



# Cyberscope

## Audit Report

# Espoir

April 2022

Type       BEP20

Network    BSC

Address    0xb04a86ee1e220b24dD33624821db4667Fefd7CaD

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>ULTW - Unlimited Liquidity to Team Wallet</b>	<b>5</b>
Description	5
Recommendation	5
<b>Contract Diagnostics</b>	<b>6</b>
<b>FSA - Fixed Swap Address</b>	<b>7</b>
Description	7
Recommendation	7
<b>L01 - Public Function could be Declared External</b>	<b>8</b>
Description	8
Recommendation	8
<b>L02 - State Variables could be Declared Constant</b>	<b>9</b>
Description	9
Recommendation	9
<b>L04 - Conformance to Solidity Naming Conventions</b>	<b>10</b>
Description	10
Recommendation	10
<b>L05 - Unused State Variable</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>L13 - Divide before Multiply Operation</b>	<b>14</b>
<b>Description</b>	<b>14</b>
<b>Recommendation</b>	<b>14</b>
<b>Contract Functions</b>	<b>15</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>	EspoirToken
<b>Compiler Version</b>	v0.8.13+commit.abaa5c0e
<b>Optimization</b>	200 runs
<b>Licence</b>	Unlicense
<b>Explorer</b>	<a href="https://bscscan.com/token/0xb04a86ee1e220b24dD33624821db4667Fefd7CaD">https://bscscan.com/token/0xb04a86ee1e220b24dD33624821db4667Fefd7CaD</a>
<b>Symbol</b>	ESP
<b>Decimals</b>	18
<b>Total Supply</b>	10,000,000
<b>Domain</b>	espoirclub.org

## Source Files

<b>Filename</b>	<b>SHA256</b>
<b>contract.sol</b>	8a84d4f31a036ac4396f65311802a1d9c8218ee15296d74f2580fe36a1ef8cfd

## Audit Updates

<b>Initial Audit</b>	14th April 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

## ULTW - Unlimited Liquidity to Team Wallet

**Criticality**

minor

**Location**

contract.sol#L626, 673 854,

### 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 setting a high fee to the `_minimumTokensBeforeSwap` variable.

```
function setNumTokensSellToAddToLiquidity(uint256 __minimumTokensBeforeSwap)
external onlyOwner()
{ _minimumTokensBeforeSwap = __minimumTokensBeforeSwap; }
```

```
contractTokenBalance = _minimumTokensBeforeSwap;
swapAndSendBnbToWallets(contractTokenBalance);
```

```
function swapAndSendBnbToWallets(uint256 contractTokenBalance) private
lockTheSwap
{
    swapTokensForEth(contractTokenBalance);
```

### Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may violate 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

Severity	Code	Description
●	FSA	Fixed Swap Address
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L05	Unused State Variable
●	L07	Missing Events Arithmetic
●	L09	Dead Code Elimination
●	L13	Divide before Multiply Operation

## FSA - Fixed Swap Address

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L476

### Description

The swap address is assigned once in the constructor and it can not be changed. The decentralized swaps sometimes create a new swap version or abandon the current. A contract that cannot change the swap address may not be able to catch-up the upgrade

```
IUniswapV2Router02 _uniswapV2Router =  
IUniswapV2Router02(0x10ED43C718714eb63d5aA57B78B54704E256024E);  
    uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory())  
        .createPair(address(this), _uniswapV2Router.WETH());  
    uniswapV2Router = _uniswapV2Router;
```

### Recommendation

It could be better to allow the swap address mutation in case of future swap updates.



## L01 - Public Function could be Declared External

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L159,164,170,174,178,185,491,495,499,503 and 11 more

### Description

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

```
setSwapAndLiquifyEnabled
excludeFromReward
_minimumTokensBeforeSwapAmount
totalFees
isExcludedFromReward
decreaseAllowance
increaseAllowance
transferFrom
approve
...
```

### Recommendation

Use the external attribute for functions never called from the contract

## L02 - State Variables could be Declared Constant

**Criticality**

minor

**Location**

contract.sol#L428,426,427,424,422

### Description

Constant state variables should be declared constant to save gas.

```
_tTotal  
_tFeeTotal  
_symbol  
_name  
_decimals
```

### Recommendation

Add the constant attribute to state variables that never change.

## L04 - Conformance to Solidity Naming Conventions

**Criticality**

minor

**Location**

contract.sol#L229,230,246,267,830,552,759,812,847,854 and 7 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.

```
_walletHoldingMaxLimit  
_maxTxAmount  
_marketingFee  
_developerFee  
_enabled  
_developerAddress  
_marketingAddress  
__minimumTokensBeforeSwap  
_mount  
...
```

### 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#L415

### Description

There are segments that contain unused state variables.

```
_isExcludedFromWhale
```

### Recommendation

Remove unused state variables.

## L07 - Missing Events Arithmetic

**Criticality**

minor

**Location**

contract.sol#L847,854

### 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.

```
_minimumTokensBeforeSwap = __minimumTokensBeforeSwap  
_maxTxAmount = _mount
```

### Recommendation

Emit an event for critical parameter changes.

## L09 - Dead Code Elimination

**Criticality**

minor

**Location**

contract.sol#L117,100,104,108,112,80,91,677

### Description

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

```
addLiquidity  
sendValue  
isContract  
functionCallWithValue  
functionCall  
_functionCallWithValue
```

### Recommendation

Remove unused functions.

## L13 - Divide before Multiply Operation

**Criticality**

minor

**Location**

contract.sol#L404

### Description

Performing divisions before multiplications may cause lose of prediction.

```
_walletHoldingMaxLimit = _tTotal.div(100).mul(2)  
_maxTxAmount = _tTotal.div(100).mul(1)
```

### Recommendation

The multiplications should be prior to the divisions.

# Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
<b>Context</b>	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
<b>IBEP20</b>	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
<b>SafeMath</b>	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
<b>Address</b>	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	



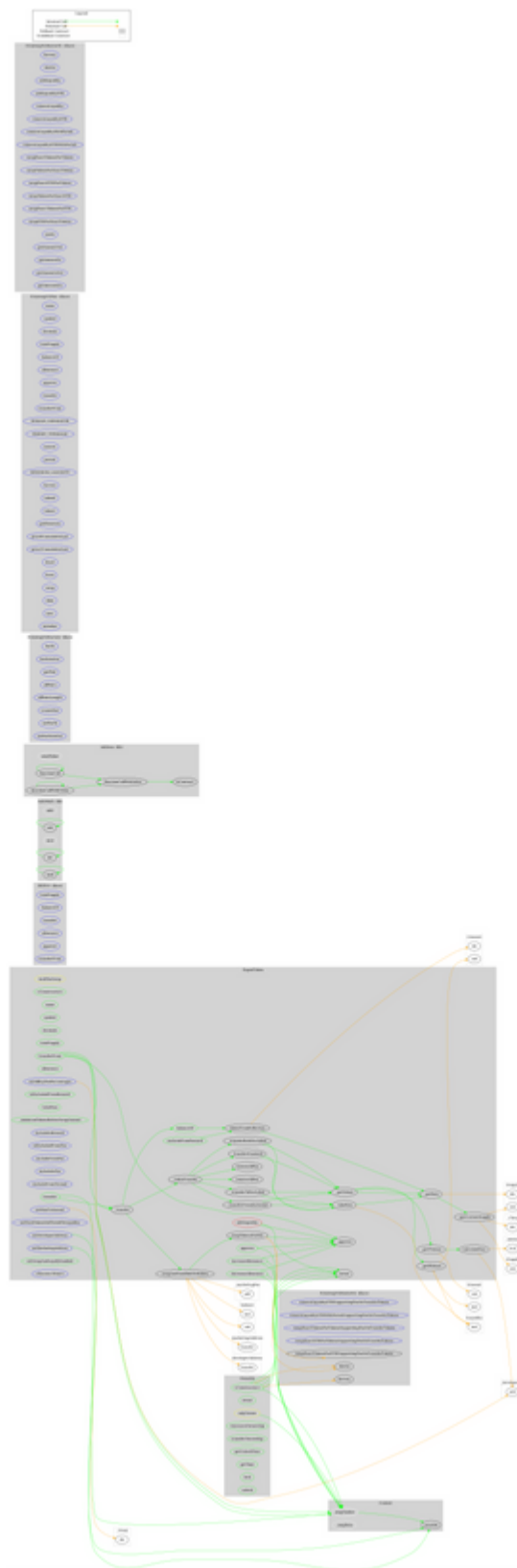
	_functionCallWithValue	Private	✓	
<b>Ownable</b>	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	getUnlockTime	Public		-
	getTime	Public		-
	lock	Public	✓	onlyOwner
	unlock	Public	✓	-
<b>IUniswapV2Factory</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>EspoirToken</b>	Implementation	Context, IBEP20, Ownable		
	<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		-
	_minimumTokensBeforeSwapAmount	Public		-
	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner
	includeInReward	External	✓	onlyOwner
	isExcludedFromFee	External		-
	excludeFromFee	External	✓	onlyOwner

	includeInFee	External	✓	onlyOwner
	_approve	Private	✓	
	_transfer	Private	✓	
	swapAndSendBnbToWallets	Private	✓	lockTheSwap
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	
	_transferBothExcluded	Private	✓	
	excludeFromTxLimit	External	✓	onlyOwner
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_takeFees	Private	✓	
	calculateFees	Private		
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	setAllBuyFeePercentages	External	✓	onlyOwner
	setMaxTxAmount	External	✓	onlyOwner
	setNumTokensSellToAddToLiquidity	External	✓	onlyOwner
	setMarketingAddress	External	✓	onlyOwner
	setDeveloperAddress	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	<Receive Ether>	External	Payable	-

# Contract Flow



## Domain Info

<b>Domain Name</b>	espoirclub.org
<b>Registry Domain ID</b>	D402200000019408750-LROR
<b>Creation Date</b>	2022-03-24T13:58:08Z
<b>Updated Date</b>	2022-03-27T05:40:19Z
<b>Registry Expiry Date</b>	2023-03-24T13:58:08Z
<b>Registrar WHOIS Server</b>	whois.namecheap.com
<b>Registrar URL</b>	<a href="http://www.namecheap.com">http://www.namecheap.com</a>
<b>Registrar</b>	NameCheap, Inc.
<b>Registrar IANA ID</b>	1068

The domain has been created 21 days 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

Espoir is an interesting project that has a friendly and growing community. 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 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.

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>