RugFreeCoins Audit

Cat Bread Token

Smart Contract Security Audit

April 25, 2022

Contents

Audit details

Audited project
Cat Bread Token

Contract Address
0x092c9dd9431328DA95Fe7F54670f569fADB422aC

Client contact
Cat Bread Team

Blockchain
Binance smart chain

Project website
https://catbreadofficial.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 Cat Bread Team to perform an audit of the smart contract.

https://bscscan.com/address/0x092c9dd9431328DA95Fe7F54670f569fADB422aC

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

Cat Bread 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 P2E high-end gaming experience with unique NFT mechanism and staking. Each transaction, purchase, and sale incur a 10% fee.

Features

- The **sustainability fee of 5% when buying and 5% when selling, which is allocated for marketing** is what allows Cat Bread to hold the aforementioned promise. Tokens will be swapped into BNB and will be sent to a marketing wallet. This way, Cat Bread 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 5% when buying and selling**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

ROADMAP

CATBREAD EVOLVES

- Catbread's Rebirth
- Community Private Exchange
- Community Private Sale
- Public Presale
- V2 Token Launch
- Community Airdrop
- Website Revamp
- Audit
- CMC/CG Updates

MID Q2 2022

- NFT Relaunch
- Legacy Airdrops
- NFT Staking
- Catbread NFT Lock System
- Leveling System
- Market Place
- NFT Reflections
- Partnership Gathering

LATE Q2 2022

- The Gameroom
- Gashapon
- Roulette
- Casino Staking
- Flappy Cat
- Scoreboard
- Daily/Weekly/Monthly Tournaments
- Logos
- Marketing Begins

EARLY/MID Q3 2022

- Gen 2 NFT Launch

Tokenomics

10% fee when buying and selling

- 5% of trade goes to marketing fund in BNB
- 5% 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 interested in trading tokens.
- Anyone who is ready to stake tokens and win rewards.
- Anyone who is ready to be part of the P2E game and win rewards.
- Anyone who’s interested in taking part with unique NFT mechanism.
- Anyone who’s interested in collecting NFTs or trading NFTs.
- Anyone who’s interested in taking part in the future plans of the Cat Bread token.
- Anyone who’s interested in making financial transactions with any other party using Cat Bread as the currency

Core concept

Catbread aims to be the go-to for players interested in casual P2E gaming.To do this, they're creating an arcade that integrates upgradeable NFTs, leveling, weekly competitions, staking, and an ever-expanding collection of virtual arcade cabinets.

- Compete in the arcade with an ever-expanding list of classic games.
- Win big in the casino and gashapon
- Bake and Level Catbread NFTs to stake, earn reflections, and much more

Sustainable mechanism

The **sustainability fee of 5% when buying and selling that allocated for marketing** is what allows Cat Bread to promote the token and use funds to further the development and marketing of the platform. Tokens will be swapped into BNB and will be sent to a marketing wallet. This way, Cat Bread 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 5% when buying and selling, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

Potential to grow with score points

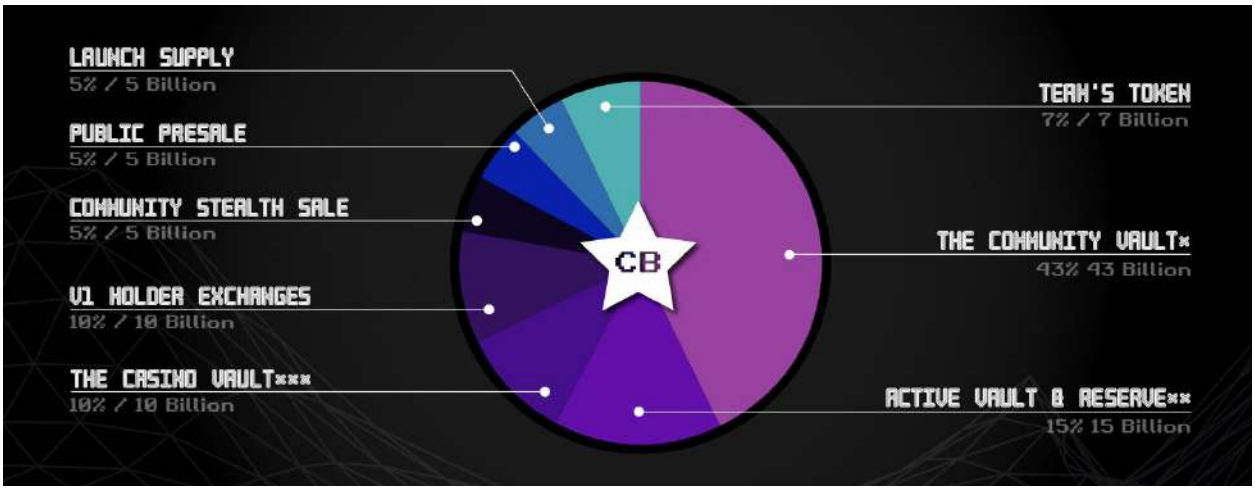
1.	Project efficiency	9/10
2.	Project uniqueness	9/10
3	Information quality	10/10
4	Service quality	9/10
5	System quality	9/10
6	Impact on the community	9/10
7	Impact on the business	9/10
8	Preparing for the future	9/10
Total Points		**9.125/10**

Contract details

Token contract details for 25th April 2022

Contract name	CatBread
Contract address	0x092c9dd9431328DA95Fe7F54670f569fADB422aC
Token supply	100,000,000,000
Token ticker	CatBread
Decimals	18
Token holders	3
Transaction count	9
Contract deployer address	0xDF5a2D6fD739927D19D7C23145357bA2a79452f1
Contract's current owner address	0xdf5a2d6fd739927d19d7c23145357ba2a79452f1

Tokens are distributed as follows:



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	High Centralization Risk
		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 🔒		
IUniswapV2 Pair	**Interface**			
L	name	External ❗		NO❗
L	symbol	External ❗		NO❗
L	decimals	External ❗		NO❗
L	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❗

└	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 Factory	**Interface**			
└	feeTo	External ❗		NO❗
└	feeToSetter	External ❗		NO❗

L	getPair	External ❗		NO❗
L	allPairs	External ❗		NO❗
L	allPairsLength	External ❗		NO❗
L	createPair	External ❗	🛑	NO❗
L	setFeeTo	External ❗	🛑	NO❗
L	setFeeToSetter	External ❗	🛑	NO❗
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❗
IERC20Metadata	**Interface**	**IERC20**		
L	name	External ❗		NO❗
L	symbol	External ❗		NO❗
L	decimals	External ❗		NO❗
ERC20	**Implementation**	**Context, IERC20 Metadata**		

└		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❗
└	_transfer	Internal 🔒	🛑	
└	_mint	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	
└	_approve	Internal 🔒	🛑	
└	_beforeTokenTransfer	Internal 🔒	🛑	
SafeMath	**Library**			
└	add	Internal 🔒		
└	sub	Internal 🔒		

L	sub	Internal		
L	mul	Internal		
L	div	Internal		
L	div	Internal		
L	mod	Internal		
L	mod	Internal		
Ownable	**Implementation**	**Context**		
L		Public		NO
L	owner	Public		NO
L	renounceOwnership	Public		onlyOwner
L	transferOwnership	Public		onlyOwner
SafeMathInt	**Library**			
L	mul	Internal		
L	div	Internal		
L	sub	Internal		
L	add	Internal		
L	abs	Internal		
L	toUint256Safe	Internal		
SafeMathUint	**Library**			

└	toInt256Safe	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❗

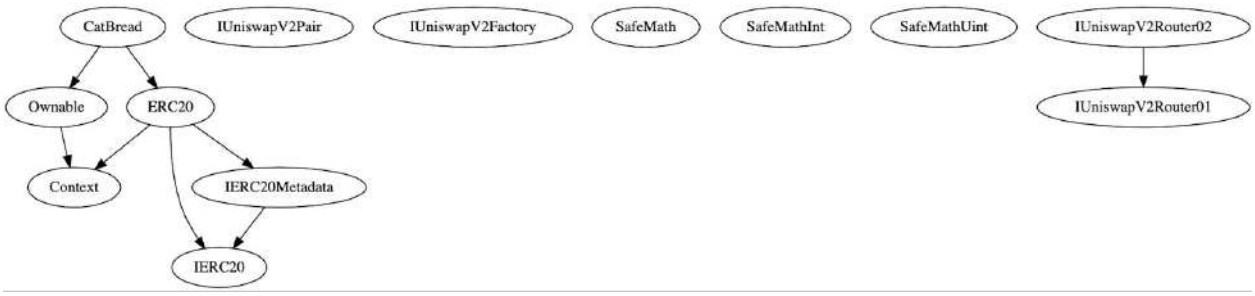
└	getAmountsIn	External ❗		NO❗
IUniswapV2 Router02	**Interface**	**IUniswap V2 Router01**		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
CatBread	**Implementation**	**ERC20, Ownable**		
└		Public ❗	🛑	ERC20
└		External ❗	💵	NO❗
└	enableTrading	External ❗	🛑	onlyOwner
└	removeLimits	External ❗	🛑	onlyOwner
└	disableTransferDelay	External ❗	🛑	onlyOwner
└	setEarlySellTax	External ❗	🛑	onlyOwner
└	updateSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	updateMaxTxnAmount	External ❗	🛑	onlyOwner
└	updateMaxWalletAmount	External ❗	🛑	onlyOwner
└	excludeFromMaxTransaction	Public ❗	🛑	onlyOwner

L	updateSwapEnabled	External ❗	🛑	onlyOwner
L	updateBuyFees	External ❗	🛑	onlyOwner
L	updateSellFees	External ❗	🛑	onlyOwner
L	excludeFromFees	Public ❗	🛑	onlyOwner
L	blacklistAccount	Public ❗	🛑	onlyOwner
L	setAutomatedMarketMakerPair	Public ❗	🛑	onlyOwner
L	_setAutomatedMarketMakerPair	Private 🔐	🛑	
L	updateMarketingWallet	External ❗	🛑	onlyOwner
L	updateDevWallet	External ❗	🛑	onlyOwner
L	isExcludedFromFees	Public ❗		NO❗
L	_transfer	Internal 🔒	🛑	
L	swapTokensForEth	Private 🔐	🛑	
L	addLiquidity	Private 🔐	🛑	
L	swapBack	Private 🔐	🛑	
L	Chire	External ❗	🛑	onlyOwner

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**

- The owner can change early sell fees without any maximum limit; early sell fees will be applied if someone sells tokens within 24 hours.

Owner privileges

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

```
ftrace | funcSig
function enableTrading() external onlyOwner {
    tradingActive = true;
    swapEnabled = true;
    launchedAt = block.number;
}
```

- The owner can remove initial limits and transactions delays

```
function removeLimits() external onlyOwner returns (bool) {
    limitsInEffect = false;
    return true;
}

// disable Transfer delay - cannot be reenabled
ftrace | funcSig
function disableTransferDelay() external onlyOwner returns (bool) {
    transferDelayEnabled = false;
    return true;
}
```

- The owner can enable/disable early taxes

```
ftrace | funcSig
function setEarlySellTax(bool onoff⬆) external onlyOwner {
    enableEarlySellTax = onoff⬆;
}
```

❖ The owner can change token swap point

```
ftrace | funcSig
function updateSwapTokensAtAmount(uint256 newAmount⬆)
    external
    onlyOwner
    returns (bool)
{
    require(
        newAmount⬆ >= (totalSupply() * 1) / 100000,
        "Swap amount cannot be lower than 0.001% total supply."
    );
    require(
        newAmount⬆ <= (totalSupply() * 5) / 1000,
        "Swap amount cannot be higher than 0.5% total supply."
    );
    swapTokensAtAmount = newAmount⬆;
    return true;
}
```

❖ The owner can change max transaction amount minimum up to 0.1%

```
ftrace | funcSig
function updateMaxTxnAmount(uint256 newNum⬆) external onlyOwner {
    require(
        newNum⬆ >= ((totalSupply() * 1) / 1000) / 1e18,
        "Cannot set maxTransactionAmount lower than 0.1%"
    );
    maxTransactionAmount = newNum⬆ * (10**18);
}
```

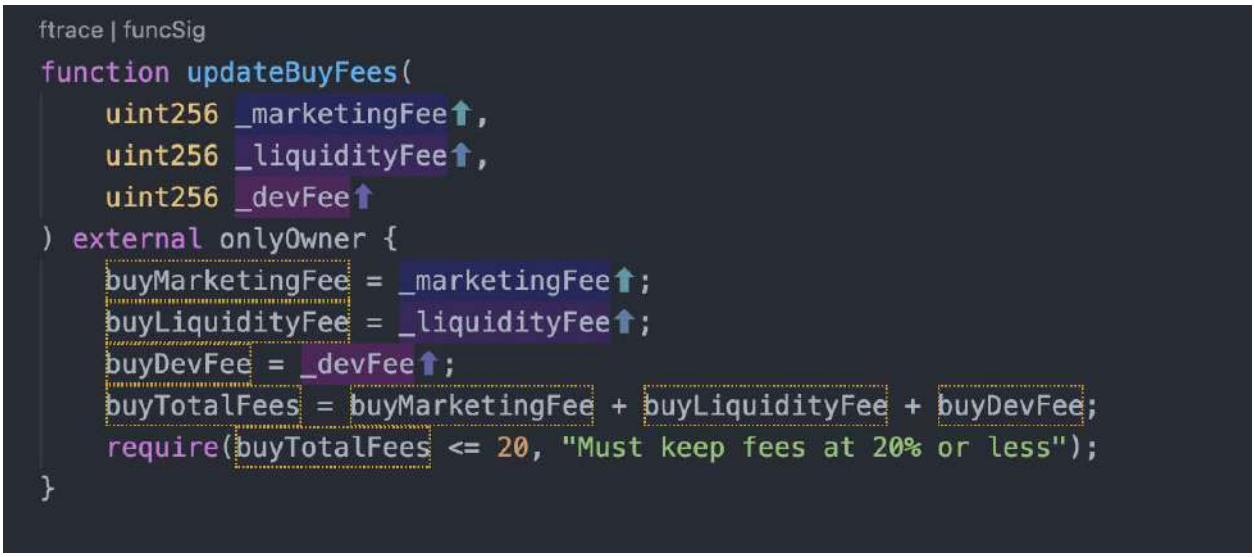
❖ The owner can change max wallet amount minimum up to 0.5%

```
ftrace | funcSig
function updateMaxWalletAmount(uint256 newNum⬆) external onlyOwner {
    require(
        newNum⬆ >= ((totalSupply() * 5) / 1000) / 1e18,
        "Cannot set maxWallet lower than 0.5%"
    );
    maxWallet = newNum⬆ * (10**18);
}
```

❖ The owner can exclude/include wallets from max transaction limit

❖ The owner can enable/disable swap point.

❖ Owner can change all buy fees maximum up to 20%



❖ Owner can change all sell fees maximum up to 25% and early sell fees without any limit

```
ftrace | funcSig
function updateSellFees(
    uint256 _marketingFee,
    uint256 _liquidityFee,
    uint256 _devFee,
    uint256 _earlySellLiquidityFee,
    uint256 _earlySellMarketingFee,
    uint256 _earlySellDevFee
) external onlyOwner {
    sellMarketingFee = _marketingFee;
    sellLiquidityFee = _liquidityFee;
    sellDevFee = _devFee;
    earlySellLiquidityFee = _earlySellLiquidityFee;
    earlySellMarketingFee = _earlySellMarketingFee;
    earlySellDevFee = _earlySellDevFee;
    sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
    require(sellTotalFees <= 25, "Must keep fees at 25% or less");
}
```

❖ Owner can include/exclude wallets from fees

```
ftrace | funcSig
function excludeFromFees(address account, bool excluded) public onlyOwner {
    _isExcludedFromFees[account] = excluded;
    emit ExcludeFromFees(account, excluded);
}
```

❖ Owner can block/unblock wallets from contract

```
ftrace | funcSig
function blacklistAccount(address account, bool isBlacklisted)
    public
    onlyOwner
{
    _blacklist[account] = isBlacklisted;
}
```

❖ Owner can change marketing and dev wallet

```
ftrace | funcSig
function updateMarketingWallet(address newMarketingWallet⬆)
    external
    onlyOwner
{
    emit marketingWalletUpdated(newMarketingWallet⬆, marketingWallet);
    marketingWallet = newMarketingWallet⬆;
}

ftrace | funcSig
function updateDevWallet(address newWallet⬆) external onlyOwner {
    emit devWalletUpdated(newWallet⬆, devWallet);
    devWallet = newWallet⬆;
}
```

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: **13**

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

Smart contract active ownership: **YES**