COINSCOPE

Audit Report

Babylon

December 2021

Type	BEP20
Address	0x4CCb3D85E7298997e937311023464Fa2127e17ab
Audited by	© coinscope

Table of Contents

Contract Review

Contract Name	Babylon
Compiler Version	v0.8.11+commit.d7f03943
Optimization	200 runs
Licence	MIT
Explorer	https://testnet.bscscan.com/address/0x4ccb3d85e7298997e937311023464fa2127e17ab
Symbol	BBL
Decimals	9
Total Supply	1,000,000,000,000,000
Website	

Audit Updates

Initial Audit	30th of December 2021
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
🔵	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

BC - Contract Owner is not able to blacklist wallets from selling

Criticality	medium
Location	contract.sol#L572

Description

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

```
require(amount <= _maxTxAmount || isTxLimitExempt[sender], "TX Limit Exceeded");
```

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
●	L07	Missing Events Arithmetic
●	L11	Unnecessary Boolean equality
●	L10	Costly Operations Inside a Loop
●	L09	Dead Code Elimination
●	L04	Conformance to Solidity Naming Conventions
●	L05	Unused State Variable
●	L02	State Variables could be Declared Constant
●	L01	Public Function could be Declared External

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L278,L718,L728 and 4 more

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.

```
minPeriod = _minPeriod
autoBuybackCap = _cap
minHoldingDays = _minHoldingDays
...
```

Recommendation

Emit an event for critical parameter changes.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L500

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compare to true or false.

```
require(bool,string)(buyBacker[msg.sender] == true,)
```

Recommendation

Remove the equality to the boolean constant.

L10 - Costly Operations Inside a Loop

Criticality	minor
Location	contract.sol#L319,L350,L319

Description

Costly operations inside a loop might waste gas, so optimizations are justified. Incrementing state variables in a loop incurs a lot of gas because of expensive SSTOREs, which might lead to an out-of-gas.

```
currentIndex = 0
totalDistributed = totalDistributed.add(amount)
currentIndex ++
```

Recommendation

Use a local variable to hold the loop computation result.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L76,L65,L83 and 5 more

Description

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

```
div
mod
mod
...
```

Recommendation

Remove unused functions.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L175,L278,L278 and 37 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
_minPeriod
_minDistribution
...
```

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#L399

Description

There are segments that contains unused state variable.

```
Babylon
```

Recommendation

Remove unused state variables.

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L399,L401,L402 and 5 more

Description

Constant state variables should be declared constant to save gas.

```
BUSD
DEAD
ZERO
...
```

Recommendation

Add the constant attribute to state variables that never change.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L132,L139,L160

Description

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

```
authorize
unauthorize
transferOwnership
```

Recommendation

Use the external attribute for functions never called from the contract

Contract Functions

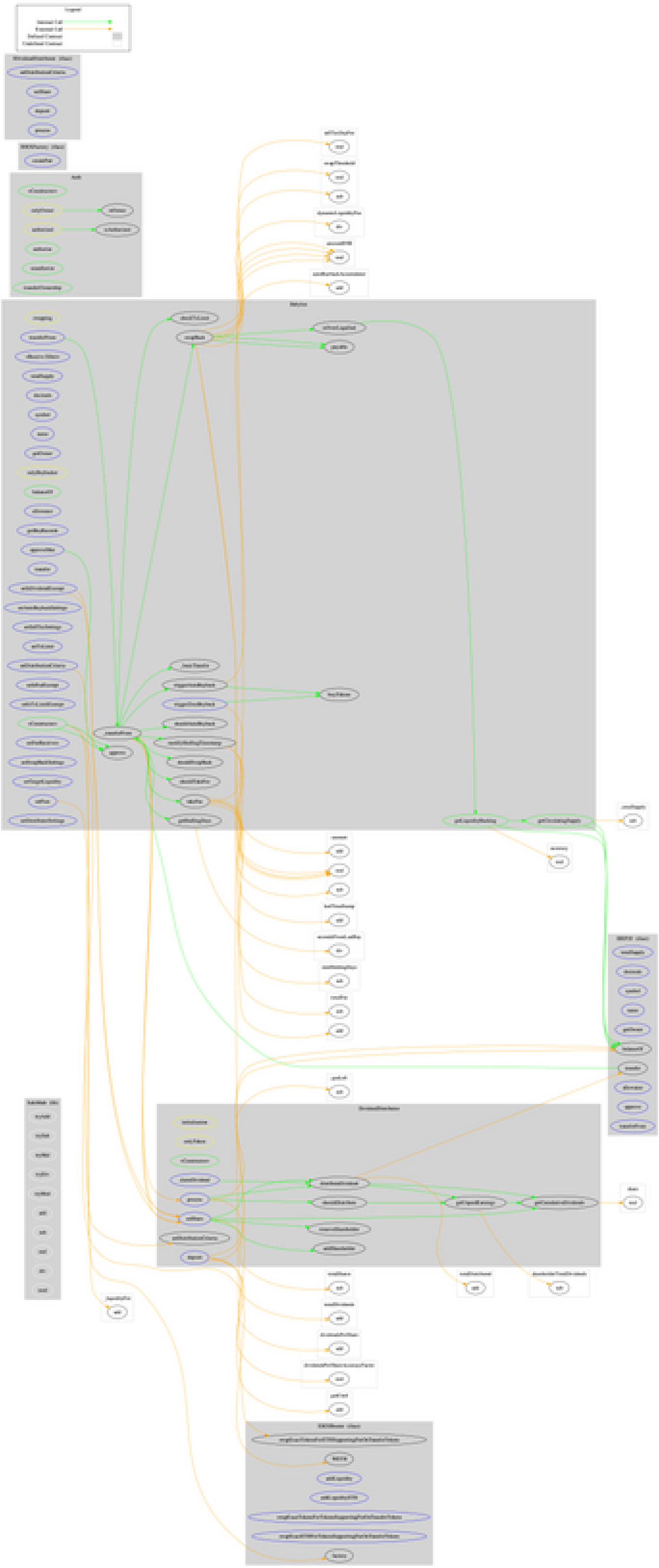
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
SafeMath	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	Internal		
IBEP20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Auth	Implementation			
	<Constructor>	Public	✓	-

	authorize	Public	✓	onlyOwner
	unauthorize	Public	✓	onlyOwner
	isOwner	Public		-
	isAuthorized	Public		-
	transferOwnership	Public	✓	onlyOwner
IDEXFactory	Interface			
	createPair	External	✓	-
IDEXRouter	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IDividendDistributor	Interface			
	setDistributionCriteria	External	✓	-
	setShare	External	✓	-
	deposit	External	Payable	-
	process	External	✓	-
DividendDistributor	Implementation	IDividendDistributor		
	<Constructor>	Public	✓	-
	setDistributionCriteria	External	✓	onlyToken
	setShare	External	✓	onlyToken
	deposit	External	Payable	onlyToken
	process	External	✓	onlyToken
	shouldDistribute	Internal		
	distributeDividend	Internal	✓	

	claimDividend	External	✓	-
	getUnpaidEarnings	Public		-
	getCumulativeDividends	Internal		
	addShareholder	Internal	✓	
	removeShareholder	Internal	✓	
Babylon	Implementation	IBEP20, Auth		
	<Constructor>	Public	✓	Auth
	<Receive Ether>	External	Payable	-
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	Public		-
	allowance	External		-
	getBuyRecords	External		-
	approve	Public	✓	-
	approveMax	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	_transferFrom	Internal	✓	
	_basicTransfer	Internal	✓	
	checkTxLimit	Internal		
	modifyHodlingTimestamp	Internal	✓	
	getHodlingDays	Internal		
	shouldTakeFee	Internal		
	takeFee	Internal	✓	
	shouldSwapBack	Internal		
	swapBack	Internal	✓	swapping
	shouldAutoBuyback	Internal		
	triggerZeusBuyback	External	✓	authorized
	triggerAutoBuyback	Internal	✓	
	buyTokens	Internal	✓	swapping
	setAutoBuybackSettings	External	✓	authorized

	setSellTaxSettings	External	✓	authorized
	setTxLimit	External	✓	authorized
	setIsDividendExempt	External	✓	authorized
	setIsFeeExempt	External	✓	authorized
	setIsTxLimitExempt	External	✓	authorized
	setFees	External	✓	authorized
	setFeeReceivers	External	✓	authorized
	setSwapBackSettings	External	✓	authorized
	setTargetLiquidity	External	✓	authorized
	setDistributionCriteria	External	✓	authorized
	setDistributorSettings	External	✓	authorized
	getCirculatingSupply	Public		-
	getLiquidityBacking	Public		-
	isOverLiquified	Public		-

Contract Flow



Summary

Babylon tokens contain a proportional tax logic. The tax is calculated according to the last trading period. The contract analysis reported only one medium level issue. The owner has the ability to blacklist contracts. The rest findings are minor and optional for the contract authors. 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.

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