



# Cyberscope

## Audit Report

# OOZE

April 2022

Type       BEP20

Network     BSC

Address     0x6d850247f9f99a85217f9da5bcb5b36ca5350b8a

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
Corrected 19th April 2022	6
<b>ELFM - Exceed Limit Fees Manipulation</b>	<b>7</b>
Description	7
Recommendation	7
Corrected 19th April 2022	8
<b>BC - Blacklisted Contracts</b>	<b>9</b>
Description	9
Recommendation	9
Corrected 19th April 2022	9
<b>Contract Diagnostics</b>	<b>10</b>
<b>L01 - Public Function could be Declared External</b>	<b>11</b>
Description	11
Recommendation	11
<b>L02 - State Variables could be Declared Constant</b>	<b>12</b>
Description	12
Recommendation	12
<b>L04 - Conformance to Solidity Naming Conventions</b>	<b>13</b>
Description	13

<b>Recommendation</b>	<b>13</b>
<b>L05 - Unused State Variable</b>	<b>14</b>
<b>Description</b>	<b>14</b>
<b>Recommendation</b>	<b>14</b>
<b>L13 - Divide before Multiply Operation</b>	<b>15</b>
<b>Description</b>	<b>15</b>
<b>Recommendation</b>	<b>15</b>
<b>L14 - Uninitialized Variables in Local Scope</b>	<b>16</b>
<b>Description</b>	<b>16</b>
<b>Recommendation</b>	<b>16</b>
<b>Contract Functions</b>	<b>17</b>
<b>Contract Flow</b>	<b>22</b>
<b>Domain Info</b>	<b>23</b>
<b>Summary</b>	<b>24</b>
<b>Corrected 19th April 2022</b>	<b>24</b>
<b>Disclaimer</b>	<b>25</b>
<b>About Cyberscope</b>	<b>26</b>

## Contract Review

<b>Contract Name</b>	CoinToken
<b>Compiler Version</b>	v0.8.10+commit.fc410830
<b>Optimization</b>	200 runs
<b>Licence</b>	None
<b>Explorer</b>	<a href="https://bscscan.com/token/0x6d850247f9f99a85217f9da5bcb5b36ca5350b8a">https://bscscan.com/token/0x6d850247f9f99a85217f9da5bcb5b36ca5350b8a</a>
<b>Symbol</b>	OOZE
<b>Decimals</b>	18
<b>Total Supply</b>	100,000,000
<b>Domain</b>	ooze.site

## Source Files

<b>Filename</b>	<b>SHA256</b>
<b>contract.sol</b>	07bf1b7844f274d1a10ddee98f577cf68a2ddd06b515b7f1fd68066d11f3b0b1

## Audit Updates

<b>Initial Audit</b>	18th April 2022
<b>Corrected</b>	19th April 2022

# Contract Analysis

● Critical   ● Medium   ● Minor   ● Pass

Severity	Code	Description	Status
●	ST	Contract Owner is not able to stop or pause transactions	Resolved
●	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%)	Resolved
●	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	Resolved

## ST - Stop Transactions

Criticality	critical
Location	contract.sol#L1001,1088

### Description

The contract owner has the authority to stop transactions for all users. The owner may take advantage of it by setting the `paused` to true.

```
require(!paused(), "CoinToken: token transfer while paused");
```

The contract owner has the authority to stop sales for all users. The owner may take advantage of it by setting the `sellTaxes` to very high values. This will result in a **HONEYPOT** behaviour.

```
    } else if(to == address(uniswapV2Pair)) {  
        tax += baseUnit * sellTaxes["marketing"];  
        tax += baseUnit * sellTaxes["dev"];  
        tax += baseUnit * sellTaxes["liquidity"];  
        tax += baseUnit * sellTaxes["charity"];  
  
        if(tax > 0) {  
            _transfer(from, address(this), tax);  
        }  
    }
```

### 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.

Corrected 19th April 2022

The team has renounced ownership and the threats have been resolved.

## ELFM - Exceed Limit Fees Manipulation

<b>Criticality</b>	critical
<b>Location</b>	contract.sol#L1078,1088

### 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 `setBuyTax` and the `setSellTax` functions with high percentage values.

```
function setBuyTax(uint256 dev, uint256 marketing, uint256 liquidity,
uint256 charity) public onlyOwner {
    buyTaxes["dev"] = dev;
    buyTaxes["marketing"] = marketing;
    buyTaxes["liquidity"] = liquidity;
    buyTaxes["charity"] = charity;
}

/**
 * @dev Sets tax for sells.
 */
function setSellTax(uint256 dev, uint256 marketing, uint256 liquidity,
uint256 charity) public onlyOwner {

    sellTaxes["dev"] = dev;
    sellTaxes["marketing"] = marketing;
    sellTaxes["liquidity"] = liquidity;
    sellTaxes["charity"] = charity;
}
```

### 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.



Corrected 19th April 2022

The team has renounced ownership and the threats have been resolved.

## BC - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L1046

### Description

The contract owner has the authority to stop contracts from transactions. The owner may take advantage of it by calling the `enableBlacklist` function.

```
require(!isBlacklisted(msg.sender), "CoinToken: sender blacklisted");  
require(!isBlacklisted(recipient), "CoinToken: recipient blacklisted");  
require(!isBlacklisted(tx.origin), "CoinToken: sender blacklisted");
```

### 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.

### Corrected 19th April 2022

The team has renounced ownership and the threats have been resolved.

# 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
●	L05	Unused State Variable
●	L13	Divide before Multiply Operation
●	L14	Uninitialized Variables in Local Scope

## L01 - Public Function could be Declared External

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L177,185,202,209,216,228,236,247,265,293,312,518,526,1016,1023,1031,1039,1046,1054,1070,1108,1116

### Description

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

```
disableTax  
enableTax  
removeExclude  
disableBlacklist  
enableBlacklist  
burn  
unpause  
pause  
triggerTax  
...
```

### Recommendation

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

## L02 - State Variables could be Declared Constant

**Criticality**

minor

**Location**

contract.sol#L838,843,848,831,835,840,845,837,842,847,836,841,846,833

### Description

Constant state variables should be declared constant to save gas.

```
swapThreshold
marketingTaxWallet
marketingTaxSell
marketingTaxBuy
liquidityTaxWallet
liquidityTaxSell
liquidityTaxBuy
devTaxWallet
devTaxSell
...
```

### Recommendation

Add the constant attribute to state variables that never change.

## L04 - Conformance to Solidity Naming Conventions

**Criticality**

minor

**Location**

contract.sol#L638,639,656,692

### 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
```

### 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#L835,836,837,838,840,841,842,843,845,846,847,848

### Description

There are segments that contain unused state variables.

```
charityTaxWallet  
liquidityTaxWallet  
marketingTaxWallet  
devTaxWallet  
charityTaxSell  
liquidityTaxSell  
marketingTaxSell  
devTaxSell  
charityTaxBuy  
...
```

### Recommendation

Remove unused state variables.

## L13 - Divide before Multiply Operation

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L890

### Description

Performing divisions before multiplications may cause lose of prediction.

```
charityETH = (ethGained * ((charityTokens * 10 ** 18) / taxSum)) / 10 ** 18
devETH = (ethGained * ((devTokens * 10 ** 18) / taxSum)) / 10 ** 18
marketingETH = (ethGained * ((marketingTokens * 10 ** 18) / taxSum)) / 10 ** 18
liquidityETH = (ethGained * ((liquidityTokens / 2 * 10 ** 18) / taxSum)) / 10
** 18
baseUnit = amount / denominator
...
```

### Recommendation

The multiplications should be prior to the divisions.



## L14 - Uninitialized Variables in Local Scope

**Criticality**

minor

**Location**

contract.sol#L896

### Description

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

```
tax
```

### Recommendation

All the local scoped variables should be initialized.

# Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-

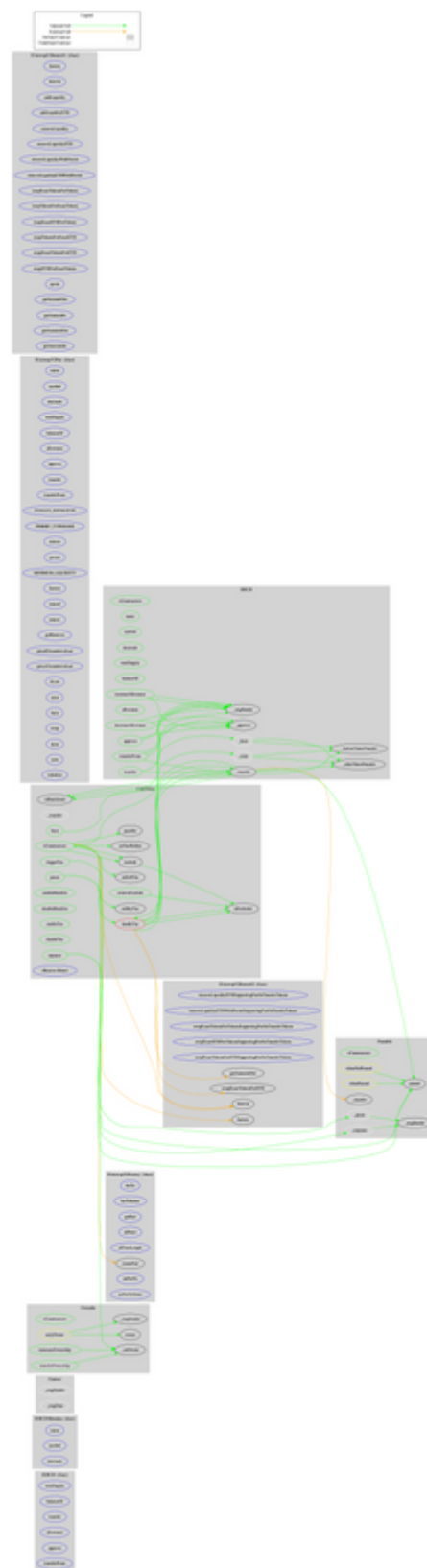
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
<b>Ownable</b>	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Internal	✓	
<b>Pausable</b>	Implementation	Context		
	<Constructor>	Public	✓	-
	paused	Public		-
	_pause	Internal	✓	whenNotPaused
	_unpause	Internal	✓	whenPaused
<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>IUniswapV2Factory</b>	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	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>CoinToken</b>	Implementation	ERC20, Ownable, Pausable		
	<Constructor>	Public	Payable	ERC20
	handleTax	Private	✓	
	_transfer	Internal	✓	
	triggerTax	Public	✓	onlyOwner
	pause	Public	✓	onlyOwner
	unpause	Public	✓	onlyOwner

	burn	Public	✓	onlyOwner
	enableBlacklist	Public	✓	onlyOwner
	disableBlacklist	Public	✓	onlyOwner
	exclude	Public	✓	onlyOwner
	removeExclude	Public	✓	onlyOwner
	setBuyTax	Public	✓	onlyOwner
	setSellTax	Public	✓	onlyOwner
	setTaxWallets	Public	✓	onlyOwner
	enableTax	Public	✓	onlyOwner
	disableTax	Public	✓	onlyOwner
	isBlacklisted	Public		-
	isExcluded	Public		-
	<Receive Ether>	External	Payable	-

# Contract Flow



## Domain Info

<b>Domain Name</b>	ooze.site
<b>Registry Domain ID</b>	D285549924-CNIC
<b>Creation Date</b>	2022-03-29T12:14:14+00:00
<b>Updated Date</b>	2022-03-29T12:14:18+00:00
<b>Registry Expiry Date</b>	2023-03-29T23:59:59+00:00
<b>Registrar WHOIS Server</b>	whois.PublicDomainRegistry.com
<b>Registrar URL</b>	<a href="https://publicdomainregistry.com">https://publicdomainregistry.com</a>
<b>Registrar</b>	PDR Ltd. d/b/a PublicDomainRegistry.com
<b>Registrar IANA ID</b>	303

The domain has been created 20 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

OOZE is an interesting project that has a friendly and growing community. There are some functions that can be abused by the owner, like manipulating fees up to 100%, blacklisting wallets from trading and stopping transactions. The contract can be converted into a **honeypot** and prevent users from selling if the owner abuses the admin functions. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

Corrected 19th April 2022

The team has renounced ownership and the threats have been resolved.

## 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>