# RugFreeCoins Audit

# Meme Royale Token
# Smart Contract Security Audit

December 9th 2022

# Contents

# Audit details

**Audited project**
Meme Royale Token

**Contract Address**

0xD701a4668ff5c3488aEdceF7BD6647701394cC8b

**Client contact**

Meme Royale Team

**Blockchain**

Binance smart chain

**Project website**

https://www.thememeroyale.com

# 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.

# Overview

✅ No mint function found, the owner cannot mint tokens after initial deployment.

✅ The owner can set a max sell limit minimum upto 0.5%

✅ The owner can’t pause trading.

✅ The owner can't set fees over 20%

✅ The owner can set a max wallet limit minimum upto 1%

❌ The owner can claim the contract’s balance of its own token.

❌ The owner can blacklist wallets.(with pink antibot)

# Background

Rugfreecoins was commissioned by the Meme Royale Team to perform an audit of the smart contract.

https://bscscan.com/token/0xD701a4668ff5c3488aEdceF7BD6647701394cC8b

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 smart contract's security posture by remediating the identified issues.

.

# ROADMAP

**Phase 1**

- Develope a solid contract for future Certik Audit
- Building community
- Marketing via Tg and twitter crypto influencers
- Presale
- Partner with crypto influencers as brand ambassador
- Game Developement started
- NFT started

**Phase 2**

- Marketing
- Building a strong #Defi infrastructure
- MRoyaleSwap
- $ROYALE staking pool
- ETH bridge
- Dogechain bridge
- CRO bridge
- BRISE bridge

**Phase 3**

- Marketing
- NFT Launchpad development
- NFT marketplace Development
- Certik Audit

**Phase 4**

- Marketing
- Game Beta Release
- Lottery for Beta tester get free Legendary items
- Hold Twitch Beta preview and Testing
- Community give away

# Tokenomics

**8% when buying & selling**

- 3% of trade goes to marketing wallet in BNB.
- 2% of trade goes to holders pockets in native tokens.
- 2% of trade goes to the liquidity pool.
- 1% of trade goes to burn wallet.

# 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 interested in taking part of the Meme Royale ecosystem.
- Anyone who’s interested in taking part in the future plans of Meme Royale Token.
- Anyone who’s interested in making financial transactions with any other party using Meme Royale Token as the currency.

# Potential to grow with score points

| | | |
|---|---|---|
| 1. | Project efficiency | 9/10 |
| 2. | Project uniqueness | 9/10 |
| 3 | Information quality | 9/10 |
| 4 | Service quality | 9/10 |
| 5 | System quality | 9/10 |
| 6 | Impact on the community | 9/10 |
| 7 | Impact on the business | 9/10 |
| 8 | Preparing for the future | 8/10 |
| 9 | Smart contract security | 10/10 |
| 10 | Smart contract functionality assessment | 10/10 |
| Total Points | | **9.1/10** |

# Contract details

## Token contract details for 9th of December 2022

| | |
|---|---|
| Contract name | MemeRoyale |
| Contract address | 0xD701a4668ff5c3488aEdceF7BD6647701394cC8b |
| Token supply | 100,000,000,000,000,000 |
| Token ticker | ROYALE |
| Decimals | 18 |
| Token holders | 1 |
| Transaction count | 1 |
| Marketing wallet | 0xa3cc159798ff27048d6be496f9571f36d50b2145 |
| Dividend tracker | 0x51dda43de237b8800b6f68041415a048bc18f738 |
| Contract deployer address | 0x72468F1B96e5c91b035B10f6D9A6b365980837fc |
| Contract’s current owner address | 0x0f59012595c820289c07d73b2c25aef70db779e2 |

# 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 |
| | | Self-destruct function security | pass |
| 3 | Business security | Access control of owners | |
| | | 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 |
| | | | | |
| **MemeRoyale** | **Implementation** | **BEP20, Auth** | | |
| └ | | Public ❗ | 🛑 | Auth<br>BEP20 |
| └ | | External ❗ | 💵 | NO ❗ |
| └ | _transfer | Internal 🔒 | 🛑 | |
| └ | _preTransferCheck | Internal 🔒 | 🛑 | |
| └ | _takeFee | Internal 🔒 | 🛑 | |
| └ | _shouldTakeFee | Internal 🔒 | | |
| └ | _shouldSwapBack | Internal 🔒 | | |
| └ | _swapBack | Internal 🔒 | 🛑 | |
| └ | _addLiquidity | Private 🔐 | 🛑 | |
| └ | _swapForNative | Internal 🔒 | 🛑 | |
| └ | getCirculatingSupply | Public ❗ | | NO ❗ |
| └ | updateClaimWait | External ❗ | 🛑 | onlyOwner |
| └ | getClaimWait | External ❗ | | NO ❗ |
| └ | updateMinimumTokenBalanceForDividends | External ❗ | 🛑 | onlyOwner |

| | | | | |
|---|---|---|---|---|
| └ | getMinimumTokenBalanceForDividends | External ❗ | | NO❗ |
| └ | claim | External ❗ | 🛑 | NO❗ |
| └ | processDividendTracker | External ❗ | 🛑 | NO❗ |
| └ | getAccountDividendsInfo | External ❗ | | NO❗ |
| └ | setReflectionExempt | External ❗ | 🛑 | authorized |
| └ | isReflectionExempt | External ❗ | | NO❗ |
| └ | setAntiBot | External ❗ | 🛑 | authorized |
| └ | setSellCooldown | External ❗ | 🛑 | authorized |
| └ | setSellCooldownExempt | External ❗ | 🛑 | authorized |
| └ | setSwapEnabled | External ❗ | 🛑 | authorized |
| └ | setSwapThreshold | External ❗ | 🛑 | authorized |
| └ | setFeeExempt | External ❗ | 🛑 | authorized |
| └ | setMaxWalletSize | External ❗ | 🛑 | authorized |
| └ | setMaxWalletExempt | External ❗ | 🛑 | authorized |
| └ | setMaxSellTxSize | External ❗ | 🛑 | authorized |
| └ | setMaxSellTxSizeExempt | External ❗ | 🛑 | authorized |
| └ | setExempt | External ❗ | 🛑 | authorized |
| └ | setAmm | External ❗ | 🛑 | authorized |
| └ | setMarketingWallet | External ❗ | 🛑 | authorized |
| └ | setFees | External ❗ | 🛑 | authorized |

| | | | | |
|---|---|---|---|---|
| └ | rescueBalance | External ❗ | 🛑 | authorized |
| └ | rescueOwnBalance | External ❗ | 🛑 | authorized |
| └ | rescueNativeBalance | External ❗ | 🛑 | authorized |
| | | | | |
| **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 🔒 | | |
| | | | | |
| **Auth** | **Implementation** | | | |
| └ | | Public ❗ | 🛑 | NO ❗ |
| └ | authorize | Public ❗ | 🛑 | onlyOwner |

| | | | | |
|---|---|---|---|---|
| └ | unauthorize | Public ❗ | 🛑 | onlyOwner |
| └ | isOwner | Public ❗ | | NO ❗ |
| └ | owner | Public ❗ | | NO ❗ |
| └ | isAuthorized | Public ❗ | | NO ❗ |
| └ | transferOwnership | Public ❗ | 🛑 | onlyOwner |
| | | | | |
| **BEP20** | **Implementation** | **Context, IBEP20, IBEP20 Metadata** | | |
| └ | | Public ❗ | 🛑 | NO ❗ |
| └ | name | Public ❗ | | NO ❗ |
| └ | symbol | Public ❗ | | NO ❗ |
| └ | decimals | Public ❗ | | NO ❗ |
| └ | totalSupply | Public ❗ | | NO ❗ |
| └ | balanceOf | Public ❗ | | NO ❗ |
| └ | transfer | Public ❗ | 🛑 | NO ❗ |
| └ | allowance | Public ❗ | | NO ❗ |
| └ | approve | Public ❗ | 🛑 | NO ❗ |
| └ | transferFrom | Public ❗ | 🛑 | NO ❗ |
| └ | increaseAllowance | Public ❗ | 🛑 | NO ❗ |
| └ | decreaseAllowance | Public ❗ | 🛑 | NO ❗ |
| └ | _transfer | Internal 🔒 | 🛑 | |

| | | | | |
|---|---|---|---|---|
| L | _mint | Internal 🔒 | 🛑 | |
| L | _burn | Internal 🔒 | 🛑 | |
| L | _approve | Internal 🔒 | 🛑 | |
| L | _spendAllowance | Internal 🔒 | 🛑 | |
| L | _beforeTokenTransfer | Internal 🔒 | 🛑 | |
| L | _afterTokenTransfer | Internal 🔒 | 🛑 | |
| | | | | |
| **IBEP20** | **Interface** | | | |
| L | totalSupply | External ❗ | | NO❗ |
| L | balanceOf | External ❗ | | NO❗ |
| L | transfer | External ❗ | 🛑 | NO❗ |
| L | allowance | External ❗ | | NO❗ |
| L | approve | External ❗ | 🛑 | NO❗ |
| L | transferFrom | External ❗ | 🛑 | NO❗ |
| | | | | |
| **IBEP20 Metadata** | **Interface** | **IBEP20** | | |
| L | name | External ❗ | | NO❗ |
| L | symbol | External ❗ | | NO❗ |
| L | decimals | External ❗ | | NO❗ |
| | | | | |
| **IDEXFactory** | **Interface** | | | |
| L | createPair | External ❗ | 🛑 | NO❗ |

| | | | | |
|---|---|---|---|---|
| | | | | |
| **IDEXRouter** | **Interface** | | | |
| └ | factory | External ❗ | | NO❗ |
| └ | WETH | External ❗ | | NO❗ |
| └ | addLiquidity | External ❗ | 🛑 | NO❗ |
| └ | addLiquidityETH | External ❗ | 💵 | NO❗ |
| └ | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ❗ | 🛑 | NO❗ |
| └ | swapExactETHForTokensSupportingFeeOnTransferTokens | External ❗ | 💵 | NO❗ |
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗ | 🛑 | NO❗ |
| | | | | |
| **IPinkAntiBot** | **Interface** | | | |
| └ | setTokenOwner | External ❗ | 🛑 | NO❗ |
| └ | onPreTransferCheck | External ❗ | 🛑 | NO❗ |
| | | | | |
| **SafeBEP20** | **Library** | | | |
| └ | safeTransfer | Internal 🔒 | 🛑 | |
| └ | safeTransferFrom | Internal 🔒 | 🛑 | |
| └ | safeApprove | Internal 🔒 | 🛑 | |
| └ | safeIncreaseAllowance | Internal 🔒 | 🛑 | |
| └ | safeDecreaseAllowance | Internal 🔒 | 🛑 | |
| └ | _callOptionalReturn | Private 🔐 | 🛑 | |
| | | | | |

| **MemeRoyale Dividend Tracker** | **Implementation** | **Auth, Dividend PayingToken** | | |
|---|---|---|---|---|
| └ | | Public ❗ | 🛑 | Auth Dividend Paying Token |
| └ | _transfer | Internal 🔒 | | |
| └ | withdrawDividend | Public ❗ | | NO ❗ |
| └ | distributeDividends | External ❗ | 🛑 | onlyOwner |
| └ | excludeFromDividends | External ❗ | 🛑 | onlyOwner |
| └ | includeInDividends | External ❗ | 🛑 | onlyOwner |
| └ | isExcludedFromDividends | Public ❗ | | NO ❗ |
| └ | updateClaimWait | External ❗ | 🛑 | onlyOwner |
| └ | updateMinimumTokenBalanceForDividends | External ❗ | 🛑 | onlyOwner |
| └ | getLastProcessedIndex | External ❗ | | NO ❗ |
| └ | getNumberOfTokenHolders | External ❗ | | NO ❗ |
| └ | getAccount | Public ❗ | | NO ❗ |
| └ | getAccountAtIndex | Public ❗ | | NO ❗ |
| └ | canAutoClaim | Private 🔐 | | |
| └ | setBalance | External ❗ | 🛑 | onlyOwner |
| └ | process | Public ❗ | 🛑 | NO ❗ |
| └ | processAccount | Public ❗ | 🛑 | onlyOwner |
| | | | | |

| | | | | |
|---|---|---|---|---|
| **Context** | **Implementation** | | | |
| └ | _msgSender | Internal 🔒 | | |
| └ | _msgData | Internal 🔒 | | |
| | | | | |
| **Address** | **Library** | | | |
| └ | isContract | Internal 🔒 | | |
| └ | sendValue | Internal 🔒 | 🛑 | |
| └ | functionCall | Internal 🔒 | 🛑 | |
| └ | functionCall | Internal 🔒 | 🛑 | |
| └ | functionCallWithValue | Internal 🔒 | 🛑 | |
| └ | functionCallWithValue | Internal 🔒 | 🛑 | |
| └ | functionStaticCall | Internal 🔒 | | |
| └ | functionStaticCall | Internal 🔒 | | |
| └ | functionDelegateCall | Internal 🔒 | 🛑 | |
| └ | functionDelegateCall | Internal 🔒 | 🛑 | |
| └ | verifyCallResult | Internal 🔒 | | |
| | | | | |
| **Dividend PayingToken** | **Implementation** | **ERC20 Mintable, Idividend PayingToken, Idividend PayingToken Optional** | | |
| └ | | Public ❗ | 🛑 | ERC20Mintable |

| | | | | |
|---|---|---|---|---|
| └ | distribute | Internal 🔒 | 🛑 | |
| └ | withdrawDividend | Public ❗ | 🛑 | NO ❗ |
| └ | _withdrawDividendOfUser | Internal 🔒 | 🛑 | |
| └ | dividendOf | Public ❗ | | NO ❗ |
| └ | withdrawableDividendOf | Public ❗ | | NO ❗ |
| └ | withdrawnDividendOf | Public ❗ | | NO ❗ |
| └ | accumulativeDividendOf | Public ❗ | | NO ❗ |
| └ | _transfer | Internal 🔒 | 🛑 | |
| └ | _mint | Internal 🔒 | 🛑 | |
| └ | _burn | Internal 🔒 | 🛑 | |
| └ | _setBalance | Internal 🔒 | 🛑 | |
| | | | | |
| **Iterable Mapping** | **Library** | | | |
| └ | get | Public ❗ | | NO ❗ |
| └ | getIndexOfKey | Public ❗ | | NO ❗ |
| └ | getKeyAtIndex | Public ❗ | | NO ❗ |
| └ | size | Public ❗ | | NO ❗ |
| └ | set | Public ❗ | 🛑 | NO ❗ |
| └ | remove | Public ❗ | 🛑 | NO ❗ |
| | | | | |

| | | | | |
|---|---|---|---|---|
| **ERC20Preset MinterPauser** | **Implementation** | **Context, AccessControl Enumerable, ERC20 Burnable, ERC20 Pausable** | | |
| L | | Public ❗ | 🛑 | ERC20 |
| L | mint | Public ❗ | 🛑 | NO❗ |
| L | pause | Public ❗ | 🛑 | NO❗ |
| L | unpause | Public ❗ | 🛑 | NO❗ |
| L | _beforeTokenTransfer | Internal 🔒 | 🛑 | |
| | | | | |
| **ERC20 Mintable** | **Implementation** | **Context, AccessControl Enumerable, ERC20 Burnable** | | |
| L | | Public ❗ | 🛑 | ERC20 |
| L | mint | Public ❗ | 🛑 | NO❗ |
| | | | | |
| **SafeMathInt** | **Library** | | | |
| L | mul | Internal 🔒 | | |
| L | div | Internal 🔒 | | |
| L | sub | Internal 🔒 | | |
| L | add | Internal 🔒 | | |
| L | abs | Internal 🔒 | | |
| L | toUint256Safe | Internal 🔒 | | |

| | | | | |
|---|---|---|---|---|
| | | | | |
| **SafeMathUint** | **Library** | | | |
| └ | toInt256Safe | Internal 🔒 | | |
| | | | | |
| **Idividend PayingToken** | **Interface** | | | |
| └ | dividendOf | External ❗ | | NO❗ |
| └ | withdrawDividend | External ❗ | 🛑 | NO❗ |
| | | | | |
| **Idividend PayingToken Optional** | **Interface** | | | |
| └ | withdrawableDividendOf | External ❗ | | NO❗ |
| └ | withdrawnDividendOf | External ❗ | | NO❗ |
| └ | accumulativeDividendOf | External ❗ | | NO❗ |
| | | | | |
| **ERC20** | **Implementation** | **Context, IERC20, IERC20 Metadata** | | |
| └ | | Public ❗ | 🛑 | NO❗ |
| └ | name | Public ❗ | | NO❗ |
| └ | symbol | Public ❗ | | NO❗ |
| └ | decimals | Public ❗ | | NO❗ |
| └ | totalSupply | Public ❗ | | NO❗ |
| └ | balanceOf | Public ❗ | | NO❗ |
| └ | transfer | Public ❗ | 🛑 | NO❗ |

|  |  |  |  |  |
|---|---|---|---|---|
| └ | allowance | Public ❗ |  | NO ❗ |
| └ | approve | Public ❗ | 🛑 | NO ❗ |
| └ | transferFrom | Public ❗ | 🛑 | NO ❗ |
| └ | increaseAllowance | Public ❗ | 🛑 | NO ❗ |
| └ | decreaseAllowance | Public ❗ | 🛑 | NO ❗ |
| └ | _transfer | Internal 🔒 | 🛑 |  |
| └ | _mint | Internal 🔒 | 🛑 |  |
| └ | _burn | Internal 🔒 | 🛑 |  |
| └ | _approve | Internal 🔒 | 🛑 |  |
| └ | _spendAllowance | Internal 🔒 | 🛑 |  |
| └ | _beforeTokenTransfer | Internal 🔒 | 🛑 |  |
| └ | _afterTokenTransfer | Internal 🔒 | 🛑 |  |
|  |  |  |  |  |
| **ERC20 Burnable** | **Implementation** | **Context, ERC20** |  |  |
| └ | burn | Public ❗ | 🛑 | NO ❗ |
| └ | burnFrom | Public ❗ | 🛑 | NO ❗ |
|  |  |  |  |  |
| **ERC20 Pausable** | **Implementation** | **ERC20, Pausable** |  |  |
| └ | _beforeTokenTransfer | Internal 🔒 | 🛑 |  |
|  |  |  |  |  |
| **AccessControl Enumerable** | **Implementation** | **IAccessControlEnumerable, AccessControl** |  |  |

| | | | | |
|---|---|---|---|---|
| └ | supportsInterface | Public ❗ | | NO ❗ |
| └ | getRoleMember | Public ❗ | | NO ❗ |
| └ | getRoleMemberCount | Public ❗ | | NO ❗ |
| └ | _grantRole | Internal 🔒 | 🛑 | |
| └ | _revokeRole | Internal 🔒 | 🛑 | |
| | | | | |
| **IERC20** | **Interface** | | | |
| └ | totalSupply | External ❗ | | NO ❗ |
| └ | balanceOf | External ❗ | | NO ❗ |
| └ | transfer | External ❗ | 🛑 | NO ❗ |
| └ | allowance | External ❗ | | NO ❗ |
| └ | approve | External ❗ | 🛑 | NO ❗ |
| └ | transferFrom | External ❗ | 🛑 | NO ❗ |
| | | | | |
| **IERC20 Metadata** | **Interface** | **IERC20** | | |
| └ | name | External ❗ | | NO ❗ |
| └ | symbol | External ❗ | | NO ❗ |
| └ | decimals | External ❗ | | NO ❗ |
| | | | | |
| **Pausable** | **Implementation** | **Context** | | |
| └ | | Public ❗ | 🛑 | NO ❗ |
| └ | paused | Public ❗ | | NO ❗ |

| | | | | |
|---|---|---|---|---|
| └ | _requireNotPaused | Internal 🔒 | | |
| └ | _requirePaused | Internal 🔒 | | |
| └ | _pause | Internal 🔒 | 🛑 | When NotPaused |
| └ | _unpause | Internal 🔒 | 🛑 | When Paused |
| | | | | |
| **IAccessControlEnumerable** | **Interface** | **IAccessControl** | | |
| └ | getRoleMember | External ❗ | | NO ❗ |
| └ | getRoleMemberCount | External ❗ | | NO ❗ |
| | | | | |
| **AccessControl** | **Implementation** | **Context, Iaccess Control, ERC165** | | |
| └ | supportsInterface | Public ❗ | | NO ❗ |
| └ | hasRole | Public ❗ | | NO ❗ |
| └ | _checkRole | Internal 🔒 | | |
| └ | _checkRole | Internal 🔒 | | |
| └ | getRoleAdmin | Public ❗ | | NO ❗ |
| └ | grantRole | Public ❗ | 🛑 | onlyRole |
| └ | revokeRole | Public ❗ | 🛑 | onlyRole |
| └ | renounceRole | Public ❗ | 🛑 | NO ❗ |
| └ | _setupRole | Internal 🔒 | 🛑 | |
| └ | _setRoleAdmin | Internal 🔒 | 🛑 | |

| | | | | |
|---|---|---|---|---|
| └ | _grantRole | Internal 🔒 | 🛑 | |
| └ | _revokeRole | Internal 🔒 | 🛑 | |
| | | | | |
| **Enumerable Set** | **Library** | | | |
| └ | _add | Private 🔐 | 🛑 | |
| └ | _remove | Private 🔐 | 🛑 | |
| └ | _contains | Private 🔐 | | |
| └ | _length | Private 🔐 | | |
| └ | _at | Private 🔐 | | |
| └ | _values | Private 🔐 | | |
| └ | add | Internal 🔒 | 🛑 | |
| └ | remove | Internal 🔒 | 🛑 | |
| └ | contains | Internal 🔒 | | |
| └ | length | Internal 🔒 | | |
| └ | at | Internal 🔒 | | |
| └ | values | Internal 🔒 | | |
| └ | add | Internal 🔒 | 🛑 | |
| └ | remove | Internal 🔒 | 🛑 | |
| └ | contains | Internal 🔒 | | |
| └ | length | Internal 🔒 | | |
| └ | at | Internal 🔒 | | |

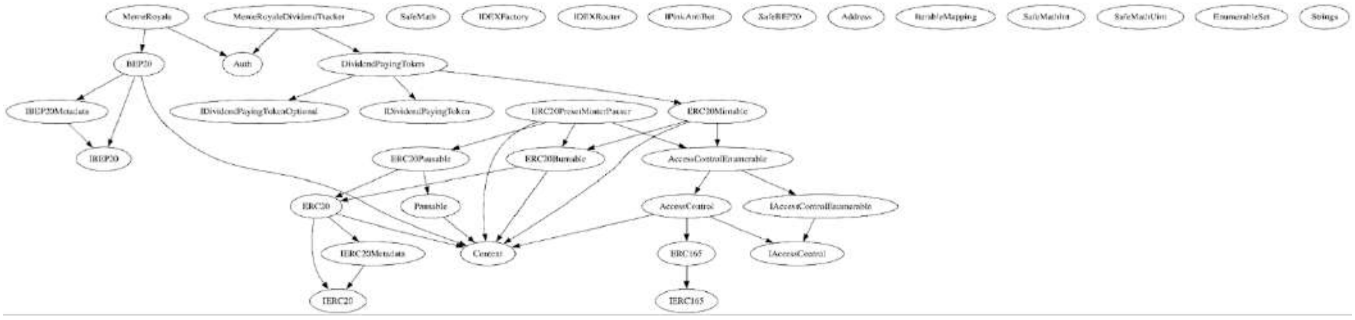
| | | | | |
|---|---|---|---|---|
| └ | values | Internal 🔒 | | |
| └ | add | Internal 🔒 | 🛑 | |
| └ | remove | Internal 🔒 | 🛑 | |
| └ | contains | Internal 🔒 | | |
| └ | length | Internal 🔒 | | |
| └ | at | Internal 🔒 | | |
| └ | values | Internal 🔒 | | |
| | | | | |
| **IaccessControl** | **Interface** | | | |
| └ | hasRole | External ❗ | | NO❗ |
| └ | getRoleAdmin | External ❗ | | NO❗ |
| └ | grantRole | External ❗ | 🛑 | NO❗ |
| └ | revokeRole | External ❗ | 🛑 | NO❗ |
| └ | renounceRole | External ❗ | 🛑 | NO❗ |
| | | | | |
| **Strings** | **Library** | | | |
| └ | toString | Internal 🔒 | | |
| └ | toHexString | Internal 🔒 | | |
| └ | toHexString | Internal 🔒 | | |
| └ | toHexString | Internal 🔒 | | |
| | | | | |

| | | | | |
|---|---|---|---|---|
| **ERC165** | **Implementation** | **IERC165** | | |
| └ | supportsInterface | Public ❗ | | NO❗ |
| | | | | |
| **IERC165** | **Interface** | | | |
| └ | supportsInterface | External ❗ | | NO❗ |

### Legend

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

## Inheritance Hierarchy



# Security issue checking status

### ❖ High severity issues

**Deposit function is missing in reward contract**

**Informed and fixed**

To calculate dividend there should be a deposit function and that function should call from the token contract when it transfers reward tokens to the reward tracker, and inside deposit function it should calculate dividend per share value, currently dividend per share value is not getting calculated. Hence, rewards are not getting processed.

### ❖ Medium severity issues

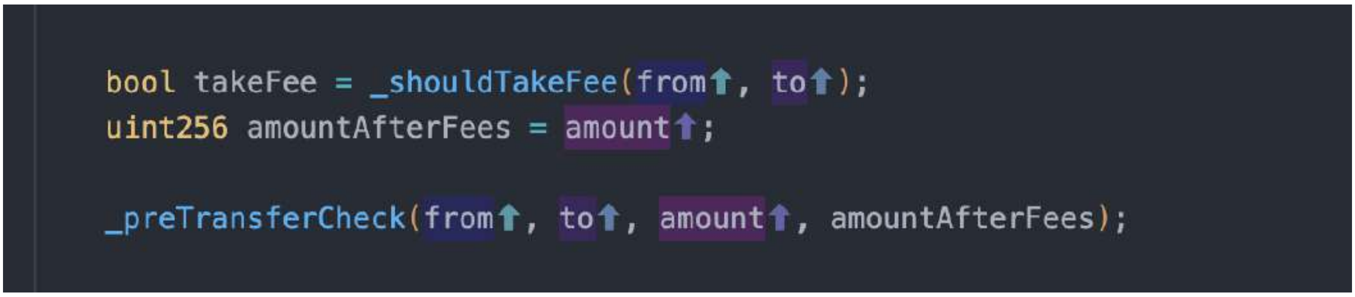
No medium severity issues found

### ❖ Low severity issues

**Duplicating variables**

**Informed and fixed**

When calling _preTransferCheck function it is passing amount and amountAfterFees variables but in this point both amount and amountAfterFees values will always same, no need to pass two variable

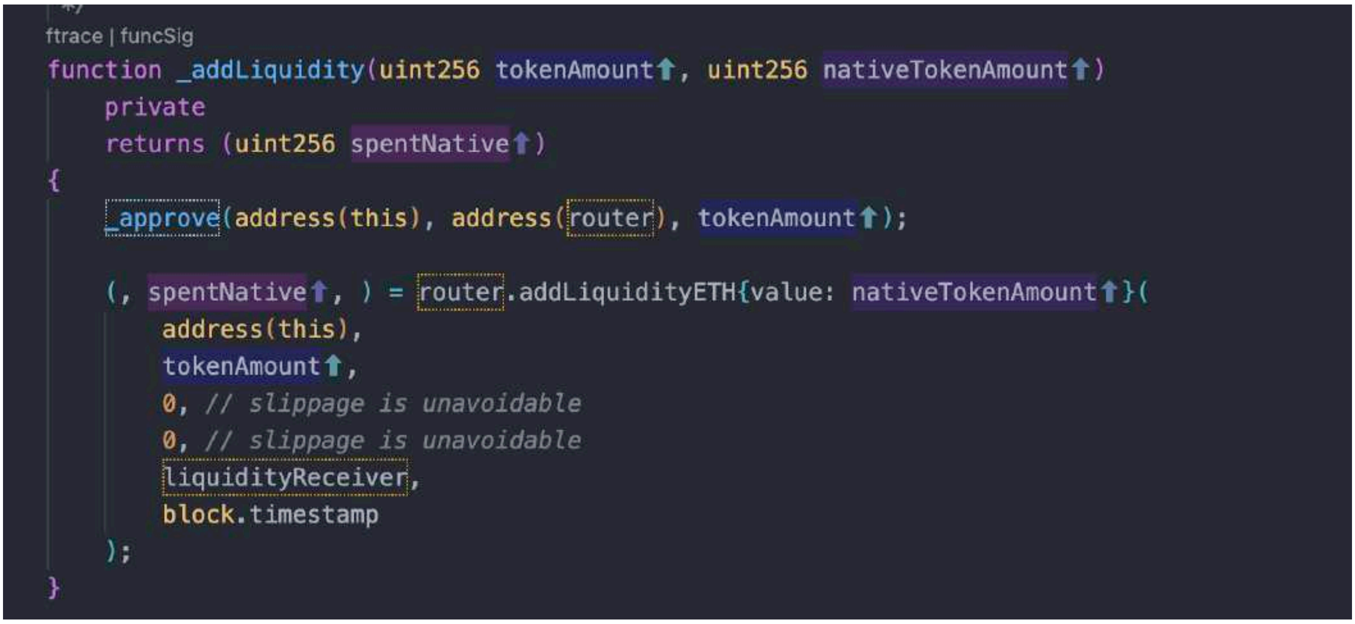


```
bool takeFee = _shouldTakeFee(from↑, to↑);
uint256 amountAfterFees = amount↑;

_preTransferCheck(from↑, to↑, amount↑, amountAfterFees);
```

### ❖ Centralization Risk

**Auto LP does not go to an unreachable address**

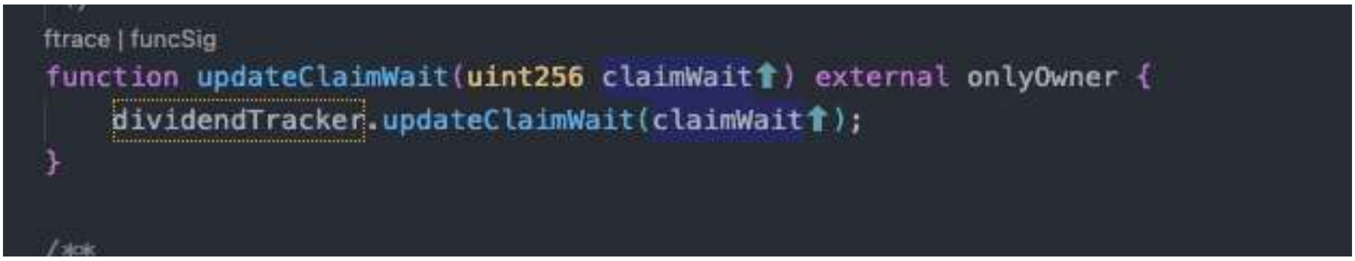
**Informed and fixed**



```
 */
ftrace | funcSig
function _addLiquidity(uint256 tokenAmount, uint256 nativeTokenAmount)
    private
    returns (uint256 spentNative)
{
    _approve(address(this), address(router), tokenAmount);

    (, spentNative, ) = router.addLiquidityETH{value: nativeTokenAmount}(
        address(this),
        tokenAmount,
        0, // slippage is unavoidable
        0, // slippage is unavoidable
        liquidityReceiver,
        block.timestamp
    );
}
```

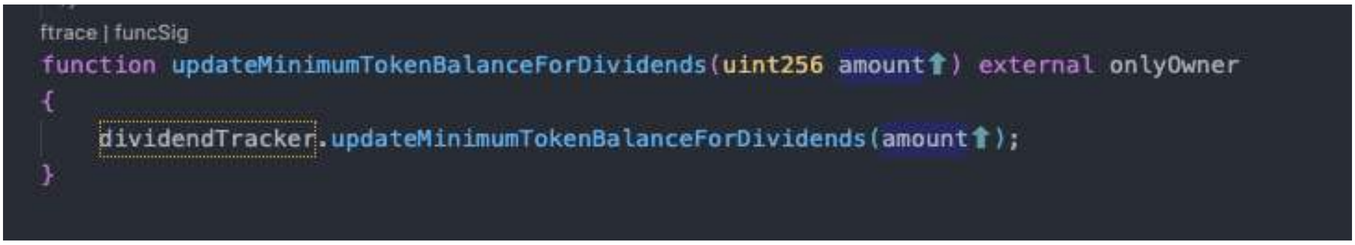
# Owner privileges

- ❖ The owner can change minimum waiting period for receive rewards



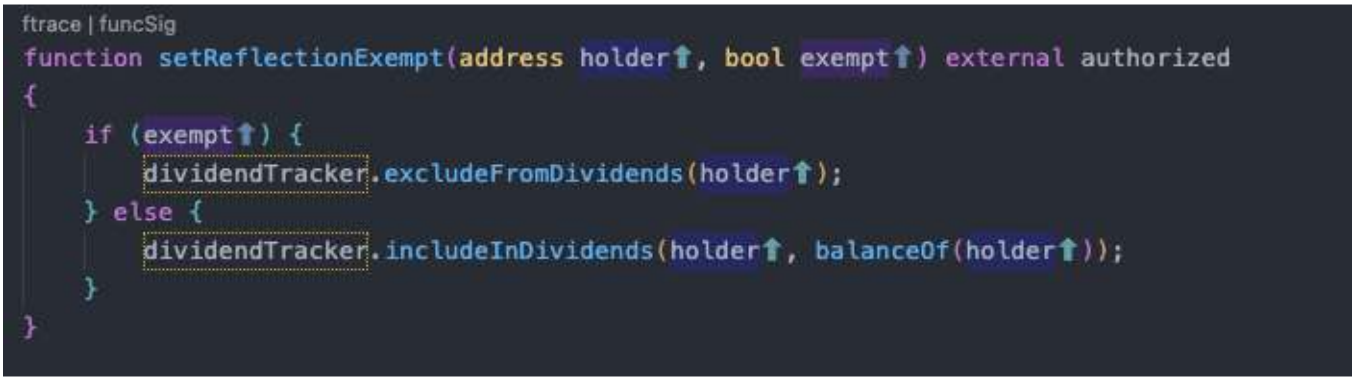
```
ftrace | funcSig
function updateClaimWait(uint256 claimWait) external onlyOwner {
    dividendTracker.updateClaimWait(claimWait);
}


/**
```

- ❖ The owner can change minimum token amount to have receive rewards



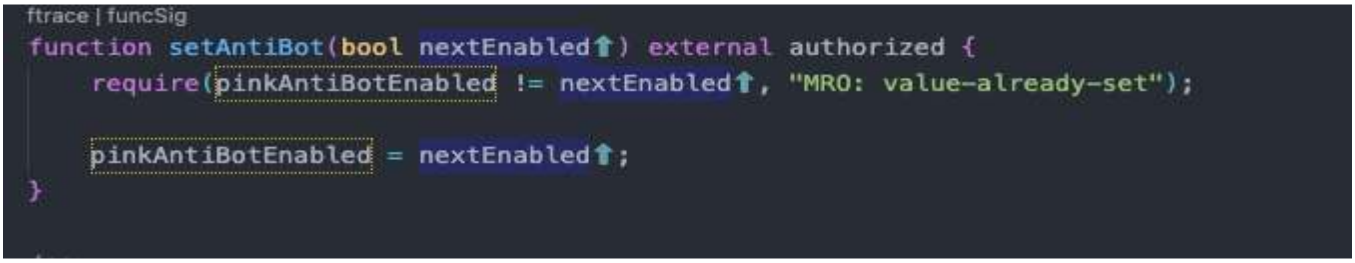
```
ftrace | funcSig
function updateMinimumTokenBalanceForDividends(uint256 amount) external onlyOwner
{
    dividendTracker.updateMinimumTokenBalanceForDividends(amount);
}
```

- ❖ The owner can enclude/exclude wallets from rewards



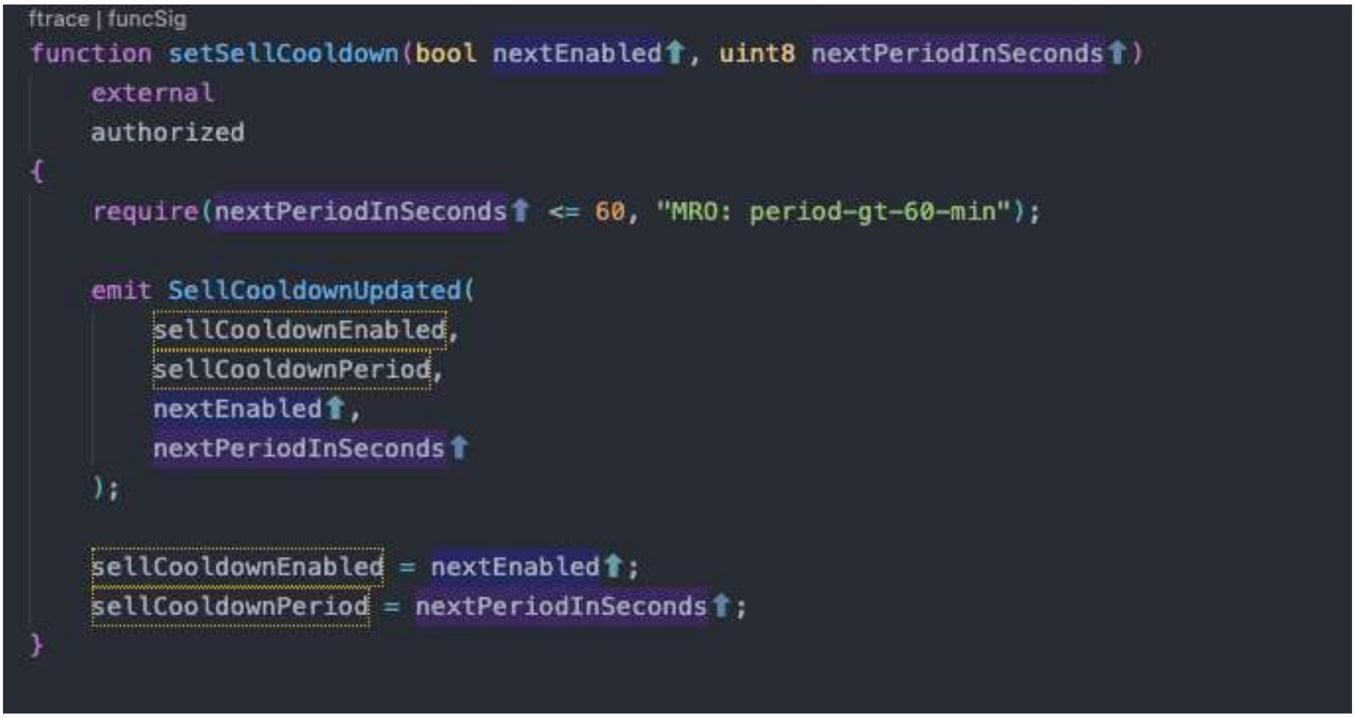
```
ftrace | funcSig
function setReflectionExempt(address holder, bool exempt) external authorized
{
    if (exempt) {
        dividendTracker.excludeFromDividends(holder);
    } else {
        dividendTracker.includeInDividends(holder, balanceOf(holder));
    }
}
```

- ❖ The owner can enable/disable pink antibot



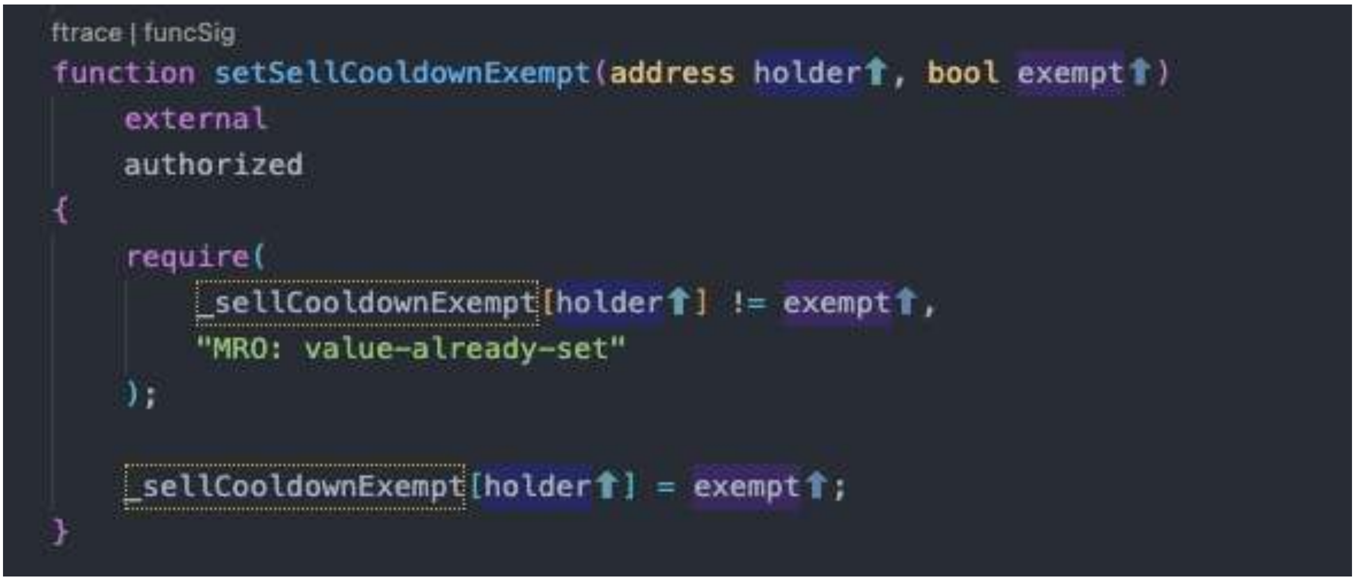
```
ftrace | funcSig
function setAntiBot(bool nextEnabled) external authorized {
    require(pinkAntiBotEnabled != nextEnabled, "MRO: value-already-set");

    pinkAntiBotEnabled = nextEnabled;
}
```

- The owner can enable/disable sell cool down and can change sell cool down time maximum upto 1 min



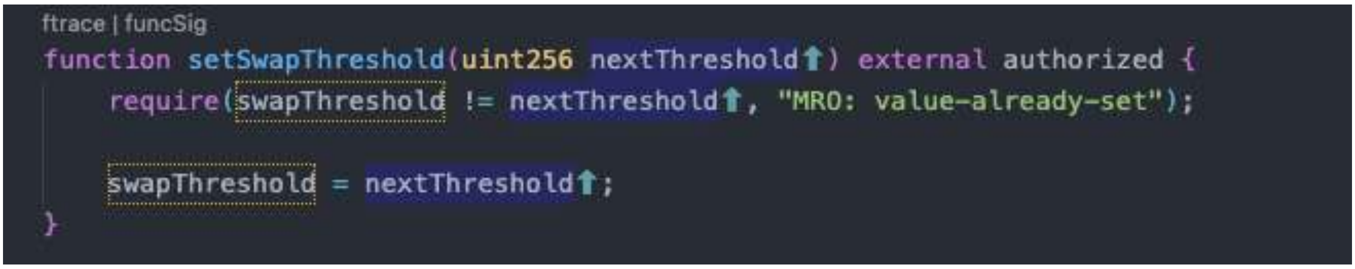
```
function setSellCooldown(bool nextEnabled, uint8 nextPeriodInSeconds)
    external
    authorized
{
    require(nextPeriodInSeconds <= 60, "MRO: period-gt-60-min");

    emit SellCooldownUpdated(
        sellCooldownEnabled,
        sellCooldownPeriod,
        nextEnabled,
        nextPeriodInSeconds
    );

    sellCooldownEnabled = nextEnabled;
    sellCooldownPeriod = nextPeriodInSeconds;
}
```

- The owner can include/exclude wallets from sell cool down



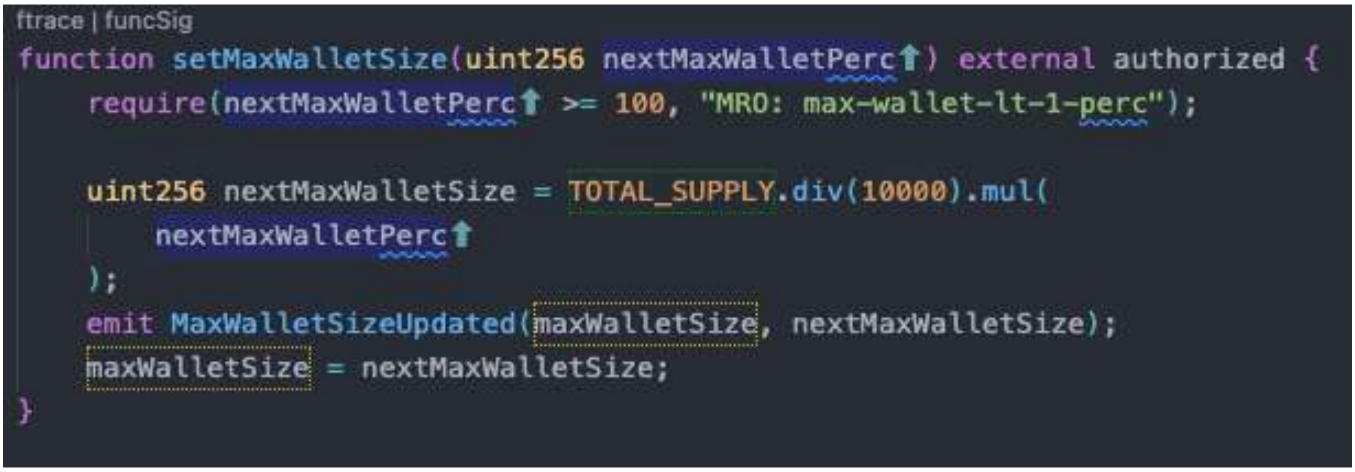
```
function setSellCooldownExempt(address holder, bool exempt)
    external
    authorized
{
    require(
        _sellCooldownExempt[holder] != exempt,
        "MRO: value-already-set"
    );

    _sellCooldownExempt[holder] = exempt;
}
```

- The owner can change the swap point



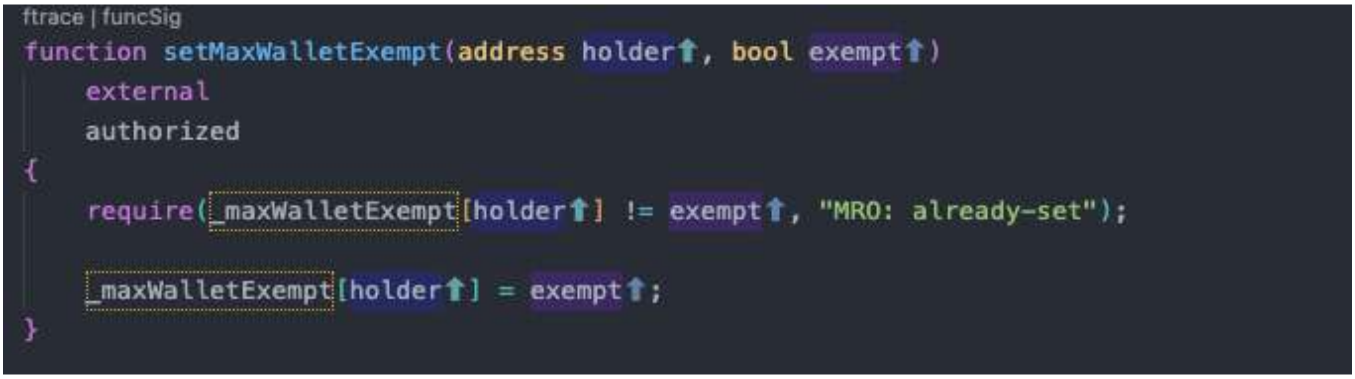
```
function setSwapThreshold(uint256 nextThreshold) external authorized {
    require(swapThreshold != nextThreshold, "MRO: value-already-set");

    swapThreshold = nextThreshold;
}
```

- ❖ The owner can change max wallet size minimum upto 1%



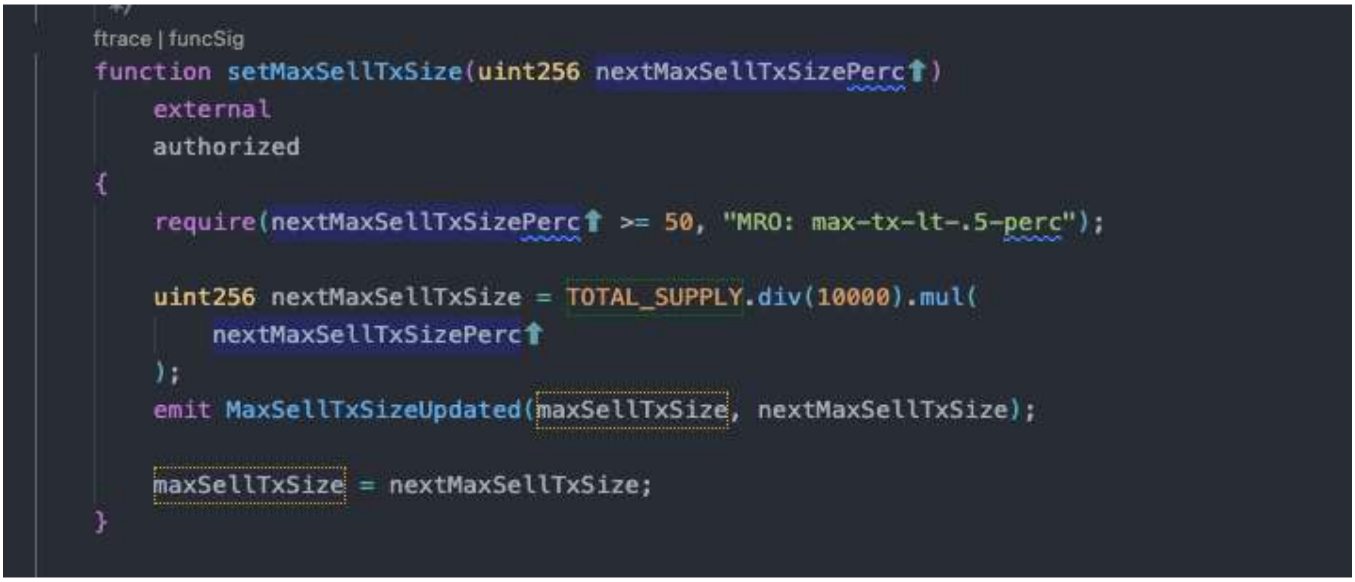
```
ftrace | funcSig
function setMaxWalletSize(uint256 nextMaxWalletPerc) external authorized {
    require(nextMaxWalletPerc >= 100, "MRO: max-wallet-lt-1-perc");

    uint256 nextMaxWalletSize = TOTAL_SUPPLY.div(10000).mul(
        nextMaxWalletPerc
    );
    emit MaxWalletSizeUpdated(maxWalletSize, nextMaxWalletSize);
    maxWalletSize = nextMaxWalletSize;
}
```

- ❖ The owner can include/exclude wallets from max wallet size



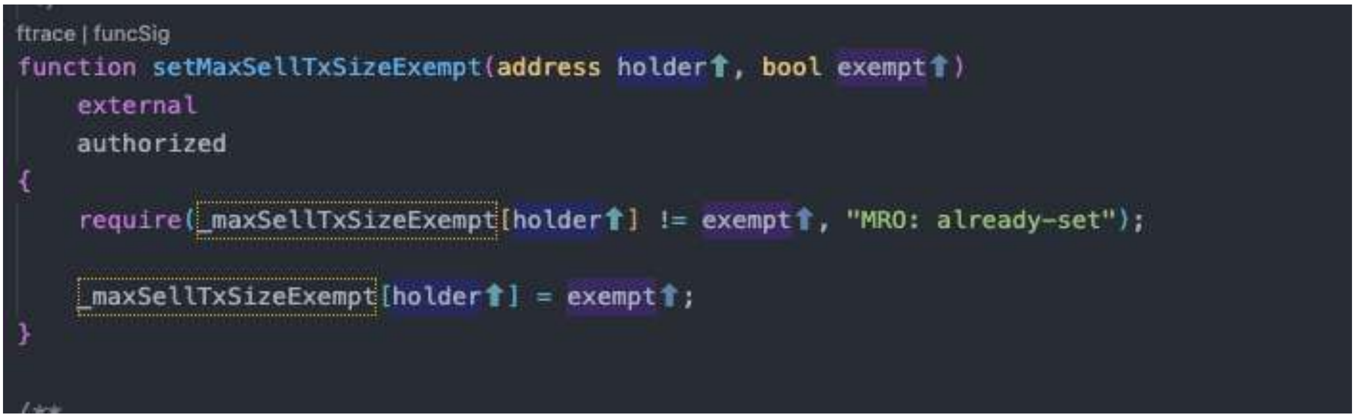
```
ftrace | funcSig
function setMaxWalletExempt(address holder, bool exempt)
    external
    authorized
{
    require(_maxWalletExempt[holder] != exempt, "MRO: already-set");

    _maxWalletExempt[holder] = exempt;
}
```

- ❖ The owner can change max sell amount maximum upto 0.5%



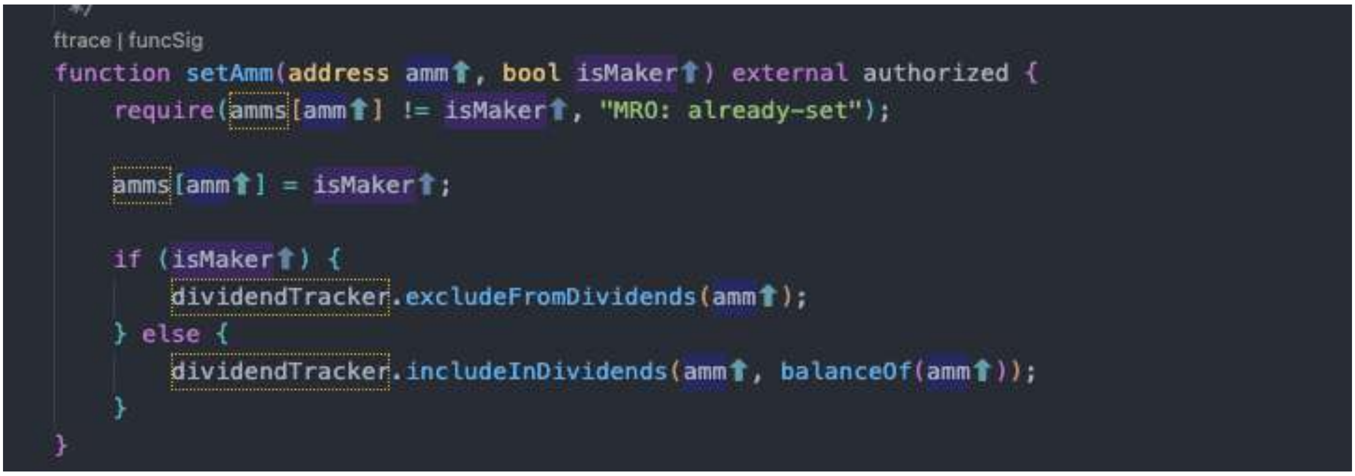
```
    */
    ftrace | funcSig
    function setMaxSellTxSize(uint256 nextMaxSellTxSizePerc)
        external
        authorized
    {
        require(nextMaxSellTxSizePerc >= 50, "MRO: max-tx-lt-.5-perc");

        uint256 nextMaxSellTxSize = TOTAL_SUPPLY.div(10000).mul(
            nextMaxSellTxSizePerc
        );
        emit MaxSellTxSizeUpdated(maxSellTxSize, nextMaxSellTxSize);

        maxSellTxSize = nextMaxSellTxSize;
    }
```

- The owner include/exclude wallets from max sell limit



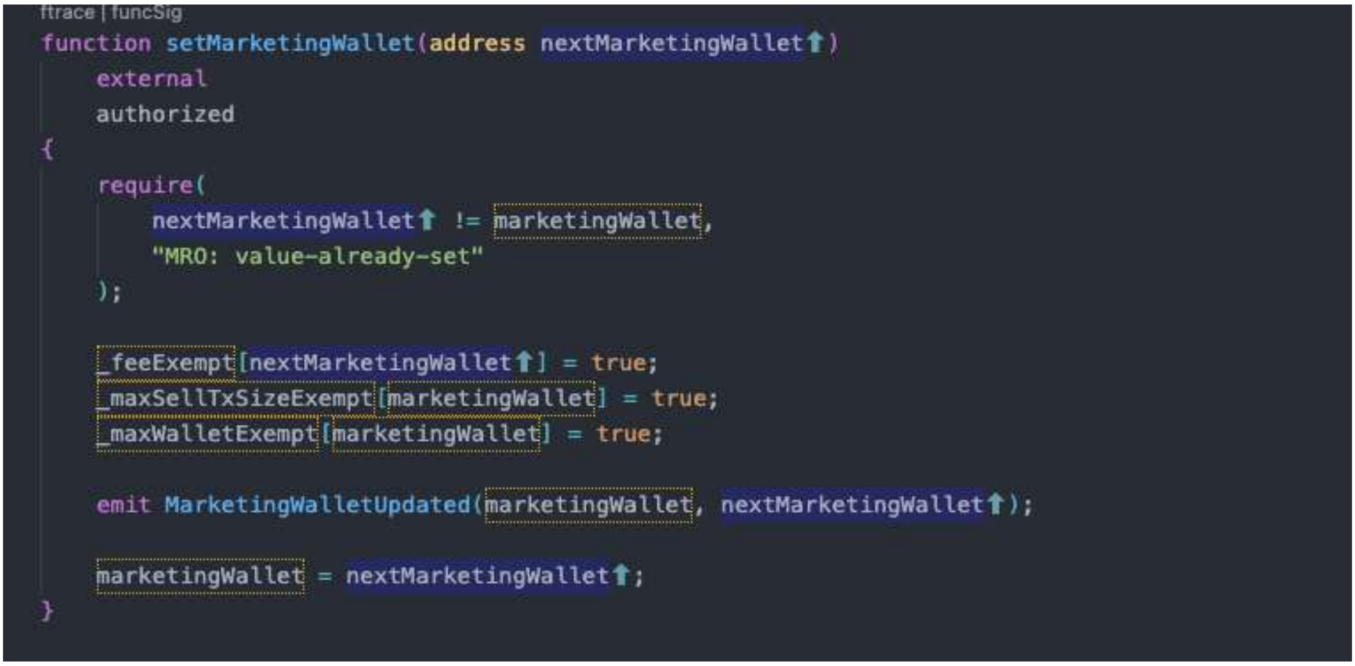
```
ftrace | funcSig
function setMaxSellTxSizeExempt(address holder, bool exempt)
    external
    authorized
{
    require(_maxSellTxSizeExempt[holder] != exempt, "MRO: already-set");

    _maxSellTxSizeExempt[holder] = exempt;
}
```

- The owner can add/remove new pairs



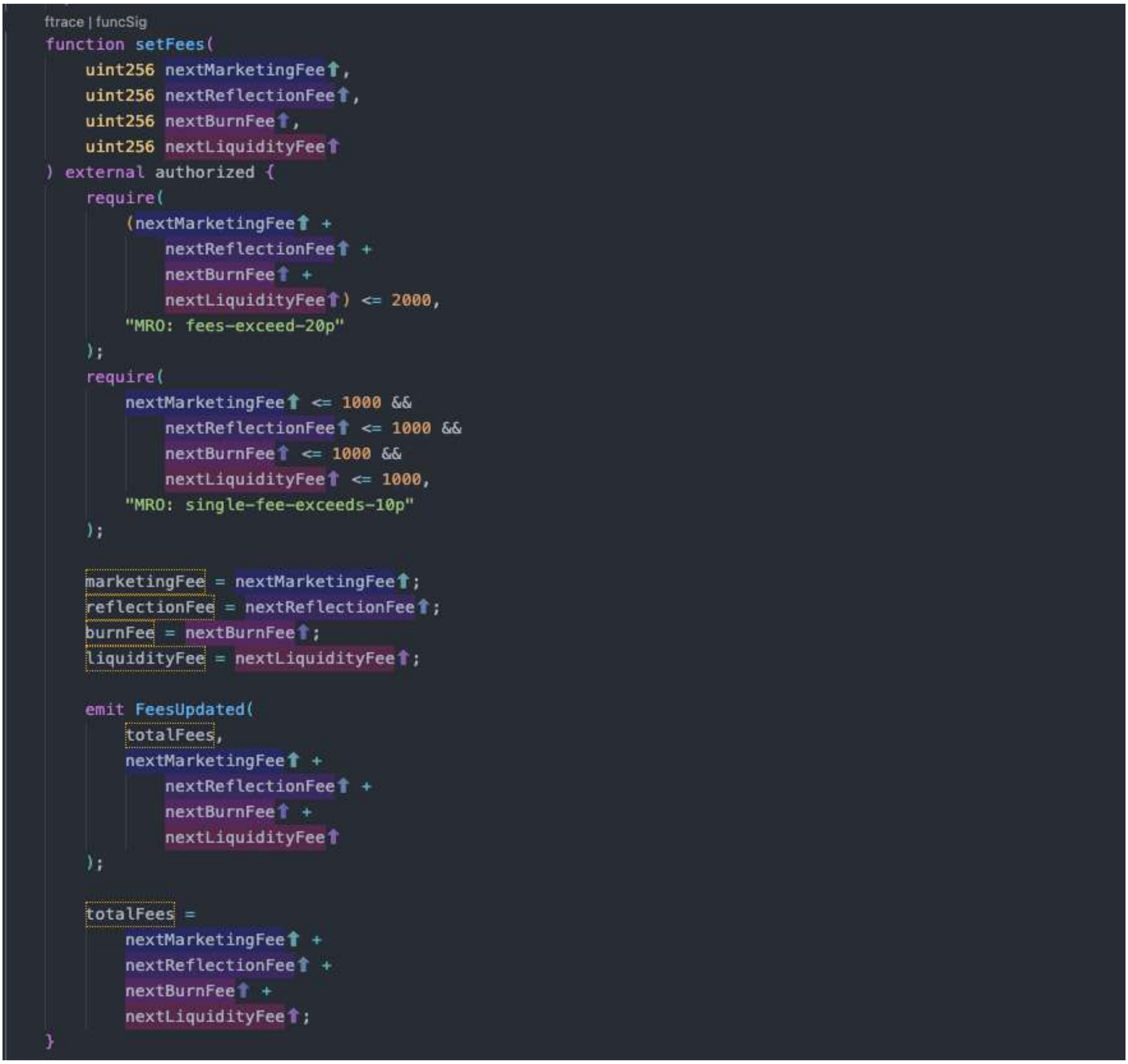
```
ftrace | funcSig
function setAmm(address amm, bool isMaker) external authorized {
    require(amms[amm] != isMaker, "MRO: already-set");

    amms[amm] = isMaker;

    if (isMaker) {
        dividendTracker.excludeFromDividends(amm);
    } else {
        dividendTracker.includeInDividends(amm, balanceOf(amm));
    }
}
```

- The owner can change marketing wallet address



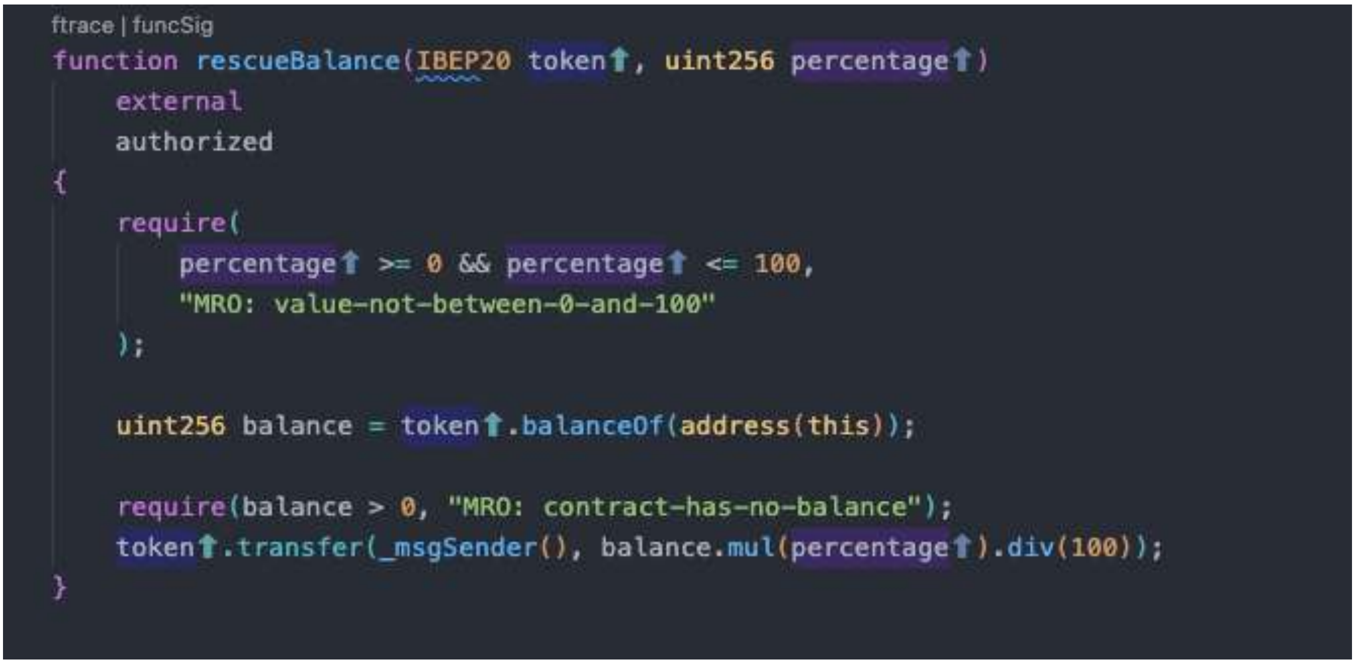
```
ftrace | funcSig
function setMarketingWallet(address nextMarketingWallet)
    external
    authorized
{
    require(
        nextMarketingWallet != marketingWallet,
        "MRO: value-already-set"
    );

    _feeExempt[nextMarketingWallet] = true;
    _maxSellTxSizeExempt[marketingWallet] = true;
    _maxWalletExempt[marketingWallet] = true;

    emit MarketingWalletUpdated(marketingWallet, nextMarketingWallet);

    marketingWallet = nextMarketingWallet;
}
```

- The owner can change all fees, total fees maximum upto 20% and each fees maximum upto 10%



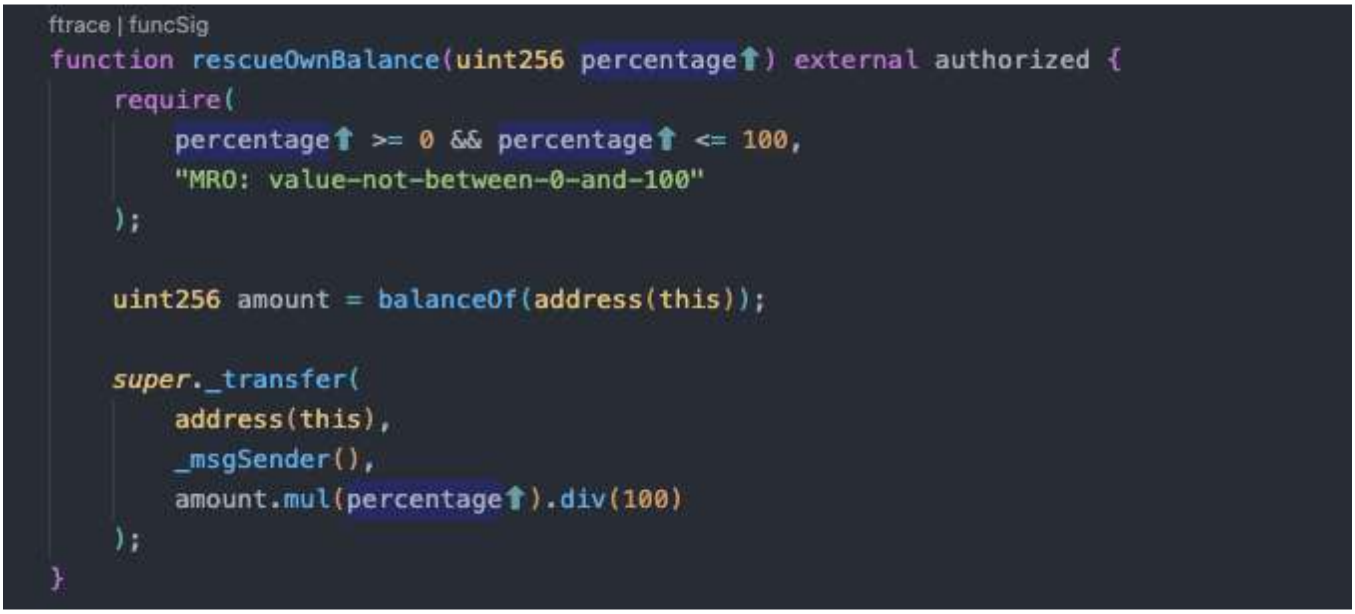
```
ftrace | funcSig
function setFees(
    uint256 nextMarketingFee,
    uint256 nextReflectionFee,
    uint256 nextBurnFee,
    uint256 nextLiquidityFee
) external authorized {
    require(
        (nextMarketingFee +
            nextReflectionFee +
            nextBurnFee +
            nextLiquidityFee) <= 2000,
        "MRO: fees-exceed-20p"
    );
    require(
        nextMarketingFee <= 1000 &&
            nextReflectionFee <= 1000 &&
            nextBurnFee <= 1000 &&
            nextLiquidityFee <= 1000,
        "MRO: single-fee-exceeds-10p"
    );

    marketingFee = nextMarketingFee;
    reflectionFee = nextReflectionFee;
    burnFee = nextBurnFee;
    liquidityFee = nextLiquidityFee;

    emit FeesUpdated(
        totalFees,
        nextMarketingFee +
            nextReflectionFee +
            nextBurnFee +
            nextLiquidityFee
    );

    totalFees =
        nextMarketingFee +
        nextReflectionFee +
        nextBurnFee +
        nextLiquidityFee;
}
```

- The owner can take any bep20 tokens in the contract



```
ftrace | funcSig
function rescueBalance(IBEP20 token, uint256 percentage)
    external
    authorized
{
    require(
        percentage >= 0 && percentage <= 100,
        "MRO: value-not-between-0-and-100"
    );

    uint256 balance = token.balanceOf(address(this));

    require(balance > 0, "MRO: contract-has-no-balance");
    token.transfer(_msgSender(), balance.mul(percentage).div(100));
}
```

- The owner can take native tokens from the contract



```
ftrace | funcSig
function rescueOwnBalance(uint256 percentage) external authorized {
    require(
        percentage >= 0 && percentage <= 100,
        "MRO: value-not-between-0-and-100"
    );

    uint256 amount = balanceOf(address(this));

    super._transfer(
        address(this),
        _msgSender(),
        amount.mul(percentage).div(100)
    );
}
```

# 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: **PASS**

Number of risk issues: **0**

Solidity code functional issue level: **PASS**

Number of owner privileges: **16**

Centralization risk correlated to the active owner: **NONE**

Smart contract active ownership: **ACTIVE**