SECURITY AUDIT OF

NFT5 RED-BLACK TREE LIBRARY CONTRACT

Public Report

Apr 19, 2022

Verichains Lab

info@verichains.io

https://www.verichains.io

Driving Technology > Forward

ABBREVIATIONS

Name	Description
Ethereum	An open source platform based on blockchain technology to create and distribute smart contracts and decentralized applications.
Ether (ETH)	A cryptocurrency whose blockchain is generated by the Ethereum platform. Ether is used for payment of transactions and computing services in the Ethereum network.
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.

EXECUTIVE SUMMARY

This Security Audit Report prepared by Verichains Lab on Apr 19, 2022. We would like to thank the NFT5 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 NFT5 Red-Black Tree Library Contract. 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 NFT5 Red-Black Tree Library Contract

NFT5 is Vietnam's first web 3.0 platform for raising capital, where you can unleash your creativity, built on the BSC (Binance Smart Chain) blockchain and co-founded by Remitano - one of the largest and oldest cryptocurrencies exchanges in Vietnam. NFT5 was born with the goal of connecting investors and creators, bringing potential creative projects to the community, and helping to create a sustainable ecosystem.

If you are a creator facing financial difficulties in producing creative content, lack the freedom to create and manage content for your project, or want to spread the impact of your creative projects to the community, NFT5 is the place for you.

If you are an investor looking for projects with high profitability, sustainable multi-dimensional income or want to own a creative project (movie, music video...) by investing and helping them to grow, NFT5 will open the gate to connect creative projects for you.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the Red-Black tree library which is used in the MarketPlace contract. It was conducted on the source code provided by NFT5 team.

The latest version of the following files were made available in the course of the review:

SHA256 Sum	File
37fb8ab59f2c85cf52b0f3f326bd5686b44034b7be3f383c0cd028703f6a2373	**RedBlackTreeLibrary.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 was 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

Please note that security auditing cannot uncover all existing vulnerabilities, and even an audit in which no vulnerabilities are found is not a guarantee for a 100% secure smart contract. However, auditing allows discovering vulnerabilities that were unobserved, overlooked during development and areas where additional security measures are necessary.

2. AUDIT RESULT

2.1. Overview

The Red-Black tree library contract was written in Solidity language, with the required version to be 0.8.9. The source code is based on the BokkyPooBahsRedBlackTreeLibrary contract at the repository *https://github.com/bokkypoobah/BokkyPooBahsRedBlackTreeLibrary/* with some minor modifications.

This is a gas-efficient Solidity library using the iterative (rather than recursive) Red-Black binary search tree algorithm to help maintain a sorted uint key index for the input data. Insertions, deletions, and searches are in O(log n) time (and ~gas). In this implementation, the key of 0 or duplicated keys are prohibited. We can use the sorted keys as indices to our mapping tables of data to access data in sorted order.

2.2. Findings

During the audit process, the audit team found no vulnerability in the given version of NFT5 Red-Black Tree Library Contract.

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Apr 19, 2022*	Public Report	Verichains Lab

Table 2. Report versions history