RugFreeCoins Audit

RichQuack 2.0 Token
Smart Contract Security Audit

July 9th ,2023

Overview

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

✅ The owner can’t pause trading once it’s enabled

✅ The 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.

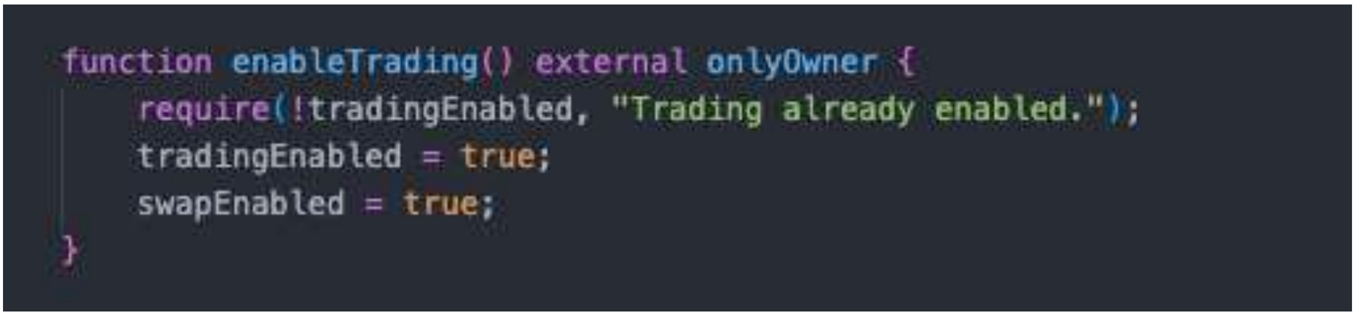
✅ The owner can't change fees by more than 20%.

✅ The owner can't set a max transaction limit

❌ The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.

- **High severity issues**

The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.



```
function enableTrading() external onlyOwner {
    require(!tradingEnabled, "Trading already enabled.");
    tradingEnabled = true;
    swapEnabled = true;
}
```

Contents

Audit details

Audited project
RichQuack 2.0 Token

Contract Address
0x963259e0675c56314d56d74a10251aeeccae50af

Client contact
RichQuack 2.0 Token Team

Blockchain
Binance Smart chain

Project website
http://richquack2.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.

Background

Rugfreecoins was commissioned by the RichQuack 2 Token Team to perform an audit of the smart contract.

https://bscscan.com/token/0x963259e0675c56314d56d74a10251aeeccae50af

This audit focuses on verifying 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.

Tokenomics

0% tax when buying

6% tax when selling

- 2% trade goes to tokens buyback and burn.
- 4% of trade goes to the marketing 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 interested in taking part in the RichQuack 2 token ecosystem.
- Anyone who’s interested in taking part in the future plans of RichQuack 2 Token.
- Anyone who’s interested in making financial transactions with any other party using RichQuack 2 Token as the currency.

Potential to grow with score points

1.	Project efficiency	8/10
2.	Project uniqueness	8/10
3	Information quality	8/10
4	Service quality	8/10
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	9/10
10	Smart contract functionality assessment	9/10
Total Points		**8.3/10**

Contract details

Token contract details for 9th of July 2023

Contract name	RichQuack 2
Contract address	0x963259e0675c56314d56d74a10251aEECCaE50af
Token supply	500,000,000,000,000,000
Token ticker	Quack 2
Decimals	9
Token holders	1
Transaction count	1
Contract deployer address	0x520Ff338966f47DaD0eE592855ba0fb68F09De99
Contract's current owner address	0x520Ff338966f47DaD0eE592855ba0fb68F09De99
Marketing wallet	0x23fda415572a428da86ecfa0d08bb16c6ec7844d

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	Low issue
		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 & centralization	Access control of owners	High Issue
		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
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 ❗

L	balanceOf	External ❗		NO❗
L	allowance	External ❗		NO❗
L	approve	External ❗	🛑	NO❗
L	transfer	External ❗	🛑	NO❗
L	transferFrom	External ❗	🛑	NO❗
L	DOMAIN_SEPARATOR	External ❗		NO❗
L	PERMIT_TYPEHASH	External ❗		NO❗
L	nonces	External ❗		NO❗
L	permit	External ❗	🛑	NO❗
L	MINIMUM_LIQUIDITY	External ❗		NO❗
L	factory	External ❗		NO❗
L	token0	External ❗		NO❗
L	token1	External ❗		NO❗
L	getReserves	External ❗		NO❗
L	price0CumulativeLast	External ❗		NO❗
L	price1CumulativeLast	External ❗		NO❗
L	kLast	External ❗		NO❗
L	mint	External ❗	🛑	NO❗
L	burn	External ❗	🛑	NO❗
L	swap	External ❗	🛑	NO❗
L	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**	**Iuniswap V2 Router01**		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
IERC20	**Interface**			
└	totalSupply	External ❗		NO ❗
└	balanceOf	External ❗		NO ❗
└	transfer	External ❗	🛑	NO ❗
└	allowance	External ❗		NO ❗
└	approve	External ❗	🛑	NO ❗
└	transferFrom	External ❗	🛑	NO ❗
IERC20Metadata	**Interface**	**IERC20**		
└	name	External ❗		NO ❗
└	symbol	External ❗		NO ❗

L	decimals	External ❗		NO❗
Address	**Library**			
L	sendValue	Internal 🔒	🛑	
Context	**Implementation**			
L	_msgSender	Internal 🔒		
L	_msgData	Internal 🔒		
Ownable	**Implementation**	**Context**		
L		Public ❗	🛑	NO❗
L	owner	Public ❗		NO❗
L	renounceOwnership	Public ❗	🛑	onlyOwner
L	transferOwnership	Public ❗	🛑	onlyOwner
ERC20	**Implementation**	**Context, IERC20, IERC20 Metadata**		
L		Public ❗	🛑	NO❗
L	name	Public ❗		NO❗
L	symbol	Public ❗		NO❗
L	decimals	Public ❗		NO❗
L	totalSupply	Public ❗		NO❗
L	balanceOf	Public ❗		NO❗

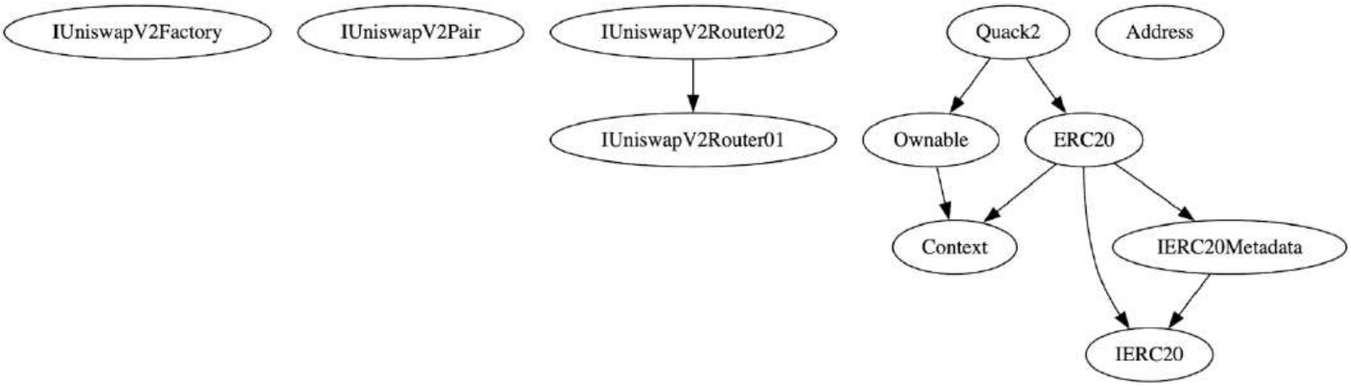
L	transfer	Public ❗	🛑	NO❗
L	allowance	Public ❗		NO❗
L	approve	Public ❗	🛑	NO❗
L	transferFrom	Public ❗	🛑	NO❗
L	increaseAllowance	Public ❗	🛑	NO❗
L	decreaseAllowance	Public ❗	🛑	NO❗
L	_transfer	Internal 🔒	🛑	
L	_mint	Internal 🔒	🛑	
L	_burn	Internal 🔒	🛑	
L	_approve	Internal 🔒	🛑	
L	_beforeTokenTransfer	Internal 🔒	🛑	
L	_afterTokenTransfer	Internal 🔒	🛑	
Quack2	**Implementation**	**ERC20, Ownable**		
L		Public ❗	🛑	ERC20
L		External ❗	💵	NO❗
L	claimStuckTokens	External ❗	🛑	onlyOwner
L	excludeFromFees	External ❗	🛑	onlyOwner
L	isExcludedFromFees	Public ❗		NO❗
L	updateFees	External ❗	🛑	onlyOwner
L	changeMarketingWallet	External ❗	🛑	onlyOwner

└	enableTrading	External ❗	🛑	onlyOwner
└	_transfer	Internal 🔒	🛑	
└	setSwapEnabled	External ❗	🛑	onlyOwner
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	swapAndSend	Private 🔐	🛑	

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

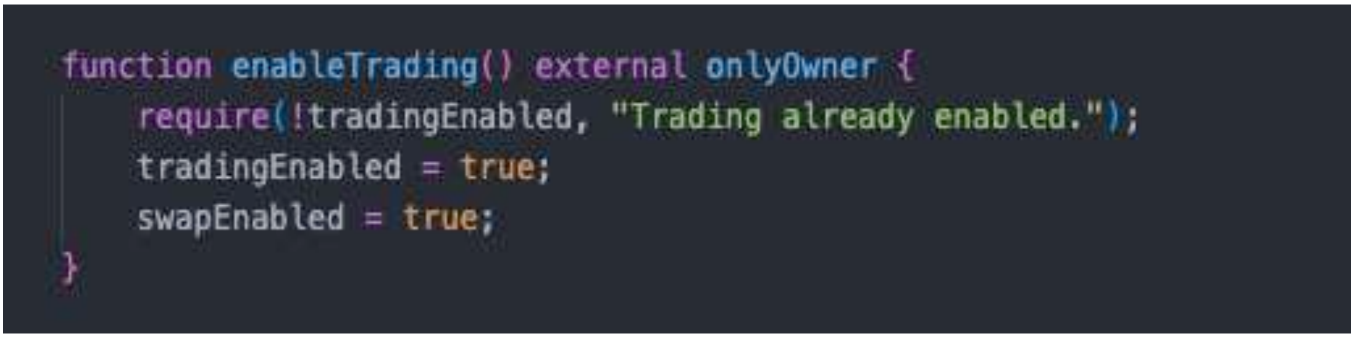
Inheritance Hierarchy



Security issue checking status

❖ **High severity issues**

The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.



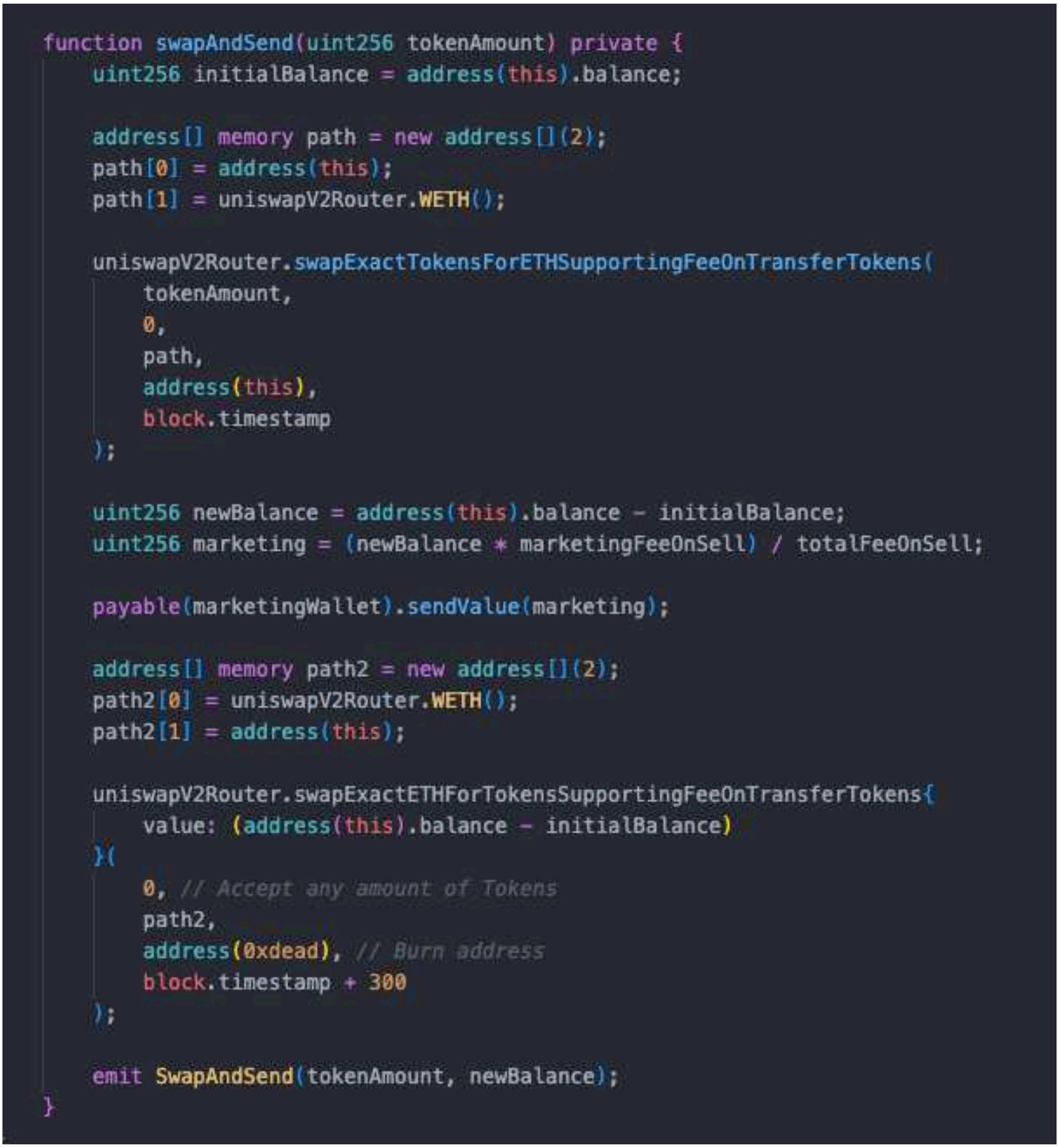
```
function enableTrading() external onlyOwner {
    require(!tradingEnabled, "Trading already enabled.");
    tradingEnabled = true;
    swapEnabled = true;
}
```

❖ **Medium severity issues**

No medium severity issues found

- **Low severity issues**

When burning tokens, instead of selling the burning tokens portion to BNB and buying tokens again, it can directly burn them without selling to BNB, thereby reducing the gas fee.

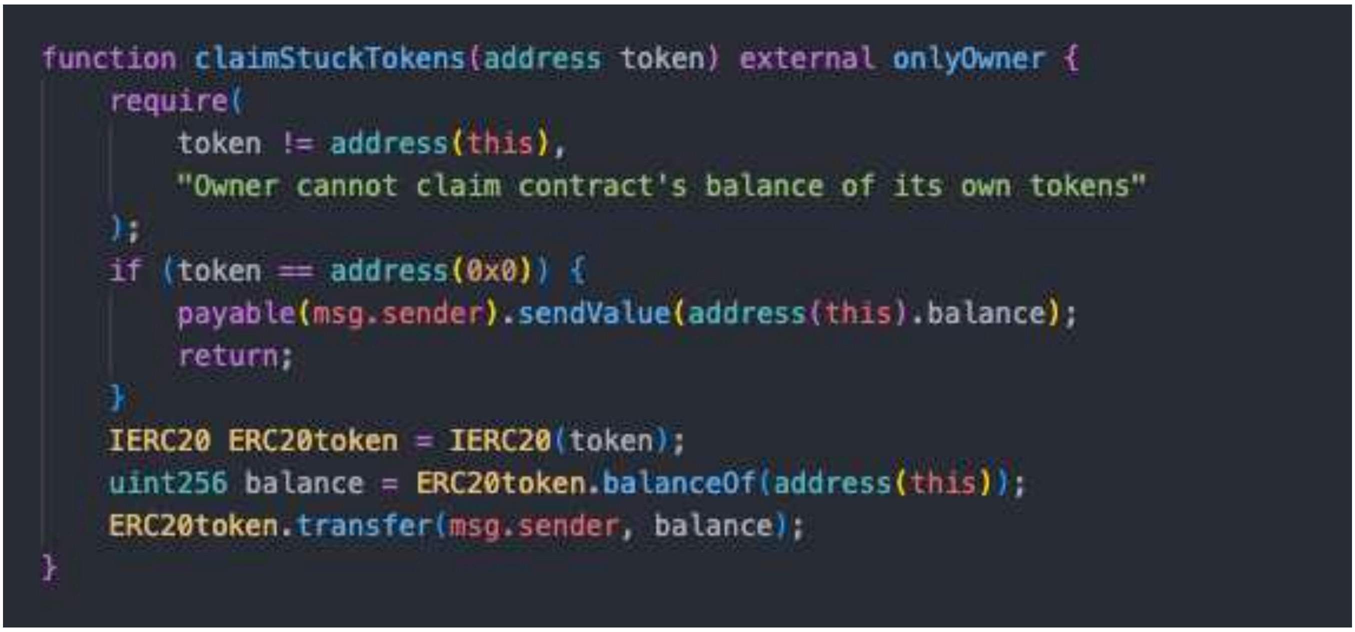


```
function swapAndSend(uint256 tokenAmount) private {
    uint256 initialBalance = address(this).balance;

    address[] memory path = new address[](2);
    path[0] = address(this);
    path[1] = uniswapV2Router.WETH();

    uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
        tokenAmount,
        0,
        path,
        address(this),
        block.timestamp
    );

    uint256 newBalance = address(this).balance - initialBalance;
    uint256 marketing = (newBalance * marketingFeeOnSell) / totalFeeOnSell;

    payable(marketingWallet).sendValue(marketing);

    address[] memory path2 = new address[](2);
    path2[0] = uniswapV2Router.WETH();
    path2[1] = address(this);

    uniswapV2Router.swapExactETHForTokensSupportingFeeOnTransferTokens{
        value: (address(this).balance - initialBalance)
    }(
        0, // Accept any amount of Tokens
        path2,
        address(0xdead), // Burn address
        block.timestamp + 300
    );

    emit SwapAndSend(tokenAmount, newBalance);
}
```

- **Centralization Risk**
  No Centralization issues found

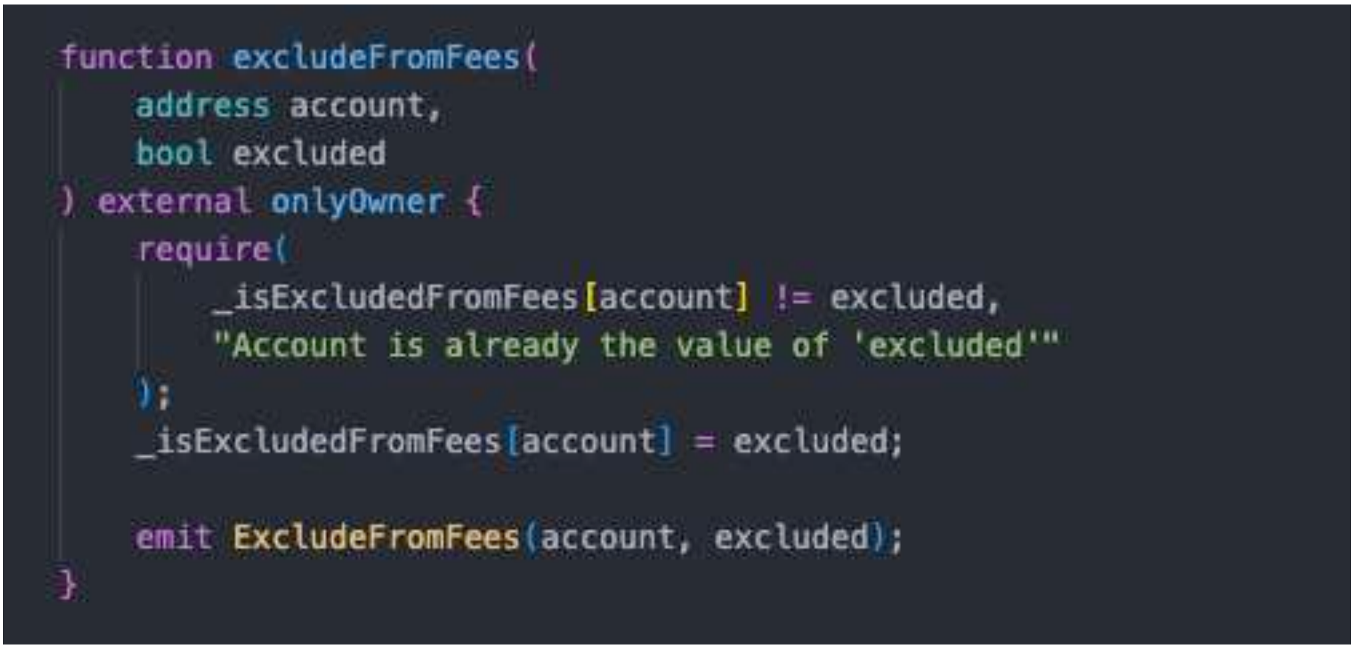
Owner privileges

- Owner can get any bep20 tokens from contract to owner wallet (Can not get native tokens)



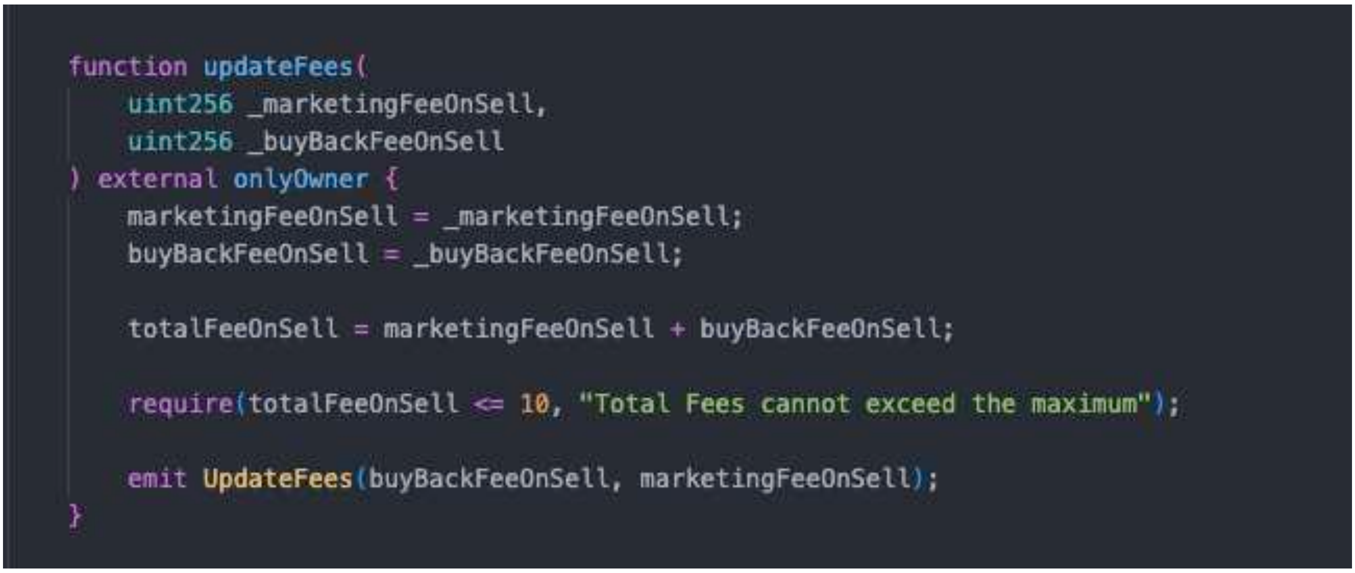
```
function claimStuckTokens(address token) external onlyOwner {
    require(
        token != address(this),
        "Owner cannot claim contract's balance of its own tokens"
    );
    if (token == address(0x0)) {
        payable(msg.sender).sendValue(address(this).balance);
        return;
    }
    IERC20 ERC20token = IERC20(token);
    uint256 balance = ERC20token.balanceOf(address(this));
    ERC20token.transfer(msg.sender, balance);
}
```

- Owner can include/exclude wallets from fees



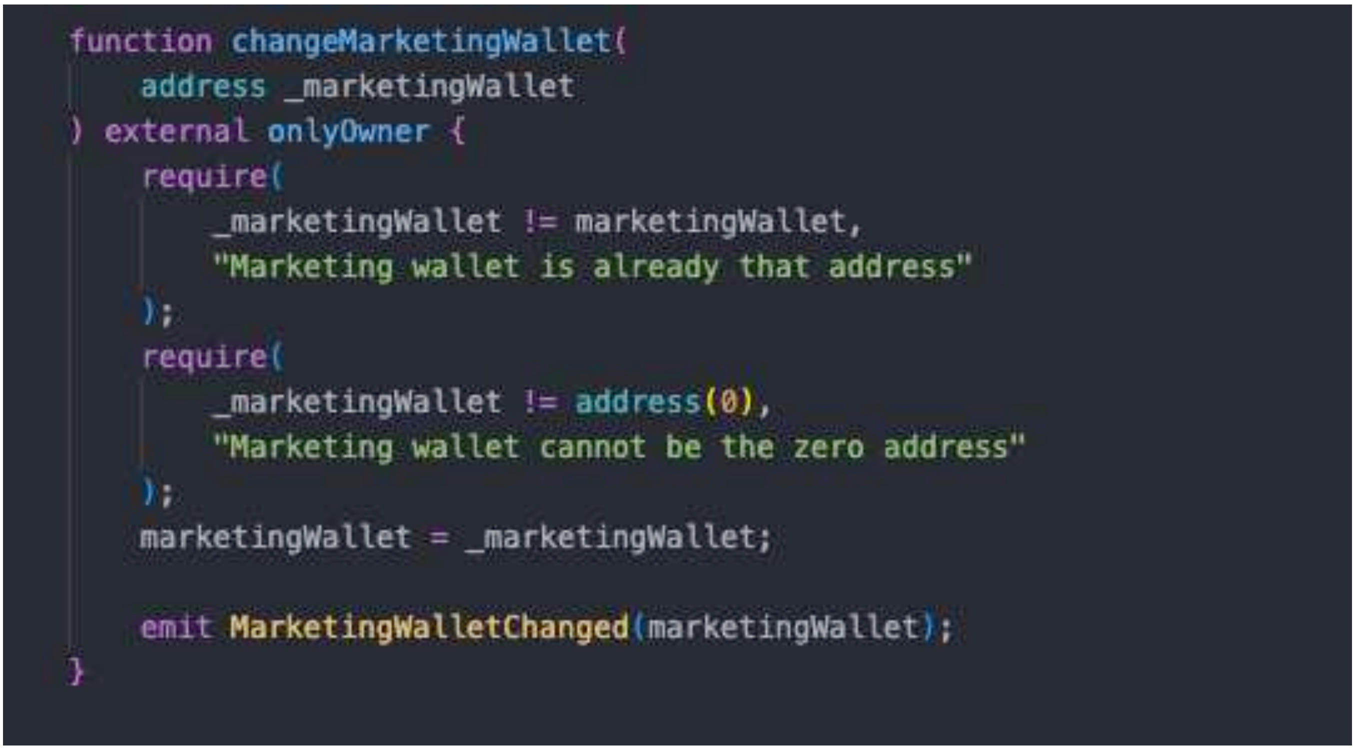
```
function excludeFromFees(
    address account,
    bool excluded
) external onlyOwner {
    require(
        _isExcludedFromFees[account] != excluded,
        "Account is already the value of 'excluded'"
    );
    _isExcludedFromFees[account] = excluded;

    emit ExcludeFromFees(account, excluded);
}
```

- Owner can change total sell fees maximum upto 10%



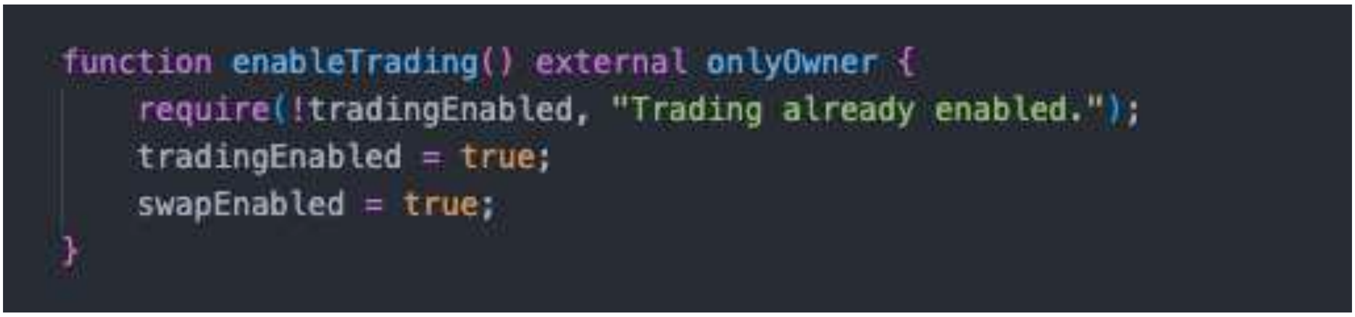
```
function updateFees(
    uint256 _marketingFeeOnSell,
    uint256 _buyBackFeeOnSell
) external onlyOwner {
    marketingFeeOnSell = _marketingFeeOnSell;
    buyBackFeeOnSell = _buyBackFeeOnSell;

    totalFeeOnSell = marketingFeeOnSell + buyBackFeeOnSell;

    require(totalFeeOnSell <= 10, "Total Fees cannot exceed the maximum");

    emit UpdateFees(buyBackFeeOnSell, marketingFeeOnSell);
}
```

- Owner can change marketing wallet



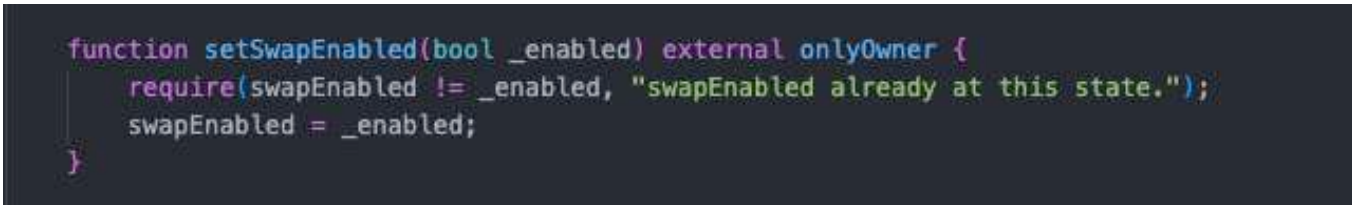
```
function changeMarketingWallet(
    address _marketingWallet
) external onlyOwner {
    require(
        _marketingWallet != marketingWallet,
        "Marketing wallet is already that address"
    );
    require(
        _marketingWallet != address(0),
        "Marketing wallet cannot be the zero address"
    );
    marketingWallet = _marketingWallet;

    emit MarketingWalletChanged(marketingWallet);
}
```

- Owner can enable trading,once enabled can not disable again



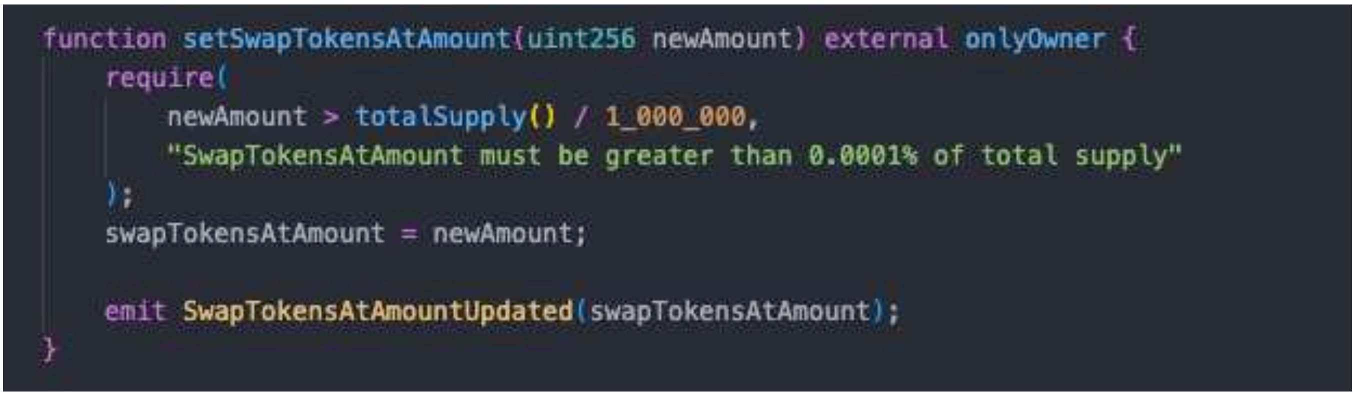
```
function enableTrading() external onlyOwner {
    require(!tradingEnabled, "Trading already enabled.");
    tradingEnabled = true;
    swapEnabled = true;
}
```

- Owner can enable/disable swapping



```
function setSwapEnabled(bool _enabled) external onlyOwner {
    require(swapEnabled != _enabled, "swapEnabled already at this state.");
    swapEnabled = _enabled;
}
```

- Owner can change swap point minimum up to 0.0001%



```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner {
    require(
        newAmount > totalSupply() / 1_000_000,
        "SwapTokensAtAmount must be greater than 0.0001% of total supply"
    );
    swapTokensAtAmount = newAmount;

    emit SwapTokensAtAmountUpdated(swapTokensAtAmount);
}
```

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: **2**

Solidity code functional issue level: **PASS**

Number of owner privileges: **7**

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

Smart contract active ownership: **ACTIVE**