



QuillAudits

Audit Report December, 2023

For



KaldiMarket™

Table of Content

Executive Summary	02
Number of Security Issues per Severity	03
Checked Vulnerabilities	04
Techniques and Methods	05
Types of Severity	06
Types of Issues	06
A. Contract - Kaldi.sol	07
High Severity Issues	07
Medium Severity Issues	07
Low Severity Issues	07
Informational Issues	07
Functional Tests	08
Automated Tests	09
Closing Summary	09
Disclaimer	09



Executive Summary

Project Name

Kaldi Coin

Overview

Kaldi coin is a token contract that inherits the standard ERC20 contract for token creation. The total supply of KALDI tokens is capped at 5 billion and allows only the owner to mint tokens until it reaches its cap.

Timeline

6th December 2023 - 7th December 2023

Method

Manual Review, Automated Testing, Functional Testing, etc.

Language

Solidity

Blockchain

Polygon

Audit Scope

The scope of this audit was to analyze Kaldi Coin Token Contract for quality, security, and correctness.

<https://gitlab.rapidinnovation.tech/root/contract-solidity-Kaldi-MarketPlace/-/blob/namechange/src/Kaldi.sol>

Branch

Namechange

Commit

1d30056abd7c6d6c9d01d241f309e57b28d05e64

Contracts in Scope

Kaldi.sol



Number of Security Issues per Severity



High

Medium

Low

Informational

	High	Medium	Low	Informational
Open Issues	0	0	0	0
Acknowledged Issues	0	0	0	0
Partially Resolved Issues	0	0	0	0
Resolved Issues	0	0	0	0

Checked Vulnerabilities

- ✓ Re-entrancy
- ✓ Timestamp Dependence
- ✓ Gas Limit and Loops
- ✓ DoS with Block Gas Limit
- ✓ Transaction-Ordering Dependence
- ✓ Use of tx.origin
- ✓ Exception disorder
- ✓ Gasless send
- ✓ Balance equality
- ✓ Byte array
- ✓ Transfer forwards all gas
- ✓ ERC20 API violation
- ✓ Malicious libraries
- ✓ Compiler version not fixed
- ✓ Redundant fallback function
- ✓ Send instead of transfer
- ✓ Style guide violation
- ✓ Unchecked external call
- ✓ Unchecked math
- ✓ Unsafe type inference
- ✓ Implicit visibility level



Techniques and Methods

Throughout the audit of smart contracts, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments, match logic and expected behaviour.
- Token distribution and calculations are as per the intended behaviour mentioned in the whitepaper.
- Implementation of ERC-20 token standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods, and tools were used to review all the smart contracts.

Structural Analysis

In this step, we have analyzed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.

Static Analysis

A static Analysis of Smart Contracts was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of smart contracts.

Code Review / Manual Analysis

Manual Analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analyzed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of the automated analysis were manually verified.

Gas Consumption

In this step, we have checked the behavior of smart contracts in production. Checks were done to know how much gas gets consumed and the possibilities of optimization of code to reduce gas consumption.

Tools and Platforms used for Audit

Remix IDE, Truffle, Solhint, Mythril, Slither, Solidity Statistic Analysis.



Types of Severity

Every issue in this report has been assigned to a severity level. There are four levels of severity, and each of them has been explained below.

High Severity Issues

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality, and we recommend these issues be fixed before moving to a live environment.

Medium Severity Issues

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems, and they should still be fixed.

Low Severity Issues

Low-level severity issues can cause minor impact and are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

Informational

These are four severity issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Types of Issues

Open

Security vulnerabilities identified that must be resolved and are currently unresolved.

Resolved

These are the issues identified in the initial audit and have been successfully fixed.

Acknowledged

Vulnerabilities which have been acknowledged but are yet to be resolved.

Partially Resolved

Considerable efforts have been invested to reduce the risk/impact of the security issue, but are not completely resolved.



A. Contract - Kaldi.sol

High Severity Issues

No issues were found.

Medium Severity Issues

No issues were found.

Low Severity Issues

No issues were found.

Informational Issues

No issues were found.



Functional Tests

Some of the tests performed are mentioned below:

- ✓ Should be able to get owner
- ✓ Should be able to get decimal
- ✓ Should be able to get Token Name
- ✓ Should be able to Total Supply
- ✓ Should be able to get balancesOf
- ✓ Should be able to IncreaseAllowance
- ✓ Should be Able to DecreaseAllowance
- ✓ Should be able to TransferFrom
- ✓ Should be Able to Transfer
- ✓ Should be able to mint
- ✓ Should not be able to mint above cap



Automated Tests

No major issues were found. Some false positive errors were reported by the tools. All the other issues have been categorized above according to their level of severity.

Closing Summary

In this report, we have considered the security of the Kaldi Coin. We performed our audit according to the procedure described above.

No issue found during the audit.

Disclaimer

QuillAudits Smart contract security audit provides services to help identify and mitigate potential security risks in Kaldi Coin smart contracts. However, it is important to understand that no security audit can guarantee complete protection against all possible security threats. QuillAudits audit reports are based on the information provided to us at the time of the audit, and we cannot guarantee the accuracy or completeness of this information. Additionally, the security landscape is constantly evolving, and new security threats may emerge after the audit has been completed.

Therefore, it is recommended that multiple audits and bug bounty programs be conducted to ensure the ongoing security of Kaldi Coin smart contracts. One audit is not enough to guarantee complete protection against all possible security threats. It is important to implement proper risk management strategies and stay vigilant in monitoring your smart contracts for potential security risks.

QuillAudits cannot be held liable for any security breaches or losses that may occur subsequent to and despite using our audit services. It is the responsibility of Kaldi Coin to implement the recommendations provided in our audit reports and to take appropriate steps to mitigate potential security risks.



About QuillAudits

QuillAudits is a secure smart contracts audit platform designed by QuillHash Technologies. We are a team of dedicated blockchain security experts and smart contract auditors determined to ensure that Smart Contract-based Web3 projects can avail the latest and best security solutions to operate in a trustworthy and risk-free ecosystem.



850+
Audits Completed



\$30B
Secured



\$30B
Lines of Code Audited



Follow Our Journey



Audit Report December, 2023

For



KaldiMarket™



QuillAudits

📍 Canada, India, Singapore, UAE, UK

🌐 www.quillaudits.com

✉️ audits@quillhash.com