



# Cyberscope

## Audit Report

# USDT DeFi

August 2022

Type       BEP20

Network    BSC

Address     0x93d00a0fbd822f249458c5f973bee6c5248670a9

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 - Transfers Liquidity to Team Wallet</b>	<b>5</b>
Description	5
Recommendation	5
<b>Contract Diagnostics</b>	<b>6</b>
<b>STC - Succeeded Transfer Check</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>L03 - Redundant Statements</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>L13 - Divide before Multiply Operation</b>	<b>13</b>
<b>Description</b>	<b>13</b>
<b>Recommendation</b>	<b>13</b>
<b>L14 - Uninitialized Variables in Local Scope</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>18</b>
<b>Domain Info</b>	<b>19</b>
<b>Summary</b>	<b>20</b>
<b>Disclaimer</b>	<b>21</b>
<b>About Cyberscope</b>	<b>22</b>

## Contract Review

<b>Contract Name</b>	USDTDeFi
<b>Compiler Version</b>	v0.8.8+commit.dddeac2f
<b>Optimization</b>	200 runs
<b>Licence</b>	Unlicense
<b>Explorer</b>	<a href="https://bscscan.com/token/0x93D00A0fBd822F249458C5f973bee6C5248670a9">https://bscscan.com/token/0x93D00A0fBd822F249458C5f973bee6C5248670a9</a>
<b>Symbol</b>	USDTd
<b>Decimals</b>	18
<b>Total Supply</b>	65,133,996,314
<b>Domain</b>	usdt-defi.io

## Source Files

<b>Filename</b>	<b>SHA256</b>
<b>contract.sol</b>	31be20d6dd3b29015fcf90a36a85fefdd47255b0b16c42bf7fe7e87842267130

## Audit Updates

<b>Initial Audit</b>	23rd August 2022
<b>Corrected</b>	

# 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

<b>Criticality</b>	minor / informative
<b>Location</b>	contract.sol#L725
<b>Status</b>	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` method.

```
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
●	STC	Succeeded Transfer Check	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

## STC - Succeeded Transfer Check

<b>Criticality</b>	minor / informative
<b>Location</b>	contract.sol#L726
<b>Status</b>	Unresolved

### Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
function rescueBSC20(address tokenAdd, uint256 amount) external onlyOwner {  
    require(tokenAdd != address(this), "Owner can't claim contract's balance of its own  
tokens");  
    IBEP20(tokenAdd).transfer(owner(), amount);  
}
```

### Recommendation

The contract should check if the result of the transfer methods is successful.



## L01 - Public Function could be Declared External

<b>Criticality</b>	minor / informative
<b>Location</b>	contract.sol#L92,165,338,135,232,84,342,148,209,116,183
<b>Status</b>	Unresolved

### Description

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

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

### Recommendation

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

## L02 - State Variables could be Declared Constant

<b>Criticality</b>	minor / informative
<b>Location</b>	contract.sol#L403
<b>Status</b>	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

<b>Criticality</b>	minor / informative
<b>Location</b>	contract.sol#L7
<b>Status</b>	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

<b>Criticality</b>	minor / informative
<b>Location</b>	contract.sol#L661,669,681,668,401,670,660,662,406,688,361,58,654,705,60,577,405
<b>Status</b>	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.

```
_Treasuryvault  
EnableTrading  
SetSellTaxes  
genesis_block  
_liquidity  
SetBuyTaxes  
deadWallet  
_deadline  
WETH  
...
```

### Recommendation

Follow the Solidity naming convention.

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

## L07 - Missing Events Arithmetic

<b>Criticality</b>	minor / informative
<b>Location</b>	contract.sol#L715,654,699,688
<b>Status</b>	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.

```
maxWalletLimit = maxWallet * 10 ** decimals()
tokenLiquidityThreshold = new_amount * 10 ** decimals()
coolDownTime = time * 1
deadline = _deadline
```

### Recommendation

Emit an event for critical parameter changes.

## L13 - Divide before Multiply Operation

<b>Criticality</b>	minor / informative
<b>Location</b>	contract.sol#L577
<b>Status</b>	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

<b>Criticality</b>	minor / informative
<b>Location</b>	contract.sol#L528,531,529
<b>Status</b>	Unresolved

### Description

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

```
feeswap  
currentTaxes  
feesum
```

### Recommendation

All the local scoped variables should be initialized.

# Contract Functions

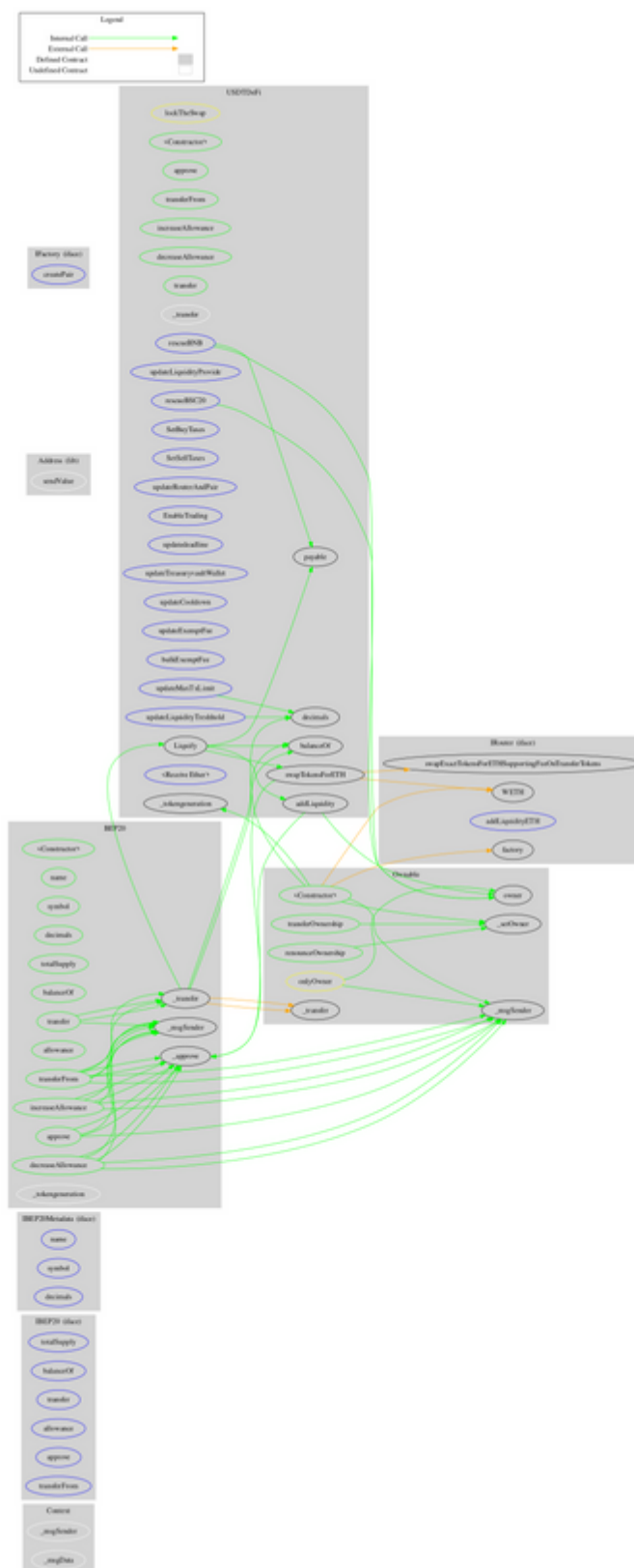
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
<b>Context</b>	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
<b>IBEP20</b>	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom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



## Domain Info

<b>Domain Name</b>	usdt-defi.io
<b>Registry Domain ID</b>	d2bc846556ab45d184c9d5073f605fba-DONUTS
<b>Creation Date</b>	2022-06-22T02:04:23Z
<b>Updated Date</b>	2022-08-20T15:53:48Z
<b>Registry Expiry Date</b>	2023-06-22T02:04:23Z
<b>Registrar WHOIS Server</b>	whois.namecheap.com
<b>Registrar URL</b>	<a href="https://www.namecheap.com/">https://www.namecheap.com/</a>
<b>Registrar</b>	NameCheap, Inc.
<b>Registrar IANA ID</b>	1068

The domain was created 2 months before the creation of the audit. It will expire in 10 months.

There is no public billing information, the creator is protected by the privacy settings.

## Summary

The Smart Contract analysis reported one minor severity issue. The contract owner has the authority to transfer funds to the team's wallet. 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 25% 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>