Cyberscope

Audit Report

Defiskeletons

May 2022

Type	BEP20
Network	BSC
Address	0xb0688e82d162df5288a0d986dffd4cf80afb7897
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	Defiskeletons
Compiler Version	v0.8.7+commit.e28d00a7
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0xb0688E82D162Df5288A0D986DfFd4CF80AFb7897
Symbol	Skeleton
Decimals	18
Total Supply	1,000,000
Domain	defiskeletons.com

Source Files

Filename	**SHA256**
contract.sol	3437c6b4eaf29c850d9b84a1d3e52c4e17c5d00083ea2818941f27e28e5ae9bd

Audit Updates

Initial Audit	23rd May 2022
Corrected	

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#L1079, 1088

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 `transferToOwner` and `withdrawToken` functions.

```
function transferToOwner(uint256 amount) external onlyOwner {

       address master = owner();
        if (amount == 0) {
            amount = address(this).balance;
        }
        payable(master).transfer(amount);
    }
```

```
function withdrawToken( uint256 _tPercent) external onlyOwner {
        IERC20 tokenContract = IERC20(address(this));
        uint256 balance=balanceOf(address(this))*_tPercent/100;
        // transfer the token from address of this contract
        // to address of the user (executing the withdrawToken() function)
        tokenContract.transfer(msg.sender, balance);
    }
```

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.

BC - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L933

Description

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

```
require(!unAuthlisted[from] && !unAuthlisted[to], 'Address is 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.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	CO	Code Optimization
	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

CO - Code Optimization

Criticality	minor
Location	contract.sol#L660, 1067, 1073

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 following variables are not used by other contract's methods.

```
uint256 private _usageAppLimit0;
uint256 private _usageAppLimit1;
uint256 private _usageAppLimit2;
```

```
function getLimits() public view returns (uint256 _app0, uint256 _app1,
uint256 _app2) {
        _app0 = _usageAppLimit0;
        _app1 = _usageAppLimit1;
        _app2 = _usageAppLimit2;
    }
```

```
function setUsageLimits(uint256 app0,uint256 app1,uint256 app2) external
onlyOwner {
        _usageAppLimit0 = app0;
        _usageAppLimit1 = app1;
        _usageAppLimit2 = app2;
    }
```

Recommendation

Rewrite some code segments so the runtime will be more performant. Try to eliminate these code segments.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L587,591,711,715,719,723,731,740,749,758,775,788,874,878,882,886,894,898,903,907,911,915,919,1062,1067

Description

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

```
getLimits
unAuthAddress
maxBalance
liquifyThreshhold
maxFees
developmentFee
liquidityFee
includeInMaxBalance
excludeFromMaxBalance
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L640,638,639

Description

Constant state variables should be declared constant to save gas.

```
_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#L67,69,100,146,1062,1088,1097,653

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.

```
_maxFees
_router
_tPercent
_value
_address
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

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L861,868,1073

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.

```
_usageAppLimit0 = app0
_maxBalance = newMaxBalance
_liquifyThreshhold = newLiquifyThreshhold
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L367,372,381,389,420,425,403,408,351,360,437

Description

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

```
verifyCallResult
sendValue
isContract
functionStaticCall
functionDelegateCall
functionCallWithValue
functionCall
...
```

Recommendation

Remove unused functions.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	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		-

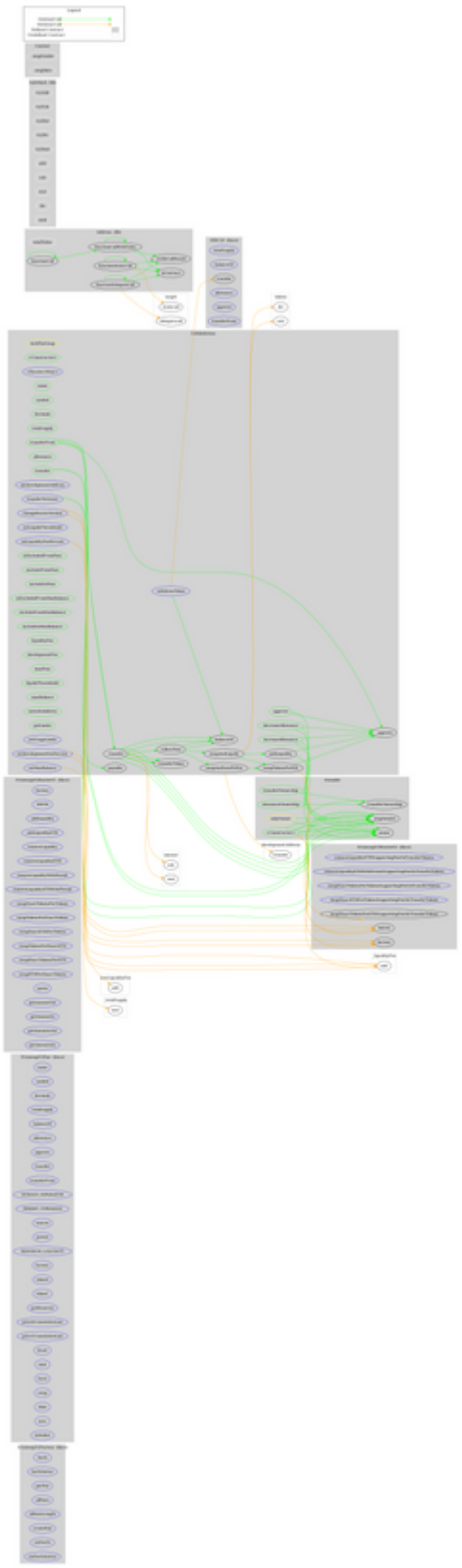
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	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		-
IUniswapV2Ro	Interface	IUniswapV2		

uter02		Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		
SafeMath	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		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Defiskeletons	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	<Receive Ether>	External	Payable	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-

	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_approve	Private	✓	
	setdevelopmentAddress	External	✓	onlyOwner
	setLiquidityFeePercent	External	✓	onlyOwner
	setdevelopmentFeePercent	External	✓	onlyOwner
	setLiquifyThreshhold	External	✓	onlyOwner
	setMaxBalance	External	✓	onlyOwner
	isExcludedFromFees	Public		-
	excludeFromFees	Public	✓	onlyOwner
	includeInFees	Public	✓	onlyOwner
	isExcludedFromMaxBalance	Public		-
	excludeFromMaxBalance	Public	✓	onlyOwner
	includeInMaxBalance	Public	✓	onlyOwner
	liquidityFee	Public		-
	developmentFee	Public		-
	maxFees	Public		-
	liquifyThreshhold	Public		-
	maxBalance	Public		-
	_transfer	Private	✓	
	collectFees	Private	✓	lockTheSwap
	swapAndLiquify	Private	✓	
	swapAndSendToFee	Private	✓	
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	transferToken	Private	✓	
	unAuthAddress	Public	✓	onlyOwner
	getLimits	Public		-
	setUsageLimits	External	✓	onlyOwner
	transferToOwner	External	✓	onlyOwner
	withdrawToken	External	✓	onlyOwner
	changeRouterVersion	External	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	defiskeletons.com
Registry Domain ID	2684398621_DOMAIN_COM-VRSN
Creation Date	2022-03-25T19:55:23Z
Updated Date	2022-03-25T20:05:28Z
Registry Expiry Date	2023-03-25T19:55:23Z
Registrar WHOIS Server	whois.sawbuck.com
Registrar URL	http://www.automattic.com/
Registrar	Automattic Inc.
Registrar IANA ID	1531

The domain has been created about 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

There are some functions that can be abused by the owner like transferring funds to the team’s wallet and blacklisting addresses. The maximum fee percentage that can be set is 15%. 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.

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