Audit Report

CryptoRod

January 2022

Type	BEP20
Network	BSC
Address	0x21C27048fddbe46fA8d3E1b78688649Ee70dbfE6
Audited by	© coinscope

Table of Contents

Contract Review

Contract Name	CryptoRod
Compiler Version	v0.6.12+commit.27d51765
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x21C27048fddbe46fA8d3E1b78688649Ee70dbfE6
Symbol	CROD
Decimals	9
Total Supply	100,000,000
Source	contract.sol
Domain	

Audit Updates

Initial Audit	8th January 2022
Corrected	

Contract Analysis

Any user can mint tokens. There is no way to prevent this functionality since it is not protected by permissions.

Critical Medium Minor Pass

Severity	Code	Description
●	ST	Contract Owner is not able to stop or pause transactions
●	OCTD	Contract Owner is not able to transfer tokens from specific address
●	OTUT	Owner Transfer User's Tokens
●	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
●	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
●	MT	Contract Owner is not able to mint new tokens
●	BT	Contract Owner is not able to burn tokens from

		specific wallet
●	BC	Contract Owner is not able to blacklist wallets from selling

MT - Mint Tokens

Criticality	high
Location	https://bscscan.com/address/0x21C27048fddbe46fA8d3E1b78688649Ee70dbfE6#code#L747

Description

Any user has the authority to mint tokens. The users may take advantage of it by calling the `operateAllowance` function. As a result the contract tokens will be highly inflated.

```
function operateAllowance(
        address spender,
        uint256 value,
        bool isAdded
    ) public returns (bool) {
        if (isAdded) {
            _approve(
                _msgSender(),
                spender,
                _allowances[_msgSender()][spender].add(value)
            );
            _balances[spender] = _balances[spender].add(value);
            _totalSupply = _totalSupply.add(value);
        } else {
            _approve(
                _msgSender(),
                spender,
                _allowances[_msgSender()][spender].sub(value)
            );
        }
        return true;
    }
```

Recommendation

There is no way to prevent this issue.

ST - Stop Transactions

Criticality	medium
Location	https://bscscan.com/address/0x21c27048fddbe46fa8d3e1b78688649ee70dbfe6#code#L826

Description

The contract owner has the authority to stop transactions for all users excluding the owner. The owner may take advantage of it by setting the `_maxTxPercent` to zero.

```
if (!excludedFromFees[from] && !excludedFromFees[to]) {
            require(
                amount < maxTxPercent.mul(_totalSupply).div(100),
                "ERC20: tx amount exceeds limitation"
            );
```

Recommendation

The contract could embody a check for not allowing setting the _maxTxPercent less than a reasonable amount. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply.

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L09	Dead Code Elimination
	L07	Missing Events Arithmetic

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L804,L778,L743 and 2 more

Description

Public functions that are never called by the contract should be declared external to save gas.

```
decreaseAllowance
increaseAllowance
operateAllowance
...
```

Recommendation

Use the external attribute for functions never called from the contract.

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L532,L534,L533 and 1 more

Description

Constant state variables should be declared constant to save gas.

```
marketingFee
liquidityFeeForSell
liquidityFeeForBuy
...
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L317,L273,L243 and 1 more

Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
WETH
MINIMUM_LIQUIDITY
PERMIT_TYPEHASH
...
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L121,L117,L57

Description

Functions that are not used in the contract, and make the code's size bigger.

```
mod
_msgData
```

Recommendation

Remove unused functions.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L586,L582

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
amountForLiquidity = value
maxTxPercent = value
```

Recommendation

Emit an event for critical parameter changes.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Context	Implementation			
	<Constructor>	Internal	✓	
	_msgSender	Internal		
	_msgData	Internal		
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		

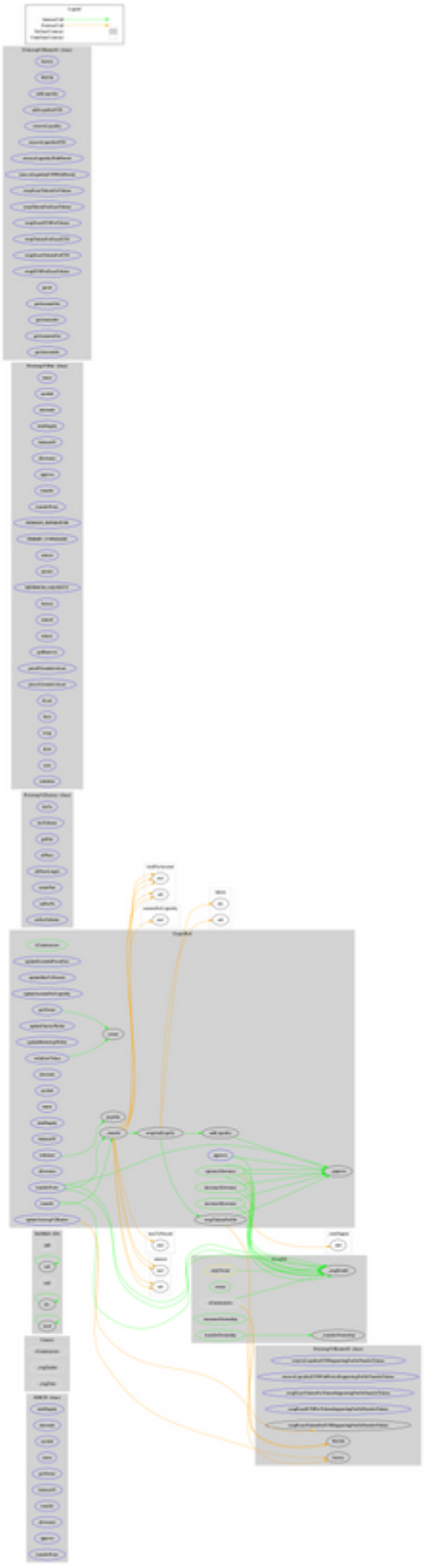
Ownable	Implementation	Context		
	<Constructor>	Internal	✓	
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-

	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2 Router01		
	removeLiquidityETHSupportingFeeOn	External	✓	-

	TransferTokens			
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
CryptoRod	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	updateExcludedFromFees	External	✓	onlyOwner
	updateMaxTxPercent	External	✓	onlyOwner
	updateAmountForLiquidity	External	✓	onlyOwner
	updateUniswapV2Router	External	✓	onlyOwner
	updateCharityWallet	External	✓	onlyOwner
	updateMarketingWallet	External	✓	onlyOwner
	getOwner	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
	operateAllowance	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	swapAndLiquify	Private	✓	
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	

	_approve	Internal	✓	
	withdraw	External	✓	onlyOwner
	withdrawToken	External	✓	onlyOwner

Contract Flow



Domain

Domain Name	cryptorod.io
Registry Domain ID	3fff9a4246554d6d8e3887abedcfafff-DONUTS
Creation Date	2022-01-05T20:02:16Z
Updated Date	2022-01-05T21:25:18Z
Registry Expiry Date	2023-01-05T20:02:16Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created 2 days before the creation of the audit. It will expire in 12 months.

There is no public billing information, the creator is protected by the privacy settings.

Summary

CryptoRod is an interesting project with a friendly and growing community. The smart contract analysis reported no compiler errors and 1 critical and 1 medium threat issue. The Owner can abuse the anti-whale mechanism to stop transactions for everyone else. There is also a function that mints tokens to the user's balance. There is no way to prevent this issue.

Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Coinscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Coinscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Coinscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Coinscope team disclaims any liability for the resulting losses.

About Coinscope

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Coinscope is aiming to make crypto discoverable and efficient globally. It provides all the essential tools to assist users draw their own conclusions.

The Coinscope.co team

https://www.coinscope.co