



Cyberscope

Audit Report

Candybar

September 2022

Type BEP20

Network BSC

Address 0xb483a090252b260f9b5f0a1bd0d90469432aef8f

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Contract Analysis	4
ULTW - Transfers Liquidity to Team Wallet	5
Description	5
Recommendation	5
Contract Diagnostics	6
BLC - Business Logic Concern	7
Description	7
Recommendation	7
L01 - Public Function could be Declared External	8
Description	8
Recommendation	8
L02 - State Variables could be Declared Constant	9
Description	9
Recommendation	9
L03 - Redundant Statements	10
Description	10
Recommendation	10
L04 - Conformance to Solidity Naming Conventions	11
Description	11
Recommendation	11
L07 - Missing Events Arithmetic	12
Description	12

Recommendation	12
L13 - Divide before Multiply Operation	13
Description	13
Recommendation	13
L14 - Uninitialized Variables in Local Scope	14
Description	14
Recommendation	14
Contract Functions	15
Contract Flow	18
Summary	19
Disclaimer	20
About Cyberscope	21

Contract Review

Contract Name	CandybarFinance
Compiler Version	v0.8.8+commit.dddeac2f
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0xb483a090252b260f9B5f0A1BD0d90469432aEF8F
Symbol	CANDYBAR
Decimals	18
Total Supply	10,000,000

Source Files

Filename	SHA256
contract.sol	91b04de87943bddf985f03304340bccca75b508286def32aeb183280efc8504c4

Audit Updates

Initial Audit	30th September 2022
Corrected	

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OCTD	Transfers Contract's Tokens	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	ULTW	Transfers Liquidity to Team Wallet	Unresolved
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

ULTW - Transfers Liquidity to Team Wallet

Criticality	minor / informative
Location	contract.sol#L747
Status	Unresolved

Description

The contract owner has the authority to transfer funds without limit to the team wallet. These funds have been accumulated from fees collected from the contract. The owner may take advantage of it by calling the `rescueBNB` methods.

```
function rescueBNB(uint256 weiAmount) external onlyOwner {  
    payable(owner()).transfer(weiAmount);  
}
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may volatile the token's price.

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 / Informative

Severity	Code	Description	Status
●	BLC	Business Logic Concern	Unresolved
●	L01	Public Function could be Declared External	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L03	Redundant Statements	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L07	Missing Events Arithmetic	Unresolved
●	L13	Divide before Multiply Operation	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved

BLC - Business Logic Concern

Criticality	minor / informative
Location	contract.sol#L401,738
Status	Unresolved

Description

Misleading starting limits. The limits are initialised with greater values than the corresponding max limits.

```
uint256 public maxBuyLimit = 1e5 * 10**18;
uint256 public maxSellLimit = 1e5 * 10**18;
uint256 public maxWalletLimit = 1e5 * 10**18;
//
function updateMaxTxLimit(uint256 maxBuy, uint256 maxSell, uint256 maxWallet) external
onlyOwner {
    require(maxBuy >= 1e4, "Cannot set max buy amount lower than 0.1%");
    require(maxSell >= 1e4, "Cannot set max sell amount lower than 0.1%");
    require(maxWallet >= 1e5, "Cannot set max wallet amount lower than 1%");
    maxBuyLimit = maxBuy * 10**decimals();
    maxSellLimit = maxSell * 10**decimals();
    maxWalletLimit = maxWallet * 10**decimals();
}
```

Recommendation

The team is advised to carefully check if the implementation follows the expected business logic.

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L207,90,336,163,181,82,133,340,230,146,114
Status	Unresolved

Description

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

```
increaseAllowance  
symbol  
renounceOwnership  
approve  
transferFrom  
name  
transfer  
transferOwnership  
decreaseAllowance  
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contract.sol#L403
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
launchtax
```

Recommendation

Add the constant attribute to state variables that never change.

L03 - Redundant Statements

Criticality	minor / informative
Location	contract.sol#L5
Status	Unresolved

Description

The contract contains statements that are not used and have no effect. As a result, those segments increase the code size of the contract unnecessarily.

Context

Recommendation

Remove the redundant statements in order to decrease the code size.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L673,672,407,675,584,682,684,674,683,359,724,681,56,695,666,702,58,401
Status	Unresolved

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.

```
_marketing  
SetBuyTaxes  
deadWallet  
_dev  
Liquify  
_liquidity  
WETH  
_address  
SetSellTaxes  
...
```

Recommendation

Follow the Solidity naming convention.

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

L07 - Missing Events Arithmetic

Criticality	minor / informative
Location	contract.sol#L718,702,666,734
Status	Unresolved

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.

```
coolDownTime = time * 1
deadline = _deadline
tokenLiquidityThreshold = new_amount * 10 ** decimals()
maxBuyLimit = maxBuy * 10 ** decimals()
```

Recommendation

Emit an event for critical parameter changes.

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	contract.sol#L584
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
unitBalance = deltaBalance / (denominator - swapTaxes.liquidity)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contract.sol#L534,536,533
Status	Unresolved

Description

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

```
feesum  
currentTaxes  
feeswap
```

Recommendation

All the local scoped variables should be initialized.

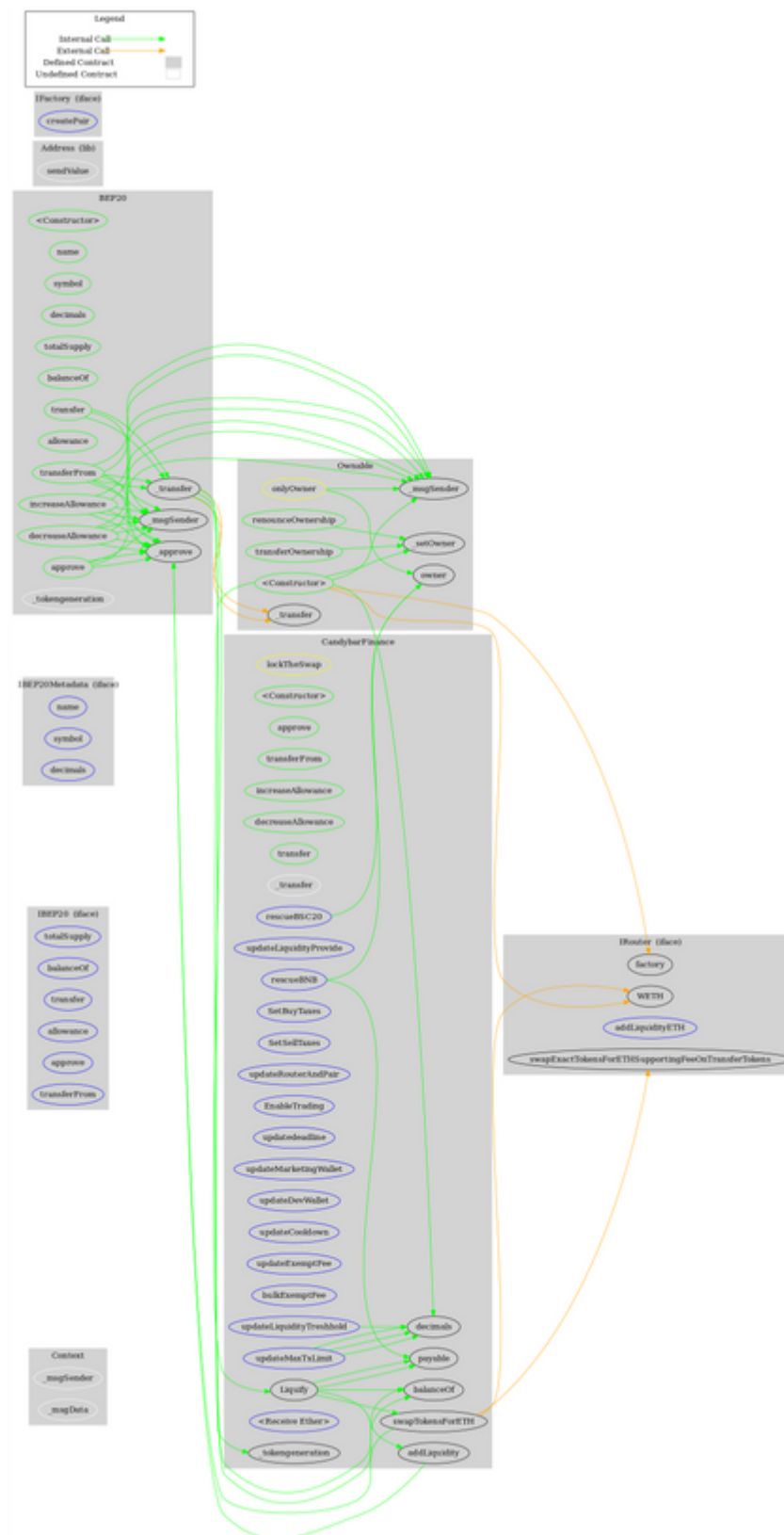
Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IBEP20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IBEP20Metadata	Interface	IBEP20		
	name	External		-
	symbol	External		-
	decimals	External		-
BEP20	Implementation	Context, IBEP20, IBEP20Meta data		
	<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	✓	
	_tokengeneration	Internal	✓	
	_approve	Internal	✓	
Address	Library			
	sendValue	Internal	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
IFactory	Interface			
	createPair	External	✓	-
IRouter	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	swapExactTokensForETHSupporting FeeOnTransferTokens	External	✓	-
CandybarFinance	Implementation	BEP20, Ownable		
	<Constructor>	Public	✓	BEP20
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-

	transfer	Public	✓	-
	_transfer	Internal	✓	
	Liquify	Private	✓	lockTheSwap
	swapTokensForETH	Private	✓	
	addLiquidity	Private	✓	
	updateLiquidityProvide	External	✓	onlyOwner
	updateLiquidityTreshhold	External	✓	onlyOwner
	SetBuyTaxes	External	✓	onlyOwner
	SetSellTaxes	External	✓	onlyOwner
	updateRouterAndPair	External	✓	onlyOwner
	EnableTrading	External	✓	onlyOwner
	updatedecline	External	✓	onlyOwner
	updateMarketingWallet	External	✓	onlyOwner
	updateDevWallet	External	✓	onlyOwner
	updateCooldown	External	✓	onlyOwner
	updateExemptFee	External	✓	onlyOwner
	bulkExemptFee	External	✓	onlyOwner
	updateMaxTxLimit	External	✓	onlyOwner
	rescueBNB	External	✓	onlyOwner
	rescueBSC20	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-

Contract Flow



Summary

The Smart Contract analysis reported one minor severity issue. The contract owner has the authority to transfer funds to the team's wallet. The contract has the ability to prevent sales with a frequency of fewer than 5 minutes. In addition, the contract applies a launch tax of 99% up to the first 5 blocks. Other than that, 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 10% 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 provide all the essential tools to assist users draw their own conclusions.



The Cyberscope team

<https://www.cyberscope.io>