# RugFreeCoins Audit

# AVA Finance Token

## Smart Contract Security Audit

## May 21, 2022

# Contents

# Audit details

**Audited project**
AVA Finance Token

**Contract Address**

0x7D40659a0fC177071b0d2f68E41A057ac417B42D

**Client contact**

AVA Finance Team

**Blockchain**

Polygon chain

**Project website**

https://avafinance.net/

# Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

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 Rugfreecoins and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Rugfreecoins) owe no duty of care towards you or any other person, nor does Rugfreecoins 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 Rugfreecoins 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, Rugfreecoins hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Rugfreecoins, 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.

# Background

Rugfreecoins was commissioned by the AVA Finance Team to perform an audit of the smart contract.

**https://polygonscan.com/token/0x7d40659a0fc177071b0d2f68e41a057ac417b42d**

The focus of this audit is to verify that the smart contract is secure, resilient, and working according to the specifications.

The information in this report should be used to understand the risk exposure of the smart contract, project feasibility, and long-term sustainability, and as a guide to improving the security posture of the smart contract by remediating the issues that were identified.

.

# About the project

AVA Finance is a token built on the Polygon Smart Chain that is with an innovative investment use case the main purpose of which is to seek out constant revenue sources, and **auto-staking protocol backed by Defi 3.0 yield farming** on Polygon. AVA Finance token will bring an unparallel, fixed APY of **9,999%** while imposing profound ease, simplicity, and accessibility upon all AVA Finance holders. Each transaction, purchase incurs a 16% fee, and sale incurs an 18% fee.

**Features**

- **3%** of the fee is directed to the AVA Buyback Assurance which helps sustain and back the Staking Rewards provided by the Positive Rebase.

- The **sustainability fee of 1% for marketing, 1% for development, and 4% for the owner** is what allows AVA Finance Token to hold the aforementioned promise. Tokens will be swapped into USDT and will be sent to those three wallets. This way, AVA 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 5%**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.
- The **rewards** will be distributed in USDT among every holder proportional to how many tokens each individual holds in values of a **2% tax fee.**

# Roadmap

**Q2 2022**

- Ava found
- Website launch
- Presale launch
- CMC/CG listing
- Website upgrade
- 10000+ holders

**Q3 2022**

- Certik audit
- CEX listing
- Launch Ava dex
- AVA launchpad
- 50000+ holders
- Partnership announcement

**Q4 2022**

- Token migration 90% for max supply
- Launch Ava laugh and earn Metaverse
- Cross-chain development
- 100,000 holders
- Major CEX listing
- Our final goal develop Ava blockchain upgrade to Dex.

# Tokenomics

**16% fee when buying**

- 1% of trade goes to the treasury in USDT
- 1% of trade goes for the development fund in USDT
- 2% of trade goes for USDT rewards among holders
- 3% of trade goes to AVA Finance Assurance fund in USDT
- 4% trade goes for the owner in USDT
- 5% of trade goes to the liquidity pool.

**18% fee when selling**

- Fee is proportionally increased by 2% from all selling taxes when selling.

# 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 30 minutes.
- Anyone who's interested in receiving fixed interest in 30 minutes and 9,999% per year.
- Anyone who's interested in taking part in the future plans of the AVA Finance token.
- Anyone who's interested in making financial transactions with any other party using AVA Finance Token as the currency.

## Core concept

### Reward mechanism

3% of all trading fees are stored in the AVA Finance Token fund, which helps sustain and back the staking rewards provided by the positive rebase.

AVA Finance Token fund which is a separate wallet in the ecosystem. The AVA Finance Token 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 30 minutes at a rate of 9,999% per year are backed by the AVA Finance Token parameter, thus ensuring a high and stable interest rate for AVA Finance Token holders.

### USDT Rewards

2% of each transaction gets converted to USDT and is split amongst all holders. Holders will be eligible to receive tokens in each transaction and rewards are proportional to how many tokens each individual holds.

### Sustainable mechanism

**The sustainability fee of 4% for the owner, 1% for the treasury, and 1% for the development** is what allows AVA Finance Token to promote the token and use funds to further the development of the platform. Tokens will be swapped into USDT and will be sent to all these 3 wallets. This way, AVA Finance Token will have access to the funds without selling tokens as the traditional way, which will enable them to consume funds without hurting the project.

**The liquidity fee of 5%,** is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

# Potential to grow with score points

| | | |
|---|---|---|
| 1. | Project efficiency | 9/10 |
| 2. | Project uniqueness | 8/10 |
| 3 | Information quality | 8/10 |
| 4 | Service quality | 8/10 |
| 5 | System quality | 9/10 |
| 6 | Impact on the community | 8/10 |
| 7 | Impact on the business | 8/10 |
| 8 | Preparing for the future | 7/10 |
| 9 | Smart contract security | 10/10 |
| 10 | Smart contract functionality assessment | 10/10 |
| Total Points | | **8.5/10** |

# Contract details

## Token contract details for 21st May 2022

| | |
|---|---|
| Contract name | AVA Finance |
| Contract address | 0x7D40659a0fC177071b0d2f68E41A057ac417B42D |
| Token supply | 50,000 |
| Token ticker | AVA |
| Decimals | 5 |
| Token holders | 1 |
| Transaction count | 1 |
| Distributor | 0x7d2c746f12bea79c98ee855ad3ebb97ed8b1b07e |
| Dev Fee Receiver | 0xa7324068a17eb6e8435d98cd6df999170e751838 |
| Freedom Fee Receiver | 0x8d09d53bab287778b490b9592551a78eff4ee32f |
| Treasury Receiver | 0x33ed9dae90df2b098729b6a20aafa5c168ef9524 |
| Owner Fee Receiver | 0x33ed9dae90df2b098729b6a20aafa5c168ef9524 |
| Contract deployer address | 0xE296869532f6A38E372444cbcBf7a745179d6378 |
| Contract’s current owner address | 0x33ed9dae90df2b098729b6a20aafa5c168ef9524 |

# Contract code function details

| No | Category | Item | Result |
|---|---|---|---|
| 1 | Coding conventions | BRC20 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 | | |
|---|---|---|---|---|
| └ | Function Name | Visibility | Mutability | Modifiers |
| | | | | |
| **AVA** | **Implementation** | **IERC20, Ownable** | | |
| └ | | Public ❗ | 🛑 | NO❗ |
| └ | name | Public ❗ | | NO❗ |
| └ | symbol | Public ❗ | | NO❗ |
| └ | decimals | Public ❗ | | NO❗ |
| └ | rebase | Internal 🔒 | 🛑 | |
| └ | transfer | External ❗ | 🛑 | validRecipient |
| └ | transferFrom | External ❗ | 🛑 | validRecipient |
| └ | _basicTransfer | Internal 🔒 | 🛑 | |
| └ | _transferFrom | Internal 🔒 | 🛑 | |
| └ | takeFee | Internal 🔒 | 🛑 | |
| └ | swapAndLiquify | Private 🔐 | 🛑 | |
| └ | swapTokensForEth | Private 🔐 | 🛑 | |
| └ | addLiquidity | Private 🔐 | 🛑 | |
| └ | swapBack | Internal 🔒 | 🛑 | swapping |

| | | | | |
|---|---|---|---|---|
| └ | swapTokensForTokens | Private 🔐 | 🛑 | |
| └ | shouldTakeFee | Internal 🔒 | | |
| └ | shouldRebase | Internal 🔒 | | |
| └ | shouldSwapBack | Internal 🔒 | | |
| └ | setAutoRebase | External ❗️ | 🛑 | onlyOwner |
| └ | setAutoAddLiquidity | External ❗️ | 🛑 | onlyOwner |
| └ | allowance | External ❗️ | | NO❗️ |
| └ | enableTrading | Public ❗️ | 🛑 | onlyOwner |
| └ | decreaseAllowance | External ❗️ | 🛑 | NO❗️ |
| └ | increaseAllowance | External ❗️ | 🛑 | NO❗️ |
| └ | approve | External ❗️ | 🛑 | NO❗️ |
| └ | checkFeeExempt | External ❗️ | | NO❗️ |
| └ | setDistributionCriteria | External ❗️ | 🛑 | onlyOwner |
| └ | updateBuyFees | Public ❗️ | 🛑 | onlyOwner |
| └ | updateSellFees | Public ❗️ | 🛑 | onlyOwner |
| └ | updateSwapPercentages | Public ❗️ | 🛑 | onlyOwner |
| └ | getHolderDetails | Public ❗️ | | NO❗️ |
| └ | getLastProcessedIndex | Public ❗️ | | NO❗️ |
| └ | getNumberOfTokenHolders | Public ❗️ | | NO❗️ |
| └ | totalDistributedRewards | Public ❗️ | | NO❗️ |
| └ | claimProcess | Public ❗️ | 🛑 | NO❗️ |

| | | | | |
|---|---|---|---|---|
| └ | setDistributorSettings | External ❗ | 🛑 | onlyOwner |
| └ | getCirculatingSupply | Public ❗ | | NO❗ |
| └ | isNotInSwap | External ❗ | | NO❗ |
| └ | manualSync | External ❗ | 🛑 | NO❗ |
| └ | setFeeReceivers | External ❗ | 🛑 | onlyOwner |
| └ | setIsDividendExempt | External ❗ | 🛑 | onlyOwner |
| └ | getLiquidityBacking | Public ❗ | | NO❗ |
| └ | setWhitelist | External ❗ | 🛑 | onlyOwner |
| └ | setBotBlacklist | External ❗ | 🛑 | onlyOwner |
| └ | setLP | External ❗ | 🛑 | onlyOwner |
| └ | totalSupply | External ❗ | | NO❗ |
| └ | balanceOf | Public ❗ | | NO❗ |
| └ | isContract | Internal 🔒 | | |
| └ | | External ❗ | 💵 | NO❗ |
| | | | | |
| **Ownable** | **Implementation** | **Context** | | |
| └ | | Public ❗ | 🛑 | NO❗ |
| └ | owner | Public ❗ | | NO❗ |
| └ | renounceOwnership | Public ❗ | 🛑 | onlyOwner |
| └ | transferOwnership | Public ❗ | 🛑 | onlyOwner |
| └ | _transferOwnership | Internal 🔒 | 🛑 | |

| | | | | |
|---|---|---|---|---|
| **IERC20** | **Interface** | | | |
| └ | totalSupply | External ❗ | | NO❗ |
| └ | balanceOf | External ❗ | | NO❗ |
| └ | transfer | External ❗ | 🛑 | NO❗ |
| └ | allowance | External ❗ | | NO❗ |
| └ | approve | External ❗ | 🛑 | NO❗ |
| └ | transferFrom | External ❗ | 🛑 | NO❗ |
| | | | | |
| **SafeMath** | **Library** | | | |
| └ | tryAdd | Internal 🔒 | | |
| └ | trySub | Internal 🔒 | | |
| └ | tryMul | Internal 🔒 | | |
| └ | tryDiv | Internal 🔒 | | |
| └ | tryMod | Internal 🔒 | | |
| └ | add | Internal 🔒 | | |
| └ | sub | Internal 🔒 | | |
| └ | mul | Internal 🔒 | | |
| └ | div | Internal 🔒 | | |
| └ | mod | Internal 🔒 | | |
| └ | sub | Internal 🔒 | | |
| └ | div | Internal 🔒 | | |

| | | | | |
|---|---|---|---|---|
| └ | mod | Internal 🔒 | | |
| | | | | |
| **SignedSafeMath** | **Library** | | | |
| └ | mul | Internal 🔒 | | |
| └ | div | Internal 🔒 | | |
| └ | sub | Internal 🔒 | | |
| └ | add | Internal 🔒 | | |
| | | | | |
| **IUniswapV2 Router02** | **Interface** | **IUniswap V2 Router01** | | |
| └ | removeLiquidityETHSupportingFeeOnTransferTokens | External ❗ | 🛑 | NO❗ |
| └ | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ❗ | 🛑 | NO❗ |
| └ | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ❗ | 🛑 | NO❗ |
| └ | swapExactETHForTokensSupportingFeeOnTransferTokens | External ❗ | 💵 | NO❗ |
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗ | 🛑 | NO❗ |
| | | | | |
| **IUniswapV2Factory** | **Interface** | | | |
| └ | feeTo | External ❗ | | NO❗ |
| └ | feeToSetter | External ❗ | | NO❗ |
| └ | getPair | External ❗ | | NO❗ |
| └ | allPairs | External ❗ | | NO❗ |
| └ | allPairsLength | External ❗ | | NO❗ |

|  |  |  |  |  |
|---|---|---|---|---|
| └ | createPair | External ❗ | 🛑 | NO❗ |
| └ | setFeeTo | External ❗ | 🛑 | NO❗ |
| └ | setFeeToSetter | External ❗ | 🛑 | NO❗ |
|  |  |  |  |  |
| **IUniswapV2Pair** | **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❗ |
| └ | permit | External ❗ | 🛑 | NO❗ |
| └ | MINIMUM_LIQUIDITY | External ❗ |  | NO❗ |
| └ | factory | External ❗ |  | NO❗ |
| └ | token0 | External ❗ |  | NO❗ |

|  |  |  |  |  |
|---|---|---|---|---|
| └ | token1 | External ❗ |  | NO❗ |
| └ | getReserves | External ❗ |  | NO❗ |
| └ | price0CumulativeLast | External ❗ |  | NO❗ |
| └ | price1CumulativeLast | External ❗ |  | NO❗ |
| └ | kLast | External ❗ |  | NO❗ |
| └ | mint | External ❗ | 🛑 | NO❗ |
| └ | burn | External ❗ | 🛑 | NO❗ |
| └ | swap | External ❗ | 🛑 | NO❗ |
| └ | skim | External ❗ | 🛑 | NO❗ |
| └ | sync | External ❗ | 🛑 | NO❗ |
| └ | initialize | External ❗ | 🛑 | NO❗ |
|  |  |  |  |  |
| **IDividend Distributor** | **Interface** |  |  |  |
| └ | setDistributionCriteria | External ❗ | 🛑 | NO❗ |
| └ | setShare | External ❗ | 🛑 | NO❗ |
| └ | deposit | External ❗ | 🛑 | NO❗ |
| └ | process | External ❗ | 🛑 | NO❗ |
| └ | purge | External ❗ | 🛑 | NO❗ |
|  |  |  |  |  |
| **Dividend Distributor** | **Implementation** | **IDividend Distributor** |  |  |
| └ |  | Public ❗ | 🛑 | NO❗ |

|  |  |  |  |  |
|---|---|---|---|---|
| └ |  | External ❗ | 💵 | NO❗ |
| └ | setDistributionCriteria | External ❗ | 🛑 | onlyToken |
| └ | purge | External ❗ | 🛑 | onlyToken |
| └ | setShare | External ❗ | 🛑 | onlyToken |
| └ | deposit | External ❗ | 🛑 | onlyToken |
| └ | process | External ❗ | 🛑 | onlyToken |
| └ | shouldDistribute | Internal 🔒 |  |  |
| └ | distributeDividend | Internal 🔒 | 🛑 |  |
| └ | claimDividend | External ❗ | 🛑 | NO❗ |
| └ | getUnpaidEarnings | Public ❗ |  | NO❗ |
| └ | getHolderDetails | Public ❗ |  | NO❗ |
| └ | getCumulativeDividends | Internal 🔒 |  |  |
| └ | getLastProcessedIndex | External ❗ |  | NO❗ |
| └ | getNumberOfTokenHolders | External ❗ |  | NO❗ |
| └ | getShareHoldersList | External ❗ |  | NO❗ |
| └ | totalDistributedRewards | External ❗ |  | NO❗ |
| └ | addShareholder | Internal 🔒 | 🛑 |  |
| └ | removeShareholder | Internal 🔒 | 🛑 |  |
|  |  |  |  |  |
| **Context** | **Implementation** |  |  |  |
| └ | _msgSender | Internal 🔒 |  |  |

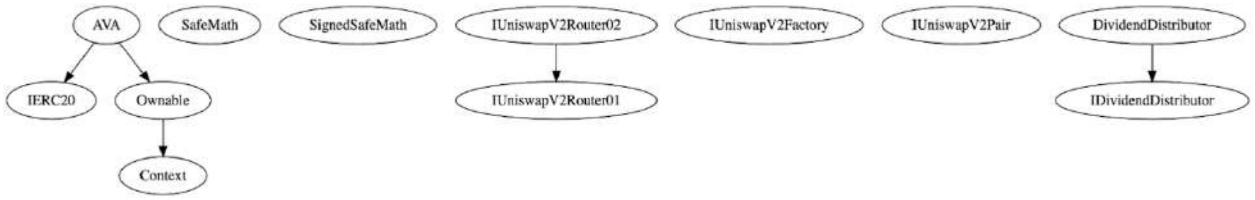
| | | | | |
|---|---|---|---|---|
| └ | _msgData | Internal 🔒 | | |
| | | | | |
| **IUniswapV2 Router01** | **Interface** | | | |
| └ | factory | External ❗ | | NO❗ |
| └ | WETH | External ❗ | | NO❗ |
| └ | addLiquidity | External ❗ | 🛑 | NO❗ |
| └ | addLiquidityETH | External ❗ | 💵 | NO❗ |
| └ | removeLiquidity | 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❗ |

Legend

| Symbol | Meaning |
|---|---|
| 🛑 | Function can modify state |
| 💵 | Function is payable |

## Inheritance Hierarchy



# Security issue checking status

- **High severity issues**

  No High severity issues found.

- **Medium severity issues**

  No medium severity issues found

- **Low severity issues**

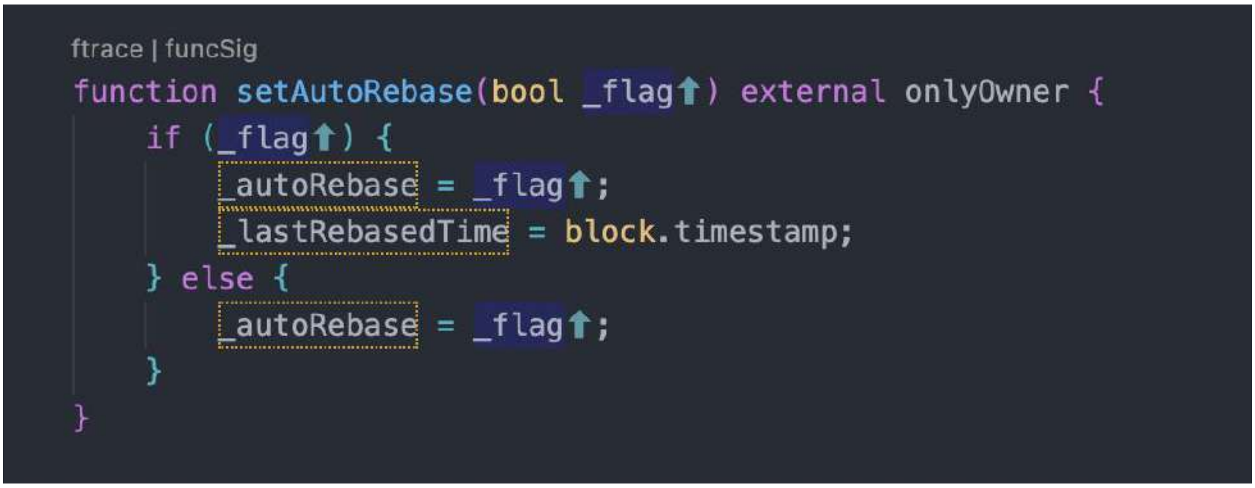
  No low severity issues found

- **Centralization risk**

  No Centralization issues found

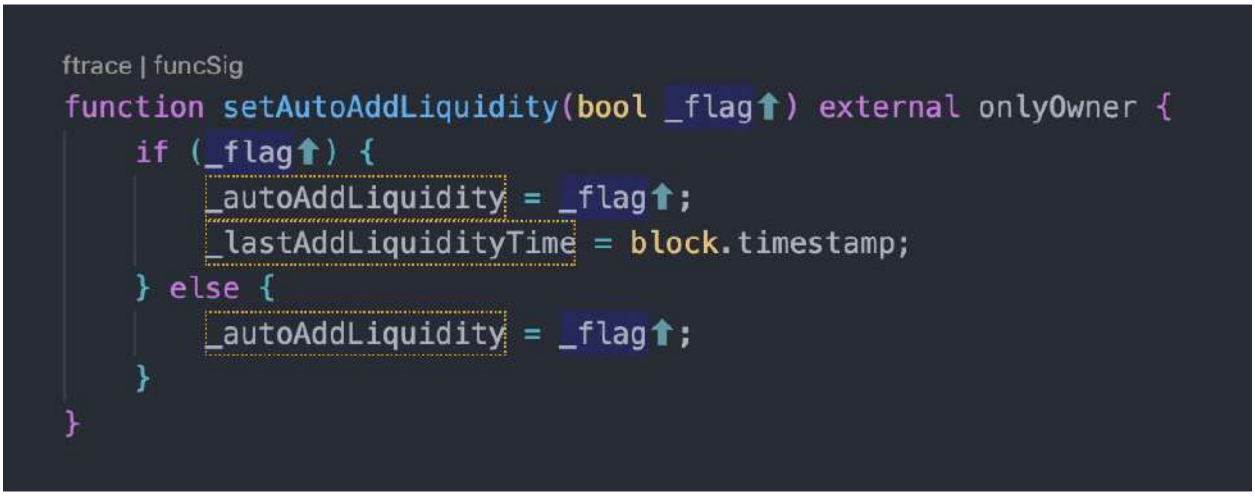
# Owner privileges

- ❖ Owner can enable/disable auto rebase



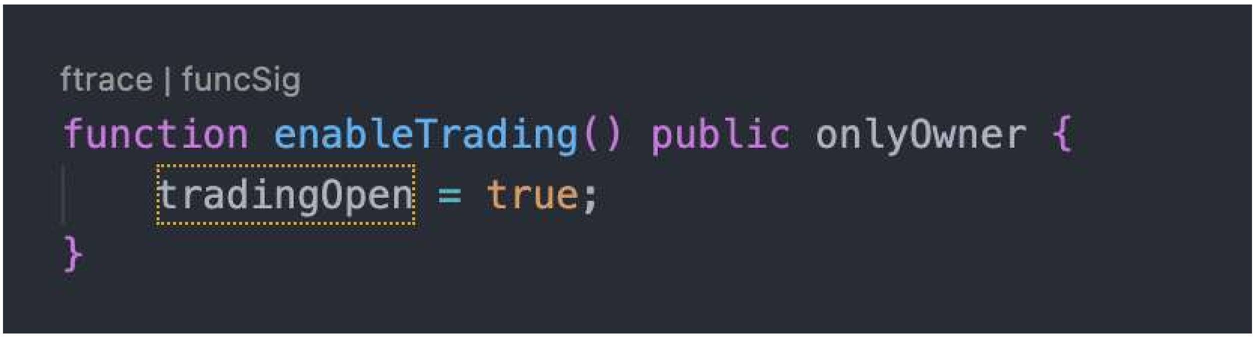
```
ftrace | funcSig
function setAutoRebase(bool _flag⬆) external onlyOwner {
    if (_flag⬆) {
        _autoRebase = _flag⬆;
        _lastRebasedTime = block.timestamp;
    } else {
        _autoRebase = _flag⬆;
    }
}
```

- ❖ Owner can enable/disable adding liquidity automatically



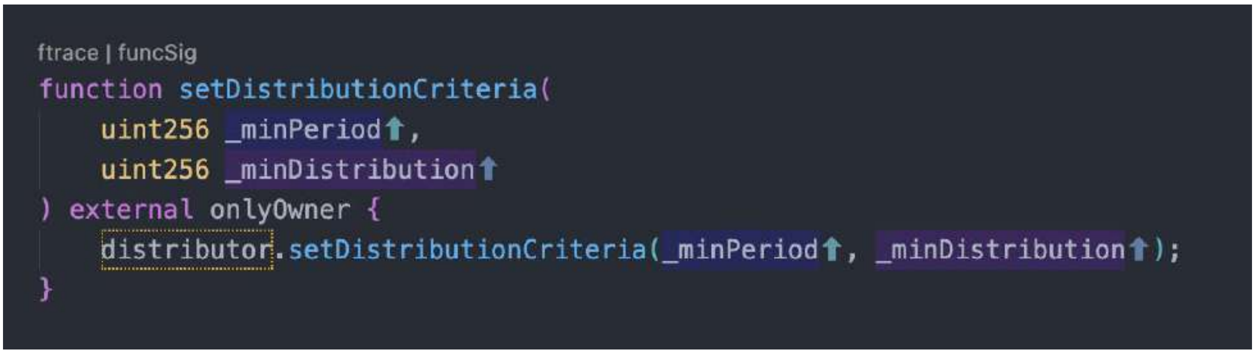
```
ftrace | funcSig
function setAutoAddLiquidity(bool _flag⬆) external onlyOwner {
    if (_flag⬆) {
        _autoAddLiquidity = _flag⬆;
        _lastAddLiquidityTime = block.timestamp;
    } else {
        _autoAddLiquidity = _flag⬆;
    }
}
```

- ❖ Owner can enable trading, once enabled cannot disable again



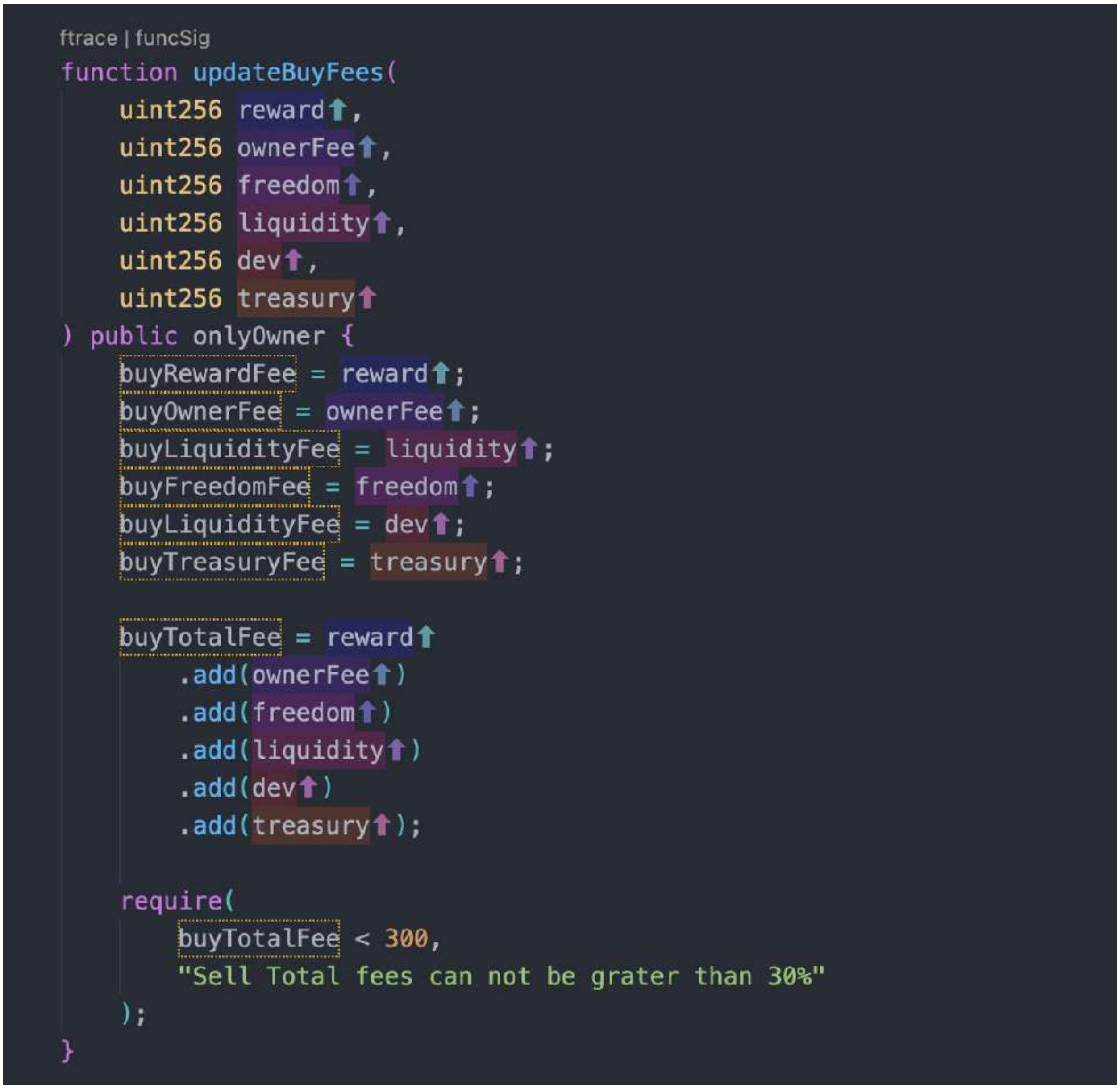
```
ftrace | funcSig
function enableTrading() public onlyOwner {
    tradingOpen = true;
}
```

- Owner can change minimum distribution time and minimum distribution amount



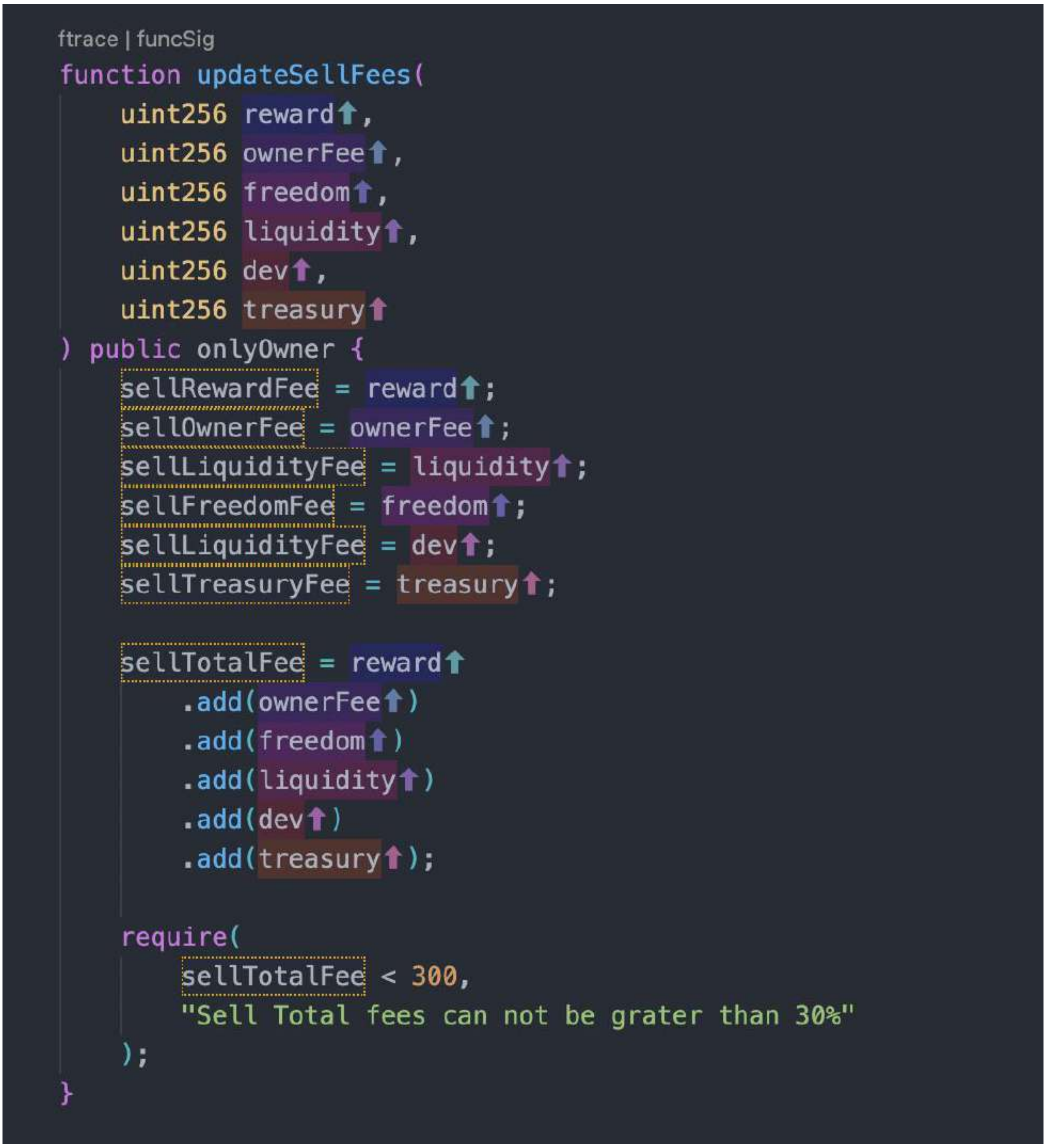
```
ftrace | funcSig
function setDistributionCriteria(
    uint256 _minPeriod,
    uint256 _minDistribution
) external onlyOwner {
    distributor.setDistributionCriteria(_minPeriod, _minDistribution);
}
```

- Owner can change all buy fees maximum up to 30%



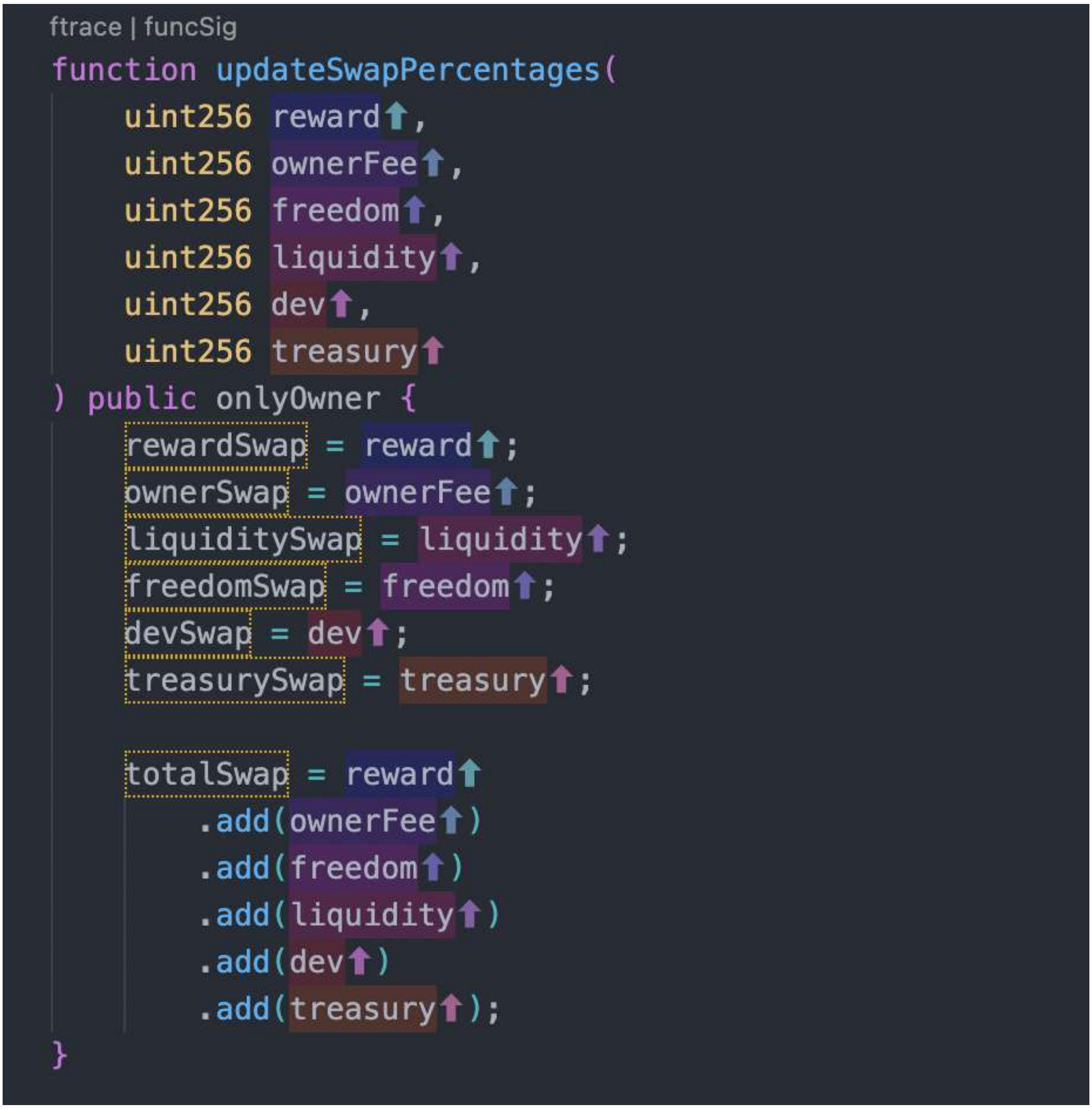
```
ftrace | funcSig
function updateBuyFees(
    uint256 reward,
    uint256 ownerFee,
    uint256 freedom,
    uint256 liquidity,
    uint256 dev,
    uint256 treasury
) public onlyOwner {
    buyRewardFee = reward;
    buyOwnerFee = ownerFee;
    buyLiquidityFee = liquidity;
    buyFreedomFee = freedom;
    buyLiquidityFee = dev;
    buyTreasuryFee = treasury;

    buyTotalFee = reward
        .add(ownerFee)
        .add(freedom)
        .add(liquidity)
        .add(dev)
        .add(treasury);

    require(
        buyTotalFee < 300,
        "Sell Total fees can not be grater than 30%"
    );
}
```

- Owner can change all sell fees maximum up to 30%



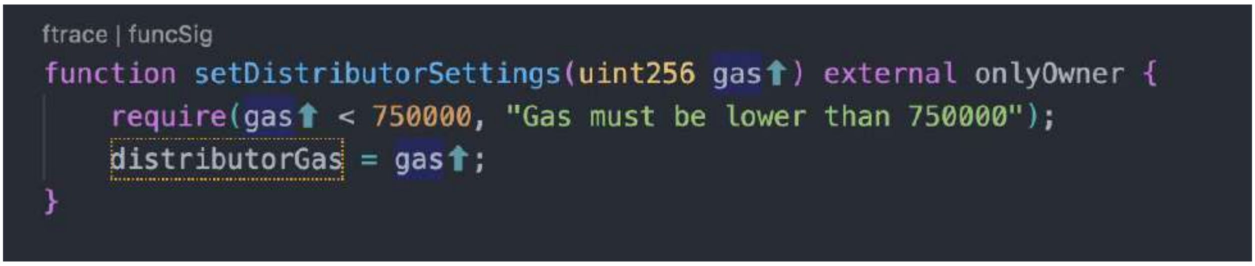
```
ftrace | funcSig
function updateSellFees(
    uint256 reward,
    uint256 ownerFee,
    uint256 freedom,
    uint256 liquidity,
    uint256 dev,
    uint256 treasury
) public onlyOwner {
    sellRewardFee = reward;
    sellOwnerFee = ownerFee;
    sellLiquidityFee = liquidity;
    sellFreedomFee = freedom;
    sellLiquidityFee = dev;
    sellTreasuryFee = treasury;

    sellTotalFee = reward
        .add(ownerFee)
        .add(freedom)
        .add(liquidity)
        .add(dev)
        .add(treasury);

    require(
        sellTotalFee < 300,
        "Sell Total fees can not be grater than 30%"
    );
}
```

- Owner can change all swap percentages



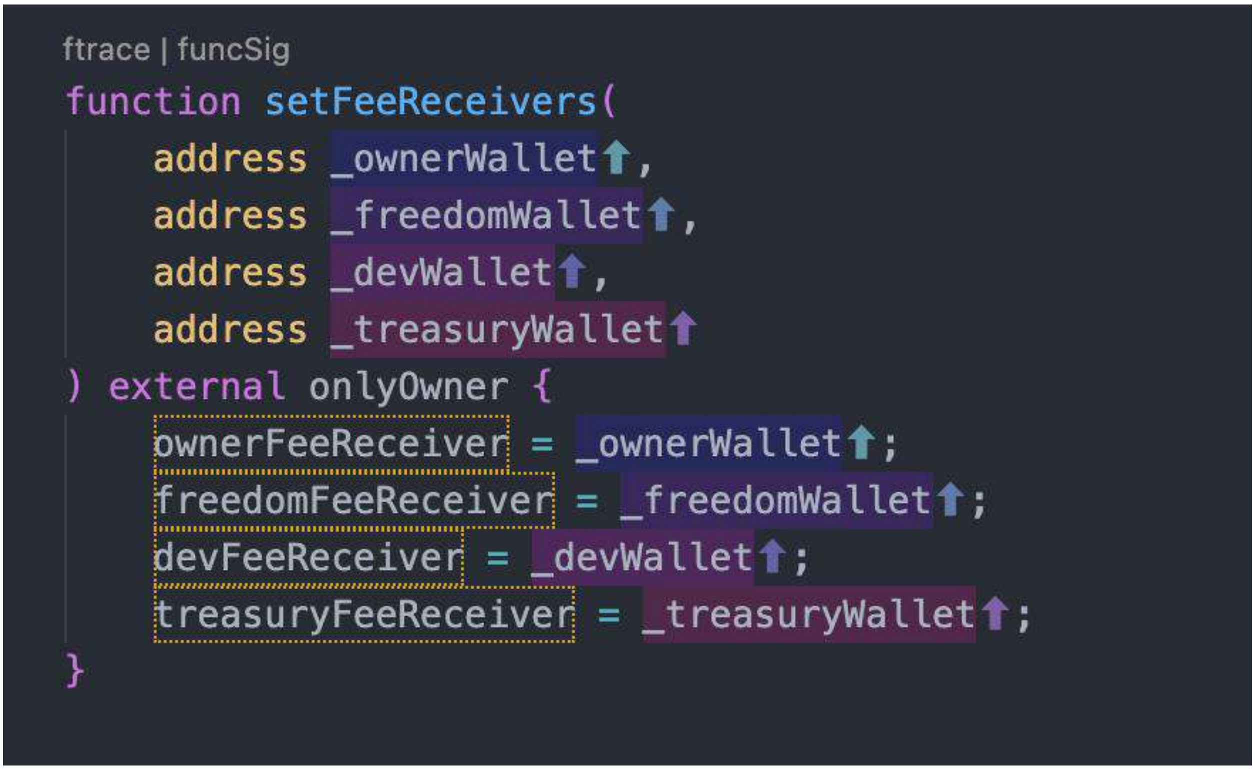
```
function updateSwapPercentages(
    uint256 reward,
    uint256 ownerFee,
    uint256 freedom,
    uint256 liquidity,
    uint256 dev,
    uint256 treasury
) public onlyOwner {
    rewardSwap = reward;
    ownerSwap = ownerFee;
    liquiditySwap = liquidity;
    freedomSwap = freedom;
    devSwap = dev;
    treasurySwap = treasury;

    totalSwap = reward
        .add(ownerFee)
        .add(freedom)
        .add(liquidity)
        .add(dev)
        .add(treasury);
}
```

- Owner can change maximum gas to use distribute dividend



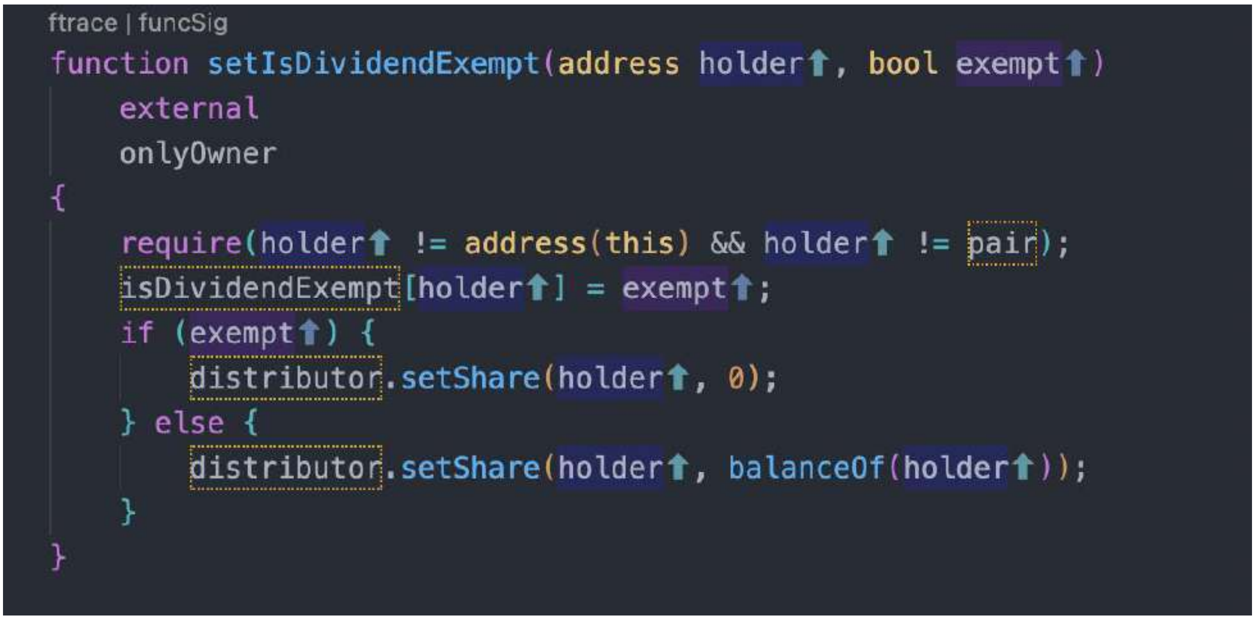
```
function setDistributorSettings(uint256 gas) external onlyOwner {
    require(gas < 750000, "Gas must be lower than 750000");
    distributorGas = gas;
}
```

- ❖ Owner can change all fee receiver address



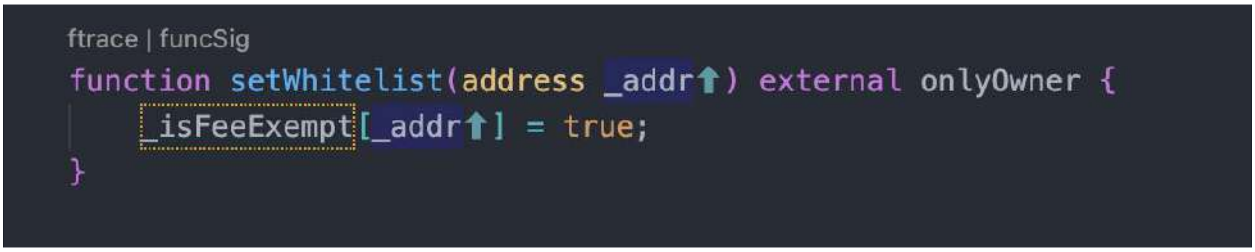
```
ftrace | funcSig
function setFeeReceivers(
        address _ownerWallet,
        address _freedomWallet,
        address _devWallet,
        address _treasuryWallet
) external onlyOwner {
        ownerFeeReceiver = _ownerWallet;
        freedomFeeReceiver = _freedomWallet;
        devFeeReceiver = _devWallet;
        treasuryFeeReceiver = _treasuryWallet;
}
```

- ❖ Owner can include/exclude wallets from dividend



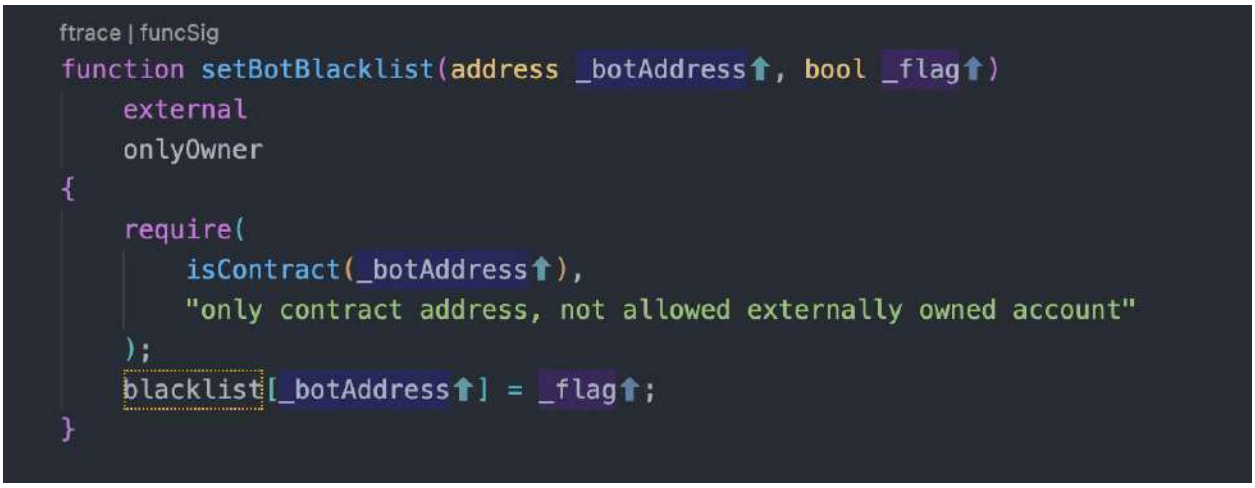
```
ftrace | funcSig
function setIsDividendExempt(address holder, bool exempt)
    external
    onlyOwner
{
    require(holder != address(this) && holder != pair);
    isDividendExempt[holder] = exempt;
    if (exempt) {
        distributor.setShare(holder, 0);
    } else {
        distributor.setShare(holder, balanceOf(holder));
    }
}
```

- Owner can exclude wallets from fees



```
ftrace | funcSig
function setWhitelist(address _addr) external onlyOwner {
    _isFeeExempt[_addr] = true;
}
```

- Owner can blacklist contracts from this contract



```
ftrace | funcSig
function setBotBlacklist(address _botAddress, bool _flag)
    external
    onlyOwner
{
    require(
        isContract(_botAddress),
        "only contract address, not allowed externally owned account"
    );
    blacklist[_botAddress] = _flag;
}
```

# Audit conclusion

RugFreeCoins team has performed in-depth testings, line by line manual code review, and automated audit of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, manipulations, and hacks. According to the smart contract audit.

Smart contract functional Status: **PASSED**

Number of risk issues: **0**

Solidity code functional issue level: **PASSED**

Number of owner privileges: **12**

Centralization risk correlated to the active owner: **LOW**

Smart contract active ownership: **YES**