RugFreeCoins Audit

ROTTCOIN Token
Smart Contract Security Audit

October 3rd, 2022

Contents

Audit details

Audited project
ROTTCOIN Token

Contract Address
0x1CC41b493EEcc0CE084776a7a4BDdfF50D77EC91

Client contact
ROTTCOIN Team

Blockchain
Binance smart chain

Project website
https://rottcoin.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’t set a max transaction limit

✅ The owner can’t pause trading.

✅ The owner can't set fees over 10%.

✅ Owner can't blacklist wallets.

✅ The owner can’t set a max wallet limit

✅ The owner can’t claim the contract’s balance of its own token.

Background

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

https://bscscan.com/token/0x1CC41b493EEcc0CE084776a7a4BDdfF50D77EC91

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

2022 Q3

- Team recruitment
- Build a social channel
- Website Launch
- Community growth and development
- Whitepaper
- Audit & KYC - Safu
- Pinksale Presale
- Marketing Campaign
- Pancake swap launch

2022 Q4

- Influencer Marketing Push
- 5.000+ Holders
- CoinGecko, CoinmarketCap Listing
- Community events Dextools & Poocoin Banner Ads
- NFT Marketplace
- RottWallet

2023 Q1

- Professional marketing campaign
- Articles on influential sites
- Influencers on Twitter, Youtube, Tiktok
- AMA with large crypto community
- NFTs Collection Release
- CEX Exchanges Listing
- RottSwap launch
- Rottchain Release more to come

Tokenomics

3% when buying & selling

- 1% of trade goes to the marketing wallet in BNB.
- 1% of trade goes to the distribution of rewards among investors in tokens.
- 1% of trade goes to the development wallet in BNB.

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 to staking and receive rewards.
- Anyone who's interested in collecting or trading NFTs
- Anyone who's interested in taking part in ROTTchain platform.
- Anyone who's interested in taking part in the future plans of ROTTCOIN Token.
- Anyone who's interested in making financial transactions with any other party using ROTTCOIN 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	8/10
6	Impact on the community	8/10
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		**8.9/10**

Contract details

Token contract details for 3rd of October 2022

Contract name	ROTTCOIN
Contract address	0x1CC41b493EEcc0CE084776a7a4BDdfF50D77EC91
Token supply	1,000,000,000,000
Token ticker	$ROTT
Decimals	9
Token holders	1
Transaction count	1
Marketing wallet	0xBcAf03A0aF480AE969f312aCfff8F9FA5Edf5C7c
Development wallet	0x639f72d9ce1f2bcC2024eAb4Ba99fa3b98C430AF
Contract deployer address	0xCFC031370451f883B467ab7C9568920899FdFF44
Contract's current owner address	0xCFC031370451f883B467ab7C9568920899FdFF44

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
ROTTCOIN	**Implementation**	**Context, IERC20, Ownable**		
└		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❗
└	isExcludedFromReward	Public ❗		NO❗
└	totalFees	Public ❗		NO❗

└	deliver	Public ❗	🛑	NO❗
└	reflectionFromToken	Public ❗		NO❗
└	tokenFromReflection	Public ❗		NO❗
└	excludeFromReward	Public ❗	🛑	onlyOwner
└	includeInReward	External ❗	🛑	onlyOwner
└	setMarketingWallet	External ❗	🛑	onlyOwner
└	setDevWallet	External ❗	🛑	onlyOwner
└	changeSwapAmount	External ❗	🛑	onlyOwner
└	setExcludedFromFee	External ❗	🛑	onlyOwner
└	tradingEnable	External ❗	🛑	onlyOwner
└	updateBuyFees	External ❗	🛑	onlyOwner
└	updateSellFees	External ❗	🛑	onlyOwner
└	updateSwapPercentages	External ❗	🛑	onlyOwner
└	setSwapAndLiquifyEnabled	Public ❗	🛑	onlyOwner
└		External ❗	💵	NO❗
└	setUniswapRouter	External ❗	🛑	onlyOwner
└	setUniswapPair	External ❗	🛑	onlyOwner
└	setAuthorizedWallets	External ❗	🛑	onlyOwner
└	setExcludedFromAutoLiquidity	External ❗	🛑	onlyOwner
└	_reflectFee	Private 🔐	🛑	
└	_getTValues	Private 🔐		

└	_getRValues	Private 🔐		
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	takeTokenFees	Private 🔐	🛑	
└	takeTransactionFee	Private 🔐	🛑	
└	calculateFee	Private 🔐		
└	isExcludedFromFee	Public ❗		NO❗
└	_approve	Private 🔐	🛑	
└	_transfer	Private 🔐	🛑	
└	swapAndSendBnb	Private 🔐	🛑	lockTheSwap
└	swapTokensForBnb	Private 🔐	🛑	
└	_tokenTransfer	Private 🔐	🛑	
└	_transferStandard	Private 🔐	🛑	
└	_transferBothExcluded	Private 🔐	🛑	
└	_transferToExcluded	Private 🔐	🛑	
└	_transferFromExcluded	Private 🔐	🛑	
Ownable	**Implementation**	**Context**		
└		Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	_checkOwner	Internal 🔒		

L	renounceOwnership	Public ❗	🛑	onlyOwner
L	transferOwnership	Public ❗	🛑	onlyOwner
L	_transferOwnership	Internal 🔒	🛑	
IERC20	**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❗
SafeMath	**Library**			
L	tryAdd	Internal 🔒		
L	trySub	Internal 🔒		
L	tryMul	Internal 🔒		
L	tryDiv	Internal 🔒		
L	tryMod	Internal 🔒		
L	add	Internal 🔒		
L	sub	Internal 🔒		
L	mul	Internal 🔒		
L	div	Internal 🔒		

└	mod	Internal 🔒		
└	sub	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
IUniswapV2Router02	**Interface**	**IUniswapV2Router01**		
└	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❗️

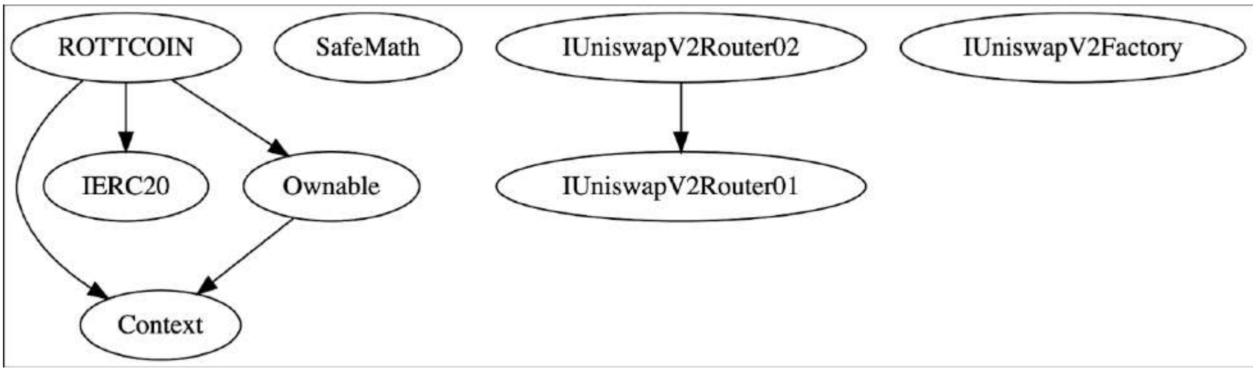
└	setFeeToSetter	External ❗	🛑	NO❗
Context	**Implementation**			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
IUniswapV2 Router01	**Interface**			
└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
└	addLiquidity	External ❗	🛑	NO❗
└	addLiquidityETH	External ❗	💵	NO❗
└	removeLiquidity	External ❗	🛑	NO❗
└	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

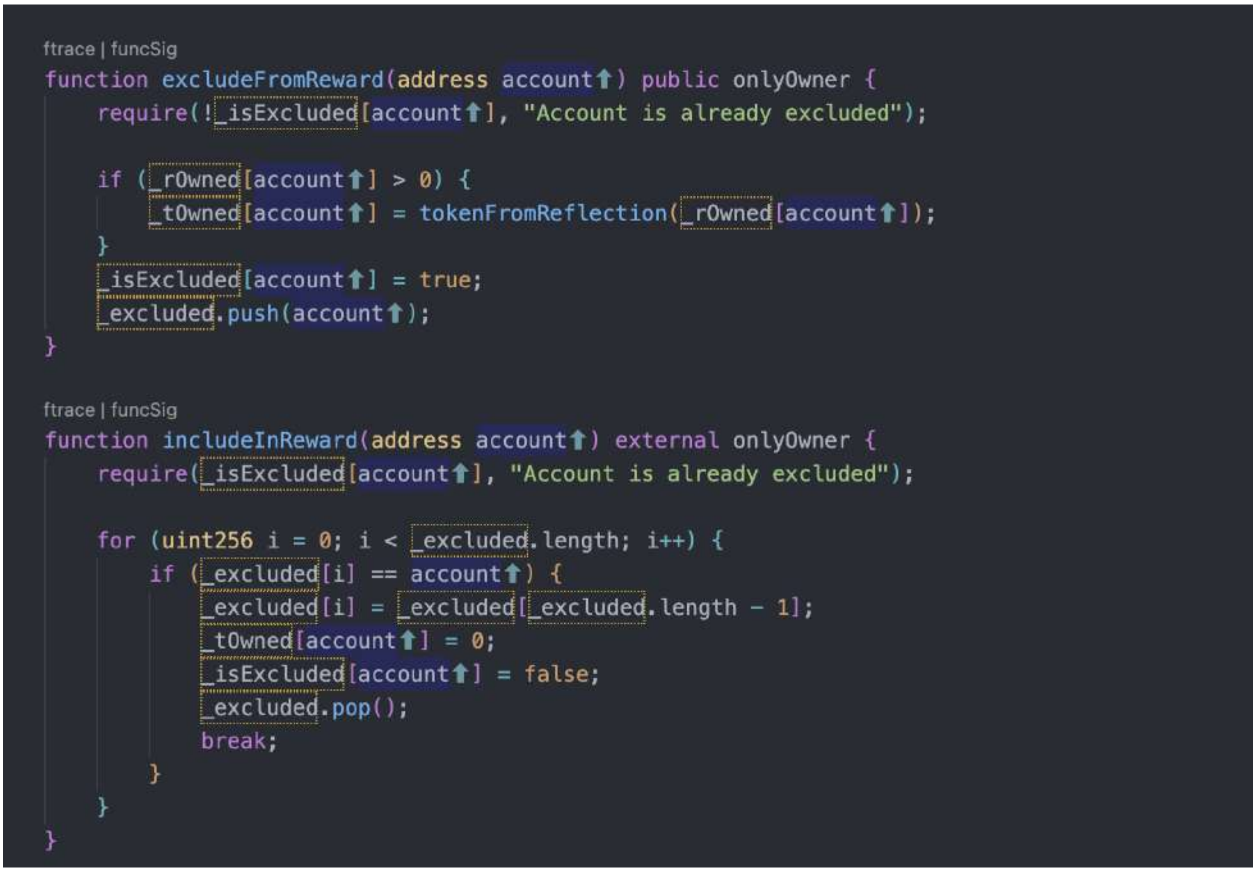
- **Medium severity issues**
  No medium severity issues found

- **Low severity issues**
  No low severity issues found

- **Centralization Risk**
  No Centralization Risk found

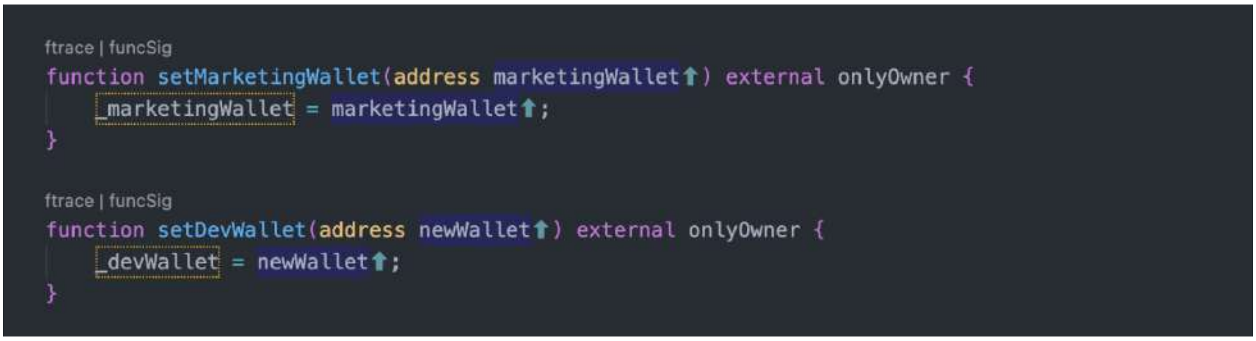
Owner privileges

❖ The owner can include/exclude wallets from the rewards



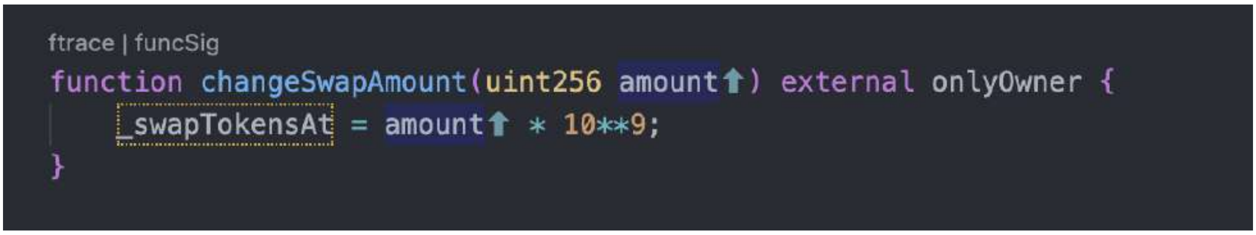
```
ftrace | funcSig
function excludeFromReward(address account) public onlyOwner {
    require(!_isExcluded[account], "Account is already excluded");

    if (_rOwned[account] > 0) {
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}

ftrace | funcSig
function includeInReward(address account) external onlyOwner {
    require(_isExcluded[account], "Account is already excluded");

    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account] = 0;
            _isExcluded[account] = false;
            _excluded.pop();
            break;
        }
    }
}
```

❖ The owner can change marketing and dev wallet



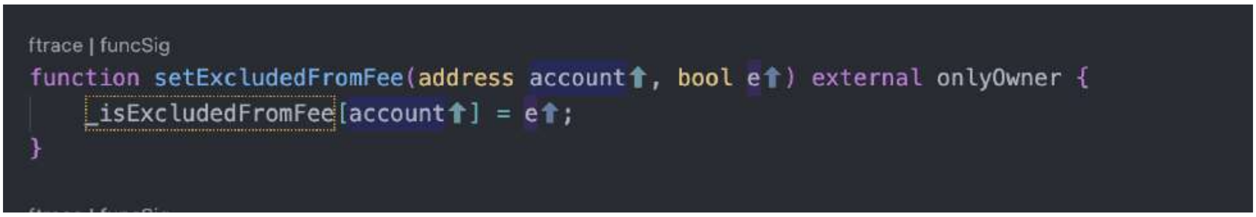
```
ftrace | funcSig
function setMarketingWallet(address marketingWallet) external onlyOwner {
    _marketingWallet = marketingWallet;
}

ftrace | funcSig
function setDevWallet(address newWallet) external onlyOwner {
    _devWallet = newWallet;
}
```

- The owner can change swap point



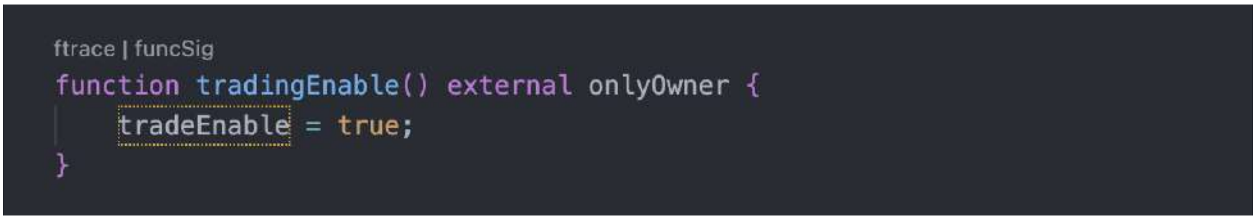
```
ftrace | funcSig
function changeSwapAmount(uint256 amount) external onlyOwner {
    _swapTokensAt = amount * 10**9;
}
```

- The owner can include/exclude wallets from the fees



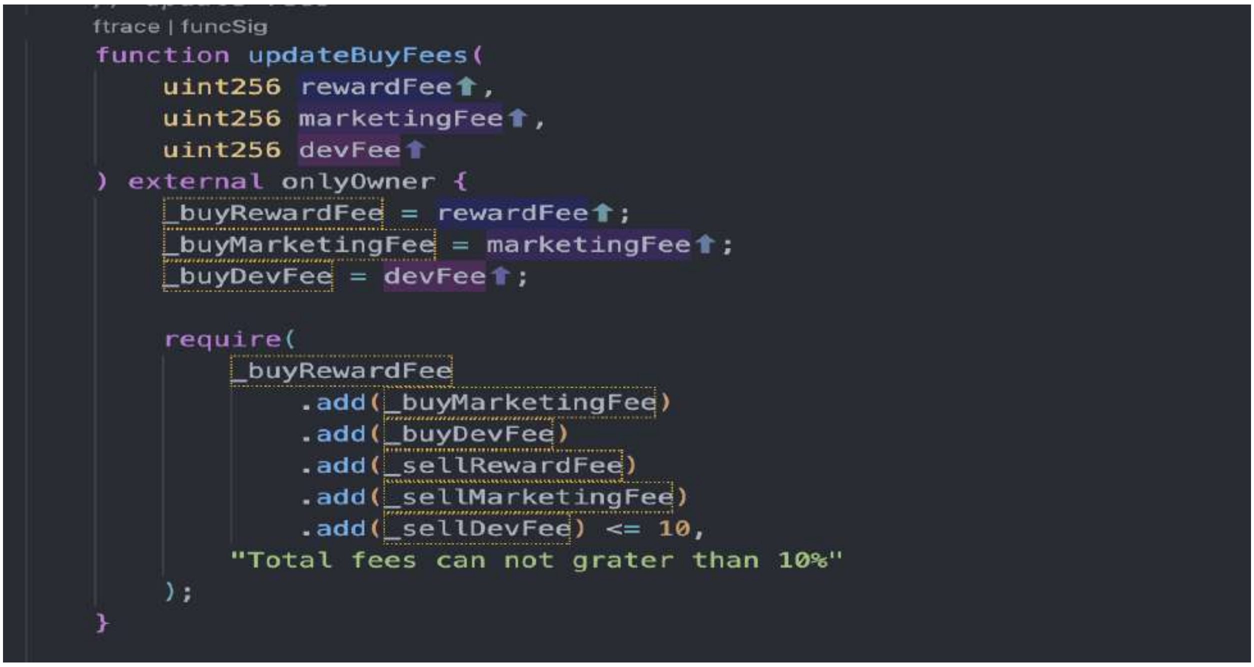
```
ftrace | funcSig
function setExcludedFromFee(address account, bool e) external onlyOwner {
    _isExcludedFromFee[account] = e;
}
```

- The owner can enable trading, once enabled cannot disable again



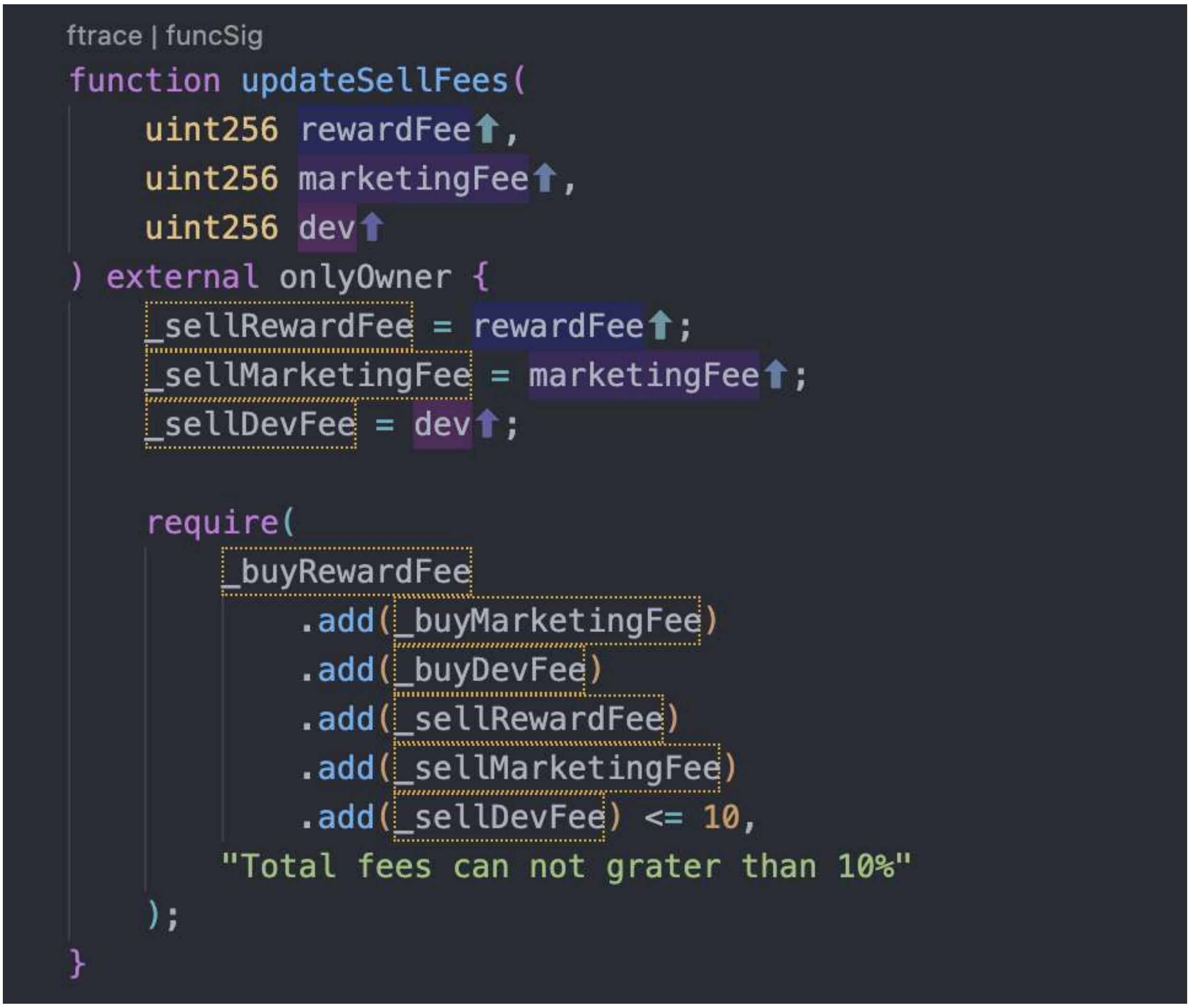
```
ftrace | funcSig
function tradingEnable() external onlyOwner {
    tradeEnable = true;
}
```

- The owner can update all buy fees, total fees maximum up to 10%



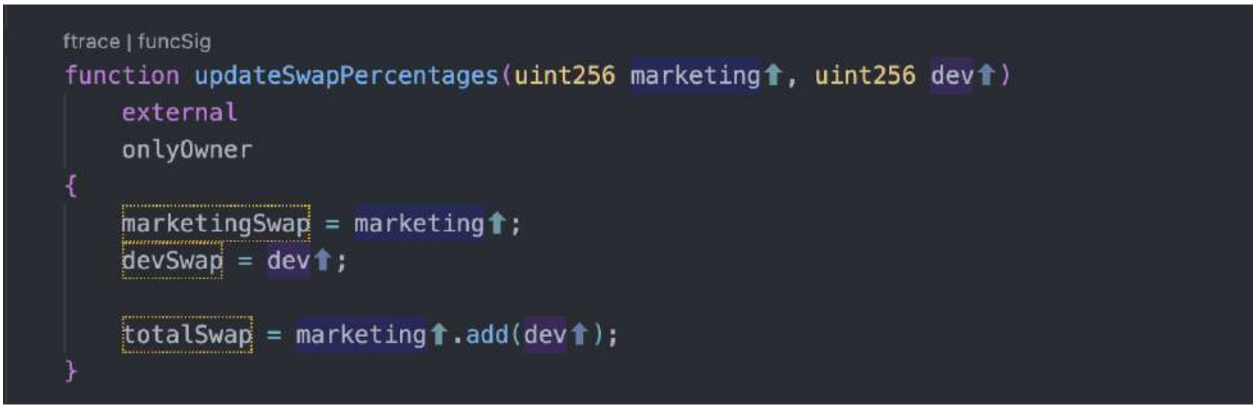
```
ftrace | funcSig
function updateBuyFees(
    uint256 rewardFee,
    uint256 marketingFee,
    uint256 devFee
) external onlyOwner {
    _buyRewardFee = rewardFee;
    _buyMarketingFee = marketingFee;
    _buyDevFee = devFee;

    require(
        _buyRewardFee
            .add(_buyMarketingFee)
            .add(_buyDevFee)
            .add(_sellRewardFee)
            .add(_sellMarketingFee)
            .add(_sellDevFee) <= 10,
        "Total fees can not grater than 10%"
    );
}
```

- ❖ The owner can change all sell fees, total fees maximum up to 10%



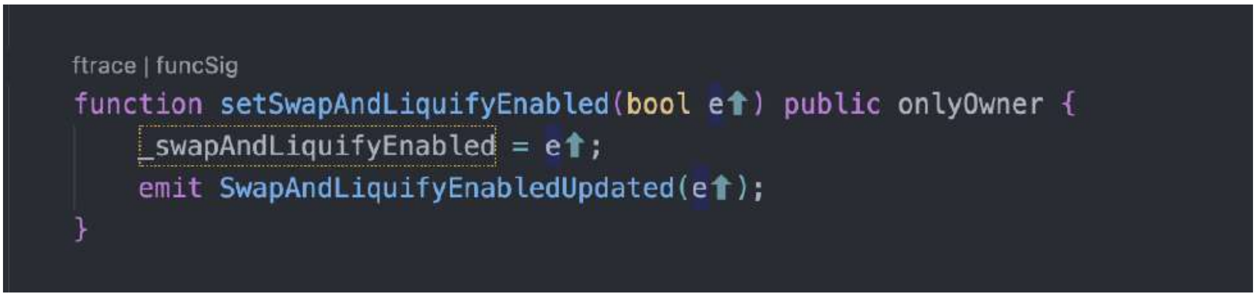
```
ftrace | funcSig
function updateSellFees(
    uint256 rewardFee,
    uint256 marketingFee,
    uint256 dev
) external onlyOwner {
    _sellRewardFee = rewardFee;
    _sellMarketingFee = marketingFee;
    _sellDevFee = dev;

    require(
        _buyRewardFee
            .add(_buyMarketingFee)
            .add(_buyDevFee)
            .add(_sellRewardFee)
            .add(_sellMarketingFee)
            .add(_sellDevFee) <= 10,
        "Total fees can not grater than 10%"
    );
}
```

- ❖ The owner can change swap percentages



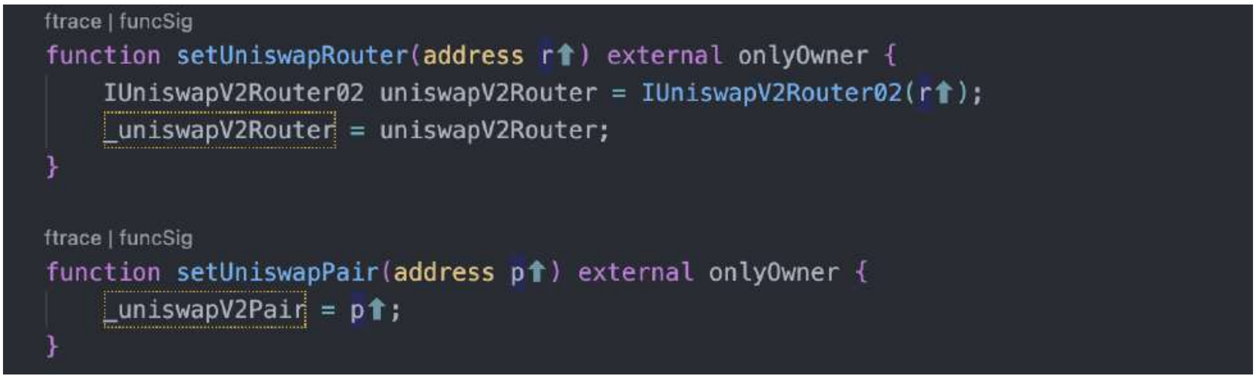
```
ftrace | funcSig
function updateSwapPercentages(uint256 marketing, uint256 dev)
    external
    onlyOwner
{
    marketingSwap = marketing;
    devSwap = dev;

    totalSwap = marketing.add(dev);
}
```

- ❖ The owner can enable/disable swapping



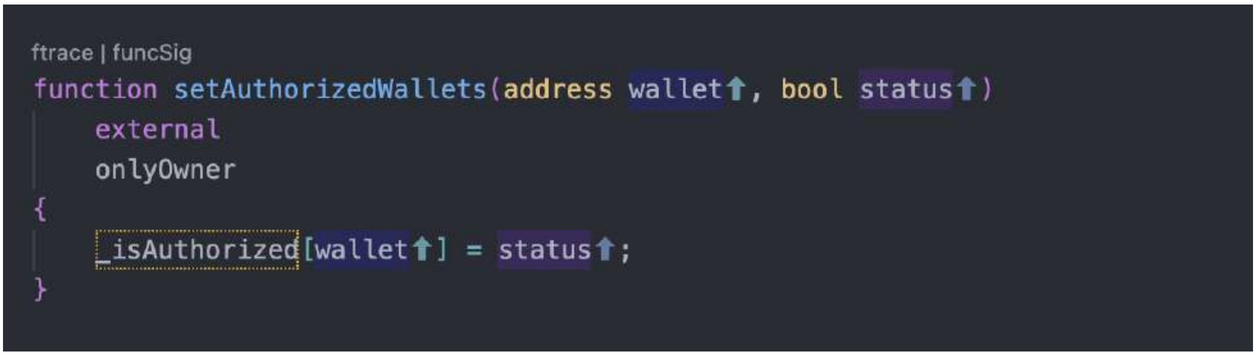
```
ftrace | funcSig
function setSwapAndLiquifyEnabled(bool e⬆) public onlyOwner {
    _swapAndLiquifyEnabled = e⬆;
    emit SwapAndLiquifyEnabledUpdated(e⬆);
}
```

- ❖ The owner can change router and pair address



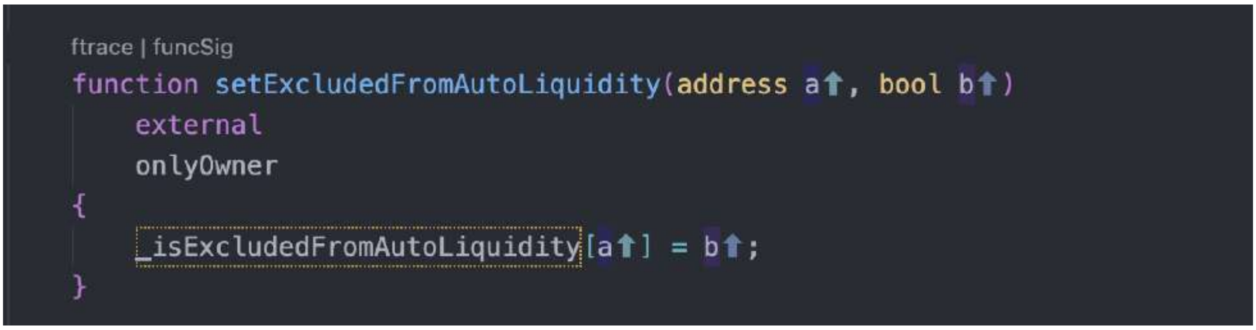
```
ftrace | funcSig
function setUniswapRouter(address r⬆) external onlyOwner {
    IUniswapV2Router02 uniswapV2Router = IUniswapV2Router02(r⬆);
    _uniswapV2Router = uniswapV2Router;
}

ftrace | funcSig
function setUniswapPair(address p⬆) external onlyOwner {
    _uniswapV2Pair = p⬆;
}
```

- ❖ The owner can add/remove authorized wallets, authorized wallets can do transactions before enable trading



```
ftrace | funcSig
function setAuthorizedWallets(address wallet⬆, bool status⬆)
    external
    onlyOwner
{
    _isAuthorized[wallet⬆] = status⬆;
}
```

- ❖ The owner can enable/disable wallets for not to trigger swapping when make transactions from the wallets getting added through calling this function.



```
ftrace | funcSig
function setExcludedFromAutoLiquidity(address a⬆, bool b⬆)
    external
    onlyOwner
{
    _isExcludedFromAutoLiquidity[a⬆] = b⬆;
}
```

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: **12**

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

Smart contract active ownership: **YES**