasing code complexity and occupying contract storage space.

Resolution: The client followed our suggestion and fixed this issue in the commit `a4c8a16f3aef99756bedc648b2f43bfaa8a51670` .

STM-04 Parameter Not Verified

Severity: Minor

Status: Fixed
Code Location: sources/Stream.move, L208.
Descriptions: There's no adequate validation for the input parameters `start_time` and `stop_time` . This could lead to issues if the current time exceeds the stop time or if the start time is greater than the stop time.
Suggestion: It is recommended to add proper input validation for the `start_time` and `stop_time` parameters in the `create` method. This can be achieved by checking if the start time is less than or equal to the stop time, and if the current time is less than or equal to the stop time. If either of these conditions fails, an error should be thrown to prevent the creation of an invalid stream. Adding proper input validation will improve the robustness and security of the contract.

Resolution: The client followed our suggestion and fixed this issue in the commit `a4c8a16f3aef99756bedc648b2f43bfaa8a51670`.

STM–05 Incorrect Error Code Category Used

Severity: Minor

Status: Fixed
Code Location: sources/Stream.move, L231, L337, L398.
Descriptions: In lines 231, 337, 398, and so on of the code, incorrect error code categories have been used, which may lead to confusion and hinder code readability.
Suggestion: The appropriate error code categories should be used to improve the code's readability and ease of use. For example, `not_found` can be used instead of `already_exists`.

Resolution: The client followed our suggestion and fixed this issue in the commit `a4c8a16f3aef99756bedc648b2f43bfaa8a51670`.

STM–06 Inaccurate Comment

Severity: Informational

Status: Fixed
Code Location: sources/Stream.move, L94.
Descriptions: The comment at line 94 may not be accurate as the deposit_amount variable is subject to change in the extend method.
Suggestion: It is recommended to modify the comment to provide a more accurate description of the code.

Resolution: The client followed our suggestion and fixed this issue in the commit `a4c8a16f3aef99756bedc648b2f43bfaa8a51670`.

STM–07 Typo

Severity: Informational

Status: Fixed
Code Location: sources/Stream.move, L68, L109.

Descriptions: There are some typos in the aforementioned lines. Such as: `pasue` .
Suggestion: It is recommended to correct the typos.

Resolution: The client followed our suggestion and fixed this issue in the commit `a4c8a16f3aef99756bedc648b2f43bfaa8a51670` .

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