Catizen Jetton Smart Contract Audit Report

Mon Aug 19 2024

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

TonBit

Catizen Jetton Smart Contract Audit Report

1 Executive Summary

1.1 Project Information

Description	A Ton jetton project.
Type	Token
Auditors	TonBit
Timeline	Sun Aug 18 2024 - Sun Aug 18 2024
Languages	FunC
Platform	Ton
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/catizendev/cattoncontract
Commits	4235b58409490a72d28968fcbee5bcc955505257

1.2 Files in Scope

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

ID	File	SHA-1 Hash
JWA	contracts/jetton/jetton-wallet.fc	9917a7110b87985088f7fadbad2cadfafccbd0f0
PAR	contracts/jetton/imports/params.fc	3e86ce82bee70992c9b0f7b4fcacf0cacfcfec1b
STD	contracts/jetton/imports/stdlib.fc	48ba5be2230d6db462adb890e7b15ff0b36b90de
OCO	contracts/jetton/imports/op-codes.fc	de6e2645c68d08535a353fa1b6bde7ac915d8ef5
DPA	contracts/jetton/imports/discovery-params.fc	6809258270f1565706bba2eb7f3bcf5b2289e0ac
UTI	contracts/jetton/imports/utils.fc	19cd144cd1353e5179c9cefdd1e9b4f484f4b016
CON	contracts/jetton/imports/constants.fc	4630656a3a259560d0f49710829754698357f4d1
JUT	contracts/jetton/imports/jetton-utils.fc	e725b3a317c7c347307c6c7a4b689119c04c8b58
JMI	contracts/jetton/jetton-minter.fc	62f1749ec13c46ea05997f3bf71ca7220203a691

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	1	1	0
Informational	0	0	0
Minor	0	0	0
Medium	1	1	0
Major	0	0	0
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 Catizen to identify any potential issues and vulnerabilities in the source code of the Catizen Jetton 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	Medium	Fixed

3 Participant Process

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

Admin

- The admin can mint any amount of jetton tokens through `mint` message.
- The admin can change the admin address through `3` message.
- The admin can change content, delete this for immutable tokens through `4` message.

User

- The user can provide a wallet address through `provide_wallet_address` message.
- The user can transfer their jetton tokens through `transfer` message.
- The user can burn their jetton tokens through `burn` message.

4 Findings

JMI-1 Centralization Risk

Severity: Medium

Status: Fixed

Code Location:

contracts/jetton/jetton-minter.fc#70

Descriptions:

Centralization risk was identified in the smart contract.

In the contract, the admin can mint any amount of jetton tokens through `mint` message.

Suggestion:

It is recommended to use a multi-signature wallet to reduce the risk of single point failure and abuse of authority.

Resolution:

The client has fixed this issue by transferring the owner to a zero address, and the catizen jetton contract address on the ton blockchain is `EQD-cvR0Nz6XAyRBvbhz-abTrRC6sI5tvHvvpeQraV9UAAD7` .

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.

TONBIT VERIFIED

TONBIT VERIFIED