



Cyberscope

Audit Report

Antcoin

July 2022

Type BEP20

Network BSC

Address 0xa52f0c0A5D0217cEE99faBAE71Db0E98ccFc106E

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Contract Analysis	4
ULTW - Unlimited Liquidity to Team Wallet	5
Description	5
Recommendation	5
Contract Diagnostics	6
STC - Succeeded Transfer Check	7
Description	7
Recommendation	7
L01 - Public Function could be Declared External	8
Description	8
Recommendation	8
L02 - State Variables could be Declared Constant	9
Description	9
Recommendation	9
L04 - Conformance to Solidity Naming Conventions	10
Description	10
Recommendation	10
L13 - Divide before Multiply Operation	11
Description	11
Recommendation	11
L15 - Local Scope Variable Shadowing	12
Description	12

Recommendation	12
Contract Functions	13
Contract Flow	16
Domain Info	17
Summary	18
Disclaimer	19
About Cyberscope	20

Contract Review

Contract Name	Antcoin
Compiler Version	v0.8.10+commit.fc410830
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xa52f0c0A5D0217cEE99faBAE71DbaE98ccFc106E
Symbol	Ants
Decimals	18
Total Supply	1,000,000,000
Domain	https://antcoin.app/en

Source Files

Filename	SHA256
contract.sol	6390dab507cd7a0d4670a89d5d07f26c0206a4b4a2409d7021fdd635c7eae906

Audit Updates

Initial Audit	15th July 2022
Corrected	18th July 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

ULTW - Unlimited Liquidity to Team Wallet

Criticality	minor
Location	contract.sol#L410

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 `LiquidityRelease` method.

```
function LiquidityRelease() public onlyOwner {
    require(block.timestamp >= _liquidityUnlockTime, "Not yet unlocked");
    IBEP20 liquidityToken = IBEP20(_pancakePairAddress);
    uint amount = liquidityToken.balanceOf(address(this));
    if(LPReleaseLimitedTo20Percent)
    {
        _liquidityUnlockTime=block.timestamp+DefaultLiquidityLockTime;
        amount=amount*2/10;
    }
    liquidityToken.transfer(msg.sender, amount);
    emit OnReleaseLP();
}
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may volatile the token's price.

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
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L13	Divide before Multiply Operation
●	L15	Local Scope Variable Shadowing

STC - Succeeded Transfer Check

Criticality

minor

Location

contract.sol#L418,L329

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.

```
liquidityToken.transfer(msg.sender, amount);  
  
(bool marketing,)=marketingWallet.call{value:marketbalance}("");  
marketing=true;  
(bool dev,)=devWallet.call{value:devbalance}("");
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality

minor

Location

contract.sol#L188,369,282,386,176,61,272,172,382,277,377

Description

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

```
ExcludeAccountFromFees  
SetOverLiquifiedTreshold  
setExcludedAccountFromLimits  
ChangeDevWallet  
setSwapTreshold  
renounceOwnership  
SetMarketingAndDevShare  
isExcludedFromLimits  
SetTaxes  
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality

minor

Location

contract.sol#L89

Description

Constant state variables should be declared constant to save gas.

```
_circulatingSupply
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L364,168,176,87,282,110,98,405,373,117,118,172,377,88,389,277,369,82,101,83,396,85,426,42

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.

```
AMM
SetAMM
WETH
_owner
_decimals
LockLiquidityForSeconds
_symbol
_liquidityUnlockTime
_name
...
```

Recommendation

Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>.

L13 - Divide before Multiply Operation

Criticality

minor

Location

contract.sol#L297

Description

Performing divisions before multiplications may cause lose of prediction.

```
tokenToSwap = _balances[_pancakePairAddress] * swapTreshold / 1000  
LiqHalf = tokenForLiquidity / 2
```

Recommendation

The multiplications should be prior to the divisions.

L15 - Local Scope Variable Shadowing

Criticality

minor

Location

contract.sol#L426

Description

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

```
_owner
```

Recommendation

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

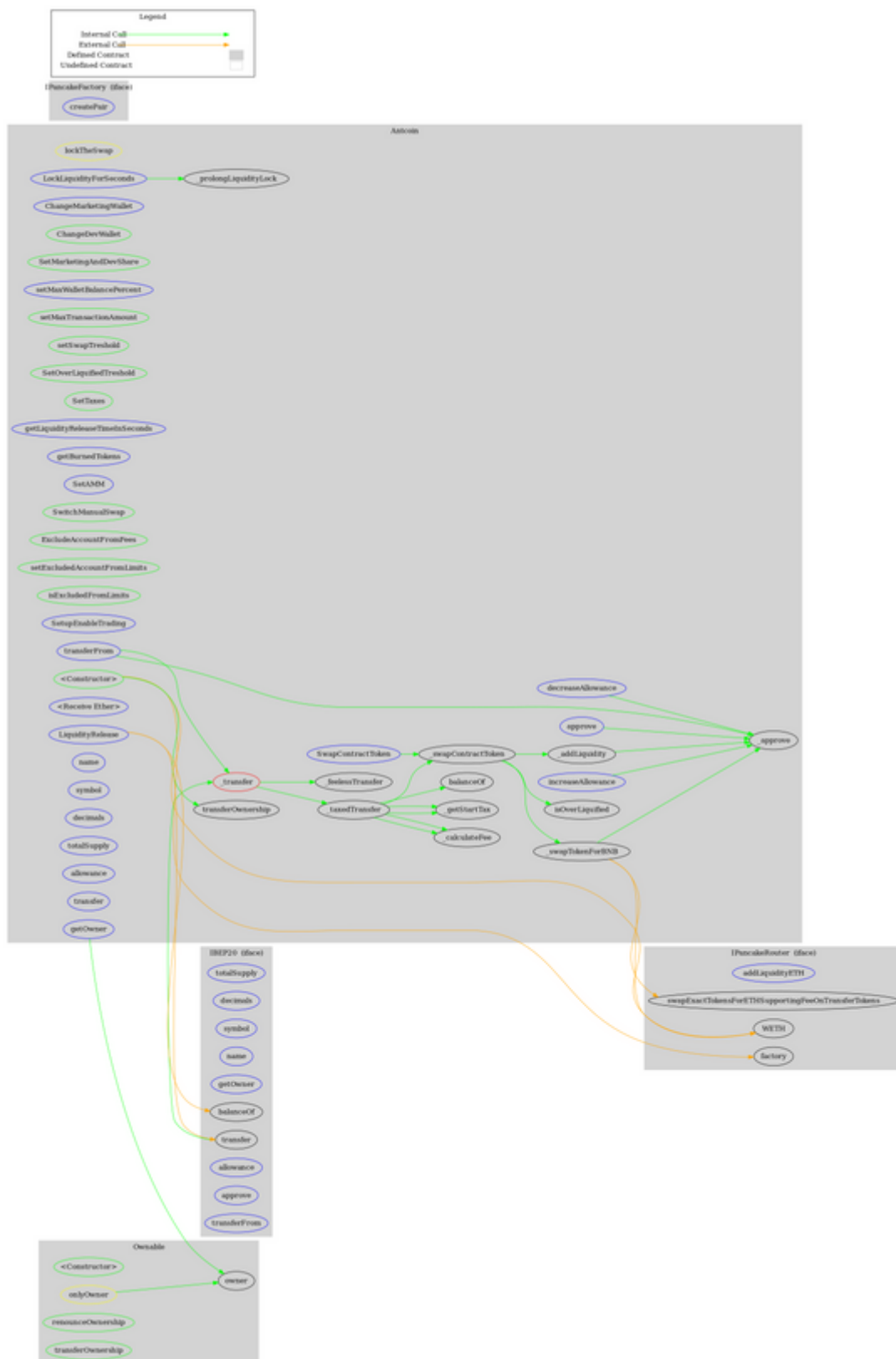
Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IBEP20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IPancakeFactory	Interface			
	createPair	External	✓	-
IPancakeRouter	Interface			
	addLiquidityETH	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	factory	External		-
	WETH	External		-
Ownable	Implementation			
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner

Antcoin	Implementation	IBEP20, Ownable		
	<Constructor>	Public	✓	-
	ChangeMarketingWallet	External	✓	onlyOwner
	ChangeDevWallet	Public	✓	onlyOwner
	SetMarketingAndDevShare	Public	✓	onlyOwner
	setMaxWalletBalancePercent	External	✓	onlyOwner
	setMaxTransactionAmount	Public	✓	onlyOwner
	_transfer	Private	✓	
	_taxedTransfer	Private	✓	
	_getStartTax	Private		
	_calculateFee	Private		
	_feelessTransfer	Private	✓	
	setSwapTreshold	Public	✓	onlyOwner
	SetOverLiquifiedTreshold	Public	✓	onlyOwner
	SetTaxes	Public	✓	onlyOwner
	isOverLiquified	Public		-
	_swapContractToken	Private	✓	lockTheSwap
	_swapTokenForBNB	Private	✓	
	_addLiquidity	Private	✓	
	getLiquidityReleaseTimeInSeconds	External		-
	getBurnedTokens	External		-
	SetAMM	External	✓	onlyOwner
	SwitchManualSwap	Public	✓	onlyOwner
	SwapContractToken	External	✓	onlyOwner
	ExcludeAccountFromFees	Public	✓	onlyOwner
	setExcludedAccountFromLimits	Public	✓	onlyOwner
	isExcludedFromLimits	Public		-
	SetupEnableTrading	External	✓	onlyOwner
	LockLiquidityForSeconds	External	✓	onlyOwner
	_prolongLiquidityLock	Private	✓	
	LiquidityRelease	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	getOwner	External		-

	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	Public		-
	allowance	External		-
	transfer	External	✓	-
	approve	External	✓	-
	_approve	Private	✓	
	transferFrom	External	✓	-
	increaseAllowance	External	✓	-
	decreaseAllowance	External	✓	-

Contract Flow



Domain Info

Domain Name	antcoin.app
Registry Domain ID	494D0CA2C-APP
Creation Date	2022-06-25T20:09:16Z
Updated Date	2022-06-30T20:09:16Z
Registry Expiry Date	2023-06-25T20:09:16Z
Registrar WHOIS Server	whois.nic.google
Registrar URL	None
Registrar	NameSilo, LLC
Registrar IANA ID	1479

The domain was created 12 months before the creation of the audit.

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

Summary

Antcoin is an interesting project that has a friendly and growing community. The Smart Contract analysis reported one issue. The contract owner has the authority to drain liquidity from the pair address. Other than that, 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 20% 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>



Cyberscope

Audit Report

Antcoin

July 2022

Type BEP20

Network BSC

Address 0xc974eefb38cb5e814f8bde916187395292aa6b27

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Contract Analysis	4
ST - Stop Transactions	5
Description	5
Recommendation	5
ELFM - Exceed Limit Fees Manipulation	6
Description	6
Recommendation	6
ULTW - Unlimited Liquidity to Team Wallet	7
Description	7
Recommendation	7
Contract Diagnostics	8
L01 - Public Function could be Declared External	9
Description	9
Recommendation	9
L02 - State Variables could be Declared Constant	10
Description	10
Recommendation	10
L04 - Conformance to Solidity Naming Conventions	11
Description	11
Recommendation	11
L07 - Missing Events Arithmetic	12
Description	12

Recommendation	12
L09 - Dead Code Elimination	13
Description	13
Recommendation	13
L13 - Divide before Multiply Operation	14
Description	14
Recommendation	14
Contract Functions	15
Contract Flow	21
Domain Info	22
Summary	23
Disclaimer	24
About Cyberscope	25

Contract Review

Contract Name	Antcoin
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xc974eefb38cb5e814f8bde916187395292aa6b27
Symbol	Ants
Decimals	9
Total Supply	1,000,000,000
Domain	antcoin.app

Source Files

Filename	SHA256
contract.sol	67d5b7803f50c7a07cc6770468bb46735e03dba95df7c7c517552dd39fe9b8ab

Audit Updates

Initial Audit	8th July 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

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L585,612

Description

The contract owner has the authority to stop the sales for all users excluding the owner. The owner may take advantage of it by setting the `_liquidityFee` to a high value. As a result, the amount will not be sufficient and the transaction will revert.

```
uint256 marketingBnb =
transferredBalance.div(_liquidityFee).mul(marketingDivisor);
uint256 liquidityBnb =
transferredBalance.div(_liquidityFee).mul(liquidityDivisor);
transferToAddressETH(marketingAddress, marketingBnb);
transferToAddressETH(liquidityAddress, liquidityBnb);
```

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

```
if(from != owner() && to != owner()) {
    require(amount <= _maxTxAmount, "Transfer amount exceeds the maxTxAmount.");
}
```

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 `swapTokens` method should not be called if the `_liquidityFee` is zero.

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

Criticality	critical
Location	contract.sol#L828,832

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 `setTaxFeePercent` function with a high percentage value.

```
function setTaxFeePercent(uint256 taxFee) external onlyOwner() {  
    _taxFee = taxFee;  
}
```

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.

ULTW - Unlimited Liquidity to Team Wallet

Criticality	minor
Location	contract.sol#L887

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 `doManualSwapTokens` and `recoverBalance` methods.

```
function recoverBalance(uint256 amount) public onlyOwner {
    payable(owner()).transfer(amount);
}

function doManualSwapTokens(uint256 tokensAmount) public
{
    swapTokens(tokensAmount);
}
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may volatile the token's price.

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
●	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
●	L13	Divide before Multiply Operation

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L157,162,168,172,176,183,455,459,463,467,476,481,485,490,496,501,506,510,514,519,529,546,816,820,824,887,892

Description

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

```
doManualSwapTokens  
recoverBalance  
includeInFee  
excludeFromFee  
isExcludedFromFee  
excludeFromReward  
reflectionFromToken  
deliver  
minimumTokensBeforeSwapAmount  
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality

minor

Location

contract.sol#L410,408,409,404,424

Description

Constant state variables should be declared constant to save gas.

```
liquidityDivisor  
_tTotal  
_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#L219,220,229,249,787,793,844,850,855,859,413,416,419,426

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.

```
_maxTxAmount  
_burnFee  
_liquidityFee  
_taxFee  
_enabled  
_newaddress  
_marketingAddress  
_minimumTokensBeforeSwap  
_amount  
...
```

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#L828,832,836,840,844

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.

```
minimumTokensBeforeSwap = _minimumTokensBeforeSwap
marketingDivisor = divisor
_maxTxAmount = maxTxAmount
_liquidityFee = liquidityFee
_taxFee = taxFee
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L115,98,102,106,110,78,89,656,639

Description

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

```
swapETHForTokens  
addLiquidity  
sendValue  
isContract  
functionCallWithValue  
functionCall  
_functionCallWithValue
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality

minor

Location

contract.sol#L607,689

Description

Performing divisions before multiplications may cause lose of prediction.

```
tBurnAmount = tAmount.div(100).mul(_burnFee)
liquidityBnb = transferredBalance.div(_liquidityFee).mul(liquidityDivisor)
marketingBnb = transferredBalance.div(_liquidityFee).mul(marketingDivisor)
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	

	_functionCallWithValue	Private	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	getUnlockTime	Public		-
	getTime	Public		-
	lock	Public	✓	onlyOwner
	unlock	Public	✓	-
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		-
	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		-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
Antcoin	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	isExcludedFromReward	Public		-
	totalFees	Public		-
	minimumTokensBeforeSwapAmount	Public		-
	deliver	Public	✓	-
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner
	includeInReward	External	✓	onlyOwner

	_approve	Private	✓	
	_transfer	Private	✓	
	swapTokens	Private	✓	lockTheSwap
	swapTokensForEth	Private	✓	
	swapETHForTokens	Private	✓	
	addLiquidity	Private	✓	
	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	
	_transferBothExcluded	Private	✓	
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	calculateTaxFee	Private		
	calculateLiquidityFee	Private		
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	isExcludedFromFee	Public		-
	excludeFromFee	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	setTaxFeePercent	External	✓	onlyOwner
	setLiquidityFeePercent	External	✓	onlyOwner
	setMaxTxAmount	External	✓	onlyOwner
	setMarketingDivisor	External	✓	onlyOwner
	setNumTokensSellToAddToLiquidity	External	✓	onlyOwner
	setMarketingAddress	External	✓	onlyOwner
	setliquidityAddress	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	prepareForPreSale	External	✓	onlyOwner
	afterPreSale	External	✓	onlyOwner

	transferToAddressETH	Public	✓	-
	recoverBalance	Public	✓	onlyOwner
	doManualSwapTokens	Public	✓	-
	<Receive Ether>	External	Payable	-

Contract Flow



Domain Info

Domain Name	antcoin.app
Registry Domain ID	494D0CA2C-APP
Creation Date	2022-06-25T20:09:16Z
Updated Date	2022-06-30T20:09:16Z
Registry Expiry Date	2023-06-25T20:09:16Z
Registrar WHOIS Server	whois.nic.google
Registrar URL	None
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 stopping transactions, manipulating fees and transferring funds to the team's wallet. 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.

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>



Cyberscope

Audit Report

Antcoin

July 2022

SHA256 d8f801d17a0755e8433cd1009ac983cac9b81bdf3649d0a074d700c430b21a61

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
ULTW - Unlimited Liquidity to Team Wallet	6
Description	6
Recommendation	6
Contract Diagnostics	7
STC - Succeeded Transfer Check	8
Description	8
Recommendation	8
L01 - Public Function could be Declared External	9
Description	9
Recommendation	9
L02 - State Variables could be Declared Constant	10
Description	10
Recommendation	10
L04 - Conformance to Solidity Naming Conventions	11
Description	11
Recommendation	11
L13 - Divide before Multiply Operation	12
Description	12

Recommendation	12
L15 - Local Scope Variable Shadowing	13
Description	13
Recommendation	13
Contract Functions	14
Contract Flow	17
Domain Info	18
Summary	19
Disclaimer	20
About Cyberscope	21

Contract Review

Contract Name	Antcoin
Testing Deploy	https://testnet.bscscan.com/address/0xEC8eea03ec866514F8e7Caa2131f1c4B4564d974
Symbol	Ants
Decimals	18
Total Supply	1,000,000,000
Domain	https://antcoin.app/en/

Source Files

Filename	SHA256
contract.sol	d8f801d17a0755e8433cd1009ac983cac9b81bdf3649d0a074d700c430b21a61

Audit Updates

Initial Audit	8th July 2022
Corrected	15th July 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

OCTD - Owner Contract Tokens Drain

Criticality	minor
Location	contract.sol#L422

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 `emergencyBNBRecovery` function.

```
function emergencyBNBRecovery(uint256 amountPercentage) external onlyOwner {  
    uint256 amountBNB = address(this).balance;  
    payable(msg.sender).transfer(amountBNB * amountPercentage / 100);  
    emit RecoverBNB();  
}
```

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.

ULTW - Unlimited Liquidity to Team Wallet

Criticality	minor
Location	contract.sol#L410

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 `LiquidityRelease` method.

```
function LiquidityRelease() public onlyOwner {
    require(block.timestamp >= _liquidityUnlockTime, "Not yet unlocked");
    IBEP20 liquidityToken = IBEP20(_pancakePairAddress);
    uint amount = liquidityToken.balanceOf(address(this));
    if(LPReleaseLimitedTo20Percent)
    {
        _liquidityUnlockTime=block.timestamp+DefaultLiquidityLockTime;
        amount=amount*2/10;
    }
    liquidityToken.transfer(msg.sender, amount);
    emit OnReleaseLP();
}
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may volatile the token's price.

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
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L13	Divide before Multiply Operation
●	L15	Local Scope Variable Shadowing

STC - Succeeded Transfer Check

Criticality

minor

Location

contract.sol#L418,L326

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.

```
liquidityToken.transfer(msg.sender, amount);  
  
(bool marketing,)=marketingWallet.call{value:marketbalance}("");  
marketing=true;  
(bool dev,)=devWallet.call{value:devbalance}("");
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L190,370,283,387,178,61,273,174,383,278,378,410,397,365,390,357,362,170,401,374

Description

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

```
SwapContractToken
LockLiquidityForSeconds
ChangeMarketingWallet
getBurnedTokens
getLiquidityReleaseTimeInSeconds
SetupEnableTrading
SetAMM
limitLiquidityReleaseTo20Percent
LiquidityRelease
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality

minor

Location

contract.sol#L114,89

Description

Constant state variables should be declared constant to save gas.

```
_circulatingSupply
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L365,170,178,87,110,283,98,410,374,117,118,174,378,88,390,278,370,82,101,83,401,85,436,42

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.

```
AMM
SetAMM
WETH
_owner
_decimals
LockLiquidityForSeconds
_symbol
_liquidityUnlockTime
_name
...
```

Recommendation

Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>.

L13 - Divide before Multiply Operation

Criticality

minor

Location

contract.sol#L298

Description

Performing divisions before multiplications may cause lose of prediction.

```
LiqHalf = tokenForLiquidity / 2
```

```
tokenToSwap = _balances[_pancakePairAddress] * swapTreshold / 1000
```

Recommendation

The multiplications should be prior to the divisions.

L15 - Local Scope Variable Shadowing

Criticality

minor

Location

contract.sol#L436

Description

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

```
_owner
```

Recommendation

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

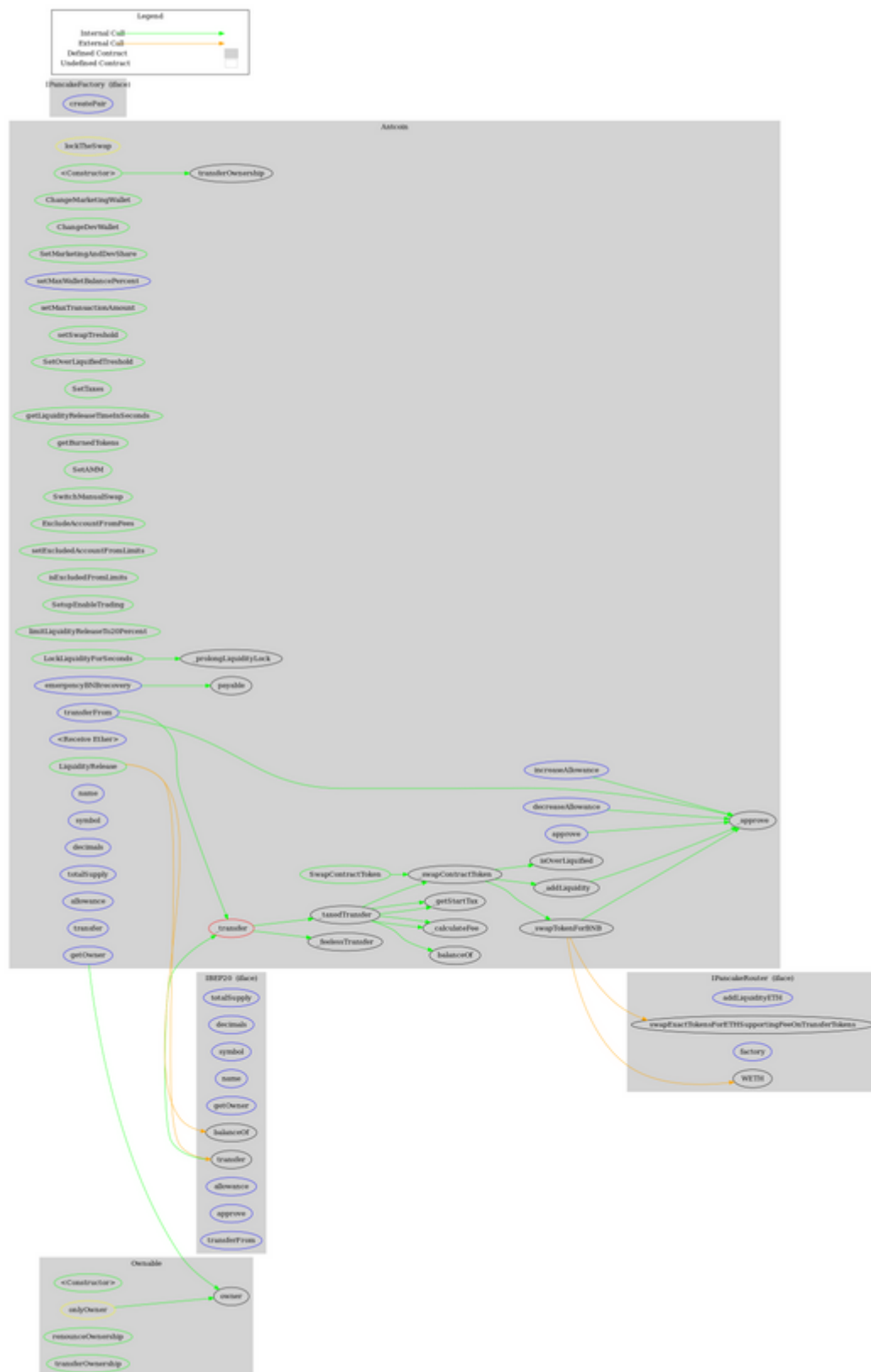
Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IBEP20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IPancakeFactory	Interface			
	createPair	External	✓	-
IPancakeRouter	Interface			
	addLiquidityETH	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	factory	External		-
	WETH	External		-
Ownable	Implementation			
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner

Antcoin	Implementation	IBEP20, Ownable		
	<Constructor>	Public	✓	-
	ChangeMarketingWallet	Public	✓	onlyOwner
	ChangeDevWallet	Public	✓	onlyOwner
	SetMarketingAndDevShare	Public	✓	onlyOwner
	setMaxWalletBalancePercent	External	✓	onlyOwner
	setMaxTransactionAmount	Public	✓	onlyOwner
	_transfer	Private	✓	
	_taxedTransfer	Private	✓	
	_getStartTax	Private		
	_calculateFee	Private		
	_feelessTransfer	Private	✓	
	setSwapTreshold	Public	✓	onlyOwner
	SetOverLiquifiedTreshold	Public	✓	onlyOwner
	SetTaxes	Public	✓	onlyOwner
	isOverLiquified	Public		-
	_swapContractToken	Private	✓	lockTheSwap
	_swapTokenForBNB	Private	✓	
	_addLiquidity	Private	✓	
	getLiquidityReleaseTimeInSeconds	Public		-
	getBurnedTokens	Public		-
	SetAMM	Public	✓	onlyOwner
	SwitchManualSwap	Public	✓	onlyOwner
	SwapContractToken	Public	✓	onlyOwner
	ExcludeAccountFromFees	Public	✓	onlyOwner
	setExcludedAccountFromLimits	Public	✓	onlyOwner
	isExcludedFromLimits	Public		-
	SetupEnableTrading	Public	✓	onlyOwner
	limitLiquidityReleaseTo20Percent	Public	✓	onlyOwner
	LockLiquidityForSeconds	Public	✓	onlyOwner
	_prolongLiquidityLock	Private	✓	
	LiquidityRelease	Public	✓	onlyOwner
	emergencyBNBRecovery	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-

	getOwner	External		-
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	Public		-
	allowance	External		-
	transfer	External	✓	-
	approve	External	✓	-
	_approve	Private	✓	
	transferFrom	External	✓	-
	increaseAllowance	External	✓	-
	decreaseAllowance	External	✓	-

Contract Flow



Domain Info

Domain Name	antcoin.app
Registry Domain ID	494D0CA2C-APP
Creation Date	2022-06-25T20:09:16Z
Updated Date	2022-06-30T20:09:16Z
Registry Expiry Date	2023-06-25T20:09:16Z
Registrar WHOIS Server	whois.nic.google
Registrar URL	None
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 transferring funds to the team's wallet. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. There is also a max limit of 20%.

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>