



Cyberscope

# Audit Report

## Dragon-li

March 2022

Type        BEP20

Network     BSC

Address     0x4d6263Dc3Bc7229bb4B2585da62Cc0d5B2C9c1a1

Audited by   © cyberscope

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>Contract Review</b>	<b>3</b>
<b>Audit Updates</b>	<b>3</b>
<b>Contract Analysis</b>	<b>4</b>
<b>Contract Diagnostics</b>	<b>5</b>
<b>L01 - Public Function could be Declared External</b>	<b>6</b>
Description	6
Recommendation	6
<b>L05 - Unused State Variable</b>	<b>7</b>
Description	7
Recommendation	7
<b>L04 - Conformance to Solidity Naming Conventions</b>	<b>8</b>
Description	8
Recommendation	8
<b>L09 - Dead Code Elimination</b>	<b>9</b>
Description	9
Recommendation	9
<b>L12 - Using Variables before Declaration</b>	<b>10</b>
Description	10
Recommendation	10
<b>L07 - Missing Events Arithmetic</b>	<b>11</b>
Description	11
Recommendation	11
<b>L15 - Local Scope Variable Shadowing</b>	<b>12</b>
Description	12
Recommendation	12

<b>L14 - Uninitialized Variables in Local Scope</b>	<b>13</b>
<b>Description</b>	<b>13</b>
<b>Recommendation</b>	<b>13</b>
<b>Contract Functions</b>	<b>14</b>
<b>Contract Flow</b>	<b>24</b>
<b>Domain Info</b>	<b>25</b>
<b>Summary</b>	<b>26</b>
<b>Disclaimer</b>	<b>27</b>
<b>About Cyberscope</b>	<b>28</b>

## Contract Review

<b>Contract Name</b>	BABYTOKEN
<b>Compiler Version</b>	v0.8.4+commit.c7e474f2
<b>Optimization</b>	200 runs
<b>Licence</b>	MIT
<b>Explorer</b>	<a href="https://bscscan.com/token/0x4d6263Dc3Bc7229bb4B2585da62Cc0d5B2C9c1a1">https://bscscan.com/token/0x4d6263Dc3Bc7229bb4B2585da62Cc0d5B2C9c1a1</a>
<b>Symbol</b>	LIHUA
<b>Decimals</b>	18
<b>Total Supply</b>	1,000,000,000
<b>Source</b>	contract.sol
<b>Domain</b>	lihuatoken.com

## Audit Updates

<b>Initial Audit</b>	12th March 2022
<b>Corrected</b>	

# 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

# Contract Diagnostics

● Critical    ● Medium    ● Minor

Severity	Code	Description
●	L01	Public Function could be Declared External
●	L05	Unused State Variable
●	L04	Conformance to Solidity Naming Conventions
●	L09	Dead Code Elimination
●	L12	Using Variables before Declaration
●	L07	Missing Events Arithmetic
●	L15	Local Scope Variable Shadowing
●	L14	Uninitialized Variables in Local Scope

## L01 - Public Function could be Declared External

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L203,211,228,235,254,262,273,291,319,338 and 33 more

### Description

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

```
isExcludedFromDividends  
dividendTokenBalanceOf  
withdrawableDividendOf  
isExcludedFromFees  
updateGasForProcessing  
setAutomatedMarketMakerPair  
excludeMultipleAccountsFromFees  
updateUniswapV2Router  
updateDividendTracker  
...
```

### Recommendation

Use the external attribute for functions never called from the contract

## L05 - Unused State Variable

**Criticality**

minor

**Location**

contract.sol#L1843,1774

### Description

There are segments that contain unused state variables.

```
__gap  
MAX_INT256
```

### Recommendation

Remove unused state variables.



## L04 - Conformance to Solidity Naming Conventions

**Criticality**

minor

**Location**

contract.sol#L931,1315,1319,1328,1386,1391,1693,1725,1730,1774 and 14 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.

```
_marketingWalletAddress  
_account  
magnitude  
_owner  
_symbol  
_name  
_rewardToken  
__DividendPayingToken_init  
MINIMUM_LIQUIDITY  
...
```

### Recommendation

Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>

## L09 - Dead Code Elimination

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L845,879,859,1315,2220,416,1558,1889

### Description

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

```
abs
_transfer
_burn
__Context_init
predictDeterministicAddress
cloneDeterministic
```

### Recommendation

Remove unused functions.

## L12 - Using Variables before Declaration

**Criticality**

minor

**Location**

contract.sol#L3036,3035,3034

### Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
iterations  
claims  
lastProcessedIndex
```

### Recommendation

The variables should be declared before any usage of them.

## L07 - Missing Events Arithmetic

**Criticality**

minor

**Location**

contract.sol#L2720,2786,2792,2798

### 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.

```
marketingFee = value
liquidityFee = value
totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee)
swapTokensAtAmount = amount
```

### Recommendation

Emit an event for critical parameter changes.

## L15 - Local Scope Variable Shadowing

**Criticality**

minor

**Location**

contract.sol#L2108,2109,2168,2175,2187,2201

### Description

There are variables that are defined in the local scope containing the same name from an upper scope.

```
_owner  
_symbol  
_name
```

### Recommendation

The local variables should have different names from the upper scoped variables.

## L14 - Uninitialized Variables in Local Scope

**Criticality**

minor

**Location**

contract.sol#L3036,3035,3034

### Description

There are variables that are defined in the local scope and are not initialized.

```
iterations  
claims  
lastProcessedIndex
```

### Recommendation

All the local scoped variables should be initialized.

# Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
<b>IERC20</b>	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
<b>IERC20Metadata</b>	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
<b>Context</b>	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
<b>ERC20</b>	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-

	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
<b>Ownable</b>	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
<b>SafeMath</b>	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	Internal		
<b>Clones</b>	Library			
	clone	Internal	✓	



	cloneDeterministic	Internal	✓	
	predictDeterministicAddress	Internal		
	predictDeterministicAddress	Internal		
<b>IUniswapV2Factory</b>	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
<b>IUniswapV2Router01</b>	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		-

<b>IUniswapV2Router02</b>	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
<b>IERC20Upgradeable</b>	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
<b>IERC20MetadataUpgradeable</b>	Interface	IERC20Upgradeable		
	name	External		-
	symbol	External		-
	decimals	External		-
<b>Initializable</b>	Implementation			
<b>ContextUpgradeable</b>	Implementation	Initializable		
	__Context_init	Internal	✓	initializer
	__Context_init_unchained	Internal	✓	initializer
	_msgSender	Internal		
	_msgData	Internal		

<b>ERC20Upgradable</b>	Implementation	Initializable, ContextUpgradable, IERC20Upgradable, IERC20MetadataUpgradable		
	__ERC20_init	Internal	✓	initializer
	__ERC20_init_unchained	Internal	✓	initializer
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
<b>OwnableUpgradable</b>	Implementation	Initializable, ContextUpgradable		
	__Ownable_init	Internal	✓	initializer
	__Ownable_init_unchained	Internal	✓	initializer
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	

<b>IUniswapV2Pair</b>	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	✓	-
<b>SafeMathInt</b>	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		

	toUint256Safe	Internal		
<b>SafeMathUint</b>	Library			
	toInt256Safe	Internal		
<b>IterableMapping</b>	Library			
	get	Public		-
	getIndexOfKey	Public		-
	getKeyAtIndex	Public		-
	size	Public		-
	set	Public	✓	-
	remove	Public	✓	-
<b>DividendPayingTokenInterface</b>	Interface			
	dividendOf	External		-
	withdrawDividend	External	✓	-
<b>DividendPayingTokenOptionalInterface</b>	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-
<b>DividendPayingToken</b>	Implementation	ERC20Upgradable, OwnableUpgradable, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface		
	__DividendPayingToken_init	Internal	✓	initializer
	distributeCAKEDividends	Public	✓	onlyOwner

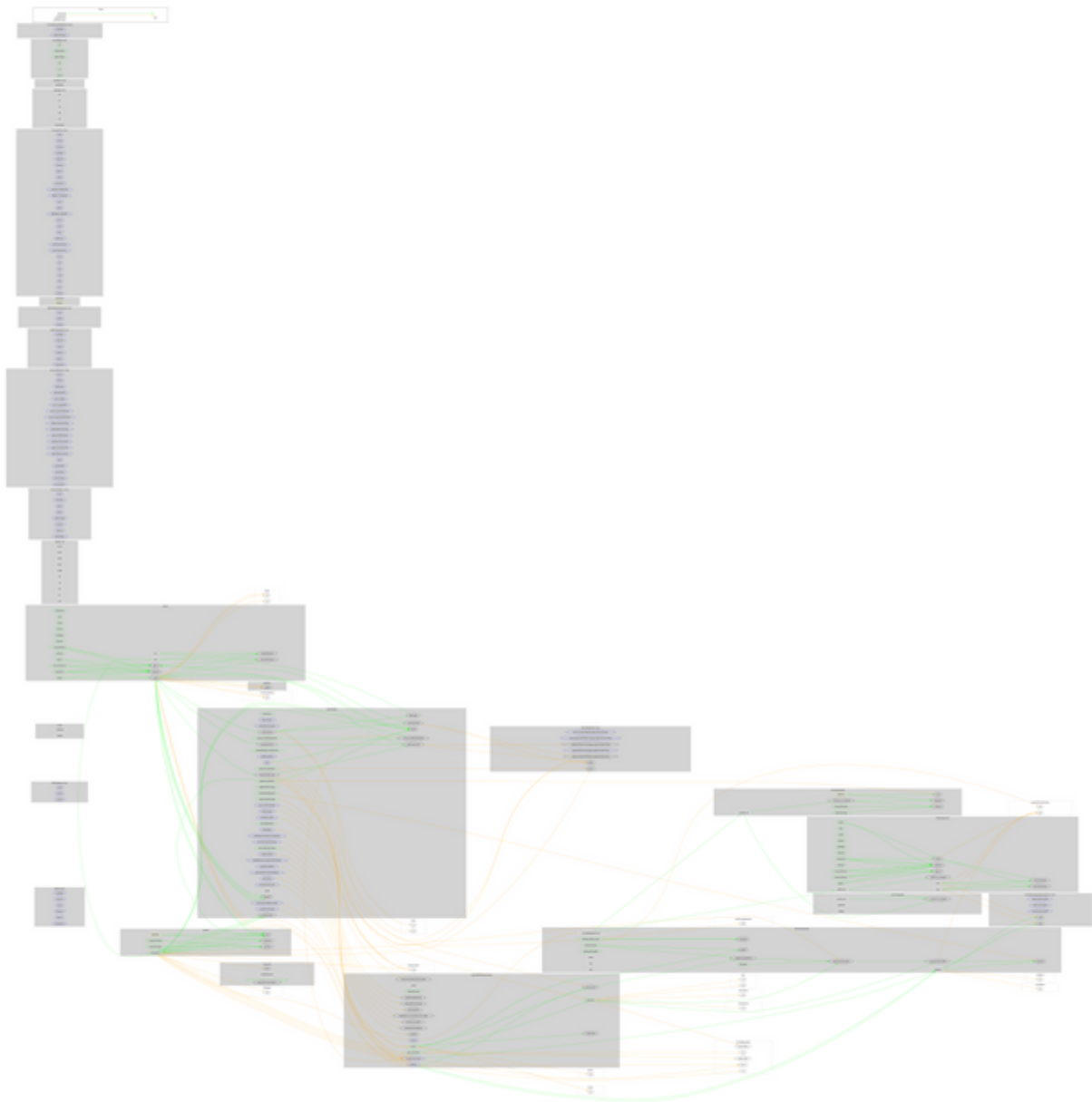
	withdrawDividend	Public	✓	-
	_withdrawDividendOfUser	Internal	✓	
	dividendOf	Public		-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	
<b>BABYTOKENDividendTracker</b>	Implementation	OwnableUp gradeable, DividendPay ingToken		
	initialize	External	✓	initializer
	_transfer	Internal		
	withdrawDividend	Public		-
	excludeFromDividends	External	✓	onlyOwner
	isExcludedFromDividends	Public		-
	updateClaimWait	External	✓	onlyOwner
	updateMinimumTokenBalanceForDividends	External	✓	onlyOwner
	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getAccount	Public		-
	getAccountAtIndex	Public		-
	canAutoClaim	Private		
	setBalance	External	✓	onlyOwner
	process	Public	✓	-
	processAccount	Public	✓	onlyOwner
<b>BaseToken</b>	Implementation			
<b>BABYTOKEN</b>	Implementation	ERC20, Ownable, BaseToken		

	<Constructor>	Public	Payable	ERC20
	<Receive Ether>	External	Payable	-
	setSwapTokensAtAmount	External	✓	onlyOwner
	updateDividendTracker	Public	✓	onlyOwner
	updateUniswapV2Router	Public	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	excludeMultipleAccountsFromFees	Public	✓	onlyOwner
	setMarketingWallet	External	✓	onlyOwner
	setTokenRewardsFee	External	✓	onlyOwner
	setLiquiditFee	External	✓	onlyOwner
	setMarketingFee	External	✓	onlyOwner
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateGasForProcessing	Public	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getClaimWait	External		-
	updateMinimumTokenBalanceForDividends	External	✓	onlyOwner
	getMinimumTokenBalanceForDividends	External		-
	getTotalDividendsDistributed	External		-
	isExcludedFromFees	Public		-
	withdrawableDividendOf	Public		-
	dividendTokenBalanceOf	Public		-
	excludeFromDividends	External	✓	onlyOwner
	isExcludedFromDividends	Public		-
	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-
	processDividendTracker	External	✓	-
	claim	External	✓	-
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	_transfer	Internal	✓	
	swapAndSendToFee	Private	✓	
	swapAndLiquify	Private	✓	
	swapTokensForEth	Private	✓	

	swapTokensForCake	Private	✓	
	addLiquidity	Private	✓	
	swapAndSendDividends	Private	✓	



# Contract Flow



## Domain Info

<b>Domain Name</b>	
<b>Registry Domain ID</b>	2678525025_DOMAIN_COM-VRSN
<b>Creation Date</b>	2022-03-01T13:59:42.00Z
<b>Updated Date</b>	0001-01-01T00:00:00.00Z
<b>Registry Expiry Date</b>	2023-03-01T13:59:42.00Z
<b>Registrar WHOIS Server</b>	whois.namecheap.com
<b>Registrar URL</b>	http://www.namecheap.com
<b>Registrar</b>	NAMECHEAP INC
<b>Registrar IANA ID</b>	1068

The domain has been created 11 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

The Smart Contract analysis reported no compiler error or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. There is also a limit of max 25% fees.

## 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>