



RugFreeCoins Audit



CHIMP INU Token Smart Contract Security Audit

July 26th ,2023

Overview

- ✅ No mint function found, the owner cannot mint tokens after initial deployment.
- ✅ The owner can't set a max transaction limit
- ✅ The owner can't pause trading once it's enabled
- ❌ The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.
- ✅ The owner can't change fees.
- ✅ 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.

- **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;  
}
```

The owner has the ability to set the swap point to a very high limit. However, it's important to note that if the owner changes this limit to an excessively high amount, there is a risk of the swap failing when the contract attempts to sell a very large amount at once.

```
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);
}
```

Contents

Overview	ii
Audit details	1
Disclaimer	2
Background	3
Target market and the concept	5
Potential to grow with score points	6
Total Points	6
Contract details	7
Contract code function details	8
Contract description table	10
Security issue checking status	17
Owner privileges	18
Audit conclusion	21

Audit details



Audited project
CHMIP INU Token



Contract Address
0x7a57c5c20e00330130497024058d9e2c82f46297



Client contact
CHMIP INU Token Team



Blockchain
Binance Smart chain



Project website
<https://linktr.ee/chimpinu>

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 CHIMP INU Token Team to perform an audit of the smart contract.

<https://bscscan.com/token/0x7a57c5c20e00330130497024058d9e2c82f46297#code>

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

3% tax when buying

- 2% of trade goes to the marketing wallet in BNB
- 1% goes to the burn wallet

10% tax when selling

- 8% of trade goes to the marketing wallet in BNB
- 2% 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 by holding tokens.
- Anyone who's interested in trading tokens.
- Anyone who's interested in taking part in the CHIMP INU token ecosystem.
- Anyone who's interested in taking part in the future plans of CHIMP INU Token.
- Anyone who's interested in making financial transactions with any other party using CHIMP INU 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	8/10
5	System quality	8/10
6	Impact on the community	8/10
7	Impact on the business	8/10
8	Preparing for the future	8/10
9	Smart contract security	8/10
10	Smart contract functionality assessment	10/10
Total Points		8.3/10

Contract details

Token contract details for 26th of July 2023

Contract name	CHIMP INU
Contract address	0x7a57c5c20E00330130497024058D9E2C82f46297
Token supply	520,000,000,000,000,000
Token ticker	CHIMP
Decimals	9
Token holders	1
Transaction count	1
Contract deployer address	0x6e43A081Bead22079E44ac447D9980036699B657
Contract 's current owner address	0x6e43a081bead22079e44ac447d9980036699b657
Marketing wallet	0xc58d1200b81c901e5d9d445e2d3d4f4672bf5ed2


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 & 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		
L	Function Name	Visibility	Mutability	Modifiers
IUniswapV2 Factory	Interface			
L	feeTo	External !		NO !
L	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 !
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 !
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 !

L	sync	External !		NO !
L	initialize	External !		NO !
IUniswapV2 Router01	Interface			
L	factory	External !		NO !
L	WETH	External !		NO !
L	addLiquidity	External !		NO !
L	addLiquidityETH	External !		NO !
L	removeLiquidity	External !		NO !
L	removeLiquidityETH	External !		NO !
L	removeLiquidityWithPermit	External !		NO !
L	removeLiquidityETHWithPermit	External !		NO !
L	swapExactTokensForTokens	External !		NO !
L	swapTokensForExactTokens	External !		NO !
L	swapExactETHForTokens	External !		NO !
L	swapTokensForExactETH	External !		NO !
L	swapExactTokensForETH	External !		NO !
L	swapETHForExactTokens	External !		NO !
L	quote	External !		NO !
L	getAmountOut	External !		NO !
L	getAmountIn	External !		NO !



L	getAmountsOut	External !		NO !
L	getAmountsIn	External !		NO !
IUniswapV2Router02	Interface	IUniswapV2Router01		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External !	🔴	NO !
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External !	🔴	NO !
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External !	🔴	NO !
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External !	🏠	NO !
L	swapExactTokensForETHSupportingFeeOnTransferTokens	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 !
IERC20 Metadata	Interface	IERC20		
L	name	External !		NO !
L	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 !

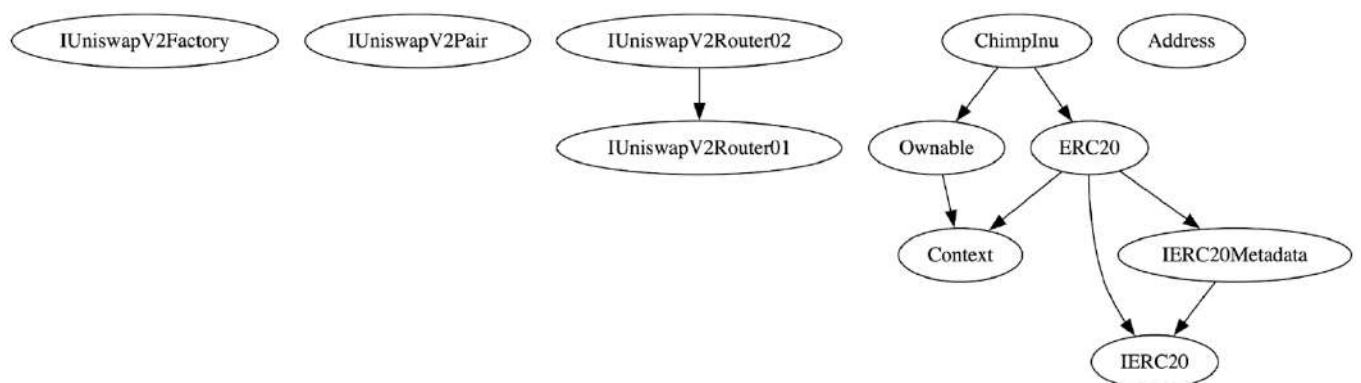
L	allowance	Public !		NO !
L	approve	Public !	🔴	NO !
L	transferFrom	Public !	🔴	NO !
L	increaseAllowance	Public !	🔴	NO !
L	decreaseAllowance	Public !	🔴	NO !
L	_transfer	Internal 🔒	🔴	
L	_mint	Internal 🔒	🔴	
L	_burn	Internal 🔒	🔴	
L	_approve	Internal 🔒	🔴	
L	_beforeTokenTransfer	Internal 🔒	🔴	
L	_afterTokenTransfer	Internal 🔒	🔴	
Chimplnu	Implementation	ERC20, Ownable		
L		Public !	🔴	ERC20
L		External !	🔒	NO !
L	claimStuckTokens	External !	🔴	onlyOwner
L	excludeFromFees	External !	🔴	onlyOwner
L	isExcludedFromFees	Public !		NO !
L	changeMarketingWallet	External !	🔴	onlyOwner
L	enableTrading	External !	🔴	onlyOwner
L	_transfer	Internal 🔒	🔴	

L	setSwapEnabled	External !		onlyOwner
L	setSwapTokensAtAmount	External !		onlyOwner
L	swapAndSendMarketing	Private 		

Legend

Symbol	Meaning
	Function can modify state
	Function is payable

Inheritance Hierarchy



Security issue checking status

❖ High severity issues

The owner has the ability to set the swap point to a very high limit. However, it's important to note that if the owner changes this limit to an excessively high amount, there is a risk of the swap failing when the contract attempts to sell a very large amount at once.

```
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);
}
```

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

No low-severity issues found

Owner privileges

- ❖ Owner can get any bep20 tokens from the contract (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 marketing wallet address

```
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

```
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: **6**

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

Smart contract active ownership: **ACTIVE**