RugFreeCoins Audit

Launch Verse Token

Smart Contract Security Audit

March 21, 2022

Contents

Audit details

Audited project
LaunchVerse Token

Contract Address
0xB98f2acc693A19330A2B8da776d99DeF05Cf077E

Client contact
LaunchVerse Team

Blockchain
Binance smart chain

Project website
https://launchverse.space/

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 Launch Verse Team to perform an audit of the smart contract.

https://bscscan.com/token/0xB98f2acc693A19330A2B8da776d99DeF05Cf077E

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, 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

Launch Verse 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 lottery system and auto burn. Each transaction, purchase, and sale incur a 10% fee.

Features

- The **BNB rewards** will be distributed among every holder proportional to how many tokens each individual holds in values of **4% when buying and selling.**

- The **sustainability fee of 4% when buying and selling for marketing and dev** is what allows Launch Verse to hold the aforementioned promise. Tokens will be swapped into BNB and will be sent to a marketing wallet. This way, Launch Verse 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 1% from buying and selling**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.
- Launch Verse has the burn strategy that a **1% fee in each transaction when selling** is getting charged that benefits and rewards those who invest long-term. This feature slowly reduces supply making each Launch Verse more and more valuable.

Roadmap

Q1 2022

- Website Development
- Contract Development & Audit
- Team Doxxed & KYC
- Setting up Socials
- Private Sale
- Presale
- Pancakeswap Listing
- CMC/CG Listing
- Massive Marketing push
- Poocoin Adds
- TG group promotions
- AMA Sessions
- 5000 Holders
- First NFT Distribution
- Tracking Tool release

Q2 2022

- CEX listings
- Influencer Marketing
- Launchpad Launching
- First IDO Launching
- Staking & Lottery release
- Community Growth
- Team Expanding
- Continuous NFT Distribution
- First IDO Shareholding
- Website Update

Q3 2022

- Trending on CMC/Dextools
- CEX Listings
- Dapps Development
- Partnerships with other Launchpads
- Listing Application for Kucoin, Gate.io and Upbit
- Roadmap Update

Tokenomics

10% fee when buying and selling

- 4% of trade goes to holders’ pockets in BNB tokens.
- 4% of trade goes to the marketing and Dev wallets.
- 1% of trade goes to the liquidity pool.
- 1% of trade goes to the 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 in BNB by holding tokens.
- Anyone who's interested in trading tokens.
- Anyone who's interested in finding launchpads and looking for legit tokens to invest through the launch verse platform.
- Anyone who is ready to hold and be eligible to win in the daily lottery
- Anyone who is ready to hold a large portion of tokens and be eligible to get a high chance of winning in the weekly lottery.
- Anyone who's interested in taking part with the future plans of the Launch Verse token.
- Anyone who's interested in making financial transactions with any other party using BNB or Launch Verse as the currency.

Core concept

The Launch Verse reward system

4% of each transaction when buying and selling get converted to BNB and is split amongst all holders. Holders will be eligible to receive tokens every one hour and rewards are proportional to how many tokens each individual holds.

Sustainable mechanism

- The **sustainability fee of 4% when buying and selling for marketing and dev** is what allows Launch Verse to promote the token and use funds to further the development of the platform. Tokens will be swapped into BNB and will be sent to a marketing wallet. This way, Launch Verse 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 1%, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

- Launch Verse has the burn strategy that 1% fee in each transaction when selling is getting charged that benefits and rewards those who invest long-term. This feature slowly reduces supply making each Launch Verse more and more valuable.

Potential to grow with score points

1.	Project efficiency	9/10
2.	Project uniqueness	9/10
3	Information quality	8/10
4	Service quality	9/10
5	System quality	8/10
6	Impact on the community	9/10
7	Impact on the business	9/10
8	Preparing for the future	8/10
Total Points		**8.625/10**

Contract details

Token contract details for 21th March 2022

Contract name	LaunchVerse
Contract address	0xB98f2acc693A19330A2B8da776d99DeF05Cf077E
Token supply	1,000,000,000
Token ticker	XLV
Decimals	9
Token holders	1
Transaction count	1
Investment wallet	0xc7a1875adf86720acb946808f543209e59ffe6cb
Marketing and development wallet	0x8ec0955bdbe4a1d9fa9cb92f05487ce404b33ea0
Contract deployer address	-
Contract’s current owner address	0x7b7943394b172be0af3961e01c33ab4bcf195d77

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
Context	**Implementation**			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
IBEP20	**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 🔒		
SafeMathInt	**Library**			
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	sub	Internal 🔒		
└	add	Internal 🔒		
└	abs	Internal 🔒		
└	toUint256Safe	Internal 🔒		
SafeMathUint	**Library**			
└	toInt256Safe	Internal 🔒		

Ownable	**Implementation**	**Context**		
└		Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
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❗
IUniswapV2 Router02	**Interface**	**IUniswapV 2Router01**		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
XLV	**Implementation**	**Context, IBEP20, Ownable**		
└		Public ❗	🛑	Ownable
└	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
└	setInvestmentWallet	External ❗	🛑	onlyOwner
└	changeMaxSellLimit	External ❗	🛑	onlyOwner
└	changeMaxWalletToken	External ❗	🛑	onlyOwner
└	changeSwapAmount	External ❗	🛑	onlyOwner
└	setExcludedFromFee	External ❗	🛑	onlyOwner
└	tradingEnable	External ❗	🛑	onlyOwner
└	updateFees	External ❗	🛑	onlyOwner
└	setSwapAndLiquifyEnabled	Public ❗	🛑	onlyOwner
└		External ❗	💵	NO❗
└	setUniswapRouter	External ❗	🛑	onlyOwner

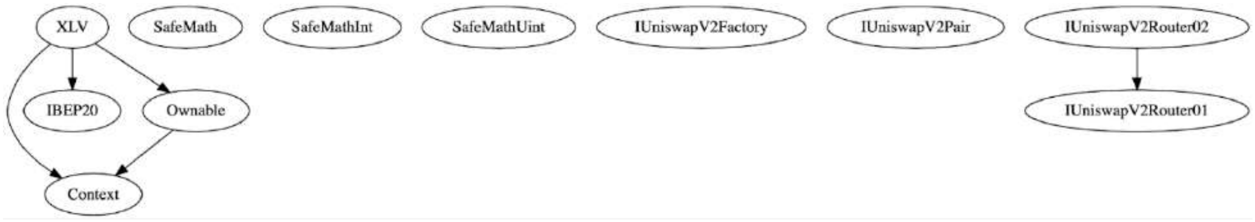
└	setUniswapPair	External ❗	🛑	onlyOwner
└	setExcludedFromAutoLiquidity	External ❗	🛑	onlyOwner
└	_reflectFee	Private 🔐	🛑	
└	_getTValues	Private 🔐		
└	_getRValues	Private 🔐		
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	takeTokenFees	Private 🔐	🛑	
└	takeTransactionFee	Private 🔐	🛑	
└	calculateFee	Private 🔐		
└	isExcludedFromFee	Public ❗		NO❗
└	_approve	Private 🔐	🛑	
└	_transfer	Private 🔐	🛑	
└	swapAndSendBnb	Private 🔐	🛑	lockTheSwap
└	swapAndLiquify	Private 🔐	🛑	
└	swapTokensForBnb	Private 🔐	🛑	
└	addLiquidity	Private 🔐	🛑	
└	_tokenTransfer	Private 🔐	🛑	
└	_transferStandard	Private 🔐	🛑	
└	_transferBothExcluded	Private 🔐	🛑	

└	_transferToExcluded	Private 🔐	🛑	
└	_transferFromExcluded	Private 🔐	🛑	

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

Inheritance Hierarchy



Security issue checking status

- **High severity issues**

  No medium severity issues found.

- **Medium severity issues**

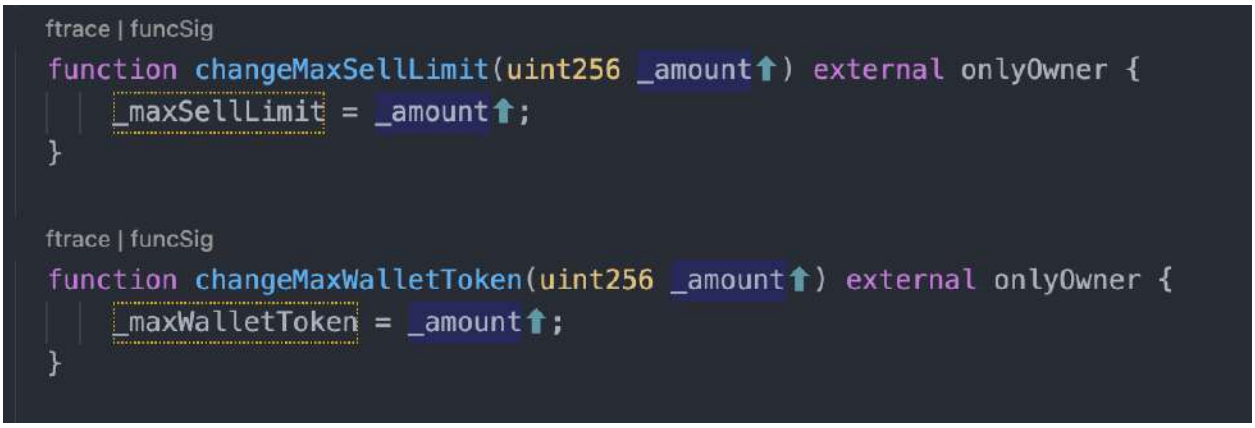
  No medium severity issues found

- **Low severity issues**

  No low severity issues found

- **Centralization risk**

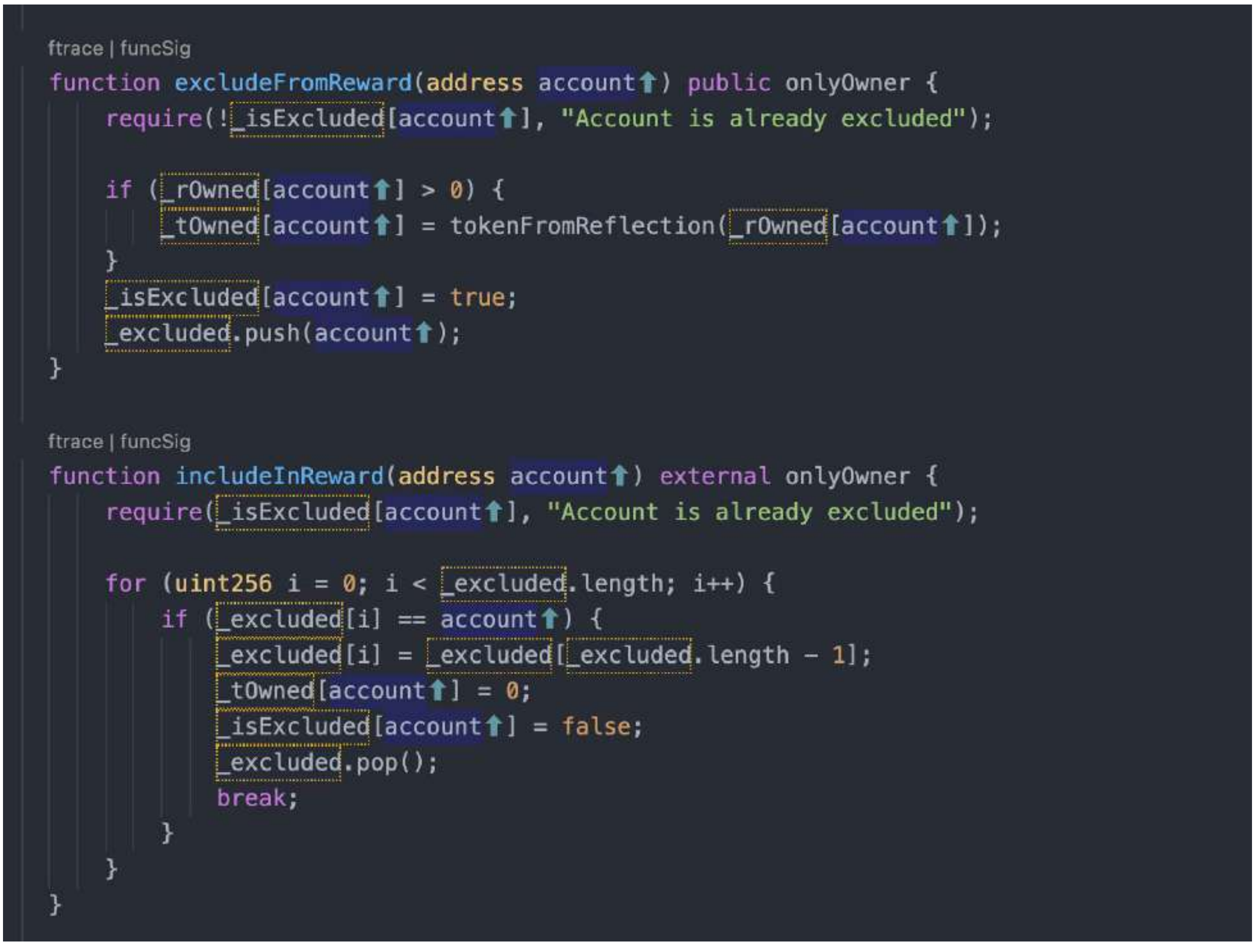
The owner can change max sell amount without minimum limit (can set 0%)



```
ftrace | funcSig
function changeMaxSellLimit(uint256 _amount) external onlyOwner {
        _maxSellLimit = _amount;
}

ftrace | funcSig
function changeMaxWalletToken(uint256 _amount) external onlyOwner {
        _maxWalletToken = _amount;
}
```

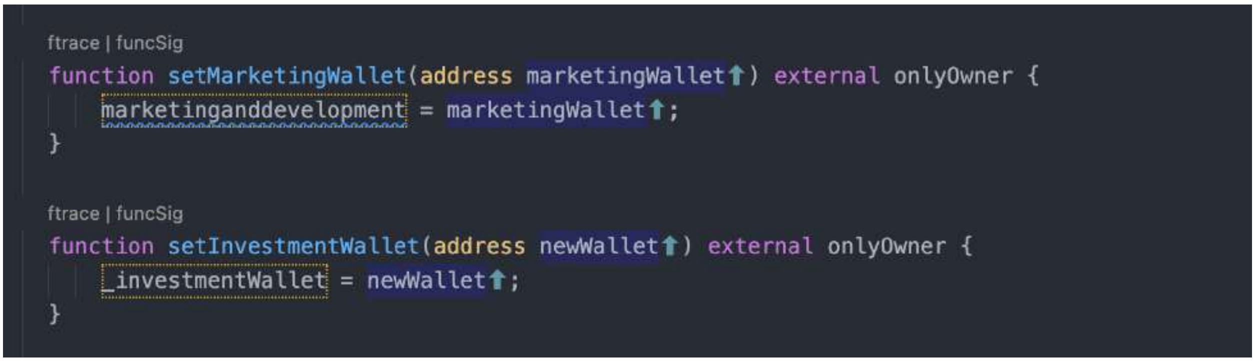
Owner privileges

❖ The owner can include/exclude wallets from fees



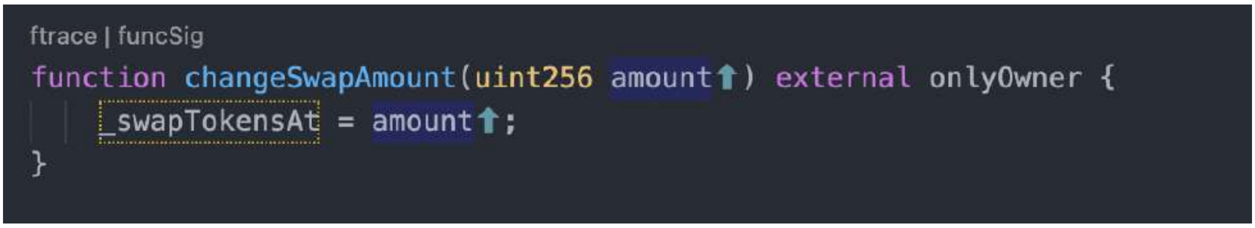
```
ftrace | funcSig
function excludeFromReward(address account) public onlyOwner {
    require(!_isExcluded[account], "Account is already excluded");

    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 investment wallets



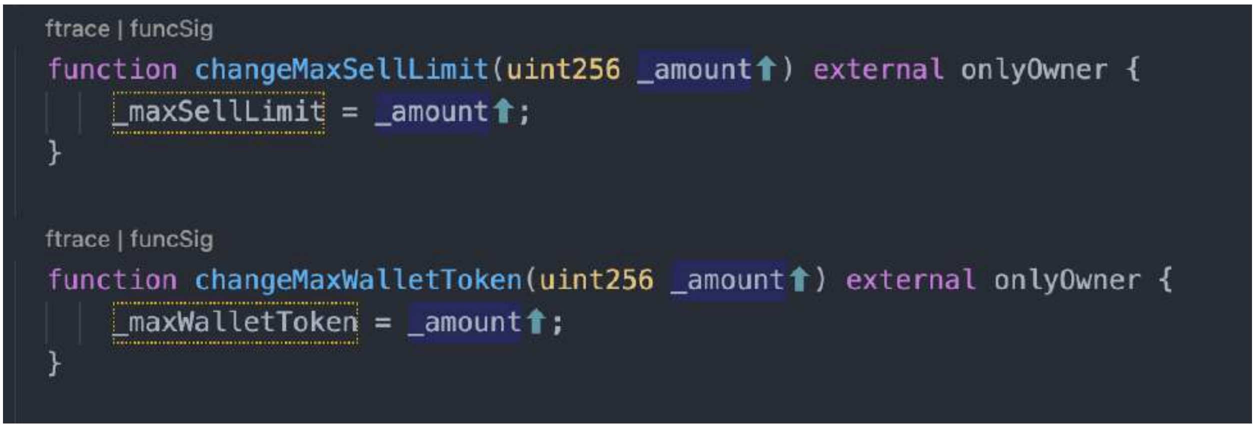
```
ftrace | funcSig
function setMarketingWallet(address marketingWallet) external onlyOwner {
    marketinganddevelopment = marketingWallet;
}

ftrace | funcSig
function setInvestmentWallet(address newWallet) external onlyOwner {
    _investmentWallet = newWallet;
}
```

- ❖ The owner can change swap point



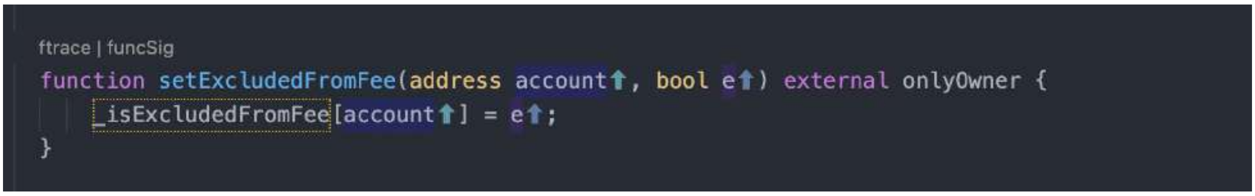
```
ftrace | funcSig
function changeSwapAmount(uint256 amount) external onlyOwner {
        _swapTokensAt = amount;
}
```

- ❖ The owner can change sell token amount and max wallet token



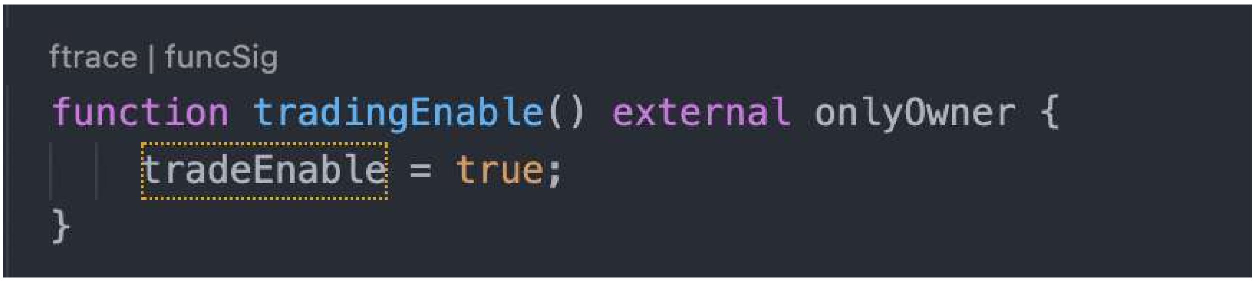
```
ftrace | funcSig
function changeMaxSellLimit(uint256 _amount) external onlyOwner {
        _maxSellLimit = _amount;
}

ftrace | funcSig
function changeMaxWalletToken(uint256 _amount) external onlyOwner {
        _maxWalletToken = _amount;
}
```

- ❖ The owner can include/exclude wallets from fees



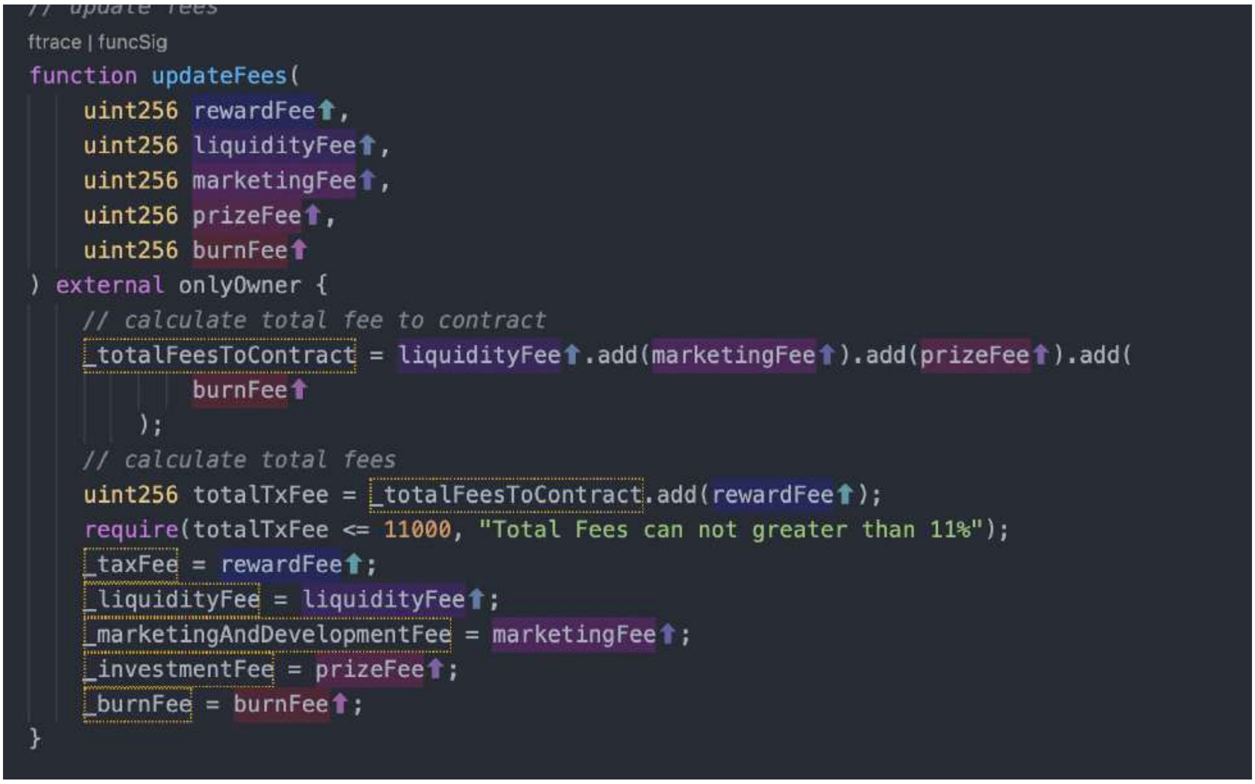
```
ftrace | funcSig
function setExcludedFromFee(address account, bool e) external onlyOwner {
        _isExcludedFromFee[account] = e;
}
```

- ❖ The owner can enable trading once enabled cannot disabled again



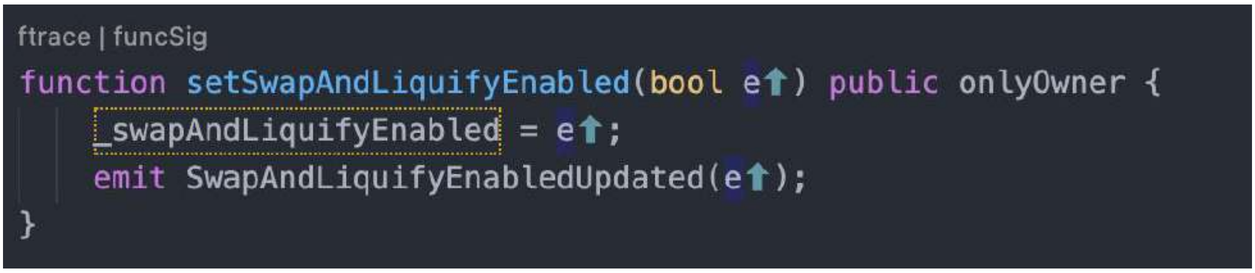
```
ftrace | funcSig
function tradingEnable() external onlyOwner {
        tradeEnable = true;
}
```

- The owner can change all fees maximum up to 11%



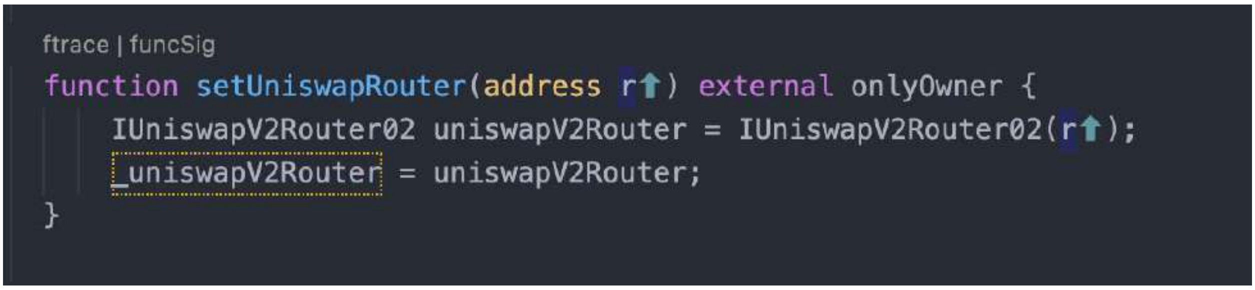
```
// update fees
ftrace | funcSig
function updateFees(
    uint256 rewardFee,
    uint256 liquidityFee,
    uint256 marketingFee,
    uint256 prizeFee,
    uint256 burnFee
) external onlyOwner {
    // calculate total fee to contract
    _totalFeesToContract = liquidityFee.add(marketingFee).add(prizeFee).add(
            burnFee
        );
    // calculate total fees
    uint256 totalTxFee = _totalFeesToContract.add(rewardFee);
    require(totalTxFee <= 11000, "Total Fees can not greater than 11%");
    _taxFee = rewardFee;
    _liquidityFee = liquidityFee;
    _marketingAndDevelopmentFee = marketingFee;
    _investmentFee = prizeFee;
    _burnFee = burnFee;
}
```

- The owner can enable/disable swapping



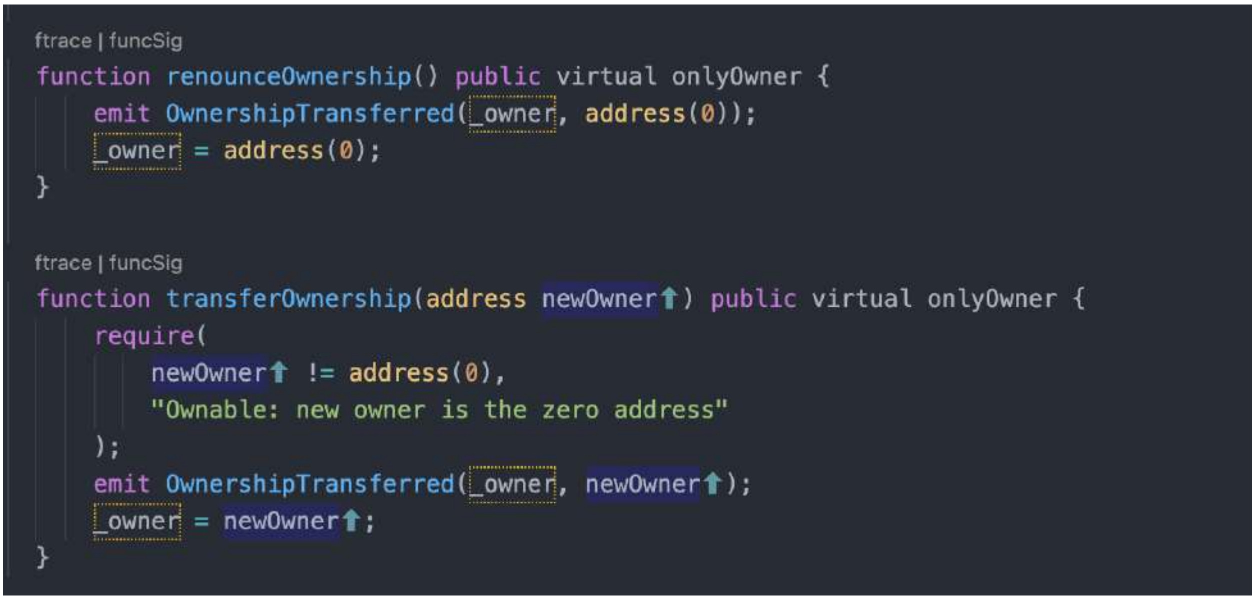
```
ftrace | funcSig
function setSwapAndLiquifyEnabled(bool e) public onlyOwner {
    _swapAndLiquifyEnabled = e;
    emit SwapAndLiquifyEnabledUpdated(e);
}
```

- The owner can change router address



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

- The owner can transfer and renounce ownership



```
ftrace | funcSig
function renounceOwnership() public virtual onlyOwner {
    emit OwnershipTransferred(_owner, address(0));
    _owner = address(0);
}

ftrace | funcSig
function transferOwnership(address newOwner) public virtual onlyOwner {
    require(
        newOwner != address(0),
        "Ownable: new owner is the zero address"
    );
    emit OwnershipTransferred(_owner, newOwner);
    _owner = newOwner;
}
```

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: **1**

Solidity code functional issue level: **PASSED**

Number of owner privileges: **10**

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

Smart contract active ownership: **YES**