



Cyberscope

## Audit Report

# Alpha Protocol

August 2022

Type       BEP20

Network     BSC

Address     0xb3eC968F5a16F4a5B935a283Af963A6659cE9cF8

Audited by  © cyberscope

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>Contract Review</b>	<b>4</b>
<b>Source Files</b>	<b>4</b>
<b>Audit Updates</b>	<b>4</b>
<b>Contract Analysis</b>	<b>5</b>
<b>ST - Stop Transactions</b>	<b>6</b>
Description	6
Recommendation	6
<b>OCTD - Owner Contract Tokens Drain</b>	<b>7</b>
Description	7
Recommendation	7
<b>ELFM - Exceed Limit Fees Manipulation</b>	<b>8</b>
Description	8
Recommendation	8
<b>Contract Diagnostics</b>	<b>9</b>
<b>STC - Succeeded Transfer Check</b>	<b>10</b>
Description	10
Recommendation	10
<b>FSA - Fixed Swap Address</b>	<b>11</b>
Description	11
Recommendation	11
<b>MAL - Misused Algorithmic Logic</b>	<b>12</b>
Description	12
Recommendation	12
<b>MTS - Manipulate Total Supply</b>	<b>13</b>
Description	13

<b>Recommendation</b>	<b>13</b>
<b>L01 - Public Function could be Declared External</b>	<b>14</b>
<b>Description</b>	<b>14</b>
<b>Recommendation</b>	<b>14</b>
<b>L02 - State Variables could be Declared Constant</b>	<b>15</b>
<b>Description</b>	<b>15</b>
<b>Recommendation</b>	<b>15</b>
<b>L04 - Conformance to Solidity Naming Conventions</b>	<b>16</b>
<b>Description</b>	<b>16</b>
<b>Recommendation</b>	<b>16</b>
<b>L05 - Unused State Variable</b>	<b>17</b>
<b>Description</b>	<b>17</b>
<b>Recommendation</b>	<b>17</b>
<b>L07 - Missing Events Arithmetic</b>	<b>18</b>
<b>Description</b>	<b>18</b>
<b>Recommendation</b>	<b>18</b>
<b>L08 - Tautology or Contradiction</b>	<b>19</b>
<b>Description</b>	<b>19</b>
<b>Recommendation</b>	<b>19</b>
<b>L09 - Dead Code Elimination</b>	<b>20</b>
<b>Description</b>	<b>20</b>
<b>Recommendation</b>	<b>20</b>
<b>L13 - Divide before Multiply Operation</b>	<b>21</b>
<b>Description</b>	<b>21</b>
<b>Recommendation</b>	<b>21</b>
<b>L14 - Uninitialized Variables in Local Scope</b>	<b>22</b>
<b>Description</b>	<b>22</b>
<b>Recommendation</b>	<b>22</b>

<b>Contract Functions</b>	<b>23</b>
<b>Contract Flow</b>	<b>29</b>
<b>Domain Info</b>	<b>30</b>
<b>Summary</b>	<b>31</b>
<b>Disclaimer</b>	<b>32</b>
<b>About Cyberscope</b>	<b>33</b>

## Contract Review

<b>Contract Name</b>	AlphaProtocol
<b>Compiler Version</b>	v0.7.6+commit.7338295f
<b>Optimization</b>	200 runs
<b>Licence</b>	MIT
<b>Explorer</b>	<a href="https://bscscan.com/token/0xb3eC968F5a16F4a5B935a283Af963A6659cE9cF8">https://bscscan.com/token/0xb3eC968F5a16F4a5B935a283Af963A6659cE9cF8</a>
<b>Symbol</b>	AlphaPro
<b>Decimals</b>	5
<b>Total Supply</b>	1,000,000
<b>Domain</b>	<a href="https://alpha-protocol.io">https://alpha-protocol.io</a>

## Source Files

<b>Filename</b>	<b>SHA256</b>
<b>contract.sol</b>	1648d4eb77970debbb7111dc9ac5eded475749801c1200e961cefc2fc565cf0

## Audit Updates

<b>Initial Audit</b>	27th June 2022
<b>Corrected</b>	4th August 2022

# 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

## ST - Stop Transactions

Criticality	medium
Location	contract.sol#L773

### Description

Users are not able to sell more than 70% of their holdings every day.

```
function checkLimit(address sender,
    address recipient,
    uint256 amount) internal {
    if (recipient == pair && !_isFeeExempt[sender]) {
        uint256 limitSell =
balanceOf(sender).mul(maxSellPerDay).div(10000);
        require(amount <= limitSell, "Alpha Protocol: Cant Sell More Than
Allowed Per Day");
        if (block.timestamp > lastSell[sender].timestamp +
TwentyFourHours) {
            lastSell[sender].timestamp = block.timestamp;
            lastSell[sender].amount = amount;
        } else if ((block.timestamp < lastSell[sender].timestamp +
TwentyFourHours) && (block.timestamp > lastSell[sender].timestamp)) {
            require(lastSell[sender].amount + amount <= limitSell, "Alpha
Protocol: Cant Sell More Than Allowed Per Day");
            lastSell[sender].amount = lastSell[sender].amount + amount;
        }
    }
}
```

### Recommendation

The team should embody a method which allows the users to sell more than a reasonable amount of tokens in a short timespan.

## OCTD - Owner Contract Tokens Drain

Criticality	minor
Location	contract.sol#L978

### 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 `withdrawAllToTreasury` function.

```
function withdrawAllToTreasury() external swapping onlyOwner {  
  
    uint256 amountToSwap =  
_gonBalances[address(this)].div(_gonsPerFragment);  
    require( amountToSwap > 0,"There is no AlphaPro token deposited in  
token contract");  
    address[] memory path = new address[](2);  
    path[0] = address(this);  
    path[1] = router.WETH();  
    router.swapExactTokensForETHSupportingFeeOnTransferTokens(  
        amountToSwap,  
        0,  
        path,  
        treasuryReceiver,  
        block.timestamp  
    );  
}
```

### 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.



## ELFM - Exceed Limit Fees Manipulation

<b>Criticality</b>	critical
<b>Location</b>	contract.sol#L1147

### 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 `setTransferTax` function with 5000. As a result, the fees will be 50%.

```
function setTransferTax(uint256 newTransferTax) external onlyOwner {  
    require(newTransferTax <= 5000, "AlphaPro: Maximum Transfer Tax Is 50%");  
    require(newTransferTax >= 0, "AlphaPro: Minimum Transfer Tax Is 0%");  
    transferTax = newTransferTax;  
}
```

### Recommendation

The contract could embody a check for the maximum acceptable value of 25%.

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
●	FSA	Fixed Swap Address
●	MAL	Misused Algorithmic Logic
●	MTS	Manipulate Total Supply
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L05	Unused State Variable
●	L07	Missing Events Arithmetic
●	L08	Tautology or Contradiction
●	L09	Dead Code Elimination
●	L13	Divide before Multiply Operation
●	L14	Uninitialized Variables in Local Scope

## STC - Succeeded Transfer Check

**Criticality**

minor

**Location**

contract.sol#L473,935

### 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.

```
BUSD.transfer(shareholder, amount);

(bool success, ) = payable(treasuryReceiver).call{
    value: amountETHToTreasuryAndSIF,
    gas: 30000
}("");
```

### Recommendation

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

## FSA - Fixed Swap Address

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L672

### Description

The swap address is assigned once in the constructor and it can not be changed. The decentralized swaps sometimes create a new swap version or abandon the current. A contract that cannot change the swap address may not be able to catch-up the upgrade.

```
router = IPancakeSwapRouter(0x10ED43C718714eb63d5aA57B78B54704E256024E);  
pair = IPancakeSwapFactory(router.factory()).createPair(  
    router.WETH(),  
    address(this)  
);
```

### Recommendation

It could be better to allow the swap address mutation in case of future swap updates.

## MAL - Misused Algorithmic Logic

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L715

### Description

The algorithmic flow does not follow the required business logic. In the following statement the third and the forth if will never be fulfilled since an unsigned integer is either less than or greater/equal to 365 days. Hence, always the first two if statements will be fulfilled.

```
if (deltaTimeFromInit <= (365 days)) {  
    rebaseRate = 19435595;  
} else if (deltaTimeFromInit >= (365 days)) {  
    rebaseRate = 9717797;  
} else if (deltaTimeFromInit >= ((15 * 365 days) / 10)) {  
    rebaseRate = 2858898;  
} else if (deltaTimeFromInit >= (7 * 365 days)) {  
    rebaseRate = 1929449;  
}
```

### Recommendation

The algorithm should be reshaped so it will match to the business logic.

## MTS - Manipulate Total Supply

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

### Description

Owner is able to manipulate total supply. This change will have a direct impact on the token price and Market Cap.

```
for (uint256 i = 0; i < times; i++) {  
    _totalSupply = _totalSupply  
        .mul((10**RATE_DECIMALS).add(rebaseRate))  
        .div(10**RATE_DECIMALS);  
}
```

### Recommendation

The contract owner should carefully manage the adjustment of the circulating supply (increases or decreases), according to the token's price fluctuations.

## L01 - Public Function could be Declared External

**Criticality**

minor

**Location**

contract.sol#L573,569,565,534,1154,521,539

### Description

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

```
transferOwnership  
owner  
getLiquidityBacking  
renounceOwnership  
name  
symbol  
decimals
```

### Recommendation

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

## L02 - State Variables could be Declared Constant

**Criticality**

minor

**Location**

contract.sol#L624,611,601,622,609,603,602,604,371,600,612

### Description

Constant state variables should be declared constant to save gas.

```
ZERO
liquidityFee
dividendsPerShareAccuracyFactor
autofirePitFee
safuDividendFee
sellFee
feeDenominator
swapEnabled
treasuryFee
...
```

### Recommendation

Add the constant attribute to state variables that never change.



## L04 - Conformance to Solidity Naming Conventions

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L586,1136,1135,612,1110,653,1164,650,154,1173,624,1168,1137,355,191,1023,649,1032,393,153,171,652,347,611,654,1095,651

### 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.

```
_initRebaseStartTime  
_addr  
_flag  
_totalSupply  
DEAD  
_token  
_lastRebasedTime  
MINIMUM_LIQUIDITY  
_minPeriod  
...
```

### 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#L20

### Description

There are segments that contain unused state variables.

```
MAX_INT256
```

### Recommendation

Remove unused state variables.

## L07 - Missing Events Arithmetic

**Criticality**

minor

**Location**

contract.sol#L1143,393,1148

### 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.

```
maxSellPerDay = newMaxSellPerDay  
minPeriod = _minPeriod  
transferTax = newTransferTax
```

### Recommendation

Emit an event for critical parameter changes.

## L08 - Tautology or Contradiction

**Criticality**

minor

**Location**

contract.sol#L1143

### 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)(newTransferTax >= 0,AlphaPro: Minimum Transfer Tax Is 0%)
```

### Recommendation

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

## L09 - Dead Code Elimination

**Criticality**

minor

**Location**

contract.sol#L48

### Description

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

```
abs
```

### Recommendation

Remove unused functions.

## L13 - Divide before Multiply Operation

**Criticality**

minor

**Location**

contract.sol#L843,1154,703

### Description

Performing divisions before multiplications may cause lose of prediction.

```
_gonBalances[autofirePit] =  
_gonBalances[autofirePit].add(gonAmount.div(feeDenominator).mul(autofirePitFee  
)  
)  
feeAmount = gonAmount.div(feeDenominator).mul(_totalFee)  
times = deltaTime.div(1800)  
_gonBalances[autoLiquidityReceiver] =  
_gonBalances[autoLiquidityReceiver].add(gonAmount.div(feeDenominator).mul(liqu  
idityFee))  
liquidityBalance = _gonBalances[pair].div(_gonsPerFragment)  
_gonBalances[address(this)] =  
_gonBalances[address(this)].add(gonAmount.div(feeDenominator).mul(_treasuryFee  
.add(safuDividendFee)))
```

### Recommendation

The multiplications should be prior to the divisions.

## L14 - Uninitialized Variables in Local Scope

**Criticality**

minor

**Location**

contract.sol#L706

### Description

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

```
rebaseRate
```

### Recommendation

All the local scoped variables should be initialized.

# Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
<b>SafeMathInt</b>	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
<b>SafeMath</b>	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
<b>IERC20</b>	Interface			
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	transfer	External	✓	-
	approve	External	✓	-
	transferFrom	External	✓	-
<b>IPancakeSwap Pair</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>IPancakeSwap Router</b>	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-

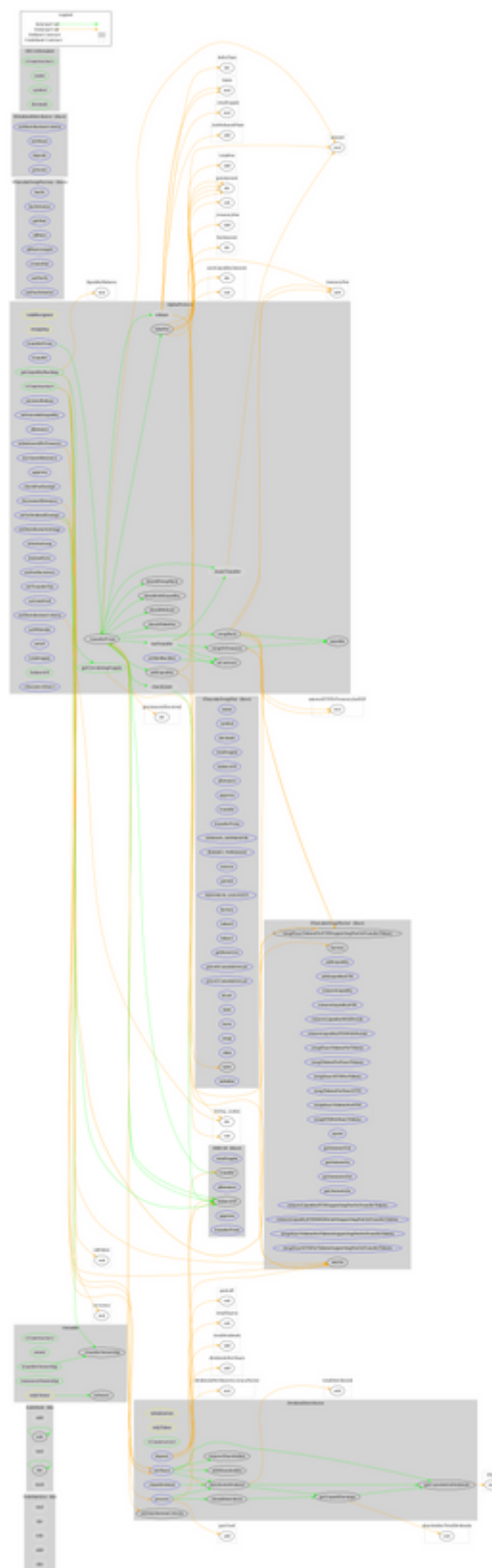
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
<b>IPancakeSwapFactory</b>	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
<b>IDividendDistributor</b>	Interface			
	setDistributionCriteria	External	✓	-
	setShare	External	✓	-

	deposit	External	Payable	-
	process	External	✓	-
<b>DividendDistributor</b>	Implementation	IDividendDistributor		
	<Constructor>	Public	✓	-
	setDistributionCriteria	External	✓	onlyToken
	setShare	External	✓	onlyToken
	deposit	External	Payable	onlyToken
	process	External	✓	onlyToken
	shouldDistribute	Internal		
	distributeDividend	Internal	✓	
	claimDividend	External	✓	-
	getUnpaidEarnings	Public		-
	getCumulativeDividends	Internal		
	addShareholder	Internal	✓	
	removeShareholder	Internal	✓	
<b>Ownable</b>	Implementation			
	<Constructor>	Public	✓	-
	owner	Public		-
	isOwner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
<b>ERC20Detailed</b>	Implementation	IERC20		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
<b>AlphaProtocol</b>	Implementation	ERC20Detailed, Ownable		

	<Constructor>	Public	✓	ERC20Detailed Ownable
	rebase	Internal	✓	
	transfer	External	✓	validRecipient
	transferFrom	External	✓	validRecipient
	_basicTransfer	Internal	✓	
	checkLimit	Internal	✓	
	taxTransfer	Internal	✓	
	_transferFrom	Internal	✓	
	takeFee	Internal	✓	
	addLiquidity	Internal	✓	swapping
	swapToTreasury	Internal	✓	swapping
	swapBack	Internal	✓	swapping
	withdrawAllToTreasury	External	✓	swapping onlyOwner
	shouldTakeFee	Internal		
	shouldRebase	Internal		
	shouldAddLiquidity	Internal		
	shouldSwapBack	Internal		
	setAutoRebase	External	✓	onlyOwner
	setAutoAddLiquidity	External	✓	onlyOwner
	allowance	External		-
	decreaseAllowance	External	✓	-
	increaseAllowance	External	✓	-
	approve	External	✓	-
	checkFeeExempt	External		-
	setIsDividendExempt	External	✓	onlyOwner
	setDistributionCriteria	External	✓	onlyOwner
	setDistributorSettings	External	✓	onlyOwner
	getCirculatingSupply	Public		-
	isNotInSwap	External		-
	manualSync	External	✓	-
	setFeeReceivers	External	✓	onlyOwner
	setTransferTax	External	✓	onlyOwner
	setLimitSell	External	✓	onlyOwner

	getLiquidityBacking	Public		-
	setWhitelist	External	✓	onlyOwner
	setBotBlacklist	External	✓	onlyOwner
	setLP	External	✓	onlyOwner
	totalSupply	External		-
	balanceOf	Public		-
	isContract	Internal		
	<Receive Ether>	External	Payable	-

# Contract Flow



## Domain Info

<b>Domain Name</b>	alpha-protocol.io
<b>Registry Domain ID</b>	713d9f39006d4d0799d73057157a0735-DONUTS
<b>Creation Date</b>	2022-06-02T10:51:16Z
<b>Updated Date</b>	2022-06-07T10:51:48Z
<b>Registry Expiry Date</b>	2023-06-02T10:51:16Z
<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 has been created in 10 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 limiting transactions, transferring tokens to the team's wallet and manipulating 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>