Cyberscope

Audit Report

Tokenreward.io

April 2022

Type	BEP20
Network	BSC
Address	0x98e7c28A86D3D50EBfC80E86dBB02EF0d38E84FD
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	TokenReward
Compiler Version	v0.8.13+commit.abaa5c0e
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x98e7c28A86D3D50EBfC80E86dBB02EF0d38E84FD
Symbol	Reward
Decimals	18
Total Supply	325,000,000
Domain	tokenreward.io

Source Files

Filename	**SHA256**
contract.sol	65e38ad01ba806805d9a06c634feec7b9547c6162f85c98d7d419daf2cdbd676

Audit Updates

Initial Audit	14th April 2022
Corrected	

Contract Analysis

🔴 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

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L663

Description

The contract owner has the authority to stop the sales for all users excluding the owner. The owner may take advantage of it by setting a high value to `sellFeet`.

```
if (recipient == pair) {
    _totalFee =
        totalFee.add(sellFeet).add(sellFeef)
        .add(sellFeel);
```

Recommendation

Check the Exceed Limit Fees Manipulation section.

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.

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L997

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the `setFees` function with a high percentage value to the `_sellFeef` variable.

```
function setFees(
    uint256 _liquidityFee,
    uint256 _treasuryFee,
    uint256 _feeDenominator,
    uint256 _firePitFee,
    uint256 _sellFeel,
    uint256 _sellFeet,
    uint256 _sellFeef,
    uint256 _riskFreeValueFee
) external onlyOwner {
    liquidityFee = _liquidityFee;
    treasuryFee = _treasuryFee;
    firePitFee = _firePitFee;
    sellFeel = _sellFeel;
    sellFeet = _sellFeet;
    sellFeef = _sellFeef;
    riskFreeValueFee = _riskFreeValueFee;
    totalFee =
liquidityFee.add(treasuryFee).add(firePitFee).add(riskFreeValueFee);
    feeDenominator = _feeDenominator;
    require(totalFee < feeDenominator / 4);
}
```

Recommendation

The contract could embody a check for the maximum acceptable value.

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.

BC - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L616

Description

The contract owner has the authority to stop contracts from transactions. The owner may take advantage of it by calling the `setBotBlacklist` function.

```
require(!blacklist[sender] && !blacklist[recipient], "in_blacklist");
```

Recommendation

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
🟠	MTS	Manipulate Total Supply
⚪	L01	Public Function could be Declared External
⚪	L02	State Variables could be Declared Constant
⚪	L04	Conformance to Solidity Naming Conventions
⚪	L05	Unused State Variable
⚪	L07	Missing Events Arithmetic
⚪	L09	Dead Code Elimination
⚪	L13	Divide before Multiply Operation

MTS - Manipulate Total Supply

Criticality	medium
Location	contract.sol#L565

Description

Owner is able to manipulate total supply. This change will have a direct impact on the token price and Market Cap.

```
function manualrebase() external onlyOwner {
    require(!inSwap, "Try again");
    rebase();
}
```

Recommendation

The contract owner should carefully manage the adjustment of the circulating supply (increases or decreases), according to the token’s price fluctuations.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L352,365,370,396,400,404,781,962

Description

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

```
setPairAddress
isOverLiquified
decimals
symbol
name
transferOwnership
renounceOwnership
owner
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L450,451,420,418,419,462

Description

Constant state variables should be declared constant to save gas.

```
swapEnabled
_symbol
_name
_decimals
ZERO
DEAD
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L143,144,161,181,785,789,840,849,912,932,933,934,935,953,957,962,966,984,988,998,999,1000,1001,1002,1003,1004,1005,1019,418,419,420,423,450,451,457,479,480,481,482,483,484

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.

```
_totalSupply
_lastAddLiquidityTime
_lastRebasedTime
_initRebaseStartTime
_autoAddLiquidity
_autoRebase
owner_address
ZERO
DEAD
...
```

Recommendation

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

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L10

Description

There are segments that contain unused state variables.

```
MAX_INT256
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L984,988,997

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.

```
liquidityFee = _liquidityFee
rebaseFrequency = _rebaseFrequency
rewardYield = _rewardYield
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L38

Description

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

```
abs
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L538,652,943

Description

Performing divisions before multiplications may cause lose of prediction.

```
liquidityBalance = _gonBalances[pair].div(_gonsPerFragment)
_gonBalances[riskFreeValueReceiver] =
_gonBalances[riskFreeValueReceiver].add(gonAmount.div(feeDenominator).mul(_riskF
reeValueFee))
_gonBalances[autoLiquidityReceiver] =
_gonBalances[autoLiquidityReceiver].add(gonAmount.div(feeDenominator).mul(_liqui
dityFee))
_gonBalances[address(this)] =
_gonBalances[address(this)].add(gonAmount.div(feeDenominator).mul(_treasuryFee))
_gonBalances[firePit] =
_gonBalances[firePit].add(gonAmount.div(feeDenominator).mul(_firePitFee))
feeAmount = gonAmount.div(feeDenominator).mul(_totalFee)
times = deltaTime.div(rebaseFrequency)
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	transfer	External	✓	-
	approve	External	✓	-
	transferFrom	External	✓	-
IPancakeSwap Pair	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		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IPancakeSwap Router	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		-
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IPancakeSwapFactory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IPinkAntiBot	Interface			
	setTokenOwner	External	✓	-
	onPreTransferCheck	External	✓	-
Ownable	Implementation			

	<Constructor>	Public	✓	-
	owner	Public		-
	isOwner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
ERC20Detailed	Implementation	IERC20		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
TokenReward	Implementation	ERC20Detailed, Ownable		
	<Constructor>	Public	✓	ERC20Detailed Ownable
	rebase	Internal	✓	
	manualrebase	External	✓	onlyOwner
	transfer	External	✓	validRecipient
	transferFrom	External	✓	validRecipient
	_basicTransfer	Internal	✓	
	_transferFrom	Internal	✓	
	takeFee	Internal	✓	
	addLiquidity	Internal	✓	swapping
	swapBack	Internal	✓	swapping
	isOverLiquified	Public		-
	setEnableAntiBot	External	✓	onlyOwner
	withdrawAllToOwner	External	✓	swapping onlyOwner
	shouldTakeFee	Internal		
	shouldRebase	Internal		
	shouldAddLiquidity	Internal		
	shouldSwapBack	Internal		
	setAutoRebase	External	✓	onlyOwner

	setAutoAddLiquidity	External	✓	onlyOwner
	allowance	External		-
	decreaseAllowance	External	✓	-
	increaseAllowance	External	✓	-
	approve	External	✓	-
	checkFeeExempt	External		-
	getCirculatingSupply	Public		-
	isNotInSwap	External		-
	manualSync	External	✓	-
	setFeeReceivers	External	✓	onlyOwner
	getLiquidityBacking	Public		-
	setWhitelist	External	✓	onlyOwner
	setBotBlacklist	External	✓	onlyOwner
	setPairAddress	Public	✓	onlyOwner
	setLP	External	✓	onlyOwner
	totalSupply	External		-
	balanceOf	External		-
	isContract	Internal		
	setRewardYield	External	✓	onlyOwner
	setRebaseFrequency	External	✓	onlyOwner
	setTargetLiquidity	External	✓	onlyOwner
	setFees	External	✓	onlyOwner
	swipe	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-

Contract Flow

Domain Info

Domain Name	tokenreward.io
Registry Domain ID	d89db0181a3646b9b879f178114e6306-DONUTS
Creation Date	2021-08-23T17:25:01Z
Updated Date	2022-04-04T16:17:20Z
Registry Expiry Date	2022-08-23T17:25:01Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created 8 months before the creation of the audit. It will expire in 4 months.

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

Summary

Tokenreward.io is an interesting project that has a friendly and growing community. There are some functions that can be abused by the owner, like manipulating fees, stopping transactions and blacklisting contracts. The contract can be converted into a honeypot and prevent users from selling if the owner abuses the admin functions. The contract cannot send tokens to the dead address. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

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.

Cyberscope 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 Cyberscope 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 Cyberscope 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 Cyberscope team disclaims any liability for the resulting losses.

About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

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.

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

The Cyberscope team

https://www.cyberscope.io