



Cyberscope

# Audit Report

## **Sleepe Token**

May 2022

Type           BEP20

Network       BSC

Address       0x4d04872e6c125149978750320a716561cba6873a

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>OCTD - Owner Contract Tokens Drain</b>	<b>6</b>
Description	6
Recommendation	6
<b>Contract Diagnostics</b>	<b>7</b>
<b>CR - Code Repetition</b>	<b>8</b>
Description	8
Recommendation	8
<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>L11 - Unnecessary Boolean equality</b>	<b>14</b>
<b>Description</b>	<b>14</b>
<b>Recommendation</b>	<b>14</b>
<b>L14 - Uninitialized Variables in Local Scope</b>	<b>15</b>
<b>Description</b>	<b>15</b>
<b>Recommendation</b>	<b>15</b>
<b>Contract Functions</b>	<b>16</b>
<b>Contract Flow</b>	<b>21</b>
<b>Domain Info</b>	<b>22</b>
<b>Summary</b>	<b>23</b>
<b>Disclaimer</b>	<b>24</b>
<b>About Cyberscope</b>	<b>25</b>

## Contract Review

<b>Contract Name</b>	Token
<b>Compiler Version</b>	v0.7.4+commit.3f05b770
<b>Optimization</b>	200 runs
<b>Licence</b>	GNU GPLv3
<b>Explorer</b>	<a href="https://bscscan.com/token/0x4d04872e6c125149978750320a716561cba6873a">https://bscscan.com/token/0x4d04872e6c125149978750320a716561cba6873a</a>
<b>Symbol</b>	SLE
<b>Decimals</b>	18
<b>Total Supply</b>	1,000,000,000
<b>Domain</b>	

## Source Files

<b>Filename</b>	<b>SHA256</b>
<b>contract.sol</b>	3e0cc6919869b298f8239d837f8981a724f9a385f783a36dc4c97cae9e0b431b

## Audit Updates

<b>Initial Audit</b>	9th May 2022
<b>Corrected</b>	15th May 2022

# 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	minor
Location	contract.sol#L612, 1436

### 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 `antiBotEnabled` to true and then whitelisting only himself.

```
function _transfer(
    address sender,
    address recipient,
    uint256 amount
) internal virtual override {
    if (
        !whitelistedBot[sender] && !whitelistedBot[recipient] &&
        antiBotEnabled
    ) {
        revert("Anti Bot");
    }
    if (
        swapWhitelisted && whitelistedPool[recipient] && !whitelistedBot[sender]
    ) {
        revert("Anti Bot");
    }
}
```

### 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.

## OCTD - Owner Contract Tokens Drain

Criticality	minor
Location	contract.sol#L983

### Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling the `transferToken` function.

```
function transferToken(  
    address coinAddress,  
    uint256 value,  
    address payable to  
) public onlyOwner {  
    if (coinAddress == address(0)) {  
        return to.transfer(value);  
    }  
    IERC20(coinAddress).transfer(to, value);  
}
```

### 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
●	CR	Code Repetition
●	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
●	L11	Unnecessary Boolean equality
●	L14	Uninitialized Variables in Local Scope



## CR - Code Repetition

**Criticality**

minor

**Location**

contract.sol#L1010, 1034

### Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily.

```
function changeFeeWallet(address feeWallet) public onlyOwner {  
    _feeWallet = feeWallet;  
}
```

```
function setFeeWallet(address newFeeWallet) public onlyOwner {  
    _feeWallet = newFeeWallet;  
}
```

### Recommendation

Remove `changeFeeWallet` or `setFeeWallet` function and keep the other code segment.

## L01 - Public Function could be Declared External

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L426,435,444,449,457,549,557,574,581,588,606,635,652,675,716,743,907,913,918,923,935,939,951,955,967,983,994,1006,1010,1014,1018,1030,1034,1038,1417

### Description

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

```
burn
setFeeTransfer
setFeeWallet
isExcludedFromPool
modifyWhiteListPool
changeFee
changeFeeWallet
isExcludedFromBot
modifyWhiteListBot
...
```

### Recommendation

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

## L02 - State Variables could be Declared Constant

**Criticality**

minor

**Location**

contract.sol#L1400

### Description

Constant state variables should be declared constant to save gas.

```
maxSupply
```

### Recommendation

Add the constant attribute to state variables that never change.

## L04 - Conformance to Solidity Naming Conventions

**Criticality**

minor

**Location**

contract.sol#L913,971,975,979,520,521,523,1114,1116,1146,1190

### 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
MINIMUM_LIQUIDITY
PERMIT_TYPEHASH
DOMAIN_SEPARATOR
_feeWallet
_feeTransfer
_numTokensSellToAddToLiquidity
_enable
_pmintable
...
```

### 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#L918

### 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.

```
_numTokensSellToAddToLiquidity = numTokensSellToAddToLiquidity
```

### Recommendation

Emit an event for critical parameter changes.

## L09 - Dead Code Elimination

**Criticality**

minor

**Location**

contract.sol#L874

### Description

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

```
_setupDecimals
```

### Recommendation

Remove unused functions.

## L11 - Unnecessary Boolean equality

**Criticality**

minor

**Location**

contract.sol#L606,675,1421

### Description

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

```
amount > _numTokensSellToAddToLiquidity && limitSell == true && recipient ==  
uniswapV2Pair  
sender != owner() && whitelistSender[_msgSender()] == false &&  
whitelistSender[sender] == false && whitelistReceiver[recipient] == false  
_msgSender() != owner() && whitelistSender[_msgSender()] == false &&  
whitelistReceiver[recipient] == false  
_msgSender() == owner() || whitelistSender[_msgSender()] == true ||  
whitelistReceiver[recipient] == true
```

### Recommendation

Remove the equality to the boolean constant.

## L14 - Uninitialized Variables in Local Scope

**Criticality**

minor

**Location**

contract.sol#L998,962,1022,1001,959,930,927,946,943,1025

### Description

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

```
index_scope_0  
index  
...
```

### Recommendation

All the local scoped variables should be initialized.



# Contract Functions

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

	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	geUnlockTime	Public		-
	lock	Public	✓	onlyOwner
	unlock	Public	✓	-
<b>ERC20</b>	Implementation	Context, IERC20, Ownable		
	_initialize	Internal	✓	
	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	✓	
	_setupDecimals	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	modifyWhiteListSender	Public	✓	onlyOwner
	isExcludedFromFee	Public		-
	modifyWhiteListReceiver	Public	✓	onlyOwner
	isExcludedToFee	Public		-
	setAntiBot	External	✓	onlyOwner
	setSwapWhiteList	External	✓	onlyOwner
	transferToken	Public	✓	onlyOwner
	modifyWhiteListBot	Public	✓	onlyOwner

	isExcludedFromBot	Public		-
	changeFeeWallet	Public	✓	onlyOwner
	changeFee	Public	✓	onlyOwner
	modifyWhiteListPool	Public	✓	onlyOwner
	isExcludedFromPool	Public		-
<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		-
	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>Token</b>	Implementation	ERC20		
	<Constructor>	Public	✓	-
	burn	Public	✓	-
	_transfer	Internal	✓	

# Contract Flow



## Domain Info

<b>Domain Name</b>	sleepe.io
<b>Registry Domain ID</b>	18529b9b6ec341fba16e599c7b315b95-DONUTS
<b>Creation Date</b>	2022-04-08T11:43:06Z
<b>Updated Date</b>	2022-04-13T11:43:08Z
<b>Registry Expiry Date</b>	2023-04-08T11:43:06Z
<b>Registrar WHOIS Server</b>	whois.discount-domain.com
<b>Registrar URL</b>	<a href="http://www.onamae.com">http://www.onamae.com</a>
<b>Registrar</b>	GMO Internet, Inc. d/b/a Onamae.com
<b>Registrar IANA ID</b>	49

The domain has been created about 1 month before the creation of the audit. It will expire in 11 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 transferring tokens to the team's wallet. 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.

Cyberscope 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>