RugFreeCoins Audit

THONIC Token
Smart Contract Security Audit

September 27th, 2022

Contents

Audit details

Audited project
THONIC Token

Contract Address

0x41DE98caD918Ad2BF2D9c1e61B53537dbbdB29d8

Client contact

THONIC Team

Blockchain

Binance smart chain

Project website

https://thonic.finance

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.

☑ There are no max tx limits and max wallet limits in the contract.

☑ The owner can’t pause trading.

☑ The owner can't set fees over 25%.

☑ Owner can't blacklist wallets.

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

Background

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

https://bscscan.com/token/00x41DE98caD918Ad2BF2D9c1e61B53537dbbdB29d8

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.

.

Roadmap

Phase 01

- Whitepaper release
- Website launch
- Contract Deployment
- KYC, dox & audit
- Raising capital

Phase 02

- Private Sale
- Social Development
- Marketing Development
- 1000 telegram followers
- Accepting requests for the whitelist presale Whitepaper release
- Website launch
- Contract Deployment
- KYC, dox & audit
- Raising capital

Phase 03

- Presale
- Coin market cap application
- Coin gecko application
- 2000 telegram followers
- 500 holders
- Add 2 ambassadors to team

Phase 04

- Poocoin ads
- Coin market cap ads
- Marketing outreach
- How to buy videos
- 2000 holders

Phase 05

- Alpha launch of the platform
- Earnings calculator
- Direct token purchase
- Swap exchange
- Add 5 analysts to the team

Phase 06

- Beta launch of platform
- Report notification and subscriptions
- Portfolio management
- Add 10 analysts to the team
- 250 platform subscribers

Tokenomics

8% when buying & selling

- 5% of trade goes to the distribution of rewards among investors in BUSD.
- 2% of trade goes to the company wallet in BUSD.
- 1% 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 ready to staking and receive rewards.
- Anyone who’s interested in taking part in the future plans of the THONIC tokenn.
- Anyone who’s interested in making financial transactions with any other party using THONIC Token as the currency.

Potential to grow with score points

1.	Project efficiency	8/10
2.	Project uniqueness	8/10
3	Information quality	9/10
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	9/10
9	Smart contract security	10/10
10	Smart contract functionality assessment	10/10
Total Points		**8.9/10**

Contract details

Token contract details for 27th of September 2022

Contract name	THONIC
Contract address	0x41DE98caD918Ad2BF2D9c1e61B53537dbbdB29d8
Token supply	100,000,000,000
Token ticker	THONIC
Decimals	9
Token holders	1
Transaction count	1
Dividend tracker	0xad3efc96ca3940204ac47e126fe0b0033c0592a2
Company fee receiver	0x97fb30609742f91620fcbfa9f082cfdf952e1307
Contract deployer address	0x16e153d10e6822dbd815b9385bf3b5aebea6e765
Contract’s current owner address	0x16e153d10e6822dbd815b9385bf3b5aebea6e765

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
THONIC	**Implementation**	**IERC20, Ownable**		
└		Public ❗	🛑	NO❗
└		External ❗	💵	NO❗
└	totalSupply	External ❗		NO❗
└	name	Public ❗		NO❗
└	symbol	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	getHolderDetails	Public ❗		NO❗
└	getLastProcessedIndex	Public ❗		NO❗
└	getNumberOfTokenHolders	Public ❗		NO❗
└	totalDistributedRewards	Public ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	Public ❗	🛑	NO❗
└	_approve	Internal 🔒	🛑	

└	approveMax	External ❗	🛑	NO❗
└	transfer	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
└	_transferFrom	Internal 🔒	🛑	
└	_basicTransfer	Internal 🔒	🛑	
└	shouldTakeFee	Internal 🔒		
└	takeFee	Internal 🔒	🛑	
└	shouldSwapBack	Internal 🔒		
└	clearStuckBalance	External ❗	🛑	onlyOwner
└	getBep20Tokens	External ❗	🛑	onlyOwner
└	updateBuyFees	Public ❗	🛑	onlyOwner
└	updateSellFees	Public ❗	🛑	onlyOwner
└	updateSwapPercentages	Public ❗	🛑	onlyOwner
└	enableTrading	Public ❗	🛑	onlyOwner
└	whitelistPreSale	Public ❗	🛑	onlyOwner
└	___claimRewards	Public ❗	🛑	NO❗
└	claimProcess	Public ❗	🛑	NO❗
└	isRewardExclude	Public ❗		NO❗
└	isFeeExclude	Public ❗		NO❗
└	swapBackInBnb	Internal 🔒	🛑	swapping
└	swapAndLiquify	Private 🔐	🛑	

└	swapTokensForEth	Private 🔐	🛑	
└	swapTokensForTokens	Private 🔐	🛑	
└	addLiquidity	Private 🔐	🛑	
└	setIsDividendExempt	External ❗	🛑	onlyOwner
└	setIsFeeExempt	External ❗	🛑	onlyOwner
└	addAuthorizedWallets	External ❗	🛑	onlyOwner
└	setcompanyWallet	External ❗	🛑	onlyOwner
└	setSwapBackSettings	External ❗	🛑	onlyOwner
└	setDistributionCriteria	External ❗	🛑	onlyOwner
└	setDistributorSettings	External ❗	🛑	onlyOwner
└	purgeBeforeSwitch	Public ❗	🛑	onlyOwner
└	includeMeinRewards	Public ❗	🛑	NO❗
└	switchToken	Public ❗	🛑	onlyOwner
IDividend Distributor	**Interface**			
└	setDistributionCriteria	External ❗	🛑	NO❗
└	setShare	External ❗	🛑	NO❗
└	deposit	External ❗	🛑	NO❗
└	process	External ❗	🛑	NO❗
└	purge	External ❗	🛑	NO❗

Dividend Distributor	**Implementation**	**IDividend Distributor**		
└		Public ❗️	🛑	NO❗️
└		External ❗️	💵	NO❗️
└	setDistributionCriteria	External ❗️	🛑	onlyToken
└	purge	External ❗️	🛑	onlyToken
└	setShare	External ❗️	🛑	onlyToken
└	deposit	External ❗️	🛑	onlyToken
└	process	External ❗️	🛑	onlyToken
└	shouldDistribute	Internal 🔒		
└	distributeDividend	Internal 🔒	🛑	
└	claimDividend	External ❗️	🛑	NO❗️
└	claimDividendTo	External ❗️	🛑	NO❗️
└	getUnpaidEarnings	Public ❗️		NO❗️
└	getHolderDetails	Public ❗️		NO❗️
└	getCumulativeDividends	Internal 🔒		
└	getLastProcessedIndex	External ❗️		NO❗️
└	getNumberOfTokenHolders	External ❗️		NO❗️
└	getShareHoldersList	External ❗️		NO❗️
└	totalDistributedRewards	External ❗️		NO❗️
└	addShareholder	Internal 🔒	🛑	

└	removeShareholder	Internal 🔒	🛑	
Ownable	**Implementation**	**Context**		
└		Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	_checkOwner	Internal 🔒		
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_transferOwnership	Internal 🔒	🛑	
IERC20	**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 🔒		
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❗

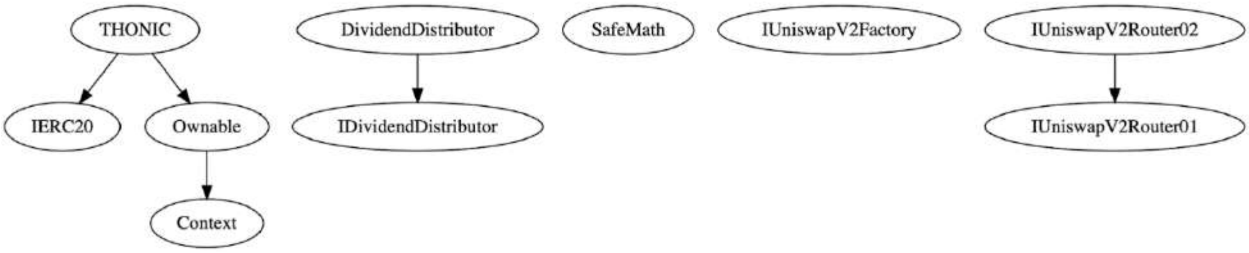
IUniswapV2Router02	Interface	IUniswapV2Router01		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
IUniswapV2Router01	**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❗
Context	**Implementation**			
└	_msgSender	Internal 🔒		
└	_msgData	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

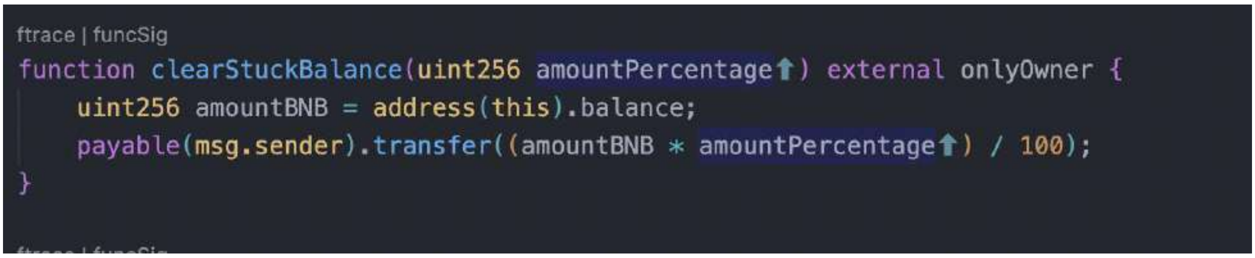
- **Medium severity issues**
  No medium severity issues found

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

- **Centralization Risk**
  No Centralization Risk found

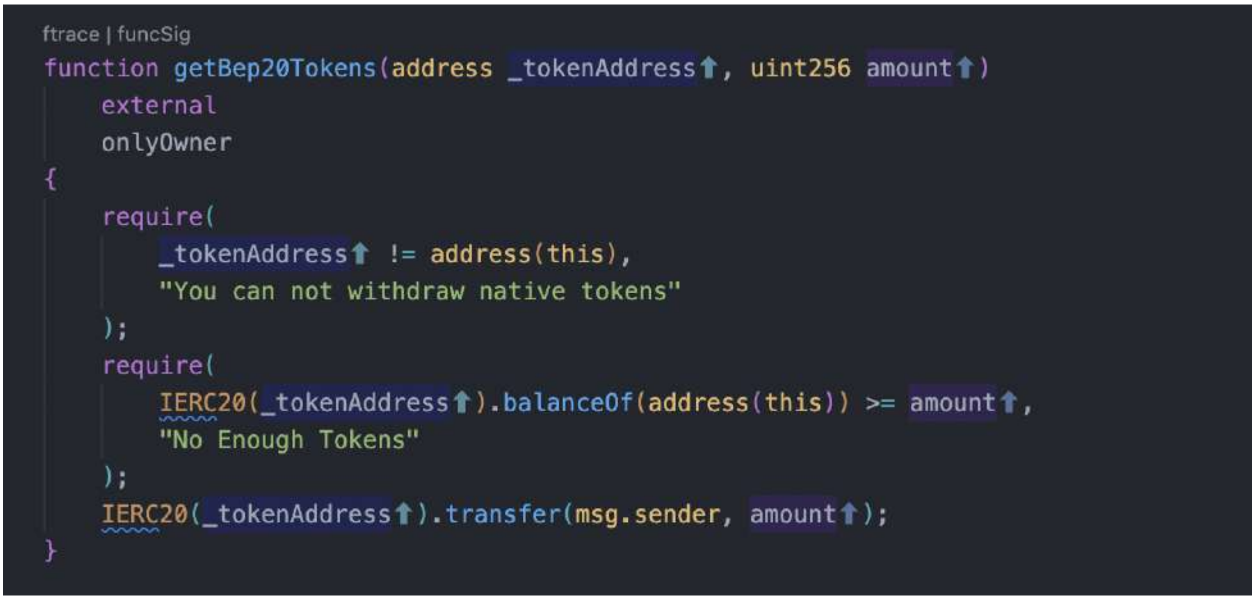
Owner privileges

❖ The owner can get the contract BNB balance to the owner wallet



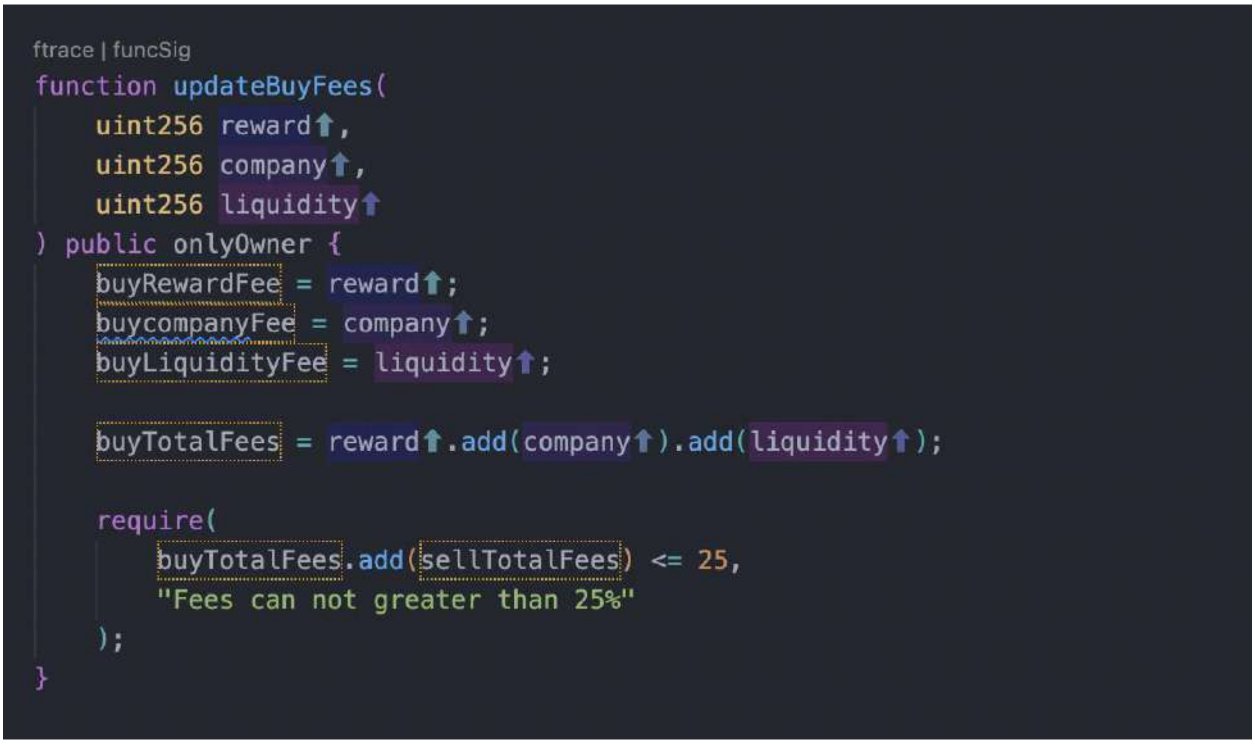
```
ftrace | funcSig
function clearStuckBalance(uint256 amountPercentage) external onlyOwner {
    uint256 amountBNB = address(this).balance;
    payable(msg.sender).transfer((amountBNB * amountPercentage) / 100);
}
```

❖ The owner can get any bep20 tokens in the contract to owner wallet (cannot get native tokens)



```
ftrace | funcSig
function getBep20Tokens(address _tokenAddress, uint256 amount)
    external
    onlyOwner
{
    require(
        _tokenAddress != address(this),
        "You can not withdraw native tokens"
    );
    require(
        IERC20(_tokenAddress).balanceOf(address(this)) >= amount,
        "No Enough Tokens"
    );
    IERC20(_tokenAddress).transfer(msg.sender, amount);
}
```

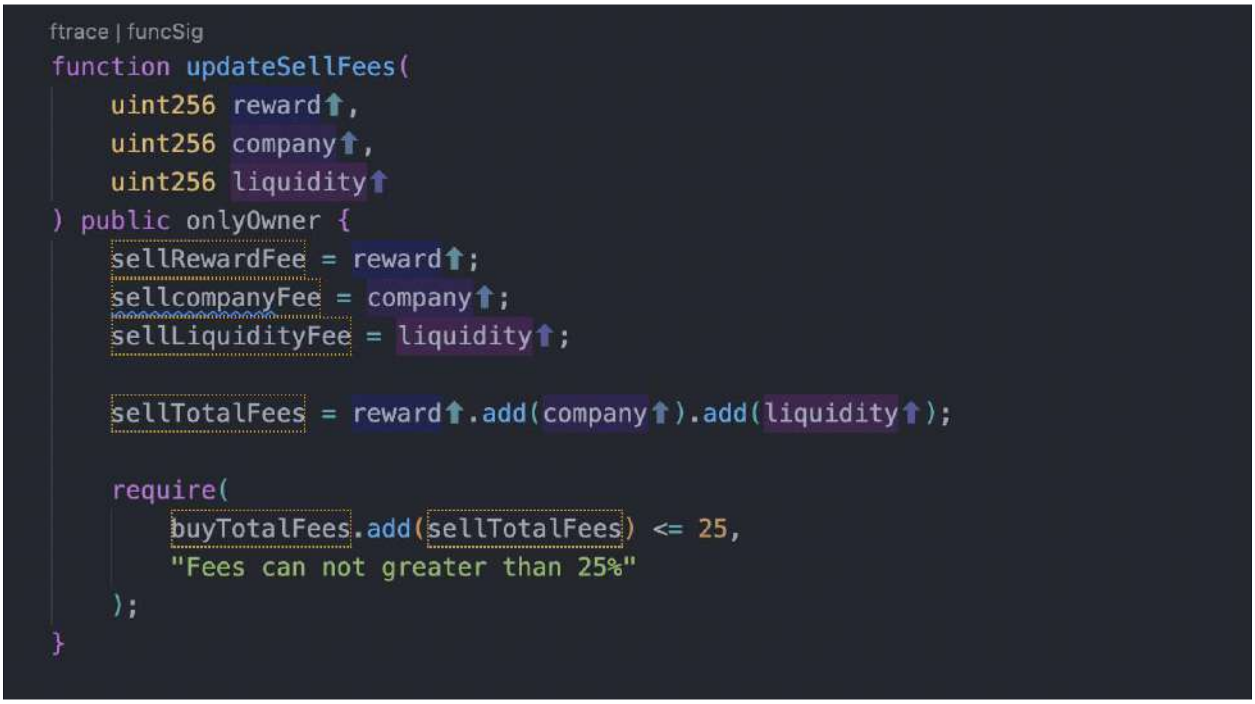
- ❖ The owner can update all buy fees, total fees maximum up to 25%



ftrace | funcSig

```
function updateBuyFees(
    uint256 reward,
    uint256 company,
    uint256 liquidity
) public onlyOwner {
    buyRewardFee = reward;
    buycompanyFee = company;
    buyLiquidityFee = liquidity;

    buyTotalFees = reward.add(company).add(liquidity);

    require(
        buyTotalFees.add(sellTotalFees) <= 25,
        "Fees can not greater than 25%"
    );
}
```

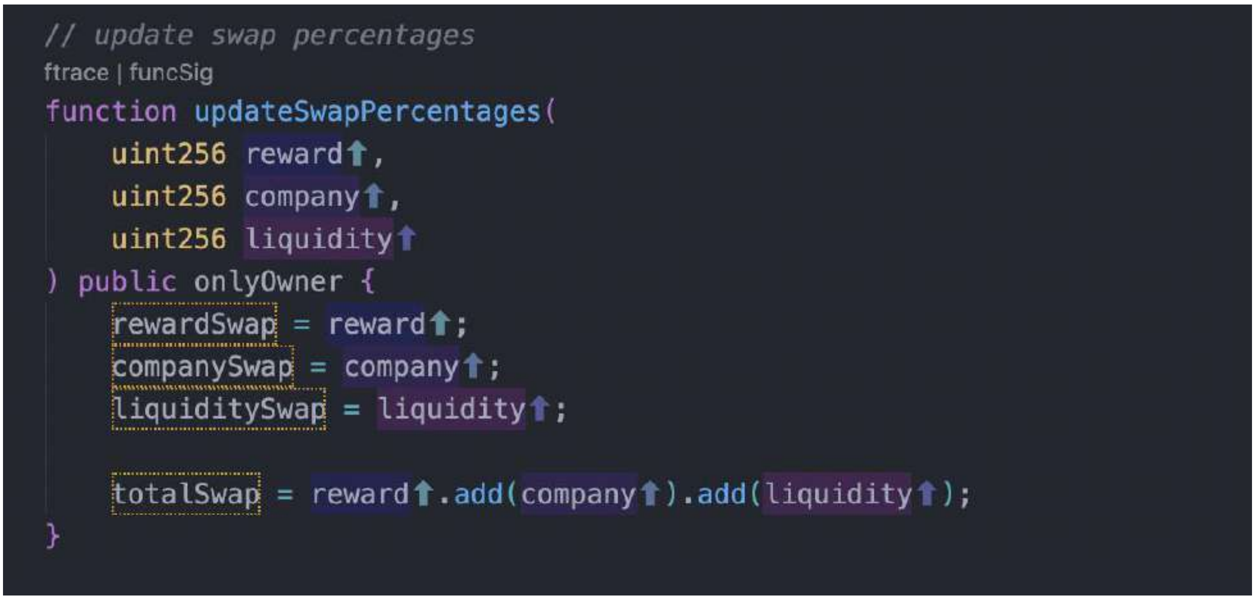
- ❖ The owner can update all sell fees, total fees maximum up to 25%



ftrace | funcSig

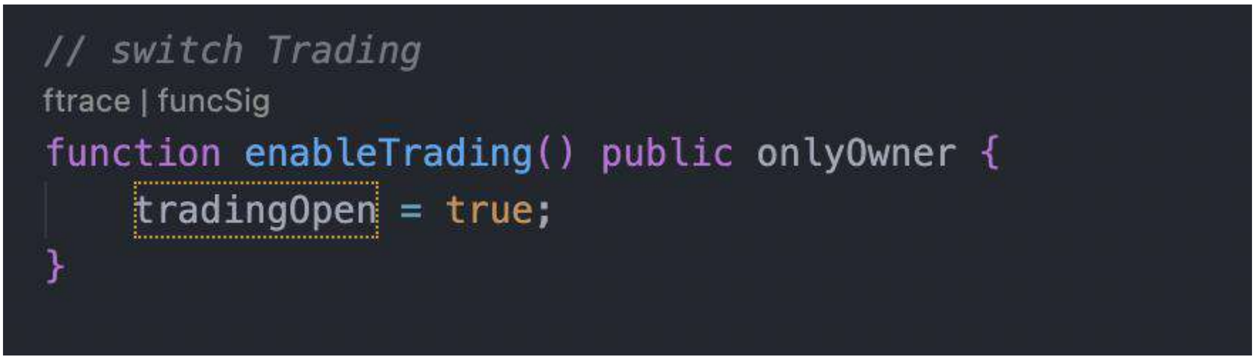
```
function updateSellFees(
    uint256 reward,
    uint256 company,
    uint256 liquidity
) public onlyOwner {
    sellRewardFee = reward;
    sellcompanyFee = company;
    sellLiquidityFee = liquidity;

    sellTotalFees = reward.add(company).add(liquidity);

    require(
        buyTotalFees.add(sellTotalFees) <= 25,
        "Fees can not greater than 25%"
    );
}
```

- The owner can change all swap percentages



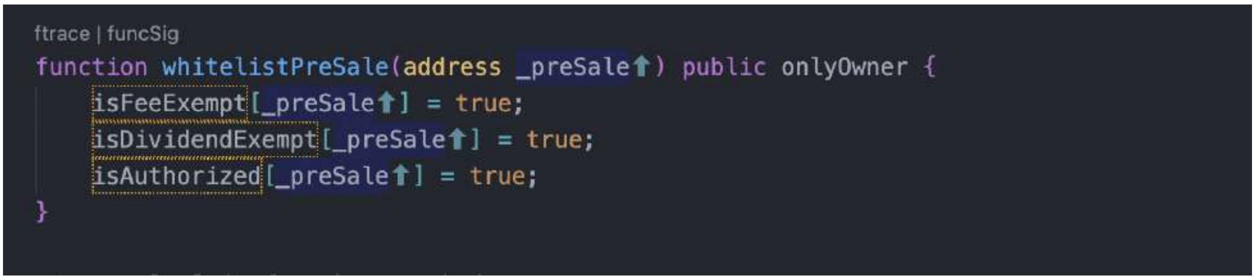
```
// update swap percentages
ftrace | funcSig
function updateSwapPercentages(
    uint256 reward,
    uint256 company,
    uint256 liquidity
) public onlyOwner {
    rewardSwap = reward;
    companySwap = company;
    liquiditySwap = liquidity;

    totalSwap = reward.add(company).add(liquidity);
}
```

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



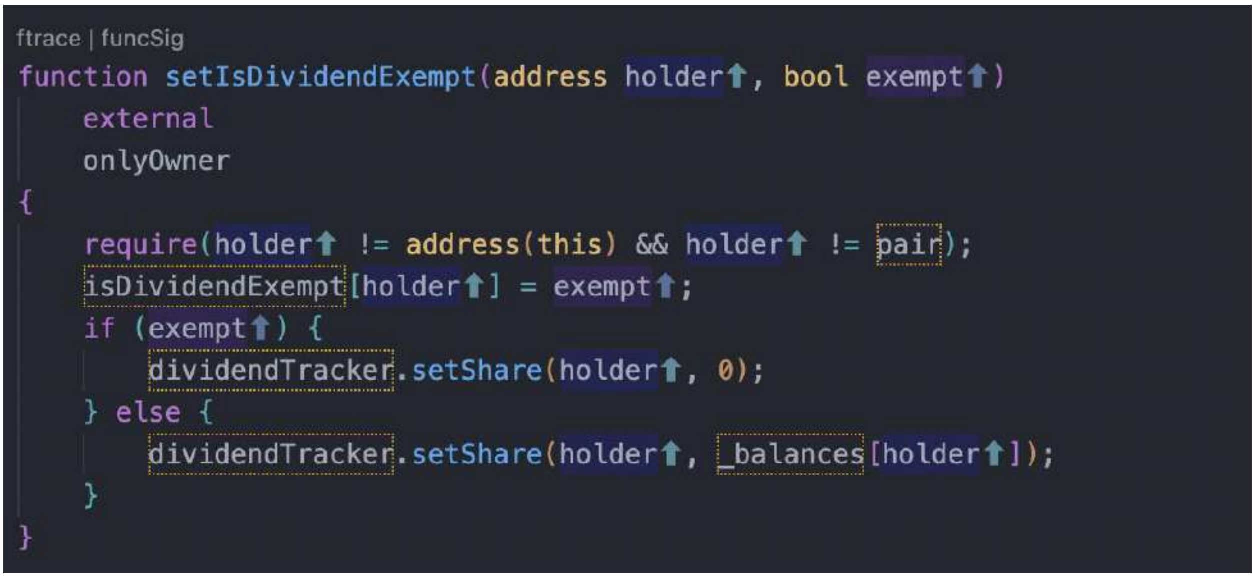
```
// switch Trading
ftrace | funcSig
function enableTrading() public onlyOwner {
    tradingOpen = true;
}
```

- The owner can whitelist presale



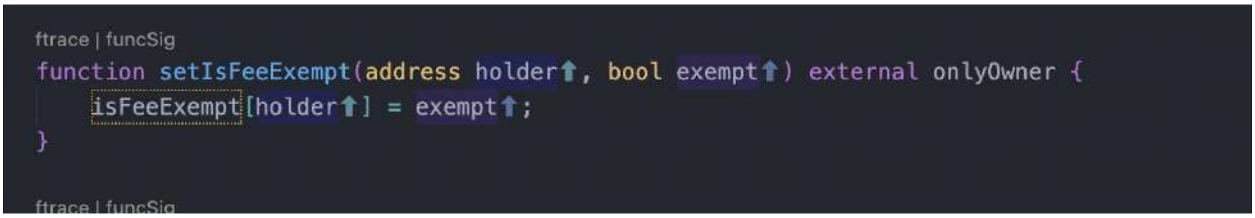
```
ftrace | funcSig
function whitelistPreSale(address _preSale) public onlyOwner {
    isFeeExempt[_preSale] = true;
    isDividendExempt[_preSale] = true;
    isAuthorized[_preSale] = true;
}
```

❖ The owner can exclude/include wallets from rewards



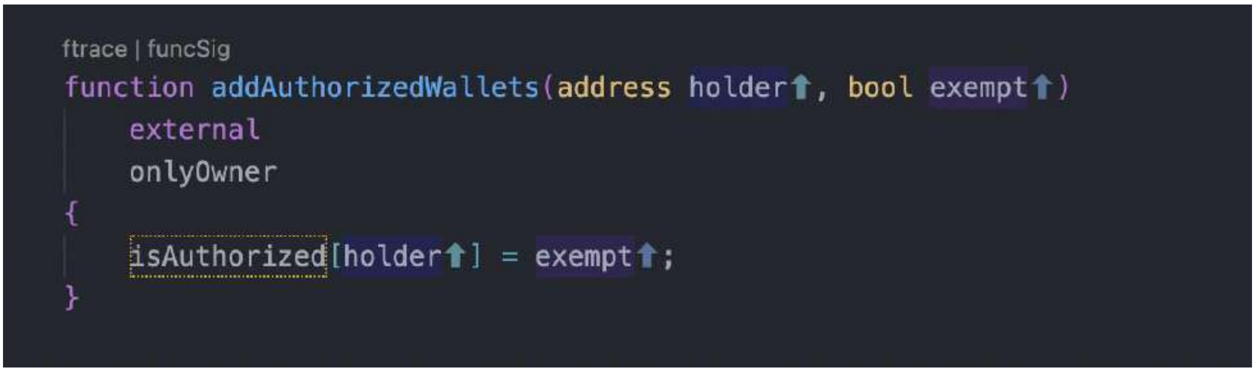
```
ftrace | funcSig
function setIsDividendExempt(address holder, bool exempt)
    external
    onlyOwner
{
    require(holder != address(this) && holder != pair);
    isDividendExempt[holder] = exempt;
    if (exempt) {
        dividendTracker.setShare(holder, 0);
    } else {
        dividendTracker.setShare(holder, _balances[holder]);
    }
}
```

❖ The owner can include/exclude wallets from fees



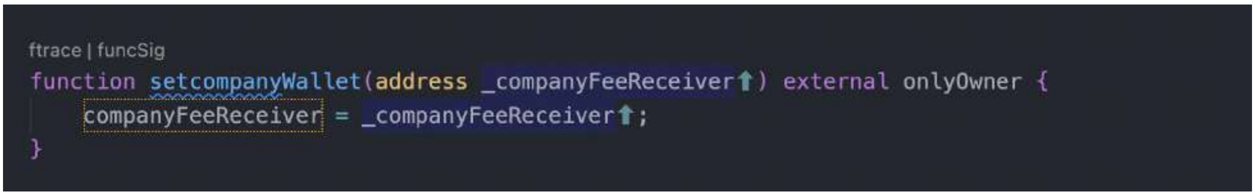
```
ftrace | funcSig
function setIsFeeExempt(address holder, bool exempt) external onlyOwner {
    isFeeExempt[holder] = exempt;
}

ftrace | funcSig
```

❖ The owner can add/remove authorized wallets



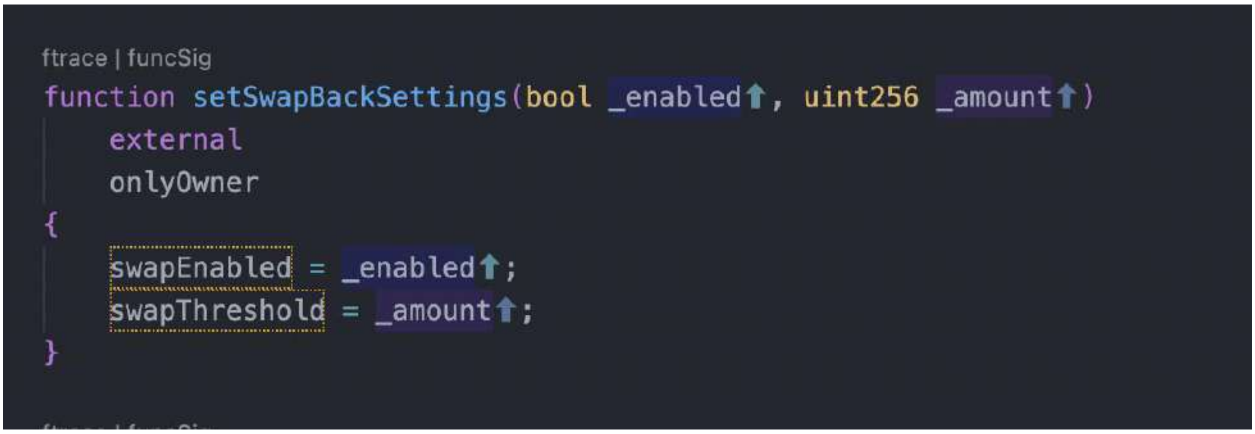
```
ftrace | funcSig
function addAuthorizedWallets(address holder, bool exempt)
    external
    onlyOwner
{
    isAuthorized[holder] = exempt;
}
```

❖ The owner can change company wallet



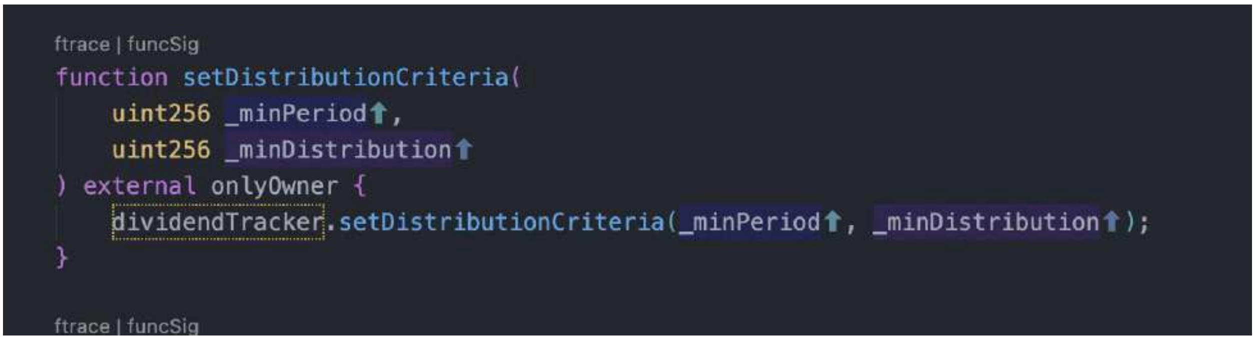
```
ftrace | funcSig
function setcompanyWallet(address _companyFeeReceiver) external onlyOwner {
    companyFeeReceiver = _companyFeeReceiver;
}
```

- The owner can change swap back settings



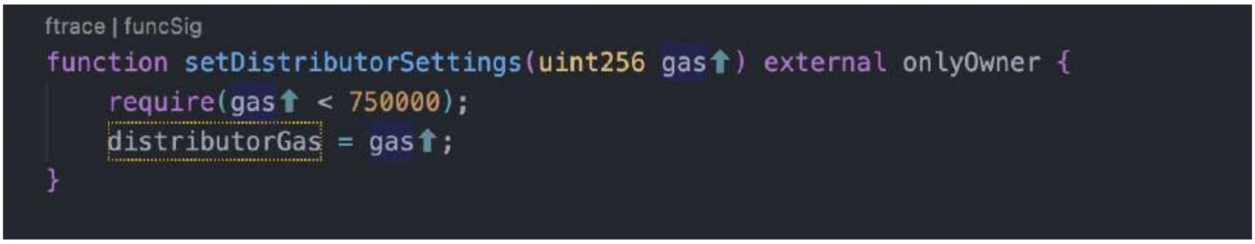
```
ftrace | funcSig
function setSwapBackSettings(bool _enabled, uint256 _amount)
    external
    onlyOwner
{
    swapEnabled = _enabled;
    swapThreshold = _amount;
}
```

- The owner can change minimum distribution amount and period



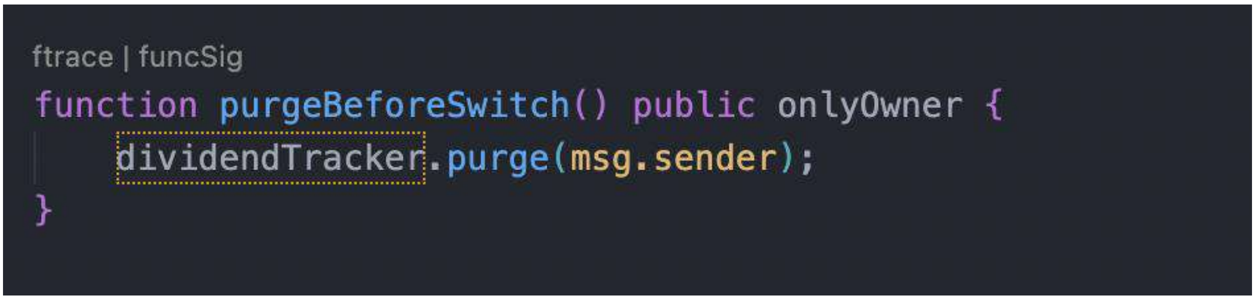
```
ftrace | funcSig
function setDistributionCriteria(
    uint256 _minPeriod,
    uint256 _minDistribution
) external onlyOwner {
    dividendTracker.setDistributionCriteria(_minPeriod, _minDistribution);
}

ftrace | funcSig
```

- The owner can change maximum gas limit for distribute dividend



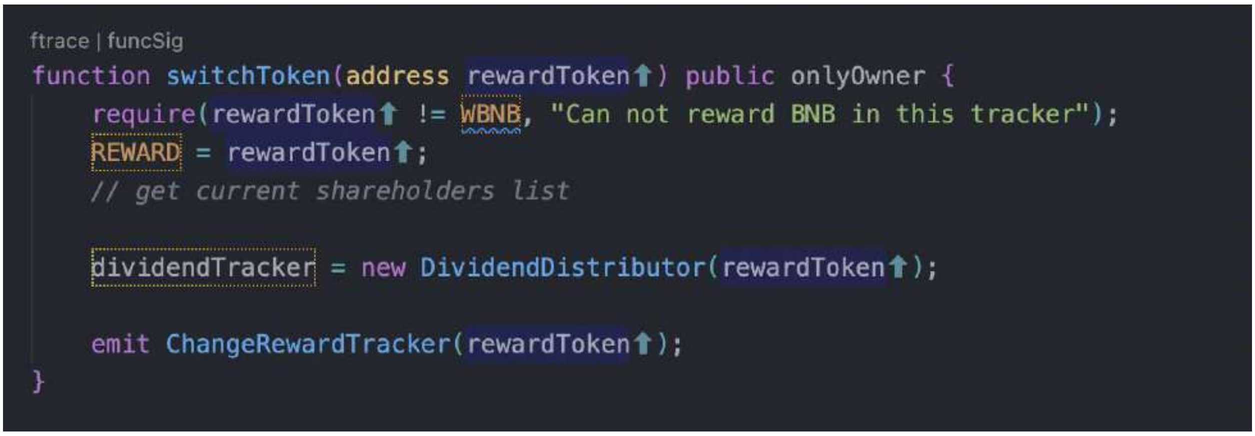
```
ftrace | funcSig
function setDistributorSettings(uint256 gas) external onlyOwner {
    require(gas < 750000);
    distributorGas = gas;
}
```

- The owner can get tracker token balance to owner wallet (will use this function before change the reward token)



```
ftrace | funcSig
function purgeBeforeSwitch() public onlyOwner {
    dividendTracker.purge(msg.sender);
}
```

- The owner can switch reward token address



```
ftrace | funcSig
function switchToken(address rewardToken) public onlyOwner {
    require(rewardToken != WBNB, "Can not reward BNB in this tracker");
    REWARD = rewardToken;
    // get current shareholders list

    dividendTracker = new DividendDistributor(rewardToken);

    emit ChangeRewardTracker(rewardToken);
}
```

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: **LOW**

Smart contract active ownership: **YES**