0x Guard

Smart contracts security assessment

Final report

Tariff: Top

Spintria

July 2024

0xguard.com

hello@0xguard.com

Contents

Introduction

The report has been prepared for **Spintria**.

The code is available at the GitHub repository and was audited after the commit 101e689.

The project under review implements a vesting functionality for user tokens. Tokens sent to the contract can be locked for a specified period, after which they are unlocked linearly over time.

The repository being audited is a fork of the TON Stablecoin Contract. Several modifications have been made to support the vesting functionality. The primary contract, `jetton-minter`, is responsible for receiving tokens and creating auxiliary contracts, known as `jetton-wallets`, which hold the users' balances. This approach of using separate contracts for managing user data adheres to the best practices in the TON blockchain ecosystem, ensuring scalability and efficiency.

Update. A recheck has been done after the commit 379e6db.

Name	Spintria
Audit date	2024-07-12 - 2024-07-18
Language	FunC
Platform	TON

Contracts checked

Name	Address
jetton-wallet	

Scope of the audit

The following files were audited:

```
contracts/jetton-minter.fc
contracts/gas.fc
contracts/jetton-utils.fc
contracts/jetton-wallet.fc
contracts/jetton.tlb
contracts/op-codes.fc
contracts/stdlib.fc
contracts/vesting-utils.fc
contracts/workchain.fc
contracts/helpers/librarian.func
```

Procedure

We perform our audit according to the following procedure:

Automated analysis

- Scanning the project's smart contracts with available automated analysis tools.
- Manual verification (reject or confirm) all the issues found by the tools.

Manual audit

- Manually analyze smart contracts for security vulnerabilities
- Smart contracts' logic check.

Centralization Analysis

- The contract admin may change metadata.

- The contract admin can update the code of the contract.

- The contract owner can block outgoing and incoming transfers.

- The contract owner can transfer users' tokens.

Classification of issue severity

High severity

High severity issues can cause a significant or full loss of funds, change of contract ownership, major interference with contract logic. Such issues require immediate attention.

Medium severity

Medium severity issues do not pose an immediate risk, but can be detrimental to the client's reputation if exploited. Medium severity issues may lead to a contract failure and can be fixed by modifying the contract state or redeployment. Such issues require attention.

Low severity

Low severity issues do not cause significant destruction to the contract's functionality. Such issues are recommended to be taken into consideration.

Issues

High severity issues

No issues were found

Medium severity issues

No issues were found

Low severity issues

1. Not optimized calculations (jetton-wallet)

Status: Open

The math operations within the `burn_jettons()` function can be optimized for efficiency. The current implementation calculates the `clawback_amount` using multiple subtraction operations, which can be simplified.

```
() burn_jettons(slice in_msg_body, slice sender_address, int msg_value) impure
inline_ref {
  ...
        int withdrawals = deposit - balance;
        ;; calculate clawback amount (unlocked jettons only)
        clawback_amount = deposit - withdrawals - locked_amount;
        balance -= clawback_amount;
  ...
}
```

The current formula, `deposit - withdrawals - locked_amount`, essentially simplifies to `balance - locked_amount`, and balance update can be reduced to balance = locked_amount.

Recommendation: Update the `burn_jettons()` function to use the optimized calculation:

```
() burn_jettons(slice in_msg_body, slice sender_address, int msg_value) impure
inline_ref {
  ...
        ;; calculate clawback amount (unlocked jettons only)
        clawback_amount = balance - locked_amount;
        balance = locked_amount;
  ...
}
```

Conclusion

Spintria jetton-wallet contract was audited. 1 low severity issue was found.

The modifications made to the TON Stablecoin Contract to support vesting functionality are well-implemented and follow best practices. However, the project is highly centralized. Users who interact with the contract have to trust the project owner. This centralization means that the project owner has significant control over the contract and its operations. For detailed insights, refer to the Centralization Analysis section.

Automated tests have been written for the core vesting functionality. However, some tests are failing. We recommend achieving full test coverage, as some bugs are difficult to detect manually but could be easily identified with comprehensive automated test suites.

Update. The Spintria team has addressed the failing tests issue. As a result, all tests are now passing successfully.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability)set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without 0xGuard prior written consent.

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts 0xGuard to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Automated tests results output

```
Test Suites: 3 passed, 3 total
Tests:       14 skipped, 65 passed, 79 total
Snapshots:   0 total
Time:        8.117 s
Ran all test suites.
```

The repository is a fork of TON Stablecoin Contract. Tests have been updated to reflect the changes made to the contract. Also, tests for the vesting functionality were written.

0x Guard