Trufin

Audit Report

MOVEBIT

contact@movebit.xyz

https://twitter.com/movebit_

Tue Apr 30 2024

Trufin Audit Report

1 Executive Summary

1.1 Project Information

Description	The TruFin Aptos staking vault offers users a reliable way of staking APT on the Aptos network.
Type	Staking
Auditors	MoveBit
Timeline	Sun Apr 07 2024 - Tue Apr 30 2024
Languages	Move
Platform	Aptos
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/TruFin-io/aptos-staker-movebit
Commits	e9b2ecb5e45c4dbb7c128a613d1ea5360deaf50f 4a00ae622e65ab6627ad0f0c67d0368db2e7d680

1.2 Files in Scope

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

ID	File	SHA-1 Hash
MWH	aptos-whitelist/sources/master_whitelist.move	c2a3bc15e8a4a2c691b69b6cb76543a5ded5b414
TAPT	aptos-staker/sources/truAPT.move	083097cf45d17da53d2bb64ff0c6c4b5bc22c5fb
STA	aptos-staker/sources/staker.move	70e850333114927d8c11a0f6c322317b94c16553
SSP	aptos-staker/sources/staker.spec.move	56a65bfe912e1d77732275b40042a4e3e5e3e443
MWH	aptos-whitelist/sources/master_whitelist.move	d9abdaa8c3f54c5a75673914af72771a00d77cd8
TAPT	aptos-staker/sources/truAPT.move	3c48f66d1c72083186c7ae63bdb085844fb96b79
STA	aptos-staker/sources/staker.move	4af2670f72c53a5f0e370d3955f801511db384b4

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	3	2	1
Informational	0	0	0
Minor	1	1	0
Medium	2	1	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 Trufin to identify any potential issues and vulnerabilities in the source code of the Trufin 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 3 issues of varying severity, listed below.

ID	Title	Severity	Status
MWH-1	Centralization Risk	Medium	Acknowledged
STA-1	May Set the Default Pool to a Wrong Pool in `initialize`	Medium	Fixed
STA-2	Duplicated Checking	Minor	Fixed

3 Participant Process

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

Admin

- The owner can call the `claim_penalty` function to collect residual rewards that accumulated upon delegation pool unlocks and transfer them to the treasury.
- The owner can utilize the `enable_pool` and `disable_pool` functions to enable and disable a delegation pool for users to stake.
- The owner can invoke the `add_pool` function to add a new delegation pool that users can stake.
- The owner can utilize the `upgrade_contract` function to upgrade the contract.
- The owner can utilize the `pause` and `unpause` functions to pause or activate the contract.

User

- Users can stake the APT to any pool by the `stake` and `stake_to_specific_pool` functions.
- Users can allocate staking rewards to another user by the `allocate` function.
- Users can distribute allocation rewards from the caller to the specified recipient by the `distribute_rewards` and `distribute_all` functions.
- Users can request to unlock a certain amount of APT from the default delegation pool by the `unlock` and `unlock_from_specific_pool` functions.
- Users can withdraw a previously requested and now unlocked APT amount from the staker by the `withdraw` and `withdraw_list` functions.
- Users can collect treasury fees accumulated on the staking rewards by the `collect_fees` function.

4 Findings

MWH-1 Centralization Risk

Severity: Medium

Status: Acknowledged

Code Location:

aptos-whitelist/sources/master_whitelist.move#132,172;

aptos-staker/sources/staker.move#880,896

Descriptions:

This contract has centralization risk:

- The admin can call the `whitelist_user` and `blacklist_user` function to arbitrarily blacklist and whitelist any account.
- The admin can call the `pause` and `unpause` functions to control the availability or unavailability of the entire contract.

Suggestion:

It is recommended to implement decentralized governance mechanisms to distribute control and mitigate centralization risks. Specifically, consider implementing a multi-signature approval process for critical actions such as minting tokens or modifying the blacklist.

Resolution:

The client already knows this problem.

STA-1 May Set the Default Pool to a Wrong Pool in `initialize`

Severity: Medium

Status: Fixed

Code Location:

aptos-staker/sources/staker.move#784

Descriptions:

In the `initialize` function, it only checks that the `default_delegation_pool` cannot be equal to a zero address, and does not check whether the corresponding pool exists and whether the pool is available, etc. If it is set incorrectly, it may lead to the failure of the initial staking.

Suggestion:

It's recommended to add a check for `default_delegation_pool` in the `initialize` function.

Resolution:

The client has added checks to resolve this issue.

STA-2 Duplicated Checking

Severity: Minor

Status: Fixed

Code Location:

aptos-staker/sources/staker.move#1604,1606

Descriptions:

The `check_deposit_amount` function checks whether the amount is greater than zero and greater than min_deposit. Since the initialization requires `min_deposit` to be set to a minimum value of 10 APT, and `min_deposit` cannot be set to a value less than 10 APT in the `set_min_deposit` function, so the check for an amount greater than or equal to min_deposit already includes the check for an amount greater than 0.

Suggestion:

It is recommended that redundant checks be deleted or commented out.

Resolution:

Client has removed redundant checks.

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