



RugFreeCoins Audit



Cartman.AI Token Smart Contract Security Audit

July 19th ,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 can't change fees over 20%
- ✓ 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.

- **Centralisation risk**

The owner can change all fees maximum of up to 50% (Up to 25% buy and up to 25% sell)

```
function setTokenRewardsFee(uint256 value) external onlyOwner {
    tokenRewardsFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}

function setLiquiditFee(uint256 value) external onlyOwner {
    liquidityFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}

function setMarketingFee(uint256 value) external onlyOwner {
    marketingFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}
```

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	26
Owner privileges	27
Audit conclusion	30

Audit details



Audited project
Cartman.AI Token



Contract Address
0xE7e41AC11A98F3b6CD7FF0C57a7Ef073dBAAB387



Client contact
Cartman.AI Token Team



Blockchain
Ethereum



Project website
<https://www.mrcartman1000x.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 Cartman.AI Token Team to perform an audit of the smart contract.

<https://etherscan.io/token/0xe7e41ac11a98f3b6cd7ff0c57a7ef073dbaab387>

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

4% tax when buying & selling (19/07/2023)

- 1% of trade goes to the marketing wallet in pepe 2.0 token.
- 1% of trade goes to the liquidity pool.
- 2% trade is distributed among holders as rewards in Pepe 2.0 tokens.

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 Cartman.AI token ecosystem.
- Anyone who's interested in taking part in the future plans of Cartman.AI Token.
- Anyone who's interested in making financial transactions with any other party using Cartman.AI 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	10/10
10	Smart contract functionality assessment	10/10
Total Points		8.5/10

Contract details

Token contract details for 19th of July 2023

Contract name	Cartman.AI
Contract address	0xE7e41AC11A98F3b6CD7FF0C57a7Ef073dBAAB387
Token supply	420,690,000,000,000
Decimals	18
Token holders	2
Transaction count	2
Contract deployer address	0x849657ca176fC22820FD76F40Df128795FB3Df46
Contract's current owner address	0x849657ca176fC22820FD76F40Df128795FB3Df46
Dividend Tracker	0x6563f523d2Ec883D278cbC9B4463aF09Ebd0607a
Reward Token	0xfb66321D7C674995dFcC2cb67A30bC978dc862AD
Marketing wallet	0xCba64A2e96fDC7A56Ee93d2CB99Ed9D7335923bC






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
		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
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 !
Context	Implementation			
L	_msgSender	Internal 		
L	_msgData	Internal 		











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 		

Address	Library			
L	isContract	Internal 🔒		
L	sendValue	Internal 🔒	🔴	
L	functionCall	Internal 🔒	🔴	
L	functionCall	Internal 🔒	🔴	
L	functionCallWithValue	Internal 🔒	🔴	
L	functionCallWithValue	Internal 🔒	🔴	
L	functionStaticCall	Internal 🔒		
L	functionStaticCall	Internal 🔒		
L	functionDelegateCall	Internal 🔒	🔴	
L	functionDelegateCall	Internal 🔒	🔴	
L	verifyCallResult	Internal 🔒		
SafeERC20	Library			
L	safeTransfer	Internal 🔒	🔴	
L	safeTransferFrom	Internal 🔒	🔴	
L	safeApprove	Internal 🔒	🔴	
L	safeIncreaseAllowance	Internal 🔒	🔴	
L	safeDecreaseAllowance	Internal 🔒	🔴	
L	_callOptionalReturn	Private 🔒	🔴	
Ownable	Implementation	Context		

L		Public !		NO !
L	owner	Public !		NO !
L	renounceOwnership	Public !		onlyOwner
L	transferOwnership	Public !		onlyOwner
L	_setOwner	Private 		
SafeMath	Library			
L	tryAdd	Internal 		
L	trySub	Internal 		
L	tryMul	Internal 		
L	tryDiv	Internal 		
L	tryMod	Internal 		
L	add	Internal 		
L	sub	Internal 		
L	mul	Internal 		
L	div	Internal 		
L	mod	Internal 		
L	sub	Internal 		
L	div	Internal 		
L	mod	Internal 		
Clones	Library			









L	clone	Internal 		
L	cloneDeterministic	Internal 		
L	predictDeterministicAddress	Internal 		
L	predictDeterministicAddress	Internal 		
SafeERC20 NoRevert	Library			
L	safeTransfer	Internal 		
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 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 !
IUniswapV2 Router02	Interface	Iuniswap V2 Router01		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External !		NO !
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External !		NO !

L	swapExactETHForTokensSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External !		NO !
IERC20 Upgradeable	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 Upgradeable	Interface	IERC20 Upgradeable		
L	name	External !		NO !
L	symbol	External !		NO !
L	decimals	External !		NO !
Initializable	Implementation			
Context Upgradeable	Implementation	Initializable		
L	__Context_init	Internal 		initializer
L	__Context_init_unchained	Internal 		initializer
L	_msgSender	Internal 		

L	_msgData	Internal 🔒		
ERC20 Upgradeable	Implementation	Initializable, Context Upgradeable, IERC20Upgradeable, IERC20 Metadata Upgradeable		
L	__ERC20_init	Internal 🔒	🔴	initializer
L	__ERC20_init_unchained	Internal 🔒	🔴	initializer
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 🔒	🔴	
Ownable Upgradeable	Implementation	Initializable, Context Upgradeable		
L	__Ownable_init	Internal 🔒	🔴	initializer
L	__Ownable_init_unchained	Internal 🔒	🔴	initializer
L	owner	Public !		NO !
L	renounceOwnership	Public !	🔴	onlyOwner
L	transferOwnership	Public !	🔴	onlyOwner
L	_setOwner	Private 🗝️	🔴	
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 !
SafeMathInt	Library			

L	mul	Internal 🔒		
L	div	Internal 🔒		
L	sub	Internal 🔒		
L	add	Internal 🔒		
L	abs	Internal 🔒		
L	toUint256Safe	Internal 🔒		
SafeMathUint	Library			
L	toInt256Safe	Internal 🔒		
Iterable Mapping	Library			
L	get	Public !		NO !
L	getIndexOfKey	Public !		NO !
L	getKeyAtIndex	Public !		NO !
L	size	Public !		NO !
L	set	Public !	🔴	NO !
L	remove	Public !	🔴	NO !
Dividend Paying Token Interface	Interface			
L	dividendOf	External !		NO !
L	withdrawDividend	External !	🔴	NO !



Dividend PayingToken Optional Interface	Interface			
L	withdrawableDividendOf	External !		NO !
L	withdrawnDividendOf	External !		NO !
L	accumulativeDividendOf	External !		NO !
Dividend PayingToken	Implementation	ERC20 Upgradeable, OwnableUpgr adeable, Dividend PayingToken Interface, Dividend PayingToken OptionalInter face		
L	__DividendPayingToken_init	Internal 🔒	●	initializer
L	distributeCAKEDividends	Public !	●	onlyOwner
L	withdrawDividend	Public !	●	NO !
L	_withdrawDividendOfUser	Internal 🔒	●	
L	dividendOf	Public !		NO !
L	withdrawableDividendOf	Public !		NO !
L	withdrawnDividendOf	Public !		NO !
L	accumulativeDividendOf	Public !		NO !
L	_transfer	Internal 🔒	●	
L	_mint	Internal 🔒	●	
L	_burn	Internal 🔒	●	

L	_setBalance	Internal 🔒	🔴	
BABYTOKEN Dividend Tracker	Implementation	OwnableUp gradeable, Dividend Paying Token		
L	initialize	External !	🔴	initializer
L	_transfer	Internal 🔒		
L	withdrawDividend	Public !		NO !
L	excludeFromDividends	External !	🔴	onlyOwner
L	isExcludedFromDividends	Public !		NO !
L	updateClaimWait	External !	🔴	onlyOwner
L	updateMinimumTokenBalanceForDivide nds	External !	🔴	onlyOwner
L	getLastProcessedIndex	External !		NO !
L	getNumberOfTokenHolders	External !		NO !
L	getAccount	Public !		NO !
L	getAccountAtIndex	Public !		NO !
L	canAutoClaim	Private 🗝️		
L	setBalance	External !	🔴	onlyOwner
L	process	Public !	🔴	NO !
L	processAccount	Public !	🔴	onlyOwner
BaseToken	Implementation			

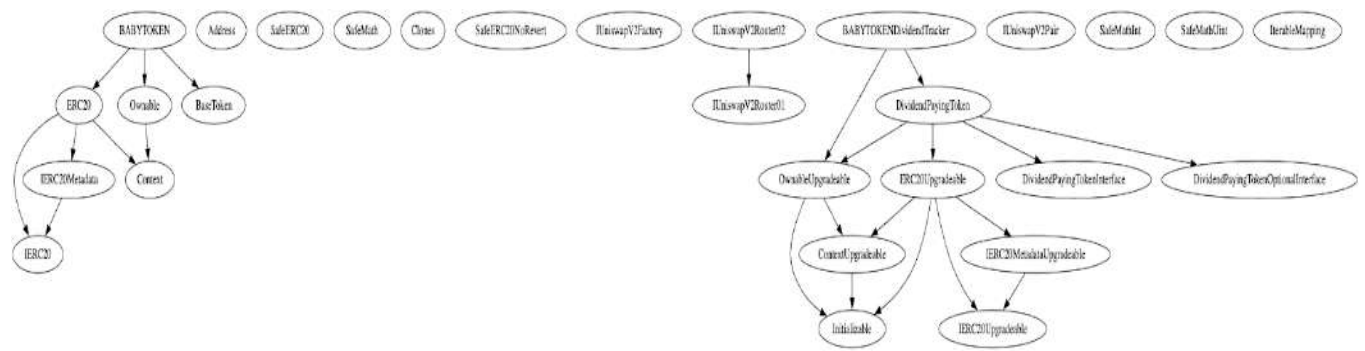
BABYTOKEN	Implementation	ERC20, Ownable, BaseToken		
L		Public !		ERC20
L		External !		NO !
L	setSwapTokensAtAmount	External !		onlyOwner
L	excludeFromFees	External !		onlyOwner
L	excludeMultipleAccountsFromFees	External !		onlyOwner
L	setMarketingWallet	External !		onlyOwner
L	setTokenRewardsFee	External !		onlyOwner
L	setLiquiditFee	External !		onlyOwner
L	setMarketingFee	External !		onlyOwner
L	_setAutomatedMarketMakerPair	Private 		
L	updateGasForProcessing	Public !		onlyOwner
L	updateClaimWait	External !		onlyOwner
L	getClaimWait	External !		NO !
L	updateMinimumTokenBalanceForDividends	External !		onlyOwner
L	getMinimumTokenBalanceForDividends	External !		NO !
L	getTotalDividendsDistributed	External !		NO !
L	isExcludedFromFees	Public !		NO !
L	withdrawableDividendOf	Public !		NO !
L	dividendTokenBalanceOf	Public !		NO !

L	excludeFromDividends	External !		onlyOwner
L	isExcludedFromDividends	Public !		NO !
L	getAccountDividendsInfo	External !		NO !
L	getAccountDividendsInfoAtIndex	External !		NO !
L	processDividendTracker	External !		NO !
L	claim	External !		NO !
L	getLastProcessedIndex	External !		NO !
L	getNumberOfDividendTokenHolders	External !		NO !
L	_transfer	Internal 		
L	swapAndSendToFee	Private 		
L	swapAndLiquify	Private 		
L	swapTokensForEth	Private 		
L	swapTokensForCake	Private 		
L	addLiquidity	Private 		
L	swapAndSendDividends	Private 		

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 all fees maximum of up to 50% (Up to 25% buy and up to 25% sell)

```
function setTokenRewardsFee(uint256 value) external onlyOwner {
    tokenRewardsFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}

function setLiquiditFee(uint256 value) external onlyOwner {
    liquidityFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}

function setMarketingFee(uint256 value) external onlyOwner {
    marketingFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}
```

Owner privileges

- ❖ The owner can change the swap point minimum up to 0.001%

```
function setSwapTokensAtAmount(uint256 amount) external onlyOwner {
    require(
        amount > totalSupply() / 10 ** 5,
        "BABYTOKEN: Amount must be greater than 0.001% of total supply"
    );
    swapTokensAtAmount = amount;
}
```

- ❖ The owner can exclude wallets from fees

```
function excludeFromFees(address account) external onlyOwner {
    require(
        !_isExcludedFromFees[account],
        "BABYTOKEN: Account is already excluded"
    );
    _isExcludedFromFees[account] = true;

    emit ExcludeFromFees(account);
}

function excludeMultipleAccountsFromFees(
    address[] calldata accounts
) external onlyOwner {
    for (uint256 i = 0; i < accounts.length; i++) {
        _isExcludedFromFees[accounts[i]] = true;
    }

    emit ExcludeMultipleAccountsFromFees(accounts);
}
```

- ❖ The owner can change the marketing wallet

```
function setMarketingWallet(address payable wallet) external onlyOwner {
    require(
        wallet != address(0),
        "BABYTOKEN: The marketing wallet cannot be the value of zero"
    );
    require(!wallet.isContract(), "Marketing wallet cannot be a contract");
    _marketingWalletAddress = wallet;
}
```

- ❖ The owner can change all fees maximum of up to 50% (Up to 25% buy and up to 25% sell)

```
function setTokenRewardsFee(uint256 value) external onlyOwner {
    tokenRewardsFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}

function setLiquiditFee(uint256 value) external onlyOwner {
    liquidityFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}

function setMarketingFee(uint256 value) external onlyOwner {
    marketingFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}
```

- ❖ The owner can change gas fee for process reward maximum up to 500000

```
function updateGasForProcessing(uint256 newValue) public onlyOwner {
    require(
        newValue >= 200000 && newValue <= 500000,
        "BABYTOKEN: gasForProcessing must be between 200,000 and 500,000"
    );
    require(
        newValue != gasForProcessing,
        "BABYTOKEN: Cannot update gasForProcessing to same value"
    );
    emit GasForProcessingUpdated(newValue, gasForProcessing);
    gasForProcessing = newValue;
}
```

- ❖ The owner can change the claim wait

```
function updateClaimWait(uint256 claimWait) external onlyOwner {
    dividendTracker.updateClaimWait(claimWait);
}
```

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: **1**

Solidity code functional issue level: **PASS**

Number of owner privileges: **6**

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

Smart contract active ownership: **ACTIVE**