RugFreeCoins Audit

Optimus Token

Smart Contract Security Audit

June 30, 2022

Contents

Audit details

Audited project
Optimus Token

Contract Address

0xcBF45858A1AC2723DEf18FaeE8302263eC996c17

Client contact

Optimus Team

Blockchain

Binance smart chain

Project website

https://optimustesla.io/

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 Optimus Token Team to perform an audit of the smart contract.

https://bscscan.com/token/0xcbf45858a1ac2723def18faee8302263ec996c17

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

Optimus 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, which in turn, powers reward combined with the most interesting games and applications. Each transaction, purchase, and sale incur a 6% fee.

Features

- The **Optimus Token** will be distributed in tokens among every holder proportional to how many tokens each individual holds in values of **1% when buying and selling.**

- **The fee of 1% is charged when buying and selling** and will be sent to the CZ wallet.

- **The fee of 2% is charged when buying and selling** and will be sent to the Elon wallet.

- The additional component included under the sustainability section is a **liquidity fee of 2% when buying and selling**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

Tokenomics

5% fee when buying and selling

- 1% of trade goes to holders pockets in token rewards
- 1% of trade goes to the CZ wallet
- 2% of trade goes to the Elon wallet
- 2% of trade goes to the liquidity pool

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 in the future plans of the Optimus Token.
- Anyone who’s interested in making financial transactions with any other party using Optimus Token as the currency.

Contract details

Token contract details for 30th June 2022

Contract name	Optimus
Contract address	0xcBF45858A1AC2723DEf18FaeE8302263eC996c17
Token supply	2,003,000,000,000,000
Token ticker	OPT
Decimals	9
Token holders	1
Transaction count	1
Elon Fund Wallet	0x58fff3935514f9492d4c4084618b690ffa3c57a6
CZ Fund Wallet	0xeffb2d8b7a531c430ea0dfa6cf258d91e9b4eb51
Contract deployer address	0x9d24eB468AC60e58c5B939a9E3711D7C2987E111
Contract’s current owner address	0x9d24eb468ac60e58c5b939a9e3711d7c2987e111

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	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
Context	**Implementation**			
L	_msgSender	Internal 🔒		
L	_msgData	Internal 🔒		
IERC20	**Interface**			
L	totalSupply	External ❗		NO❗
L	decimals	External ❗		NO❗
L	symbol	External ❗		NO❗
L	name	External ❗		NO❗
L	getOwner	External ❗		NO❗
L	balanceOf	External ❗		NO❗
L	transfer	External ❗	🛑	NO❗
L	allowance	External ❗		NO❗
L	approve	External ❗	🛑	NO❗
L	transferFrom	External ❗	🛑	NO❗

SafeMath	**Library**			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	mod	Internal 🔒		
Address	**Library**			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	_functionCallWithValue	Private 🔐	🛑	
IUniswapV2 Factory	**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❗
IUniswapV2 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❗

└	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❗
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❗
IUniswapV2Router02	**Interface**	**IUniswapV2Router01**		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗

└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
Ownable	**Implementation**	**Context**		
└		Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
OPT	**Implementation**	**Context, IERC20, Ownable**		
└		Public ❗	🛑	NO❗
└	totalSupply	External ❗		NO❗
└	decimals	External ❗		NO❗
└	symbol	External ❗		NO❗
└	name	External ❗		NO❗
└	getOwner	External ❗		NO❗
└	allowance	External ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	transfer	Public ❗	🛑	NO❗
└	approve	Public ❗	🛑	NO❗
└	transferFrom	Public ❗	🛑	NO❗

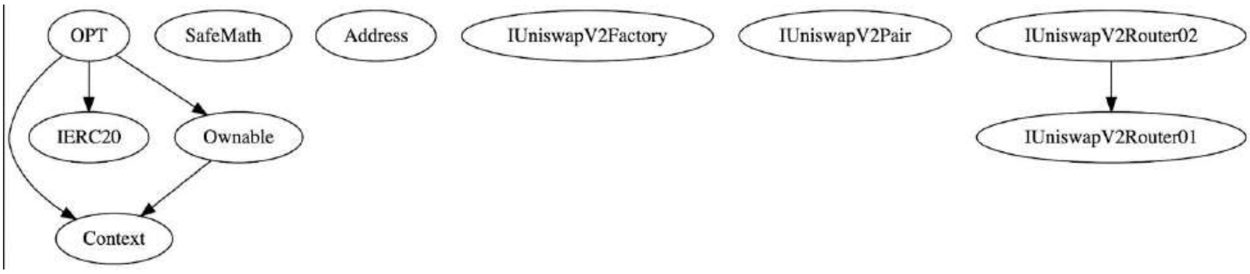
└	increaseAllowance	Public ❗	🛑	NO❗
└	decreaseAllowance	Public ❗	🛑	NO❗
└	setNewRouter	Public ❗	🛑	onlyOwner
└	isExcludedFromReward	Public ❗		NO❗
└	isExcludedFromFee	Public ❗		NO❗
└	setBuyTaxes	External ❗	🛑	onlyOwner
└	setSellTaxes	External ❗	🛑	onlyOwner
└	setBNBRatio	Private 🔐	🛑	onlyOwner
└	setMaxBuyTxPercent	External ❗	🛑	onlyOwner
└	setElonFundWallet	External ❗	🛑	onlyOwner
└	setCZFundWallet	External ❗	🛑	onlyOwner
└	setSwapAndLiquifyEnabled	Public ❗	🛑	onlyOwner
└	excludeFromFee	Public ❗	🛑	onlyOwner
└	includeInFee	External ❗	🛑	onlyOwner
└	totalFees	Public ❗		NO❗
└	_hasLimits	Private 🔐		
└	deliver	Private 🔐	🛑	
└	reflectionFromToken	Private 🔐		
└	tokenFromReflection	Private 🔐		
└	excludeFromReward	Public ❗	🛑	onlyOwner
└	includeInReward	External ❗	🛑	onlyOwner

└		External ❗	💵	NO❗
└	_approve	Private 🔐	🛑	
└	_transfer	Private 🔐	🛑	
└	swapAndLiquify	Private 🔐	🛑	lockTheSwap
└	swapTokensForEth	Private 🔐	🛑	
└	addLiquidity	Private 🔐	🛑	
└	_checkLiquidityAdd	Private 🔐	🛑	
└	_tokenTransfer	Private 🔐	🛑	
└	_finalizeTransfer	Private 🔐	🛑	
└	_getValues	Private 🔐		
└	_getTValues	Private 🔐		
└	_getRValues	Private 🔐		
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	_takeReflect	Private 🔐	🛑	
└	_takeLiquidity	Private 🔐	🛑	
└	_takeczFund	Private 🔐	🛑	
└	calculateTaxFee	Private 🔐		
└	calculateLiquidityFee	Private 🔐		
└	calculateczFund	Private 🔐		
└	adjustTaxes	Internal 🔒	🛑	

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

Inheritance Hierarchy



Security issue checking status

- **High severity issues**
  No High severity issues found

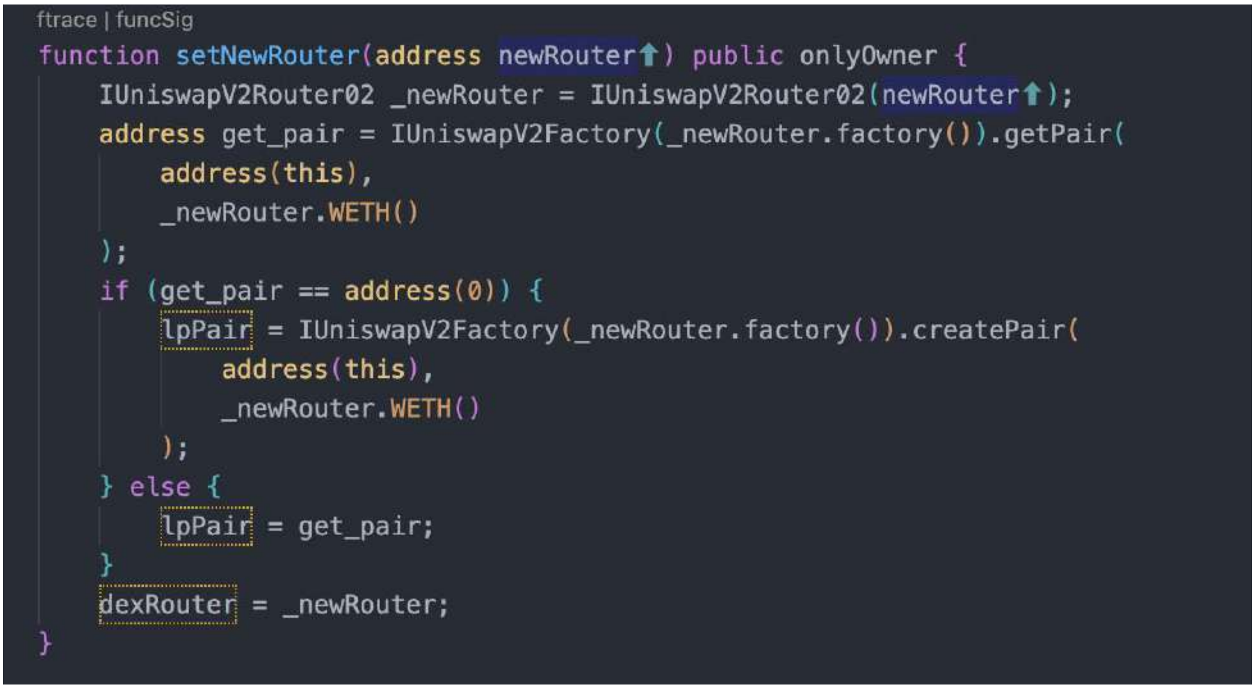
- **Medium severity issues**
  No medium severity issues found

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

- **Centralization Risk**
  No centralization issues found

Owner privileges

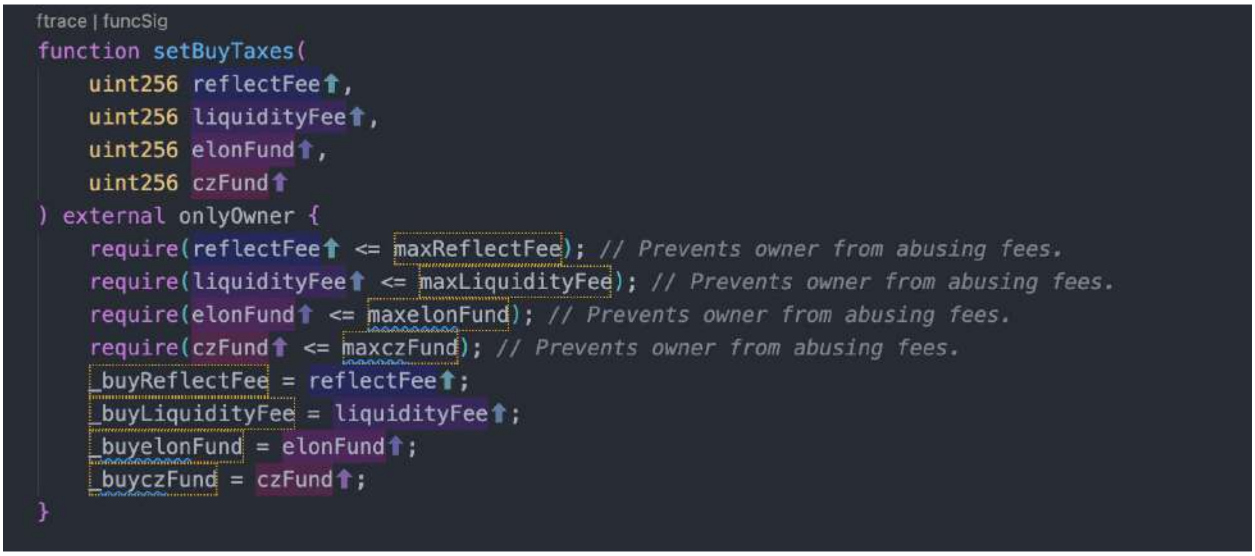
❖ The owner can change the router address



ftrace | funcSig

```
function setNewRouter(address newRouter) public onlyOwner {
    IUniswapV2Router02 _newRouter = IUniswapV2Router02(newRouter);
    address get_pair = IUniswapV2Factory(_newRouter.factory()).getPair(
        address(this),
        _newRouter.WETH()
    );
    if (get_pair == address(0)) {
        lpPair = IUniswapV2Factory(_newRouter.factory()).createPair(
            address(this),
            _newRouter.WETH()
        );
    } else {
        lpPair = get_pair;
    }
    dexRouter = _newRouter;
}
```

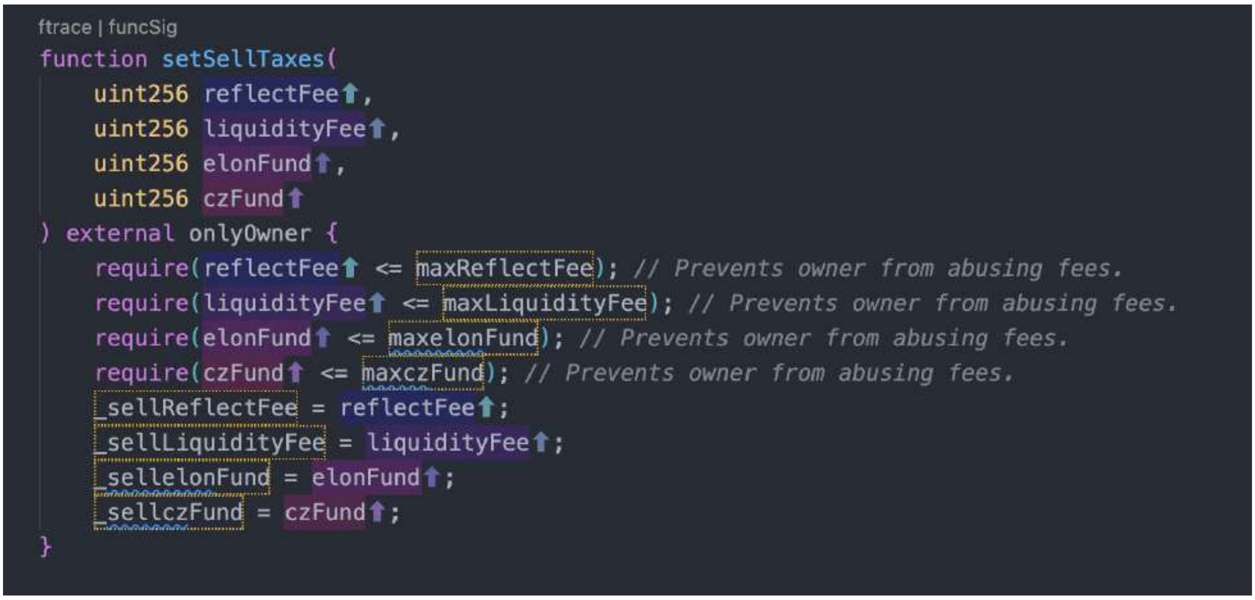
❖ The owner can change all buy fees up to maximum fee of each fee



ftrace | funcSig

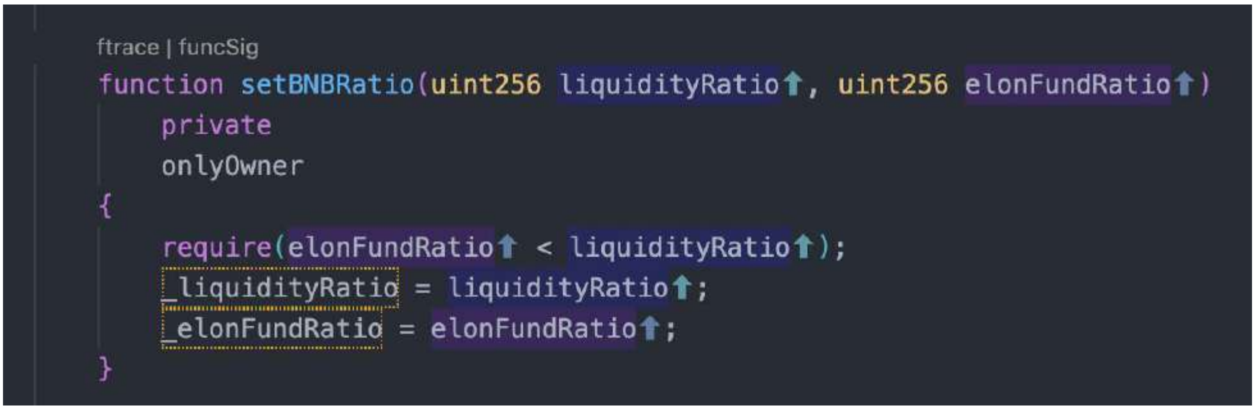
```
function setBuyTaxes(
    uint256 reflectFee,
    uint256 liquidityFee,
    uint256 elonFund,
    uint256 czFund
) external onlyOwner {
    require(reflectFee <= maxReflectFee); // Prevents owner from abusing fees.
    require(liquidityFee <= maxLiquidityFee); // Prevents owner from abusing fees.
    require(elonFund <= maxelonFund); // Prevents owner from abusing fees.
    require(czFund <= maxczFund); // Prevents owner from abusing fees.
    _buyReflectFee = reflectFee;
    _buyLiquidityFee = liquidityFee;
    _buyelonFund = elonFund;
    _buyczFund = czFund;
}
```

- The owner can change all sell fees up to maximum fee of each fee



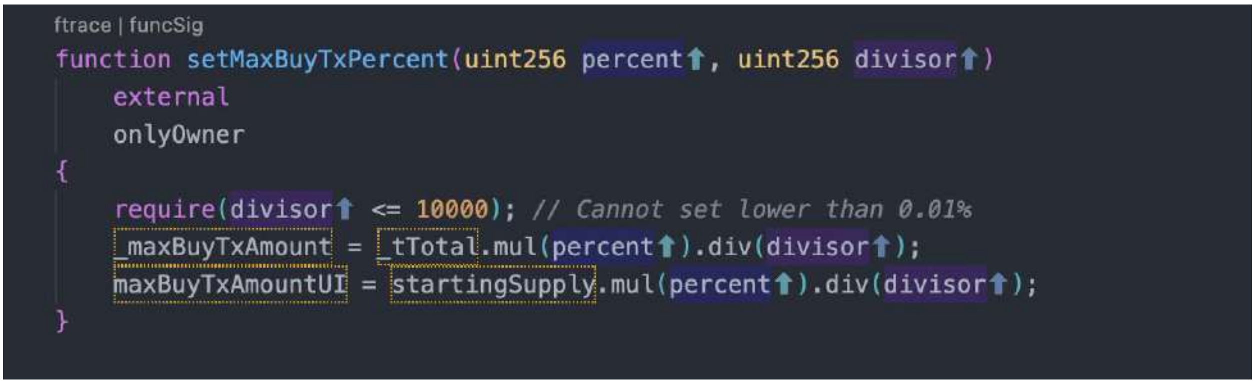
```
ftrace | funcSig
function setSellTaxes(
    uint256 reflectFee,
    uint256 liquidityFee,
    uint256 elonFund,
    uint256 czFund
) external onlyOwner {
    require(reflectFee <= maxReflectFee); // Prevents owner from abusing fees.
    require(liquidityFee <= maxLiquidityFee); // Prevents owner from abusing fees.
    require(elonFund <= maxelonFund); // Prevents owner from abusing fees.
    require(czFund <= maxczFund); // Prevents owner from abusing fees.
    _sellReflectFee = reflectFee;
    _sellLiquidityFee = liquidityFee;
    _sellelonFund = elonFund;
    _sellczFund = czFund;
}
```

- The owner can change bnb to token ration to add liquidity



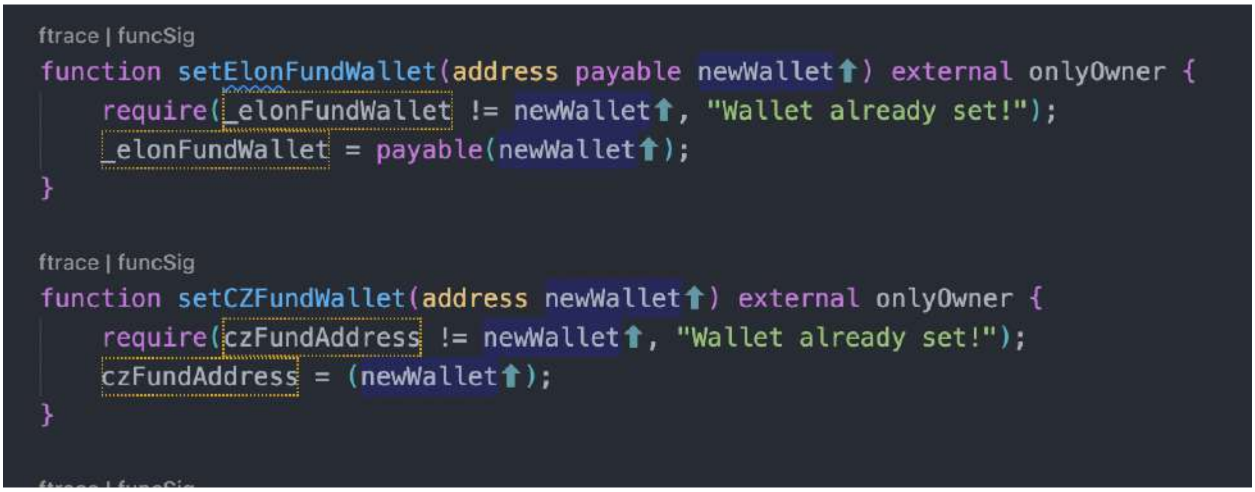
```
ftrace | funcSig
function setBNBRatio(uint256 liquidityRatio, uint256 elonFundRatio)
    private
    onlyOwner
{
    require(elonFundRatio < liquidityRatio);
    _liquidityRatio = liquidityRatio;
    _elonFundRatio = elonFundRatio;
}
```

- The owner can change maximum buy token amount minimum up to 0.01%



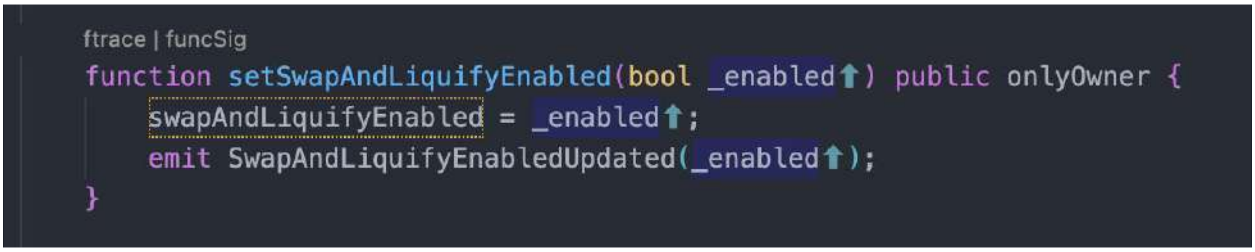
```
ftrace | funcSig
function setMaxBuyTxPercent(uint256 percent, uint256 divisor)
    external
    onlyOwner
{
    require(divisor <= 10000); // Cannot set lower than 0.01%
    _maxBuyTxAmount = _tTotal.mul(percent).div(divisor);
    maxBuyTxAmountUI = startingSupply.mul(percent).div(divisor);
}
```

- The owner can change elon fund and CZ fund wallets



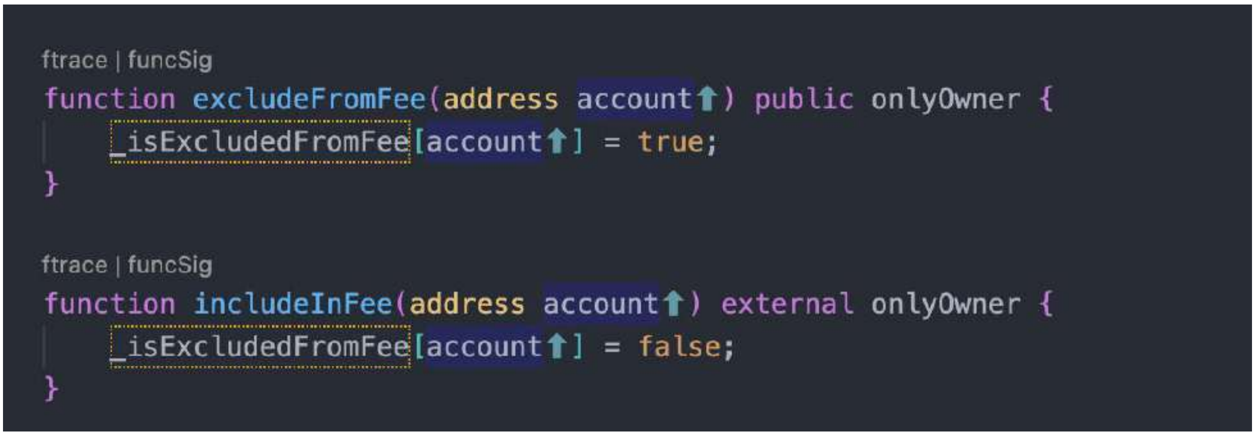
```
ftrace | funcSig
function setElonFundWallet(address payable newWallet) external onlyOwner {
    require(_elonFundWallet != newWallet, "Wallet already set!");
    _elonFundWallet = payable(newWallet);
}

ftrace | funcSig
function setCZFundWallet(address newWallet) external onlyOwner {
    require(czFundAddress != newWallet, "Wallet already set!");
    czFundAddress = (newWallet);
}
```

- The owner can enable/disable swapping



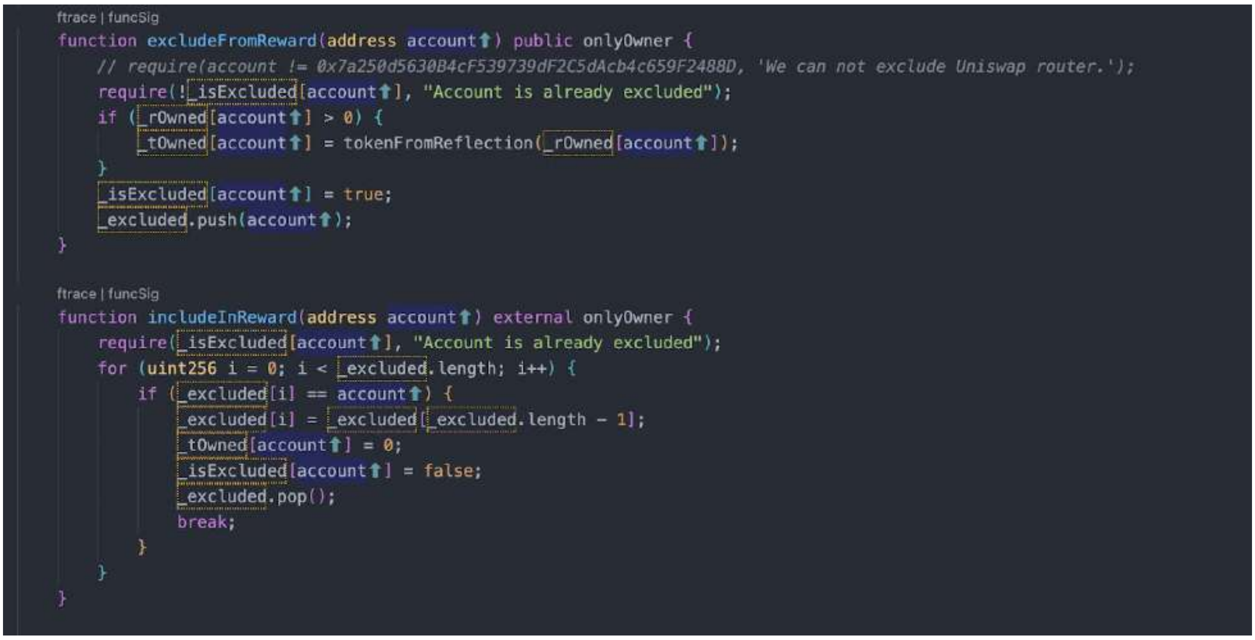
```
ftrace | funcSig
function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
    swapAndLiquifyEnabled = _enabled;
    emit SwapAndLiquifyEnabledUpdated(_enabled);
}
```

- The owner can include/exclude wallets from fee



```
ftrace | funcSig
function excludeFromFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = true;
}

ftrace | funcSig
function includeInFee(address account) external onlyOwner {
    _isExcludedFromFee[account] = false;
}
```

- The owner can include/exclude wallets from rewards



```
ftrace | funcSig
function excludeFromReward(address account) public onlyOwner {
    // require(account != 0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D, 'We can not exclude Uniswap router.');
    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;
        }
    }
}
```

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: **9**

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

Smart contract active ownership: **YES**