RugFreeCoins Audit

Clash Of Realms Token

Smart Contract Security Audit

April 06, 2022

Contents

Audit details

Audited project
Clash Of Realms Token

Contract Address
0xac97f474845d68b946b5b93bbf6658c014c7720f

Client contact
Clash Of Realms Team

Blockchain
Ethereum smart chain

Project website
https://www.clashofrealms.net/

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 Meta Shiba Inu Team to perform an audit of the smart contract.

https://etherscan.io/token/0xac97f474845d68b946b5b93bbf6658c014c7720f

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

Clash Of Realms 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 P2E gaming and staking. Each transaction, purchase, and sale incur a 9% fee.

Features

- The **BUSD rewards** will be distributed among every holder proportional to how many tokens each individual holds in values of **6% when buying and selling.**

- The **sustainability fee of 2% when buying and selling for marketing** is what allows Meta Shiba Inu to hold the aforementioned promise. Tokens will be swapped into BNB and will be sent to a marketing wallet per transaction. This way, Meta Shiba Inu 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.

- Meta Shiba Inu has the buyback and burn strategy that a **2% fee in each transaction when buying selling** is getting charged that benefits and rewards those who invest long-term. This feature slowly reduces supply making each Meta Shiba Inu more and more valuable.

Roadmap

1st phase

- Design Tokenomics
- Character and Game logic Design
- Lucid Chart + Medium Articles for base
- Website Launch
- Token Launch
- Organic community growth
- Phase 1 mint (1000 NFTs)

2nd phase

- Staking ecosystem Launch
- Launch token on centralized exchange
- $150,000 Paid to Game Devs
- Game developers organized
- Partnerships and Collaborations
- Phase 2 (2000 NFT's)

3rd phase

- Marketing strategy adjusted AMA's, Partnerships,
- Partnerships with top tier exchanges
- Exclusive AMA's and HNW Investors
- Character animations reveal
- Phase 3 mint (2000 NFT's)

4th phase

- Marketplace for NFTs

5th phase

- Beta Game release

Tokenomics

9% fee when buying and selling

- 4% of trade allocated for game development in ETH.
- 2% of trade allocated for marketing in ETH.
- 2% of trade allocated for staking in ETH.
- 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 interested in taking part in staking and earning rewards.
- Anyone who’s interested in playing the P2E game and earning rewards
- Anyone who’s interested in taking part with the future plans of the Clash Of Realms token.
- Anyone who’s interested in making financial transactions with any other party using Clash Of Realms as the currency.

Core concept

Sustainable mechanism

The **sustainability fee of 8% when buying and selling is allocated for game development 4% and 2% for marketing and 2% for staking** is what allows Clash Of Realms to promote the token and use funds to further the development of the platform. Tokens will be swapped ETH and will be sent to a marketing wallet. This way, Clash Of Realms 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% 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	9/10
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		**9/10**

Contract details

Token contract details for 6th April 2022

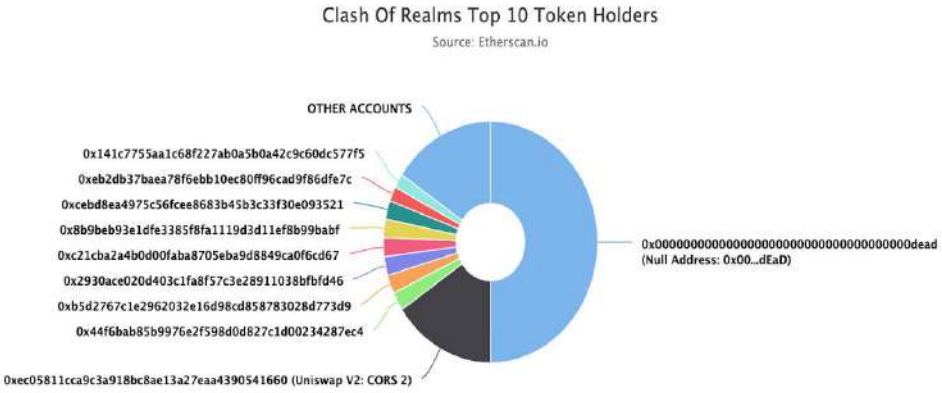
Contract name	Clash Of Realms
Contract address	0xac97f474845d68b946b5B93BbF6658C014C7720F
Token supply	10,000,000,000
Token ticker	CORS
Decimals	9
Token holders	150
Transaction count	444
Marketing wallet	Not public
Contract deployer address	0xb69052e69e7cd9DB693db7a7fcE8e6B8dA254A5E
Contract's current owner address	0xb69052e69e7cd9db693db7a7fce8e6b8da254a5e

Top token holders

Top 10 Token Holders

The top 10 holders collectively own 84.45% (8,445,084,113.10 Tokens) of Clash Of Realms

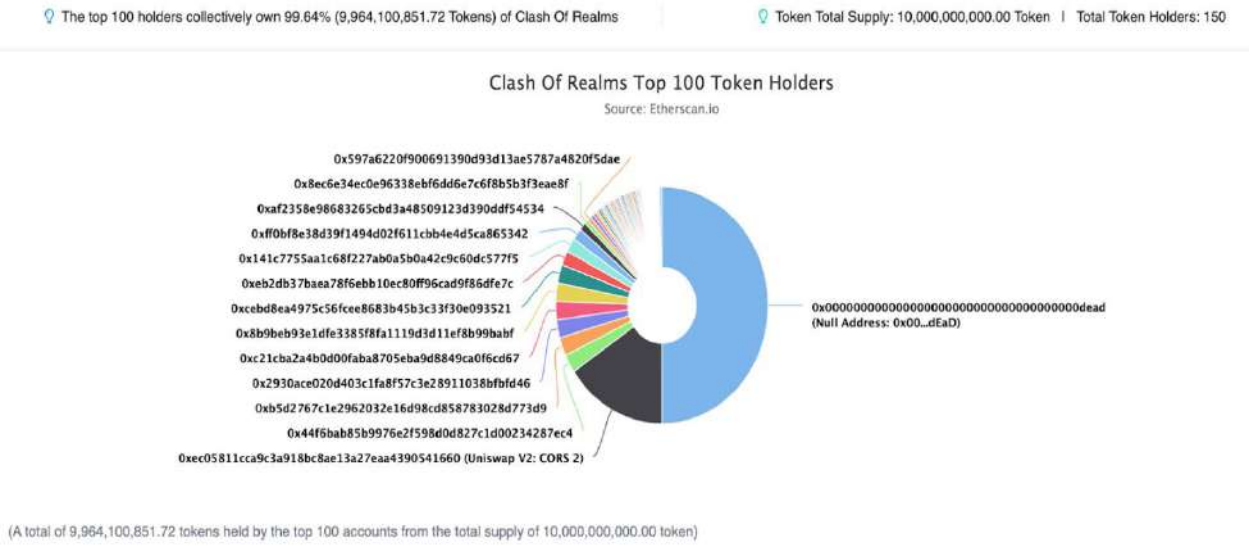
Token Total Supply: 10,000,000,000.00 Token | Total Token Holders: 150



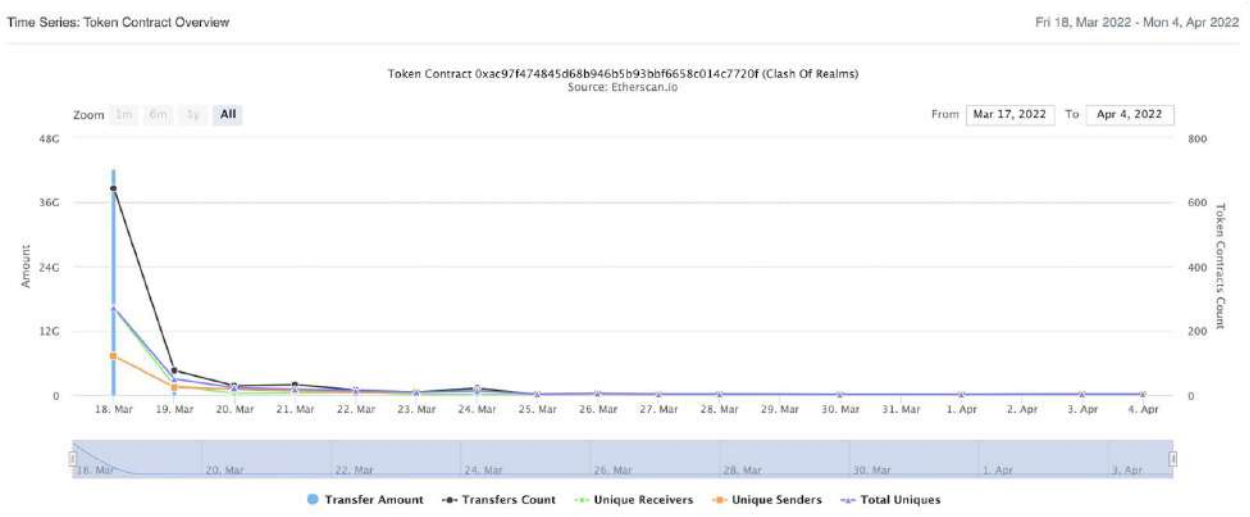
(A total of 8,445,084,113.10 tokens held by the top 10 accounts from the total supply of 10,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	Null Address: 0x00...dEaD	5,000,000,000	50.0000%
2	Uniswap V2: CORS 2	1,555,604,113.100190309	15.5560%
3	0x44f6bab85b9976e2f598d0d827c1d00234287ec4	250,000,000	2.5000%
4	0xb5d2767c1e2962032e16d98cd858783028d773d9	250,000,000	2.5000%
5	0x2930ace020d403c1fa8f57c3e28911038bfbfd46	250,000,000	2.5000%
6	0xc21cba2a4b0d00faba8705eba9d8849ca0f6cd67	250,000,000	2.5000%
7	0x8b9beb93e1dfe3385f8fa1119d3d11ef8b99babf	250,000,000	2.5000%
8	0xcebd8ea4975c56fcee8683b45b3c33f30e093521	250,000,000	2.5000%
9	0xeb2db37baea78f6ebb10ec80ff96cad9f86dfe7c	198,380,000	1.9838%
10	0x141c7755aa1c68f227ab0a5b0a42c9c60dc577f5	191,100,000	1.9110%

Top 100 Token Holders



Contract interaction details



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	Centralized high severity
		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 🔒		
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❗
Ownable	**Implementation**	**Context**		
L		Public ❗	🛑	NO❗

L	owner	Public ❗		NO❗
L	renounceOwnership	Public ❗	🛑	onlyOwner
L	transferOwnership	Public ❗	🛑	onlyOwner
SafeMath	**Library**			
L	add	Internal 🔒		
L	sub	Internal 🔒		
L	sub	Internal 🔒		
L	mul	Internal 🔒		
L	div	Internal 🔒		
L	div	Internal 🔒		
IUniswapV2Factory	**Interface**			
L	createPair	External ❗	🛑	NO❗
IUniswapV2Router02	**Interface**			
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
L	factory	External ❗		NO❗

L	WETH	External ❗		NO❗
L	addLiquidityETH	External ❗	💵	NO❗
ClashOfRealms	**Implementation**	**Context, IERC20, Ownable**		
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	tokenFromReflection	Private 🔐		
L	removeAllFee	Private 🔐	🛑	
L	restoreAllFee	Private 🔐	🛑	
L	_approve	Private 🔐	🛑	

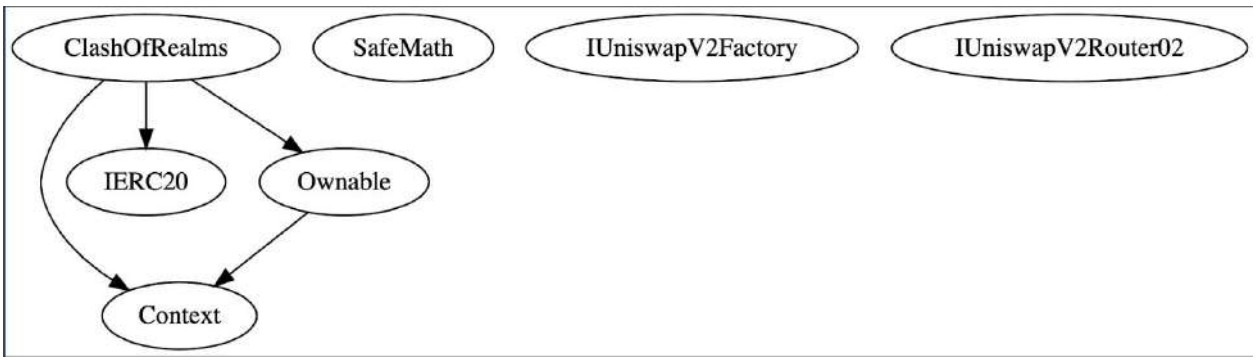
└	_transfer	Private 🔐	🛑	
└	swapTokensForEth	Private 🔐	🛑	lockTheSwap
└	swapDistributeAndLiquify	Private 🔐	🛑	
└	addLiquidity	Private 🔐	🛑	
└	sendETHToFee	Private 🔐	🛑	
└	setTrading	Public ❗	🛑	onlyOwner
└	manualswap	External ❗	🛑	NO❗
└	manualsend	External ❗	🛑	NO❗
└	blockBots	Public ❗	🛑	onlyOwner
└	unblockBot	Public ❗	🛑	onlyOwner
└	_tokenTransfer	Private 🔐	🛑	
└	_transferStandard	Private 🔐	🛑	
└	_takeTeam	Private 🔐	🛑	
└	_reflectFee	Private 🔐	🛑	
└		External ❗	💵	NO❗
└	_getValues	Private 🔐		

└	_getTValues	Private		
└	_getRValues	Private		
└	_getRate	Private		
└	_getCurrentSupply	Private		
└	setFee	Public		onlyOwner
└	excludeMultipleAccountsFromFees	Public		onlyOwner
└	setMinSwapTokensThreshold	Public		onlyOwner
└	setMarketingWallet	Public		onlyOwner
└	toggleSwap	Public		onlyOwner
└	setMaxTxnAmount	Public		onlyOwner
└	setMaxWalletSize	Public		onlyOwner
└	allowPreTrading	Public		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**

High severity centralization issues

❖ The owner can enable/disable trading anytime

❖ The owner can change all fees without maximum limitation (can set to 100%)

```
ftrace | funcSig
function setFee(
    uint256 redisFeeOnBuy,
    uint256 redisFeeOnSell,
    uint256 marketingFeeOnBuy,
    uint256 marketingFeeOnSell,
    uint256 liquidityFeeOnBuy,
    uint256 liquidityFeeOnSell
) public onlyOwner {
    _redisFeeOnBuy = redisFeeOnBuy;
    _redisFeeOnSell = redisFeeOnSell;

    _marketingFeeOnBuy = marketingFeeOnBuy;
    _marketingFeeOnSell = marketingFeeOnSell;

    _liquidityFeeOnBuy = liquidityFeeOnBuy;
    _liquidityFeeOnSell = liquidityFeeOnSell;
}
```

❖ The owner can change max transaction and wallet amount without minimum limitation (can set 0%)

```
ftrace | funcSig
function setMaxTxnAmount(uint256 maxTxAmount) public onlyOwner {
    _maxTxAmount = maxTxAmount;
}

ftrace | funcSig
function setMaxWalletSize(uint256 maxWalletSize) public onlyOwner {
    _maxWalletSize = maxWalletSize;
}
```

Owner privileges

- The owner can enable/disable trading

```
ftrace | funcSig
function setTrading(bool _tradingOpen) public onlyOwner {
    tradingOpen = _tradingOpen;
}
```

- The owner and marketing wallet can manually trigger the swap

```
ftrace | funcSig
function manualswap() external {
    require(_msgSender() == _marketingAddress || _msgSender() == owner());
    uint256 contractBalance = balanceOf(address(this));
    swapTokensForEth(contractBalance);
}
```

- The owner and marketing wallet can manually send contract ETH to the marketing wallet

```
ftrace | funcSig
function manualsend() external {
    require(_msgSender() == _marketingAddress || _msgSender() == owner());
    uint256 contractETHBalance = address(this).balance;
    sendETHToFee(contractETHBalance);
}
```

- The owner can block/unblock wallets

```
ftrace | funcSig
function blockBots(address[] memory bots_) public onlyOwner {
    for (uint256 i = 0; i < bots_.length; i++) {
        bots[bots_[i]] = true;
    }
}

ftrace | funcSig
function unblockBot(address notbot) public onlyOwner {
    bots[notbot] = false;
}
```

❖ The owner can change all buy and sell fees

```
ftrace | funcSig
function setFee(
    uint256 redisFeeOnBuy⬆,
    uint256 redisFeeOnSell⬆,
    uint256 marketingFeeOnBuy⬆,
    uint256 marketingFeeOnSell⬆,
    uint256 liquidityFeeOnBuy⬆,
    uint256 liquidityFeeOnSell⬆
) public onlyOwner {
    _redisFeeOnBuy = redisFeeOnBuy⬆;
    _redisFeeOnSell = redisFeeOnSell⬆;

    _marketingFeeOnBuy = marketingFeeOnBuy⬆;
    _marketingFeeOnSell = marketingFeeOnSell⬆;

    _liquidityFeeOnBuy = liquidityFeeOnBuy⬆;
    _liquidityFeeOnSell = liquidityFeeOnSell⬆;
}
```

❖ The owner can include/exclude wallets from fee

```
ftrace | funcSig
function excludeMultipleAccountsFromFees(
    address[] calldata accounts⬆,
    bool excluded⬆
) public onlyOwner {
    for (uint256 i = 0; i < accounts⬆.length; i++) {
        _isExcludedFromFee[accounts⬆[i]] = excluded⬆;
    }
}
```

❖ The owner can change the swap point.

```
//Set minimum tokens required to swap.
ftrace | funcSig
function setMinSwapTokensThreshold(uint256 swapTokensAtAmount)
    public
    onlyOwner
{
    _swapTokensAtAmount = swapTokensAtAmount;
}
```

❖ The owner can change the marketing wallet address

```
//Set marketing wallet
ftrace | funcSig
function setMarketingWallet(address payable marketingWallet)
    public
    onlyOwner
{
    _marketingAddress = marketingWallet;
}
```

❖ The owner can enable/disable swapping

```
//Set minimum tokens required to swap.
ftrace | funcSig
function toggleSwap(bool _swapEnabled) public onlyOwner {
    swapEnabled = _swapEnabled;
}
```

❖ The owner can change the max transaction and wallet amount

```
ftrace | funcSig
function setMaxTxnAmount(uint256 maxTxAmount⬆) public onlyOwner {
    _maxTxAmount = maxTxAmount⬆;
}

ftrace | funcSig
function setMaxWalletSize(uint256 maxWalletSize⬆) public onlyOwner {
    _maxWalletSize = maxWalletSize⬆;
}
```

❖ The owner can allow users to trade when trading is disabled

```
ftrace | funcSig
function allowPreTrading(address account⬆, bool allowed⬆) public onlyOwner {
    require(preTrader[account⬆] != allowed⬆, "TOKEN: Already enabled.");
    preTrader[account⬆] = allowed⬆;
}
```

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: **3**

Solidity code functional issue level: **PASSED**

Number of owner privileges: **11**

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

Smart contract active ownership: **YES**