SecondLive-Ton Audit Report

Wed Sep 25 2024

contact@bitslab.xyz https://twitter.com/tonbit_

TonBit

SecondLive-Ton Audit Report

1 Executive Summary

1.1 Project Information

Description	A Jetton protocol with additional features
Type	DeFi
Auditors	TonBit
Timeline	Mon Sep 23 2024 - Wed Sep 25 2024
Languages	FunC
Platform	Ton
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/SecondLive23/LiveToken-Ton
Commits	42660e3c9349b5855d20b45c83a1a991fae16425

1.2 Files in Scope

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

ID	File	SHA-1 Hash
GAS	contracts/gas.fc	f41eb715f9d9d9c2afc724372407977253be66d6
STD	contracts/stdlib.fc	4293738fb6071a57c37ec55a4bc21434575efa0f
OCO	contracts/op-codes.fc	951a6645f6b46bd714c1bc438599003bf43c7d2f
JWA	contracts/jetton-wallet.fc	a04de3d690354b2497115c40acf26c3314715a1f
JMI	contracts/jetton-minter.fc	b139562a3c905f405bf3a52aea883a73afc1542d
WOR	contracts/workchain.fc	5cc87f9a156325873f58e99eca69c8ff3b3cd29a
JUT	contracts/jetton-utils.fc	5416df683bb942a05db030ddc1f2b7d43c8bf154

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	1	0	1
Informational	0	0	0
Minor	0	0	0
Medium	0	0	0
Major	1	0	1
Critical	0	0	0

1.4 TonBit Audit Breakdown

TonBit 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

1.5 Methodology

The security team adopted the **"Testing and Automated Analysis"**, **"Code Review"** 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) 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 SecondLive to identify any potential issues and vulnerabilities in the source code of the Live Token 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 1 issues of varying severity, listed below.

ID	Title	Severity	Status
JMI-1	Centralization Risk	Major	Acknowledged

3 Participant Process

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

Admin

- The admin can mint tokens via the `op::mint` message.
- The admin can change the admin via the `op::change_admin` message.
- The admin can change the metadata URI via the `op::change_metadata_uri` message.
- The admin can upgrade the contract via the `op::upgrade` message.
- The admin can transfer user tokens, burn user tokens, and set the user's status via the `op::call_to` message.

User

- The user can transfer tokens via the `op::transfer` message.

4 Findings

JMI-1 Centralization Risk

Severity: Major

Status: Acknowledged

Code Location:

contracts/jetton-minter.fc#227-234

Descriptions:

The admin has the ability to mint, change the admin, call `call_to()` , and upgrade the contract, which introduces centralization risks.

```
if (op == op::upgrade) {
    throw_unless(error::not_owner, equal_slices_bits(sender_address, admin_address));
    (cell new_data, cell new_code) = (in_msg_body~load_ref(), in_msg_body~load_ref());
    in_msg_body.end_parse();
    set_data(new_data);
    set_code(new_code);
    return ();
}
```

Suggestion:

It is recommended to take ways to reduce the risk of centralization.

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.