Cyberscope

Audit Report

FIFA-USDT-BETTING

August 2022

Type	BEP20
Network	BSC
Address	0x56C08E2F266Ff4EbF64eb130F0189B3Bb1d0DC55
Audited by	© cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 4
OCTD - Owner Contract Tokens Drain 5
Description 5
Recommendation 5
BC - Blacklisted Contracts 6
Description 6
Recommendation 6
Contract Diagnostics 7
STC - Succeeded Transfer Check 8
Description 8
Recommendation 8
BLC - Business Logic Concern 9
Description 9
Recommendation 9
CR - Code Repetition 10
Description 10
Recommendation 10
L01 - Public Function could be Declared External 11
Description 11
Recommendation 11
L04 - Conformance to Solidity Naming Conventions 12
Description 12

Recommendation 12
L05 - Unused State Variable 13
Description 13
Recommendation 13
L07 - Missing Events Arithmetic 14
Description 14
Recommendation 14
L09 - Dead Code Elimination 15
Description 15
Recommendation 15
L12 - Using Variables before Declaration 16
Description 16
Recommendation 16
L14 - Uninitialized Variables in Local Scope 17
Description 17
Recommendation 17
L15 - Local Scope Variable Shadowing 18
Description 18
Recommendation 18
Contract Functions 19
Contract Flow 28
Domain Info 29
Summary 30
Disclaimer 31
About Cyberscope 32

Contract Review

Contract Name	BABYTOKEN
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x56C08E2F266Ff4EbF64eb130F0189B3Bb1d0DC55
Symbol	FUB
Decimals	18
Total Supply	100,000,000
Domain	fifa-usdt-betting.club

Source Files

Filename	**SHA256**
contract.sol	d8bfa8d965c2d7ce0dfb32589f7f26a4d7730f1e1e95d4430e847cfc967068a5

Audit Updates

Initial Audit	1st August 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

OCTD - Owner Contract Tokens Drain

Criticality	minor
Location	contract.sol#L2175

Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling the `swapManual` function.

```
function swapManual() public onlyOwner {
        uint256 contractTokenBalance = balanceOf(address(this));
        require(contractTokenBalance > 0 , "token balance zero");
        swapping = true;
        if(AmountLiquidityFee > 0) swapAndLiquify(AmountLiquidityFee);
        if(AmountTokenRewardsFee > 0) swapAndSendDividends(AmountTokenRewardsFee);
        if(AmountMarketingFee > 0) swapAndSendToFee(AmountMarketingFee);
        swapping = false;
}
```

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	critical
Location	contract.sol#L2034

Description

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

```
function multipleBotlistAddress(address[] calldata accounts, bool excluded) public onlyOwner {
        for (uint256 i = 0; i < accounts.length; i++) {
            _isBlacklisted[accounts[i]] = excluded;
        }
}
```

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
	STC	Succeeded Transfer Check
	BLC	Business Logic Concern
	CR	Code Repetition
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L05	Unused State Variable
	L07	Missing Events Arithmetic
	L09	Dead Code Elimination
	L12	Using Variables before Declaration
	L14	Uninitialized Variables in Local Scope
	L15	Local Scope Variable Shadowing

STC - Succeeded Transfer Check

Criticality	minor
Location	contract.sol#L2302

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 swapAndSendToFee(uint256 tokens) private  {
        uint256 initialCAKEBalance = IERC20(rewardToken).balanceOf(address(this));
        swapTokensForCake(tokens);
        uint256 newBalance =
(IERC20(rewardToken).balanceOf(address(this))).sub(initialCAKEBalance);
        IERC20(rewardToken).transfer(_marketingWalletAddress, newBalance);
        AmountMarketingFee = AmountMarketingFee - tokens;
    }
```

Recommendation

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

BLC - Business Logic Concern

Criticality	medium
Location	contract.sol#L2192

Description

The business logic seems peculiar. The implementation may not follow the expected behaviour.

```
function setDeadWallet(address addr) public onlyOwner {
        deadWallet = addr;
}
```

Recommendation

Dead wallet should not be able to change, it may mislead the meaning of burn fee.

The team is advised to carefully check if the implementation follows the expected business logic.

CR - Code Repetition

Criticality	minor
Location	contract.sol#L2214

Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily.

The `swapManual` method can be reused in the `_tansfer` method.

```
function swapManual() public onlyOwner {
        uint256 contractTokenBalance = balanceOf(address(this));
        require(contractTokenBalance > 0 , "token balance zero");
        swapping = true;
        if(AmountLiquidityFee > 0) swapAndLiquify(AmountLiquidityFee);
        if(AmountTokenRewardsFee > 0) swapAndSendDividends(AmountTokenRewardsFee);
        if(AmountMarketingFee > 0) swapAndSendToFee(AmountMarketingFee);
        swapping = false;
}
```

This code segment is repetitive in the contract.

```
            LFee = amount.mul(buyLiquidityFee).div(100);
            AmountLiquidityFee += LFee;
            RFee = amount.mul(buyTokenRewardsFee).div(100);
            AmountTokenRewardsFee += RFee;
            MFee = amount.mul(buyMarketingFee).div(100);
            AmountMarketingFee += MFee;
            DFee = amount.mul(buyDeadFee).div(100);
            fees = LFee.add(RFee).add(MFee).add(DFee);
```

Recommendation

Create an internal function that contains the code segment and remove it from all the sections.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L1203,2089,1463,872,499,887,2108,147,32,1115,436,2030,1515,876,540,419,462,521,1123,2112,2181,1231,41,2025,895,36,2188,1666,411,1185,1250,139,2045,2116,2184,1140,2171,443,2073,481,2061,1174,1816,1683,1534,1768,2124,1166,470

Description

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

```
allowance
transfer
isExcludedFromDividends
getAccountAtIndex
withdrawnDividendOf
process
excludeMultipleAccountsFromFees
approve
setAutomatedMarketMakerPair
...
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L1455,97,1719,827,108,844,1454,1433,157,1905,826,1107,2181,1904,1910,1102,88,1409,1515,1534,1907,1456,1522,113,662,1903,1548,84,1453

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.

```
__DividendPayingToken_init
__Context_init
_owner
AmountLiquidityFee
WETH
__Ownable_init_unchained
_symbol
_marketingWalletAddress
__gap
...
```

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#L249,157

Description

There are segments that contain unused state variables.

```
__gap
MAX_INT256
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L2184,2202,2193

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.

```
buyTokenRewardsFee = rewardsFee
sellTokenRewardsFee = rewardsFee
swapTokensAtAmount = amount
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L373,295,353,604,1274,339,1567,84

Description

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

```
__Context_init
_transfer
cloneDeterministic
_burn
predictDeterministicAddress
abs
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L2289

Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
lastProcessedIndex
iterations
claims
```

Recommendation

The variables should be declared before any usage of them.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L2255,2289,2251

Description

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

```
claims
fees
lastProcessedIndex
iterations
DFee
```

Recommendation

All the local scoped variables should be initialized.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L1455,1515,1534,1982,1548,1456,1522

Description

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

```
_owner
_symbol
totalSupply
_name
```

Recommendation

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

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	getTime	Public		-
	_transferOwnership	Internal	✓	
Initializable	Implementation			
ContextUpgradeable	Implementation	Initializable		
	__Context_init	Internal	✓	initializer
	__Context_init_unchained	Internal	✓	initializer
	_msgSender	Internal		
	_msgData	Internal		
OwnableUpgradeable	Implementation	Initializable, ContextUpgradeable		
	__Ownable_init	Internal	✓	initializer
	__Ownable_init_unchained	Internal	✓	initializer
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	

IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
	toUint256Safe	Internal		
SafeMathUint	Library			
	toInt256Safe	Internal		

Clones	Library			
	clone	Internal	✓	
	cloneDeterministic	Internal	✓	
	predictDeterministicAddress	Internal		
	predictDeterministicAddress	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	✓	
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		-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
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	✓	-
IterableMapping	Library			
	get	Public		-
	getIndexOfKey	Public		-
	getKeyAtIndex	Public		-
	size	Public		-

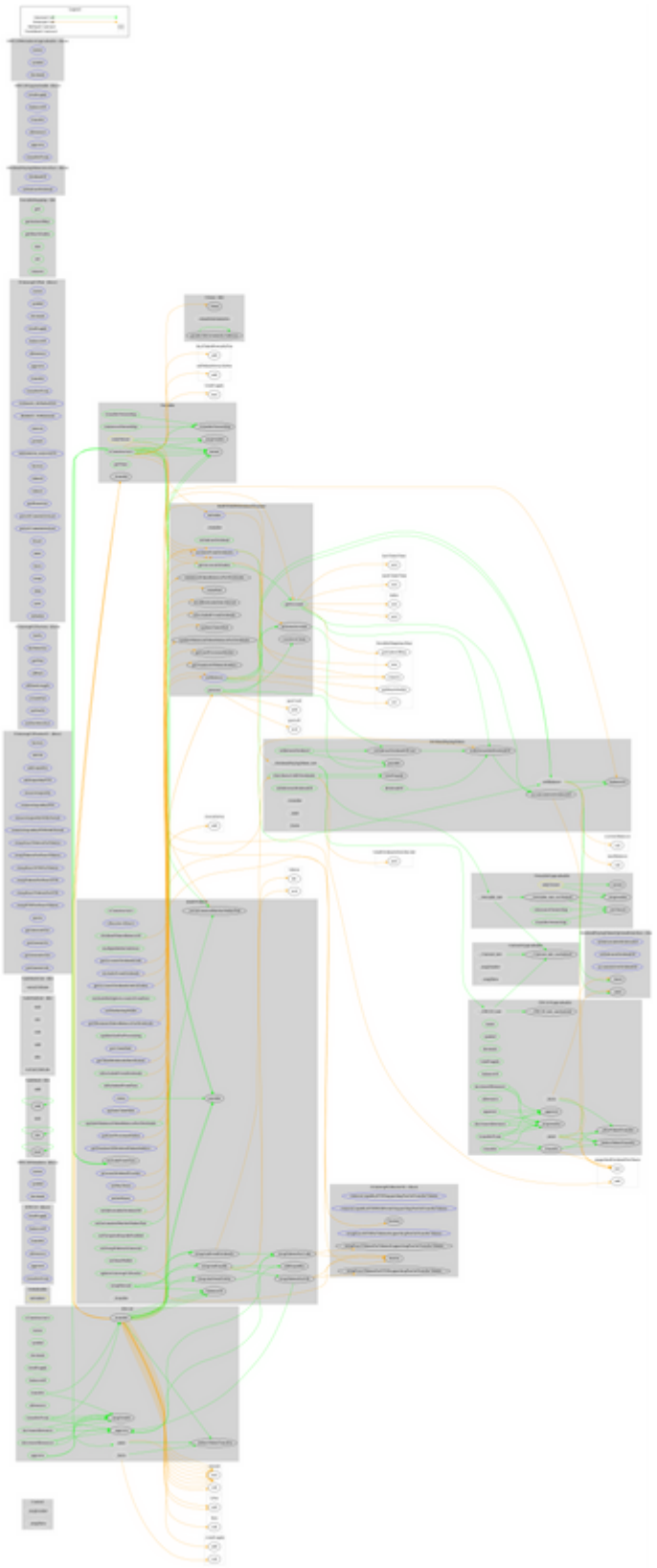
	set	Public	✓	-
	remove	Public	✓	-
DividendPayingTokenInterface	Interface			
	dividendOf	External		-
	withdrawDividend	External	✓	-
DividendPayingTokenOptionalInterface	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-
IERC20Upgradeable	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20MetadataUpgradeable	Interface	IERC20Upgradeable		
	name	External		-
	symbol	External		-
	decimals	External		-
ERC20Upgradeable	Implementation	Initializable, ContextUpgradeable, IERC20Upgradeable, IERC20MetadataUpgradeable		

	__ERC20_init	Internal	✓	initializer
	__ERC20_init_unchained	Internal	✓	initializer
	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	✓	
DividendPayingToken	Implementation	ERC20Upgradeable, OwnableUpgradeable, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface		
	__DividendPayingToken_init	Internal	✓	initializer
	distributeCAKEDividends	Public	✓	onlyOwner
	withdrawDividend	Public	✓	-
	_withdrawDividendOfUser	Internal	✓	
	dividendOf	Public		-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-

	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	
BABYTOKENDividendTracker	Implementation	OwnableUpgradeable, DividendPayingToken		
	initialize	External	✓	initializer
	_transfer	Internal		
	withdrawDividend	Public		-
	excludeFromDividends	External	✓	onlyOwner
	isExcludedFromDividends	Public		-
	updateClaimWait	External	✓	onlyOwner
	updateMinimumTokenBalanceForDividends	External	✓	onlyOwner
	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getAccount	Public		-
	getAccountAtIndex	Public		-
	canAutoClaim	Private		
	setBalance	External	✓	onlyOwner
	process	Public	✓	-
	processAccount	Public	✓	onlyOwner
BABYTOKEN	Implementation	ERC20, Ownable		
	<Constructor>	Public	Payable	ERC20
	<Receive Ether>	External	Payable	-
	updateMinimumTokenBalanceForDividends	Public	✓	onlyOwner
	multipleBotlistAddress	Public	✓	onlyOwner
	getMinimumTokenBalanceForDividends	External		-
	updateUniswapV2Router	Public	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	excludeMultipleAccountsFromFees	Public	✓	onlyOwner

	setMarketingWallet	External	✓	onlyOwner
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateGasForProcessing	Public	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getClaimWait	External		-
	getTotalDividendsDistributed	External		-
	isExcludedFromFees	Public		-
	withdrawableDividendOf	Public		-
	dividendTokenBalanceOf	Public		-
	excludeFromDividends	External	✓	onlyOwner
	isExcludedFromDividends	Public		-
	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-
	processDividendTracker	External	✓	-
	claim	External	✓	-
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	swapManual	Public	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	setSwapTokensAtAmount	Public	✓	onlyOwner
	setDeadWallet	Public	✓	onlyOwner
	setBuyTaxes	External	✓	onlyOwner
	setSelTaxes	External	✓	onlyOwner
	_transfer	Internal	✓	
	swapAndSendToFee	Private	✓	
	swapAndLiquify	Private	✓	
	swapTokensForEth	Private	✓	
	swapTokensForCake	Private	✓	
	addLiquidity	Private	✓	
	swapAndSendDividends	Private	✓	

Contract Flow



Domain Info

Domain Name	fifa-usdt-betting.club
Registry Domain ID	D4862E861921A472DAAD9055C45C645EE-GDREG
Creation Date	2022-07-20T23:46:50Z
Updated Date	2022-07-25T23:46:50Z
Registry Expiry Date	2023-07-20T23:46:50Z
Registrar WHOIS Server	whois.namesilo.com
Registrar URL	www.namesilo.com
Registrar	NameSilo, LLC
Registrar IANA ID	1479

The domain has been created in 12 months before the creation of the audit.

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 tokens to the team’s wallet and massively blacklisting addresses. There is also a limit of max 25% fees.

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