Catizen Smart Contarct Audit Report

Mon Aug 26 2024

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

TonBit

Catizen Smart Contarct Audit Report

1 Executive Summary

1.1 Project Information

Description	Catizen is a cat-themed social entertainment experience on Telegram.
Type	SocialFi
Auditors	TonBit
Timeline	Sat Aug 17 2024 - Sat Aug 17 2024
Languages	Tact
Platform	Ton
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/catizendev/cattoncontract.git
Commits	10325df695955f1ced1424f221e9f87f4ccfca7d 4235b58409490a72d28968fcbee5bcc955505257

1.2 Files in Scope

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

ID	File	SHA-1 Hash
CTS	contracts/cat_ton_signing.tact	1f641970a131fac95d9fff0c117bf1957059d6d7
CTB	contracts/cat_ton_booster.tact	51371126f0ed5441d8ba0930e236b21d2a059bb5
STD	contracts/imports/stdlib.fc	2f104cd568a4cebb1c4112ecf8979800f0672575

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	2	0	2
Informational	2	0	2
Minor	0	0	0
Medium	0	0	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 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 2 issues of varying severity, listed below.

ID	Title	Severity	Status
CTS-1	Lack of Permission Check and Parameter Validation	Informational	Acknowledged
CTS-2	Unused Variable	Informational	Acknowledged

3 Participant Process

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

Admin

- The Admin can withdraw the asset in the contract through `OwnerWithdraw` and `OwnerWithdrawJetton` messages.

User

- The user can send a message to the contract and get a reply through `SignAction1` and `SignAction2` messages.

4 Findings

CTS-1 Lack of Permission Check and Parameter Validation

Severity: Informational

Status: Acknowledged

Code Location:

contracts/cat_ton_signing.tact#42,48;

contracts/cat_ton_booster.tact#42,48

Descriptions:

The `SignAction1` and `SignAction2` functions lack proper permission checks and parameter validation, allowing any user to invoke them with arbitrary `comment` inputs, which in turn can modify the global variable `total_count`.

Suggestion:

It is recommended to confirm it aligns with your design.

CTS-2 Unused Variable

Severity: Informational

Status: Acknowledged

Code Location:

contracts/cat_ton_signing.tact#43,49;

contracts/cat_ton_booster.tact#43,49

Descriptions:

There is an unused variable in the `SignAction1` and `SignAction2` functions.

```
let ctx: Context = context();
```

Suggestion:

It is recommended to remove the unused variable.

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