Kaito Token
Audit Report

Wed Feb 05 2025

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

ScaleBit

Kaito Token Audit Report

1 Executive Summary

1.1 Project Information

Description	It is an ERC20 token.
Type	Token
Auditors	ScaleBit
Timeline	Wed Feb 05 2025 - Wed Feb 05 2025
Languages	Solidity
Platform	Others
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/OpenKaito/kaito-contracts
Commits	1d7ee6466f2ea12071b2e755e6a5107594a28767 18542915b886739b3d2e13fb5c6b41c1559e6646

1.2 Files in Scope

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

ID	File	SHA-1 Hash
KAI	contracts/Kaito.sol	228c83991b98829e6a01ac8e44c08fa4eed4d3e9

1.3 Issue Statistic

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

1.4 ScaleBit Audit Breakdown

ScaleBit 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
- Number of rounding errors
- Unchecked External Call
- Unchecked CALL Return Values
- Functionality Checks
- Reentrancy
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic issues
- Gas usage
- Fallback function usage
- tx.origin authentication
- Replay attacks
- Coding style issues

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) 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 Kaito to identify any potential issues and vulnerabilities in the source code of the Kaito 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
KAI-1	Centralization Risk	Major	Fixed

3 Participant Process

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

User

- Users can transfer tokens to another address through the `transfer()` function inherited from the `ERC20.sol` contract.
- Users can approve other addresses to spend tokens on their behalf through the `approve()` function inherited from the `ERC20.sol` contract.
- Users can transfer the token from an approved address to another address through the `transferFrom()` function inherited from the `ERC20.sol` contract.
- Users can burn their tokens through the `burn()` function.
- Users can use off-chain signatures to authorize other addresses to use their tokens through the `permit()` function inherited from the `ERC20Permit.sol` contract.

4 Findings

KAI-1 Centralization Risk

Severity: Major

Status: Fixed

Code Location:

contracts/Kaito.sol#22,31

Descriptions:

Centralization risk was identified in the smart contract:

- The minter can mint any amount of tokens for any address.

Suggestion:

It is recommended that measures be taken to reduce the risk of centralization, such as a multi-signature mechanism.

Resolution:

The client removed the minter-related functions and fixed this issue. All of the tokens are distributed to a few addresses when deploying the smart contract.

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.