Cyberscope

Audit Report

Obolo

September 2022

Type	BEP20
Network	BSC
Address	0x7bbf116D9d283d24aB9E00502443E800E0349f0D
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	OboToken
Compiler Version	v0.8.7+commit.e28d00a7
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x7bbf116D9d283d24aB9E00502443E800E0349f0D
Symbol	OBO
Decimals	9
Total Supply	100,000,000
Domain	https://obolo.finance

Source Files

Filename	**SHA256**
contract.sol	75052c71c0aee260f846adc768a01f96fc1600ccfee0856d0e7bb0d8e3413f8d

Audit Updates

Initial Audit	30th September 2022
Corrected	

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	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Contract Diagnostics

🔴 Critical 🟠 Medium ● Minor / Informative

Severity	Code	Description	Status
🔴	US	Untrusted Source	Unresolved
●	BLC	Business Logic Concern	Unresolved
●	CO	Code Optimization	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
●	L05	Unused State Variable	Unresolved
●	L08	Tautology or Contradiction	Unresolved
●	L09	Dead Code Elimination	Unresolved
●	L15	Local Scope Variable Shadowing	Unresolved

US - Untrusted Source

Criticality	critical
Location	contract.sol#L882,883,887
Status	Unresolved

Description

The contract uses an external contract in order to determine the transaction's flow. The external contract is untrusted. As a result it may produce security issues and harm the transactions.

```
if (
     !isInPresale &&
    transfer_to_profile_contract_address != address(0) &&
     has_profile_transfers_enabled &&
ProfileTransferRouter(transfer_to_profile_contract_address).addressHasProfile(recipient)
&& ProfileTransferRouter(transfer_to_profile_contract_address).addressHasProfile(sender)
       ) {
           if (has_external_profile_transfers) {
               ProfileTransferRouter(transfer_to_profile_contract_address).transferToProfile(
                   sender,
                   recipient,
                   amount
               );
```

Recommendation

The contract should use a trusted external source. A trusted source could be either a commonly recognized or an audited contract. The pointing addresses should not be able to change after the initialization.

If the contract logic required an untrusted source, then the untrusted source should be surrounded by a try-catch statement and it should not allow it to effect the contract’s flow.

BLC - Business Logic Concern

Criticality	minor / informative
Location	contract.sol#L1768,1158
Status	Unresolved

Description

The burn functionality emits a burn event from the contract address to the burn address. According to the contract flow, the transfer is performed from the sender to the burn address.

```
function _burn(
        uint256 amount,
        uint256 fee,
        uint256 index
    ) private {
        uint256 tBurn = amount.mul(fee).div(FEES_DIVISOR);

        _burnTokens(address(this), tBurn);
//
function _burnTokens(address sender, uint256 tBurn) internal {
        _balances[burnAddress] = _balances[burnAddress].add(tBurn);

        /**
         * @dev Emit the event so that the burn address balance is updated (on bscscan)
         */
        emit Transfer(sender, burnAddress, tBurn);
}
```

The ExternalToETH fee type is the same as the regular fee type. The ExternalToETH fee type merely delegates the method to the regular fee type.

```
            if (name == FeeType.Burn) {
                _burn(amount, value, index);
            } else if (name == FeeType.ExternalToETH) {
                _takeFeeToETH(amount, value, recipient, index);
            } else {
                _takeFee(amount, value, recipient, index);
            }
//
function _takeFeeToETH(
        uint256 amount,
        uint256 fee,
        address recipient,
        uint256 index
    ) private {
        _takeFee(amount, fee, recipient, index);
    }
```

Recommendation

The burn event should emit the sender's address instead of the contract's address.

The ExternalToETH case could be removed from the contract.

CO - Code Optimization

Criticality	minor / informative
Location	contract.sol#L913,111
Status	Unresolved

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations.

The method ‘_getSumOfFees’ is getting called even if the takeFee is not enabled. This produces an unnecessary function call.

```
        uint256 sumOfFees = _getSumOfFees(sender, amount);
        if (!takeFee) {
            sumOfFees = 0;
        }
```

The method ‘_getSumOfFees’ accepts two arguments that are not used.

```
function _getSumOfFees(address sender, uint256 amount)
        internal
        view
        override
        returns (uint256)
    {
        return sumOfFees;
    }
```

Recommendation

The method ‘_getSumOfFees’ should be called only when the takeFee is enabled. The two parameters could be eliminated since they are not used by the method.

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L129,811,96,100,824,114,108,105
Status	Unresolved

Description

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

```
manager
increaseAllowance
renounceOwnership
transferOwnership
decreaseAllowance
unlock
lock
getUnlockTime
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contract.sol#L217,501,639,504,222,640,499,214
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
_mainnetRouterV2Address
projectAddress
_name
burnAddress
_testnetRouterAddress
_symbol
priceAddress
_mainnetRouterV1Address
```

Recommendation

Add the constant attribute to state variables that never change.

L03 - Redundant Statements

Criticality	minor / informative
Location	contract.sol#L23
Status	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

Criticality	minor / informative
Location	contract.sol#L642,733,145,645,488,635,226,461,165,742,738,633,647,646,784,225
Status	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.

```
_isExcludedFromFee
address_to_exclude
WETH
transfer_to_profile_contract_address
numberOfTokensToSwapToLiquidity
_allowances
_pair
maxTransactionAmount
address_to_check
...
```

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 / informative
Location	contract.sol#L639,222,449,211,214,640
Status	Unresolved

Description

There are segments that contain unused state variables.

```
_name
_testnetRouterAddress
MAX
_env
_mainnetRouterV1Address
_symbol
```

Recommendation

Remove unused state variables.

L08 - Tautology or Contradiction

Criticality	minor / informative
Location	contract.sol#L572
Status	Unresolved

Description

Detects expressions that are tautologies or contradictions. For instance, an uint variable will always be greater than or equal to zero.

```
require(bool,string)(index >= 0 && index < fees.length,FeesSettings._getFeeStruct: Fee index out of bounds)
```

Recommendation

Fix the incorrect comparison by changing the value type or the comparison.

L09 - Dead Code Elimination

Criticality	minor / informative
Location	contract.sol#L551,39,45,1109,350,63,1203,60,46,307,1015,52,1131,68,268,38,44,330,55,43
Status	Unresolved

Description

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

```
_addFees
sendValue
functionCallWithValue
_isV2Pair
_addLiquidity
functionDelegateCall
_approveDelegate
_swapAndLiquify
_burn
...
```

Recommendation

Remove unused functions.

L15 - Local Scope Variable Shadowing

Criticality	minor / informative
Location	contract.sol#L929,1150,909
Status	Unresolved

Description

The are variables that are defined in the local scope containing the same name from an upper scope.

```
sumOfFees
name
```

Recommendation

The local variables should have different names from the upper scoped variables.

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		
SafeMath	Library			
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	

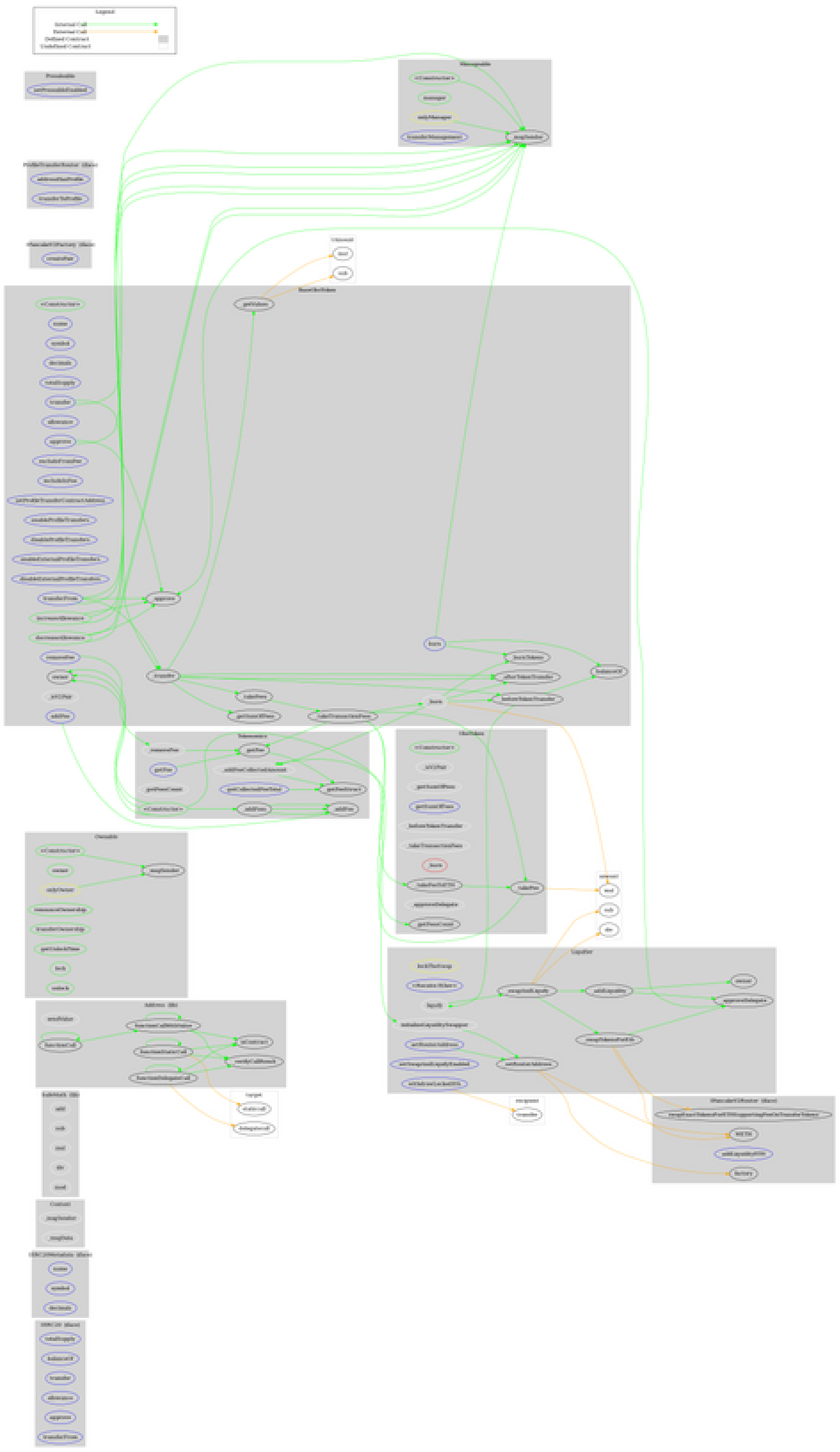
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	_verifyCallResult	Private		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	getUnlockTime	Public		-
	lock	Public	✓	onlyOwner
	unlock	Public	✓	-
Manageable	Implementation	Context		
	<Constructor>	Public	✓	-
	manager	Public		-
	transferManagement	External	✓	onlyManager
IPancakeV2Factory	Interface			
	createPair	External	✓	-
IPancakeV2Router	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-

ProfileTransfer Router	Interface			
	addressHasProfile	External		-
	transferToProfile	External	✓	-
Presaleable	Implementation	Manageable		
	setPreseableEnabled	External	✓	onlyManager
Liquifier	Implementation	Ownable, Manageable		
	<Receive Ether>	External	Payable	-
	initializeLiquiditySwapper	Internal	✓	
	liquify	Internal	✓	
	_setRouterAddress	Private	✓	
	_swapAndLiquify	Private	✓	lockTheSwap
	_swapTokensForEth	Private	✓	
	_addLiquidity	Private	✓	
	setRouterAddress	External	✓	onlyManager
	setSwapAndLiquifyEnabled	External	✓	onlyManager
	withdrawLockedEth	External	✓	onlyManager
	_approveDelegate	Internal	✓	
Tokenomics	Implementation			
	<Constructor>	Public	✓	-
	_addFee	Internal	✓	
	_removeFee	Internal	✓	
	_addFees	Private	✓	
	_getFeesCount	Internal		
	_getFeeStruct	Private		
	getFee	External		-
	_getFee	Internal		
	_addFeeCollectedAmount	Internal	✓	
	getCollectedFeeTotal	External		-

BaseOboToken	Implementation	IERC20, IERC20Metadata, Ownable, Presaleable, Tokenomics		
	<Constructor>	Public	✓	-
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	Public		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	excludeFromFee	External	✓	onlyOwner
	includeInFee	External	✓	onlyOwner
	setProfileTransferContractAddress	External	✓	onlyOwner
	enableProfileTransfers	External	✓	onlyOwner
	disableProfileTransfers	External	✓	onlyOwner
	enableExternalProfileTransfers	External	✓	onlyOwner
	disableExternalProfileTransfers	External	✓	onlyOwner
	transferFrom	External	✓	-
	addFee	External	✓	onlyOwner
	removeFee	External	✓	onlyOwner
	_getSumOfFees	Internal		
	_takeTransactionFees	Internal	✓	
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_takeFees	Private	✓	
	_getValues	Internal		
	burn	External	✓	-
	_burnTokens	Internal	✓	
	_isV2Pair	Internal		
	_burn	Internal	✓	

	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
OboToken	Implementation	BaseOboToken, Liquifier		
	<Constructor>	Public	✓	-
	_isV2Pair	Internal		
	_getSumOfFees	Internal		
	getSumOfFees	External		-
	_beforeTokenTransfer	Internal	✓	
	_takeTransactionFees	Internal	✓	
	_burn	Private	✓	
	_takeFee	Private	✓	
	_takeFeeToETH	Private	✓	
	_approveDelegate	Internal	✓	

Contract Flow



Domain Info

Domain Name	obolo.finance
Registry Domain ID	41999ef56cfb4fddb2453f7c1abece26-DONUTS
Creation Date	2022-05-10T12:04:05Z
Updated Date	2022-06-07T19:01:46Z
Registry Expiry Date	2023-05-10T12:04:05Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain was created 5 months before the creation of the audit. It will expire in 7 months.

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

Summary

Obolo Token is an interesting project that has a friendly and growing community. The ProfileTransferRouter contract is out of the scope of this audit. The Smart Contract analysis reported no compiler error. 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.

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