REHASH CRYPTO

Smart Contract Security Audit

# REHASHCRYPTO

JUSTIN YIELD FINANCE TOKEN

# Contents

# Disclaimer

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and ReHashCrypto and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (ReHashCrypto) owe no duty of care towards you or any other person, nor does ReHashCrypto make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and ReHashCrypto hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, ReHashCrypto hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against ReHashCrypto, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

# Audit Details

Project Name: JUSTIN YIELD FINANCE
Website: https://justinyield.com/

Token Name: JUSTIN YIELD FINANCE

Platform: Binance Smart Chain
Type of Token: BEP20
Initial Supply: 500,000

Token ticker: JIY

Decimals: 5

Contract address: 0x0E5E088242c81267F5668909C17ddeC33C56B026

Link: https://bscscan.com/token/0x0E5E088242c81267F5668909C17ddeC33C56B026
Languages: Solidity (Smart contract)
Platforms & Tools: Remix IDE, Truffle, Ganache, Solhint, Mythril, Contract Library
Compiler Version: v0.7.4+commit.3f05b770
Block chain: Binance Smart Chain Project

The audit items and results: (Other unknown security vulnerabilities are not included in the audit responsibility scope)
Audit Result: Passed
Audit Date: May 20, 2022
Audit Team: REHASH AUDIT TEAM
REHASHCRYPTO received the application for a smart contract security audit of JUSTIN YIELD FINANCE TOKEN on MAY 20, 2022. The following are the details and results of this smart contract security audit:

The audit items and results:
(Other unknown security vulnerabilities are not included in the audit responsibility scope)
Audit Result: Passed Ownership: Not renounced (Can be Renounced)
(The contract contains ownership functionality and ownership is not renounced which allows the creator or current owner to modify contract behavior)
(Contract owner can modify fees and can modify max tx)

Audit Team: REHASH CRYPTO https://rehashcrypto.com/

# Introduction

This Audit Report mainly focuses on the overall security of JUSTIN YIELD FINANCE Smart Contract. With this report, we have tried to ensure the reliability and correctness of their smart contract by complete and rigorous assessment of their system's architecture and the smart contract codebase.

## Auditing Approach and Methodologies applied

The REHASH CRYPTO team has performed rigorous testing of the project starting with analyzing the code design patterns in which we reviewed the smart contract architecture to ensure it is structured and safe use of third-party smart contracts and libraries.

Our team then performed a formal line by line inspection of the Smart Contract to find any potential issue like race conditions, transaction-ordering dependence, timestamp dependence, and denial of service attacks.

In the Unit testing Phase, we coded/conducted custom unit tests written for each function in the contract to verify that each function works as expected.

In Automated Testing, we tested the Smart Contract with our in-house developed tools to identify vulnerabilities and security flaws.

The code was tested in collaboration of our multiple team members and this included -

- Testing the functionality of the Smart Contract to determine proper logic has been followed throughout the whole process.
- Analyzing the complexity of the code in depth and detailed, manual review of the code, line- by-line.
- Deploying the code on test net using multiple clients to run live tests.
- Analyzing failure preparations to check how the Smart Contract performs in case of any bugs and vulnerabilities.
- Checking whether all the libraries used in the code are on the latest version.
- Analyzing the security of the on-chain data.

# About the project

JustInYIELDFinance Token is a token built on the Binance Smart Chain that is with an innovative investment use case. The main purpose of which is to seek out constant revenue sources, Autocompounding Autostaking protocol backed by Defi 3.0 yield farming on BSC. The Autocompounding Autostaking Protocol will bring an unparallel, fixed APY of 915501%, the highest of its kind onto the BSC blockchain, while imposing profound ease, simplicity, and accessibility upon all Protocol holders. Each transaction, purchase incurs 15% fee, and sale incurs a 20% fee.

## Features and Tokenomics

- 5% of the buy and sales fees is directed to the insurance which helps sustain and back the Staking Rewards provided by the Positive Rebase.

- 5% of all trading fees are stored in the JustIn YIELD Emergency Savings which helps sustain and back the staking rewards provided by the positive rebase. JES keeps holders safe by: To Prevent price instability Enabling long-term sustainability and future growth of the JustIn YIELD Protocol which is allocated for marketing is what allows JustinYIELDfinance Token to hold the aforementioned promise. Tokens will be swapped into BNB and will be sent to a marketing wallet per transaction. This way, Justin YIELD finance Token will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.
- The additional component included under the sustainability section is a liquidity fee of 4% when buying and selling, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.
- 2.5% of all Freedom Protocol tokens traded are burnt in the Blackhole. The more that is traded, the more get put into the Blackhole causing the Blackhole to grow in size, larger and larger through self-fulfilling auto-compounding which in return acts to reduce the circulating supply of Autocompounding Autostaking Protocol and keeps the JustinYIELDfinance stable.profound ease, simplicity, and accessibility upon all JustinYIELDfinance holders. Each transaction,
- 2.5% of all JustinYIELDfinance tokens traded are sent in the Blackhole. The more that is supply of JustinYIELDfinance and keeps the JustinYIELDfinance stable.

# Target market and the concept

## Target Market

- Anyone who’s interested in the Crypto space with long-term investment plans.
- Anyone who’s ready to earn a passive income by holding tokens.
- Anyone who’s interested in trading tokens.
- Anyone who’s ready in receiving automatic staking and compound rewards every 15
  minutes.
- Anyone who’s interested in receiving fixed interest of 0.026033267**%** per 15 Minutes and 915,501% per year.
- Anyone who’s interested in taking part with the future plans of the JIY token.
- Anyone who’s interested in making financial transactions with any other party using JIY as the currency.

## Core Concept

## Reward mechanism

5% of all trading fees are stored in the JustIn YIELD Emergency Savings fund which helps sustain and back the staking rewards provided by the positive rebase.

JIY fund which is a separate wallet in the ecosystem. The JIY fund uses an algorithm that backs the Rebase Rewards and is supported by a portion of the buy and sell trading fees that accrue in the wallet.

In simple terms, the staking rewards (rebase rewards) which are distributed every 15 Minutes at a rate of 0.026033267**%** are backed by the JIY parameter, thus ensuring a high and stable interest rate to JIY holders.

## Sustainable mechanism

JES: JES is the Short for JustIn YIELD Emergency Savings, which is a separate wallet in JustIn YIELD's Ecosystem. JES uses an algorithm that backs the Rebase Rewards and is supported by a portion of the buy and sell trading fees that accrue in the JES wallet. 5% of all trading fees are stored in the JustIn YIELD Emergency Savings which helps sustain and back the staking rewards provided by the positive rebase. JES keeps holders safe by: To Prevent price instability Enabling long-term sustainability and future growth of the JustIn YIELD Protocol

Treasury: The Treasury plays an Important role in the JustIn YIELD protocol.
Its main function is to support the sustainability and growth of the JustIn YIELD protocol.

The treasury may also be used to fund new JustIn YIELD products, services, and projects that will expand and provide more value to the JustIn YIELD community.
In addition, it will serve to provide funds for marketing, developer payments and salaries.

## Potential to grow with score points

| | | |
|---|---|---|
| 1. | Project efficiency | 10/10 |
| 2. | Project uniqueness | 9/10 |
| 3 | Information quality | 9/10 |
| 4 | Service quality | 10/10 |
| 5 | System quality | 9/10 |
| 6 | Impact on the community | 9/10 |
| 7 | Impact on the business | 9/10 |
| 8 | Preparing for the future | 9/10 |
| Total Points | | 9.25/10 |

# Audit Goals

The focus of the audit was to verify that the Smart Contract System is secure, resilient and working according to the specifications. The audit activities can be grouped in the following three categories:

Security

Identifying security related issues within each contract and the system of contract.

Sound Architecture

Evaluation of the architecture of this system through the lens of established smart contract best practices and general software best practices.

Code Correctness and Quality

A full review of the contract source code. The primary areas of focus include:

- Accuracy
- Readability
- Sections of code with high complexity
- Quantity and quality of test coverage

Issue Categories

Every issue in this report was assigned a severity level from the following:

High level severity issues

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium level severity issues

Issues on this level could potentially bring problems and should eventually be fixed.

Low level severity issues

Issues on this level are minor details and warnings that can remain unfixed but would be better fixed at some point in the future.

# Contract details

Token contract details for 20th May 2022

| | |
|---|---|
| Contract name | JustIn Yield Finance |
| Contract address | 0x0E5E088242c81267F5668909C17ddeC33C56B026 |
| Token supply | 500,000 |
| Token ticker | JIY |
| Decimals | 5 |
| Token holders | 1 |
| Transaction count | 1 |
| Auto liquidity receiver | 0xa68a6efce9d43b3426cf46ff1ecdad88f94f28ee |
| BlackHole | 0x27278367c265a32f12c4c5fa1b71b7f6fbde89eb |
| Justin Yield insurance fund receiver | 0x05a58da5f02c75272c1eb1bcfb0b94d276d3f0e7 |
| Treasury Receiver | 0x8034497001d3baa6737aded9cfffe58746b3e7ad |
| Contract deployer address | 0xb968f4ed35dbe0686cc645eebf10a3737118d425 |
| Contract’s current owner address | 0x8034497001d3baa6737aded9cfffe58746b3e7ad |
| Pancakeswap V2 pair | 0x105948172ab58096585b258c2a5d627a271f1f9c |

## Contract code function details

| No | Category | Item | Result |
|---|---|---|---|
| 1 | Coding conventions | ERC20 Token standards | pass |
| | | compile errors | pass |
| | | Compiler version security | pass |
| | | visibility specifiers | pass |
| | | Gas consumption | pass |
| | | SafeMath features | pass |
| | | Fallback usage | pass |
| | | tx.origin usage | pass |
| | | deprecated items | pass |
| | | Redundant code | pass |
| | | Overriding variables | pass |
| 2 | Function call audit | Authorization of function call | pass |
| | | Low level function (call/delegate call) security | pass |
| | | Returned value security | pass |
| | | Selfdestruct function security | pass |
| 3 | Business security | Access control of owners | pass |
| | | Business logics | pass |
| | | Business implementations | pass |
| 4 | Integer overflow/underflow | | pass |
| 5 | Reentrancy | | pass |
| 6 | Exceptional reachable state | | pass |
| 7 | Transaction ordering dependence | | pass |
| 8 | Block properties dependence | | pass |
| 9 | Pseudo random number generator (PRNG) | | pass |
| 10 | DoS (Denial of Service) | | pass |
| 11 | Token vesting implementation | | pass |

|  |  |  |  |
|---|---|---|---|
| 12 | Fake deposit |  | pass |
| 13 | Event security |  | pass |

# Contract description table

The below table represents the summary of the contracts and methods in the token contract. We scanned the whole contract and listed down all the Interfaces, functions, and implementations with their visibility and mutability.

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | Function Name | Visibility | Mutability | Modifiers |
| | | | | |
| **SafeMathInt** | **Library** | | | |
| L | mul | Internal | | |
| L | div | Internal | | |
| L | sub | Internal | | |
| L | add | Internal | | |
| L | abs | Internal | | |
| | | | | |
| **SafeMath** | **Library** | | | |
| L | add | Internal | | |
| L | sub | Internal | | |
| L | sub | Internal | | |
| L | mul | Internal | | |
| L | div | Internal | | |
| L | div | Internal | | |
| L | mod | Internal | | |
| | | | | |

| | | | | |
|---|---|---|---|---|
| IERC20 | Interface | | | |
| └ | totalSupply | External | | NO |
| └ | balanceOf | External | | NO |
| └ | allowance | External | | NO |
| └ | transfer | External | # | NO |
| └ | approve | External | # | NO |
| └ | transferFrom | External | # | NO |
| | | | | |
| IPancakeSwap Pair | Interface | | | |
| └ | name | External | | NO |
| └ | symbol | External | | NO |
| └ | decimals | External | | NO |
| └ | totalSupply | External | | NO |
| └ | balanceOf | External | | NO |
| └ | allowance | External | | NO |
| └ | approve | External | # | NO |
| └ | transfer | External | # | NO |
| └ | transferFrom | External | # | NO |
| └ | DOMAIN_SEPARATOR | External | | NO |
| └ | PERMIT_TYPEHASH | External | | NO |
| └ | nonces | External | | NO |

| | | | | |
|---|---|---|---|---|
| L | permit | External | # | NO |
| L | MINIMUM_LIQUIDITY | External | | NO |
| L | factory | External | | NO |
| L | token0 | External | | NO |
| L | token1 | External | | NO |
| L | getReserves | External | | NO |
| L | price0CumulativeLast | External | | NO |
| L | price1CumulativeLast | External | | NO |
| L | kLast | External | | NO |
| L | mint | External | # | NO |
| L | burn | External | # | NO |
| L | swap | External | # | NO |
| L | skim | External | # | NO |
| L | sync | External | # | NO |
| L | initialize | External | # | NO |
| | | | | |
| **IPancakeSwap Router** | **Interface** | | | |
| L | factory | External | | NO |
| L | WETH | External | | NO |
| L | addLiquidity | External | # | NO |
| L | addLiquidityETH | External | ($) | NO |

| | | | | |
|---|---|---|---|---|
| └ | removeLiquidity | External | # | NO |
| └ | removeLiquidityETH | External | # | NO |
| └ | removeLiquidityWithPermit | External | # | NO |
| └ | removeLiquidityETHWithPermit | External | # | NO |
| └ | swapExactTokensForTokens | External | # | NO |
| └ | swapTokensForExactTokens | External | # | NO |
| └ | swapExactETHForTokens | External | ($) | NO |
| └ | swapTokensForExactETH | External | # | NO |
| └ | swapExactTokensForETH | External | # | NO |
| └ | swapETHForExactTokens | External | ($) | NO |
| └ | quote | External | | NO |
| └ | getAmountOut | External | | NO |
| └ | getAmountIn | External | | NO |
| └ | getAmountsOut | External | | NO |
| └ | getAmountsIn | External | | NO |
| └ | removeLiquidityETHSupportingFeeOnTransferTokens | External | # | NO |
| └ | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External | # | NO |
| └ | swapExactTokensForTokensSupportingFeeOnTransferTokens | External | # | NO |
| └ | swapExactETHForTokensSupportingFeeOnTransferTokens | External | ($) | NO |
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External | # | NO |

| | | | | |
|---|---|---|---|---|
| | | | | |
| **IPancakeSwap Factory** | **Interface** | | | |
| L | feeTo | External | | NO |
| L | feeToSetter | External | | NO |
| L | getPair | External | | NO |
| L | allPairs | External | | NO |
| L | allPairsLength | External | | NO |
| L | createPair | External | # | NO |
| L | setFeeTo | External | # | NO |
| L | setFeeToSetter | External | # | NO |
| | | | | |
| **Ownable** | **Implementation** | | | |
| L | | Public | # | NO |
| L | owner | Public | | NO |
| L | isOwner | Public | | NO |
| L | renounceOwnership | Public | # | onlyOwner |
| L | transferOwnership | Public | # | onlyOwner |
| L | _transferOwnership | Internal | # | |
| | | | | |
| **ERC20Detailed** | **Implementation** | **IERC20** | | |
| L | | Public | # | NO |
| L | name | Public | | NO |

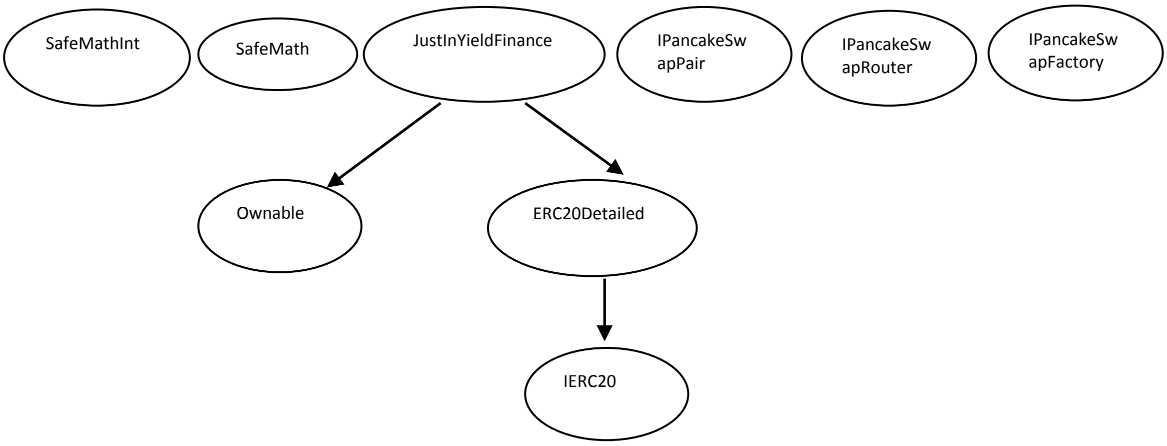
| | | | | |
|---|---|---|---|---|
| L | symbol | Public | | NO |
| L | decimals | Public | | NO |
| | | | | |
| **JustIn Yield Finance** | **Implementation** | **ERC20Detailed, Ownable** | | |
| L | | Public | # | ERC20Detailed<br>Ownable |
| L | rebase | Internal | # | |
| L | transfer | External | # | validRecipient |
| L | transferFrom | External | # | validRecipient |
| L | _basicTransfer | Internal | # | |
| L | _transferFrom | Internal | # | |
| L | takeFee | Internal | # | |
| L | addLiquidity | Internal | # | swapping |
| L | swapBack | Internal | # | swapping |
| L | withdrawAllToVault | External | # | swapping<br>onlyOwner |
| L | shouldTakeFee | Internal | | |
| L | shouldRebase | Internal | | |
| L | shouldAddLiquidity | Internal | | |
| L | shouldSwapBack | Internal | | |
| L | setAutoRebase | External | # | onlyOwner |

|  |  |  |  |  |
|---|---|---|---|---|
| L | changeTradingStatus | External | # | onlyOwner |
| L | setAutoAddLiquidity | External | # | onlyOwner |
| L | allowance | External |  | NO |
| L | decreaseAllowance | External | # | NO |
| L | increaseAllowance | External | # | NO |
| L | approve | External | # | NO |
| L | checkFeeExempt | External |  | NO |
| L | getCirculatingSupply | Public |  | NO |
| L | isNotInSwap | External |  | NO |
| L | manualSync | External | # | NO |
| L | setFeeReceivers | External | # | onlyOwner |
| L | changeFees | External | # | onlyOwner |
| L | getLiquidityBacking | Public |  | NO |
| L | setWhitelist | External | # | onlyOwner |
| L | setBotBlacklist | External | # | onlyOwner |
| L | setPairAddress | Public | # | onlyOwner |
| L | setLP | External | # | onlyOwner |
| L | totalSupply | External |  | NO |
| L | balanceOf | External |  | NO |
| L | isContract | Internal |  |  |
| L |  | External | ($) | NO |

Legend

| Symbol | Meaning |
|---|---|
| # | Function can modify state |
| ($) | Function is payable |

## Inheritance Hierarchy



## Automated Audit

### Remix Compiler Warnings

It throws warnings by Solidity's compiler. If it encounters any errors the contract cannot be compiled and deployed. No issues found.

### Number of issues per severity

| Critical | High | Medium | Low | Note |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |

### Issues Checking Status

| № | Issue description. | Checking |
|---|---|---|
| 1 | Compiler warnings. | Passed |
| 2 | Race conditions and Reentrancy. Cross-function race | Passed |
| 3 | Possible delays in data delivery. | Passed |
| 4 | Oracle calls. | Passed |
| 5 | Front running. | Passed |
| 6 | Timestamp dependence. | Passed |
| 7 | Integer Overflow and Underflow. | Passed |
| 8 | DoS with Revert. | Passed |
| 9 | DoS with block gas limit. | Passed |
| 10 | Methods execution permissions. | Passed |
| 11 | Economy model. | Passed |
| 12 | The impact of the exchange rate on the logic. | Passed |
| 13 | Private user data leaks. | Passed |
| 14 | Malicious Event log. | Passed |
| 15 | Scoping and Declarations. | Passed |
| 16 | Uninitialized storage pointers. | Passed |
| 17 | Arithmetic accuracy. | Passed |

| | | |
|---|---|---|
| 18 | Design Logic. | Passed |
| 19 | Cross-function race conditions. | Passed |
| 20 | Safe Zeppelin module. | Passed |
| 21 | Fallback function security. | Passed |

## Critical Severity Issues

No critical severity issues found.

## High Severity Issues

No high severity issues found.

## Medium Severity Issues

No medium severity issues found.

## Low Severity Issues

No Low severity issues found.

**Recommendation**:

Check that the excluded array length is not too big.

## Owner privileges (In the period when the ownership is notrenounced)

- 499 renounceOwnership
- 504 transferOwnership
- 870 The owner can withdraw tokens in contract by swapping them into BNB
- 919 The owner can enable/disable rebase
- 928 The owner can enable/disable auto liquidity adding
- 1010 The owner can change all fee receiver wallet address
- 1030 The owner can exclude wallet from fees (once excluded cannot include them again)
- 1034 The owner can add/remove contracts from blacklist
- 1045 The owner can change pair address and pair contract
- 1049 setLP

# Concluding Summary

- Owner cannot set fees
- No mint function found
- Owner cannot set max tx amount
- Owner cannot pause trading

Smart contracts do not contain any severe Security issues!

Smart contract functional Status: PASSED

Number of risk issues: 0

Solidity code functional issue level: PASSED

Centralization risk correlated to the active owner: LOW

Smart contract active ownership: YES

Note: Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner. The analysis of the contract does not give complete security and includes only the analysis that is indicated in the report. We do not analyze locked tokens or LP tokens, the presence of KYC in other companies, and so on. Also, our audit is not a recommendation for investment. All responsibility for the loss of investment lies with you!