QuillAudits

AUDIT REPORT

January, 2025

For

RDM

Table of Content

Executive Summary

Project name	RDM
Overview	RDM Token is an ERC20 token with a capped supply of 10 billion tokens, leveraging OpenZeppelin's contracts for security and standard compliance. It introduces a unique minting feature controlled by the owner, ensuring the total supply never exceeds the cap. Initially, 2 billion tokens are minted to the owner's address. This project emphasizes ownership control and supply management, making it suitable for various decentralized applications.
Project Website	https://www.rdmtoken.com
Audit Scope	The scope of this audit was to analyse the RDM Token for quality, security, and correctness. https://testnet.bscscan.com/address/0xF5701E77941544a6A1F301257ff6Fc139831daE3#code
Blockchain	BSC
Method	Manual Review, Functional Testing, Automated Testing, etc. All the raised flags were manually reviewed and re-tested to identify any false positives.
Review 1	14 January 2025
Updated Code Received	19th January 2024
Review 2	21st January 2025

Fixed In	https://github.com/Ke5haav007/rdm-Token/blob/main/contracts/RDMToken.sol commit: 0c5109910d484eefeb2d1373b103670968730ff2

Number of Issues per Severity

High	0 (0.00%)
Medium	0 (0.00%)
Low	0 (0.00%)
Informational	1 (100.00%)

Severity

Issues

	High	Medium	Low	Informational
Open	0	0	0	0
Resolved	0	0	0	**1**
Acknowledged	0	0	0	0
Partially Resolved	0	0	0	0

Checked Vulnerabilities

- ☑ Re-entrancy
- ☑ Timestamp Dependence
- ☑ DoS with Block Gas Limit
- ☑ Gas Limit and Loops
- ☑ Transaction-Ordering Dependence
- ☑ Transfer Forwards All Gas
- ☑ Compiler Version Not Fixed
- ☑ Redundant Fallback Function
- ☑ Send Instead of Transfer
- ☑ Style Guide Violation
- ☑ Unchecked Math
- ☑ Unchecked External Call
- ☑ Implicit Visibility Level
- ☑ Access Management
- ☑ Arbitrary Write to Storage
- ☑ Centralization of Control
- ☑ Improper or Missing Events
- ☑ Race Conditions/Front Running
- ☑ Malicious Libraries
- ☑ Revert/Require Functions
- ☑ Missing Zero Address Validation
- ☑ Private Modifier

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 behavior.
- Token distribution and calculations are as per the intended behavior mentioned in the whitepaper.
- Implementation of ERC 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, Foundry, Solhint, Mythril, Slither, Solidity statistical 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 Issues

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

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

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.

Informational Severity Issues

Unlocked pragma

Resolved

Description

Contract has a floating solidity pragma version. This is present also in inherited contracts. Locking the pragma helps to ensure that the contract does not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively. The recent solidity pragma version also possesses its own unique bugs.

Recommendation

Making the contract use a stable solidity pragma version prevents bugs occurrence that could be ushered in by prospective versions. It is recommended, therefore, to use a fixed solidity pragma version while deploying to avoid deployment with versions that could expose the contract to attack.

Functional Tests

Some of the tests performed are mentioned below:

- Should get the name of the token
- should get the symbol of the token
- should get the total supply of the token when deployed
- should get the Max supply of the token when deployed
- should get balance of the owner when contract is deployed
- should transfer tokens to other address
- Should be able to transfer From
- should get the Decimals
- Only Owner should able to Mint Tokens
- Should be able to check Allowance
- Should be able to check balanceof
- Should be able to Transfer Ownership
- Should be able to Renounce ownership

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 RDM Token contract. We performed our audit according to the procedure described above.

One issues of informational severity was found, which RDM Team Resolved.

Disclaimer

QuillAudits Smart contract security audit provides services to help identify and mitigate potential security risks in RDM smart contract. 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 RDM smart contract. 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 the RDM Team to implement the recommendations provided in our audit reports and to take appropriate steps to mitigate potential security risks.

About QuillAudits

QuillAudits is a leading name in Web3 security, offering top-notch solutions to safeguard projects across DeFi, GameFi, NFT gaming, and all blockchain layers. With six years of expertise, we've secured over 1000 projects globally, averting over $30 billion in losses. Our specialists rigorously audit smart contracts and ensure DApp safety on major platforms like Ethereum, BSC, Arbitrum, Algorand, Tron, Polygon, Polkadot, Fantom, NEAR, Solana, and others, guaranteeing your project's security with cutting-edge practices.

6+ Years of Expertise	**1M+** Lines of Code Audited
$30B+ Secured in Digital Assets	**1K+** Projects Secured

Follow Our Journey

AUDIT REPORT

January, 2025

For

RDM

QuillAudits

Canada, India, Singapore, UAE, UK

www.quillaudits.com audits@quillaudits.com