



Cyberscope

# Audit Report

## **SincereBat**

May 2022

Type       BEP20

Network     BSC

Address     0x48ebb7cb09F1D4753C1d3A35c55d2507c64D34cb

Audited by  © cyberscope

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>Contract Review</b>	<b>3</b>
<b>Source Files</b>	<b>3</b>
<b>Audit Updates</b>	<b>3</b>
<b>Contract Analysis</b>	<b>4</b>
<b>ST - Stop Transactions</b>	<b>5</b>
Description	5
Recommendation	5
<b>ELFM - Exceed Limit Fees Manipulation</b>	<b>6</b>
Description	6
Recommendation	6
<b>BT - Burn Tokens</b>	<b>7</b>
Description	7
Recommendation	7
<b>Contract Diagnostics</b>	<b>8</b>
<b>L01 - Public Function could be Declared External</b>	<b>9</b>
Description	9
Recommendation	9
<b>L02 - State Variables could be Declared Constant</b>	<b>10</b>
Description	10
Recommendation	10
<b>L04 - Conformance to Solidity Naming Conventions</b>	<b>11</b>
Description	11
Recommendation	11
<b>L07 - Missing Events Arithmetic</b>	<b>12</b>
Description	12

<b>Recommendation</b>	<b>12</b>
<b>L09 - Dead Code Elimination</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>20</b>
<b>Domain Info</b>	<b>21</b>
<b>Summary</b>	<b>22</b>
<b>Disclaimer</b>	<b>23</b>
<b>About Cyberscope</b>	<b>24</b>

## Contract Review

<b>Contract Name</b>	SINCEREBAT
<b>Compiler Version</b>	v0.8.3+commit.8d00100c
<b>Optimization</b>	200 runs
<b>Licence</b>	None
<b>Explorer</b>	<a href="https://bscscan.com/token/0x48ebb7cb09F1D4753C1d3A35c55d2507c64D34cb">https://bscscan.com/token/0x48ebb7cb09F1D4753C1d3A35c55d2507c64D34cb</a>
<b>Symbol</b>	SBAT
<b>Decimals</b>	9
<b>Total Supply</b>	1,000,000,000,000
<b>Domain</b>	sincerebat.com

## Source Files

<b>Filename</b>	<b>SHA256</b>
<b>contract.sol</b>	003e8820333340a34650fe14318f3e5aa1ca78144bbe497ecd6576335c722f34

## Audit Updates

<b>Initial Audit</b>	6th May 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

## ST - Stop Transactions

Criticality	medium
Location	contract.sol#L1183

### 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 `_maxTxAmount` to zero.

```
if(from != owner() && to != owner())  
    require(amount <= _maxTxAmount, "Transfer amount exceeds the maxTxAmount.");
```

### Recommendation

The contract could embody a check for not allowing setting the `_maxTxAmount` 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.

## ELFM - Exceed Limit Fees Manipulation

<b>Criticality</b>	critical
<b>Location</b>	contract.sol#L1011

### 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 `setTaxFeePercent` function with a high percentage value.

```
function setTaxFeePercent(uint256 taxFee) external onlyOwner() {  
    _taxFee = taxFee;  
}
```

### 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.

## BT - Burn Tokens

<b>Criticality</b>	critical
<b>Location</b>	contract.sol#L850

### Description

The contract owner has the authority to decrease the total supply without limit. The owner may take advantage of it by calling the `burn` function.

```
function _burn(address account, uint256 amount) public virtual onlyOwner{
    require(account != address(0), "ERC20: burn from the zero address");

    // _beforeTokenTransfer(account, address(0), amount);

    _tTotal -= amount;
    // require(accountBalance >= amount, "ERC20: burn amount exceeds balance");
    //_balances[account] = accountBalance - amount;
    // _totalSupply -= amount;
}
```

### Recommendation

The owner should carefully manage the credentials of the owner's account. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one. The owner could also renounce the contract ownership for a period of time or pass the access to the zero address.



# 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
●	L07	Missing Events Arithmetic
●	L09	Dead Code Elimination

## L01 - Public Function could be Declared External

**Criticality**

minor

**Location**

contract.sol#L566,575,850,883,887,891,895,904,909,913,918,924,929,934,938,942,951,968,1003,1007,1033,1162

### Description

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

```
isExcludedFromFee  
setSwapAndLiquifyEnabled  
includeInFee  
excludeFromFee  
excludeFromReward  
reflectionFromToken  
deliver  
totalFees  
isExcludedFromReward  
...
```

### Recommendation

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

## L02 - State Variables could be Declared Constant

**Criticality**

minor

**Location**

contract.sol#L807,816,806,814,815,835

### Description

Constant state variables should be declared constant to save gas.

```
numTokensSellToAddToLiquidity
_symbol
_name
_developerWalletAddress
_decimals
_charityAddress
```

### Recommendation

Add the constant attribute to state variables that never change.

## L04 - Conformance to Solidity Naming Conventions

**Criticality**

minor

**Location**

contract.sol#L613,614,631,651,850,1033,1117,1123,1129,1135,834

### 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.

```
_maxTxAmount  
_amount  
_enabled  
_burn  
WETH  
MINIMUM_LIQUIDITY  
PERMIT_TYPEHASH  
DOMAIN_SEPARATOR  
...
```

### 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

**Location**

contract.sol#L850,1011,1015,1019,1023,1027

### 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.

```
_maxTxAmount = _tTotal.mul(maxTxPercent).div(10 ** 2)
_liquidityFee = liquidityFee
_charityFee = charityFee
_developerFee = developerFee
_taxFee = taxFee
_tTotal -= amount
```

### Recommendation

Emit an event for critical parameter changes.

## L09 - Dead Code Elimination

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L499,407,417,432,442,481,491,457,467,354,381

### Description

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

```
sendValue  
isContract  
functionStaticCall  
functionDelegateCall  
functionCallWithValue  
functionCall  
_verifyCallResult  
...
```

### Recommendation

Remove unused functions.

# 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>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>Context</b>	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
<b>Address</b>	Library			
	isContract	Internal		

	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	_verifyCallResult	Private		
<b>Ownable</b>	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
<b>IUniswapV2Factory</b>	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
<b>IUniswapV2Pair</b>	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-



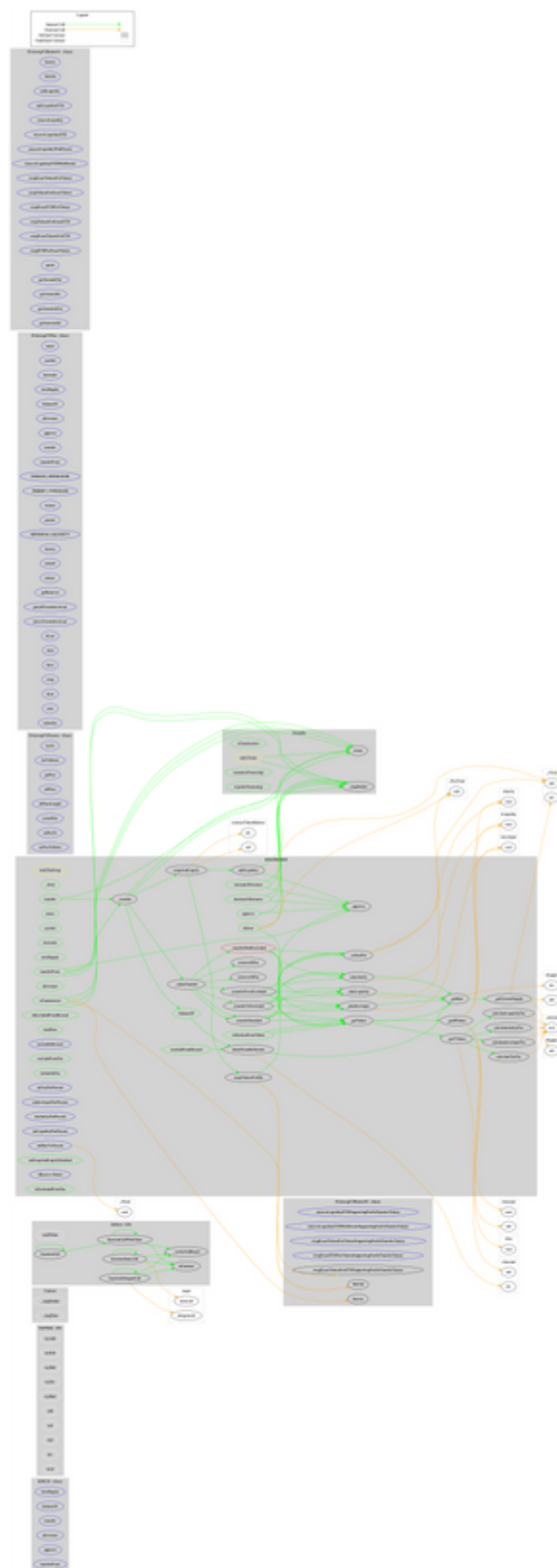
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
<b>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>SINCEREBAT</b>	Implementation	Context, IERC20, Ownable		
	_burn	Public	✓	onlyOwner
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	isExcludedFromReward	Public		-

	totalFees	Public		-
	deliver	Public	✓	-
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner
	includeInReward	External	✓	onlyOwner
	_transferBothExcluded	Private	✓	
	excludeFromFee	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	setTaxFeePercent	External	✓	onlyOwner
	setdeveloperFeePercent	External	✓	onlyOwner
	setcharityFeePercent	External	✓	onlyOwner
	setLiquidityFeePercent	External	✓	onlyOwner
	setMaxTxPercent	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	_takedeveloper	Private	✓	
	_takecharity	Private	✓	
	calculateTaxFee	Private		
	calculatedeveloperFee	Private		
	calculatecharityFee	Private		
	calculateLiquidityFee	Private		
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	isExcludedFromFee	Public		-
	_approve	Private	✓	
	_transfer	Private	✓	
	swapAndLiquify	Private	✓	lockTheSwap

	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	

# Contract Flow



## Domain Info

<b>Domain Name</b>	sincerebat.com
<b>Registry Domain ID</b>	2693120327_DOMAIN_COM-VRSN
<b>Creation Date</b>	2022-05-01T05:07:20Z
<b>Updated Date</b>	2022-05-01T05:07:22Z
<b>Registry Expiry Date</b>	2023-05-01T05:07:20Z
<b>Registrar WHOIS Server</b>	whois.publicdomainregistry.com
<b>Registrar URL</b>	www.publicdomainregistry.com
<b>Registrar</b>	PDR Ltd. d/b/a PublicDomainRegistry.com
<b>Registrar IANA ID</b>	303

The domain has been created 5 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

There are some functions that can be abused by the owner like stopping transactions and manipulating fees. The contract owner has the ability to decrease the total supply without limit. 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>