StreamFlow2

Audit Report

contact@movebit.xyz

https://twitter.com/movebit_

Tue Mar 26 2024

StreamFlow2 Audit Report

1 Executive Summary

1.1 Project Information

Description	A crypto asset streaming protocol.
Type	DeFi
Auditors	MoveBit
Timeline	Wed Mar 06 2024 - Fri Mar 08 2024
Languages	Move
Platform	Aptos
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/streamflow-finance/aptos-streamflow-module
Commits	e10006ae3e7929625f7e8b21ac7e6671461f87df 7e16e45b0c812641e6176ad6c8996a9a574b0a60 7e16e45b0c812641e6176ad6c8996a9a574b0a60

1.2 Files in Scope

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

ID	File	SHA-1 Hash
TUT	sources/test_utils.move	2bd0a3cfdab296f0cb7dba28d7f13e2d04cecd1a
STR	sources/strmt.move	87e2f9e0065c5bec7e13006675a475bc8b774478
UTI	sources/utils.move	0863762fa4dae6b77a2430f0ed9baefb54e90e74
FMA	sources/fee_manager.move	bbb5730f0e6b9352c1c7e55e2e0294834596e01d
ADM	sources/admin.move	1db37833bcfd8c5c5b83e6b43c1317cf2d8e80eb
PRO	sources/protocol.move	1122d6808a61ac3b7218c11727d9a6559eaa6ca0
SCT	sources/StrmCoinType.move	49d428d41f0f4eb3e8db67293367ee7420e46c1f
ADM	sources/admin.move	cba533135e20ffb7169cb406517719ebd1acf766
FEE	sources/fees.move	b7999abc28372e62eb3eaf1f2074b3138a45887a
MAT	aptos-streamflow-module/temp/sources/math.move	750624e5676d9bc10cb9fef2dbaa8229002493e4
PTE	aptos-streamflow-module/temp/sources/protocol_tests.move	c11519e8ab7ea6a69ffd6d6b049fccbae6e26a79

PRO	aptos-streamflow-module/temp/sources/protocol.move	e7185f98148abda1842742841c6943d28a794215
ADM	sources/admin.move	6ed3bec51b4f9b28d1b640cb4bd01e7a66fc5196

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	7	4	3
Informational	1	1	0
Minor	5	3	2
Medium	1	0	1
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

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 StreamFlow2 to identify any potential issues and vulnerabilities in the source code of the StreamFlow2 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
ADM-1	Missing Events For Important Parameter Updates	Minor	Fixed
ADM-2	Lack of `#[test]` Attribute	Minor	Acknowledged
PRO-1	`pausable` and `current_pause_start` Parameters cannot be Modified	Medium	Acknowledged
PRO-2	Unused Constants Should Be Removed	Minor	Fixed
PRO-3	Unused Event Should Be Removed	Minor	Fixed
PRO-4	Deprecated Function can Still be Used	Minor	Acknowledged
PRO-5	Unnecessary `create_signer_with_capability`	Informational	Fixed

3 Participant Process

Here are the relevant actors with their respective abilities within the StreamFlow2 Smart Contract :

Admin

- Admin can update the `streamflow_fee` and `tx_fee` through `change_tx_fee` and `change_streamflow_fee` .
- Admin can update the `treasury` address through `change_treasury` .
- Admin can add a `FeeValue` to `Fee_Table` through `fees_write` .

User

- User can create a `Contract` through `create` .
- User can update the information of `Contract` through `update` .
- User can get the `Contract` coin through `withdraw` .
- User can cancel the `Contract` and withdraw the left coin through `cancel` .
- User can extend the Contract time through `topup` .
- User can pause and unpause the `Contract` through `pause` and `unpause` .

4 Findings

ADM-1 Missing Events For Important Parameter Updates

Severity: Minor

Status: Fixed

Code Location:

sources/admin.move#60,70,80,90,100;

sources/fees.move#30,37

Descriptions:

We found that when important parameters are updated in the project, the function doesn't emit the update event, so we suggest emitting the emit event in time so as to notify the user or chain off programs.

Suggestion:

It is recommended to emit the corresponding event in time when updating the important parameter.

Resolution:

The client has added events for key actions.

ADM-2 Lack of `#[test]` Attribute

Severity: Minor

Status: Acknowledged

Code Location:

sources/admin.move#36

Descriptions:

The `init_module_test` function is missing the test `#[test]` attribute tags, missing them would cause the function to be compiled into the program and, since the permissions are public, any user can call the function.

Suggestion:

It is suggested to add `#[test]` attribute tag to the `init_module_test` function.

Resolution:

The client already knows that this issue does not pose a security risk.

PRO-1 `pausable` and `current_pause_start` Parameters cannot be Modified

Severity: Medium

Status: Acknowledged

Code Location:

sources/protocol.move#48

Descriptions:

In the `create` function we can initialize the `pausable` and `current_pause_start` parameters, but the `validate_contract_params` function requires that the determined `pausable` parameter must be false, and there is no way to modify `pausable` and `current_pause_start` anywhere else. In addition, there is no specific implementation of the pause method in the contract.

Suggestion:

It is recommended to confirm whether this is in line with the design concept.

Resolution:

The client says that the current user interface does not utilize this functionality and there are no plans to do so at this time.

PRO-2 Unused Constants Should Be Removed

Severity: Minor

Status: Fixed

Code Location:

sources/protocol.move#27

Descriptions:

There are unused constants that may be removed.

```
// protocol.move
const EBAD_INPUT_UPDATE_RATE: u64 = 8;
```

Suggestion:

It is recommended to remove those constants if not used, or keep them but comment them out for future use.

Resolution:

The client has removed unused variables.

PRO-3 Unused Event Should Be Removed

Severity: Minor

Status: Fixed

Code Location:

sources/protocol.move#282-284

Descriptions:

There are unused events that may be removed, such as `EscrowInitEvent` .

Suggestion:

It is recommended to remove that event if not used, or keep them but comment them out for future use.

Resolution:

The client has removed unused events.

PRO-4 Deprecated Function can Still be Used

Severity: Minor

Status: Acknowledged

Code Location:

sources/protocol.move#497-525

Descriptions:

The comments of the `collect_fees` function indicate that this function has been deprecated and should not be used, but the function can still be called normally, which may cause unnecessary losses.

Suggestion:

It is recommended to delete or comment out this function.

Resolution:

The client says this is a backward incompatible change and will not result in a financial loss.

PRO-5 Unnecessary `create_signer_with_capability`

Severity: Informational

Status: Fixed

Code Location:

sources/protocol.move#606

Descriptions:

For the `coin::transfer` function, the recipient parameter does not need a singer, so in the `topup` function, Contract's `contract_signer` is not needed when transferring the coin to Contract, and the `contract_address` is used directly.

Suggestion:

It is recommended to just use `contract_address`.

Resolution:

The client has taken our suggestions to optimize the code.

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.