SECURITY AUDIT OF

REDHEAL TOKEN

RED HEAL

Public Report

May 09, 2025

Verichains Lab

info@verichains.io

https://www.verichains.io

Driving Technology > Forward

ABBREVIATIONS

Name	Description
Smart contract	A computer protocol intended to digitally facilitate, verify or enforce the negotiation or performance of a contract.
Solidity	A contract-oriented, high-level language for implementing smart contracts for the Ethereum platform.
Solc	A compiler for Solidity.
ERC20	ERC20 (BEP20 in Binance Smart Chain or *x*RP20 in other chains) tokens are blockchain-based assets that have value and can be sent and received. The primary difference with the primary coin is that instead of running on their own blockchain, ERC20 tokens are issued on a network that supports smart contracts such as Ethereum or Binance Smart Chain.
Polygon	Polygon is a protocol and a framework for building and connecting Ethereum-compatible blockchain networks. Aggregating scalable solutions on Ethereum supporting a multi-chain Ethereum ecosystem.

EXECUTIVE SUMMARY

This Security Audit Report was prepared by Verichains Lab on May 09, 2025. We would like to thank the REDHeal Company for trusting Verichains Lab in auditing smart contracts. Delivering high-quality audits is always our top priority.

This audit focused on identifying security flaws in code and the design of the REDHeal Token. The scope of the audit is limited to the source code files provided to Verichains. Verichains Lab completed the assessment using manual, static, and dynamic analysis techniques.

During the audit process, the audit team had identified no vulnerable issue in the smart contract code.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY

1.1. About REDHeal Token

The REDHeal project is a DeFi ecosystem built on the BNB Smart Chain (BSC) that utilizes blockchain technology to offer decentralized financial services. Central to this ecosystem is the REDH token, which functions both as a platform token for value exchange and rewards, and as a governance token that enables participation in the platform's DAO. The REDHeal platform aims to create a sustainable liquidity pool where users can deposit tokens to earn profits. It also supports DeFi services such as staking and loans secured by crypto assets, enhancing transparency, efficiency, and accessibility through smart contracts and distributed ledger technology.

For more information, visit the official REDHeal website at https://redheal.io/.

1.2. Audit Scope

This audit focused on identifying security flaws in code and the design of the REDHeal Token. It was conducted on commit `a0d6ba32f53df8a5a5912c98fad3aed06ffc5c5a` from git repository: https://github.com/redhealcompany/Redheal

SHA256 Sum	File
8e9098c444490e4353b87341b4099d4e3c6f57a8f596b26dfd0d16ffd4de7738	redhealToken.sol

1.3. Audit Methodology

Our security audit process for smart contract includes two steps:

- Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using public and RK87, our in-house smart contract security analysis tool.
- Manual audit of the codes for security issues. The contracts are manually analyzed to look for any potential problems.

Following is the list of commonly known vulnerabilities that were considered during the audit of the smart contract:

- Integer Overflow and Underflow
- Timestamp Dependence
- Race Conditions
- Transaction-Ordering Dependence
- DoS with (Unexpected) revert
- DoS with Block Gas Limit

- Gas Usage, Gas Limit and Loops
- Redundant fallback function
- Unsafe type Inference
- Reentrancy
- Explicit visibility of functions state variables (external, internal, private and public)
- Logic Flaws

For vulnerabilities, we categorize the findings into categories as listed in table below, depending on their severity level:

SEVERITY LEVEL	DESCRIPTION
CRITICAL	A vulnerability that can disrupt the contract functioning; creates a critical risk to the contract; required to be fixed immediately.
HIGH	A vulnerability that could affect the desired outcome of executing the contract with high impact; needs to be fixed with high priority.
MEDIUM	A vulnerability that could affect the desired outcome of executing the contract with medium impact in a specific scenario; needs to be fixed.
LOW	An issue that does not have a significant impact, can be considered as less important.

Table 1. Severity levels

1.4. Disclaimer

REDHeal Company acknowledges that the security services provided by Verichains, are conducted to the best of their professional abilities but cannot guarantee 100% coverage of all security vulnerabilities. REDHeal Company understands and accepts that despite rigorous auditing, certain vulnerabilities may remain undetected. Therefore, REDHeal Company agrees that Verichains shall not be held responsible or liable, and shall not be charged for any hacking incidents that occur due to security vulnerabilities not identified during the audit process.

1.5. Acceptance Minute

This final report served by Verichains to the REDHeal Company will be considered an Acceptance Minute. Within 7 days, if no any further responses or reports is received from the REDHeal Company, the final report will be considered fully accepted by the REDHeal Company without the signature.

2. AUDIT RESULT

2.1. Overview

The REDHeal Token was written in `Solidity` language, with the required version to be `^0.8.29`. The source code was written based on **OpenZeppelin's library**.

The token contract extends both the `ERC20` and `Ownable` contracts from version `4.9.0`. Through `Ownable`, the contract deployer sets the Owner through the constructor, and the total supply is minted to this Owner. The token is an `ERC20` implementation with the following properties:

PROPERTY	VALUE
Name	Redheal
Symbol	REDH
Decimals	18
Total Supply	1,500,000,000 (x10^{18}) Note: the number of decimals is 18, so the total representation token will be 1,500,000,000 or 1,5 billion.

Table 2. The REDHeal Token properties

2.2. Findings

During the audit process, the audit team had identified no vulnerable issue in the smart contract code.

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*May 09, 2025*	Public Report	Verichains Lab

Table 3. Report versions history