



# Audit Report

## Cash Machine

February 2022

Type	ERC-20
Network	ETH
Address	0xf0b4eC000e1E2Ca0f8E0C1299e5EFDBAc1F94198
Audited by	© coinscope

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>Contract Review</b>	<b>3</b>
<b>Audit Updates</b>	<b>3</b>
<b>Contract Analysis</b>	<b>4</b>
<b>ST - Stop Transactions</b>	<b>5</b>
Description	5
Recommendation	5
<b>ELFM - Exceed Limit Fees Manipulation</b>	<b>6</b>
Description	6
Recommendation	6
<b>BC - Blacklisted Contracts</b>	<b>7</b>
Description	7
Recommendation	7
<b>Contract Diagnostics</b>	<b>8</b>
<b>L01 - Public Function could be Declared External</b>	<b>9</b>
Description	9
Recommendation	9
<b>L02 - State Variables could be Declared Constant</b>	<b>10</b>
Description	10
Recommendation	10
<b>L05 - Unused State Variable</b>	<b>11</b>
Description	11
Recommendation	11
<b>L04 - Conformance to Solidity Naming Conventions</b>	<b>12</b>
Description	12
Recommendation	12

<b>L09 - Dead Code Elimination</b>	<b>13</b>
Description	13
Recommendation	13
<b>L07 - Missing Events Arithmetic</b>	<b>14</b>
Description	14
Recommendation	14
<b>L14 - Uninitialized Variables in Local Scope</b>	<b>15</b>
Description	15
Recommendation	15
<b>L13 - Divide before Multiply Operation</b>	<b>16</b>
Description	16
Recommendation	16
<b>Contract Functions</b>	<b>17</b>
<b>Contract Flow</b>	<b>24</b>
<b>Summary</b>	<b>25</b>
<b>Disclaimer</b>	<b>26</b>
<b>About Coinscope</b>	<b>27</b>

## Contract Review

<b>Contract Name</b>	CMAC
<b>Compiler Version</b>	v0.8.10+commit.fc410830
<b>Optimization</b>	200 runs
<b>Licence</b>	none
<b>Explorer</b>	<a href="https://etherscan.io/token/0xf0b4eC000e1E2Ca0f8E0C1299e5EFDBAc1F94198">https://etherscan.io/token/0xf0b4eC000e1E2Ca0f8E0C1299e5EFDBAc1F94198</a>
<b>Symbol</b>	CMAC
<b>Decimals</b>	18
<b>Total Supply</b>	1,000,000,000

## Audit Updates

<b>Initial Audit</b>	19th of February 2022
<b>Corrected</b>	

# Contract Analysis

● Critical
 ● Medium
 ● Minor
 ● Pass

Severity	Code	Description
<span style="color: red;">●</span>	ST	Contract Owner is not able to stop or pause transactions
<span style="color: darkblue;">●</span>	OCTD	Contract Owner is not able to transfer tokens from specific address
<span style="color: darkblue;">●</span>	OTUT	Owner Transfer User's Tokens
<span style="color: red;">●</span>	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
<span style="color: darkblue;">●</span>	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
<span style="color: darkblue;">●</span>	MT	Contract Owner is not able to mint new tokens
<span style="color: darkblue;">●</span>	BT	Contract Owner is not able to burn tokens from specific wallet
<span style="color: red;">●</span>	BC	Contract Owner is not able to blacklist wallets from selling

## ST - Stop Transactions

Criticality	critical
Location	contract.sol#L364

### Description

The contract owner has the authority to stop users from selling by increasing the `totalFeeOnSellBPS` to very high value.

```
uint256 fees = (amount * totalFeeOnSellBPS) / 10000;  
uint256 burnAmt = (amount * burnFeeOnSellBPS) / 10000;  
amount -= fees;  
_executeTransfer(sender, DEAD, burnAmt);  
_executeTransfer(sender, address(this), fees - burnAmt);
```

### Recommendation

The contract could embody a check for not allowing setting the `totalFeeOnSellBPS` less than a reasonable amount. 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#L535,546

### 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 `setFeeOnSell` function with high percentage values.

```
function setFeeOnSell(  
    uint256 _treasuryFee,  
    uint256 _liquidityFee,  
    uint256 _dividendFee,  
    uint256 _burnFee  
) external onlyOwner {  
    treasuryFeeOnSellBPS = _treasuryFee;  
    liquidityFeeOnSellBPS = _liquidityFee;  
    dividendFeeOnSellBPS = _dividendFee;  
    burnFeeOnSellBPS = _burnFee;  
    totalFeeOnSellBPS = _treasuryFee + _liquidityFee + _dividendFee +  
    _burnFee;  
}
```

### 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.

## BC - Blacklisted Contracts

Criticality	critical
Location	contract.sol#L701

### Description

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

```
function blackListMany(address[] memory _users) public onlyOwner {  
    for (uint8 i = 0; i < _users.length; i++) {  
        isBlacklisted[_users[i]] = true;  
    }  
}
```

### 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
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L05	Unused State Variable
●	L04	Conformance to Solidity Naming Conventions
●	L09	Dead Code Elimination
●	L07	Missing Events Arithmetic
●	L14	Uninitialized Variables in Local Scope
●	L13	Divide before Multiply Operation

## L01 - Public Function could be Declared External

<b>Criticality</b>	minor
<b>Location</b>	@openzeppelin/contracts/access/Ownable.sol#L54,62  @openzeppelin/contracts/token/ERC20/ERC20.sol#L62,70,87,178,197  contracts/CMAC.sol#L224,192,197,230,171,175,179,203,213,490 and 26 more

### Description

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

```
decimals
symbol
name
getLastClaimTime
getAccountInfo
withdrawnDividendOf
compoundAccount
processAccount
isExcludedFromDividends
...
```

### Recommendation

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

## L02 - State Variables could be Declared Constant

<b>Criticality</b>	minor
<b>Location</b>	contracts/CMAC.sol#L30,29,31,33,34,717,718,720

### Description

Constant state variables should be declared constant to save gas.

```
lastProcessedIndex  
_symbol  
_name  
ZERO  
UNISWAPROUTER  
DEAD
```

### Recommendation

Add the constant attribute to state variables that never change.

## L05 - Unused State Variable

**Criticality**

minor

**Location**

contracts/CMAC.sol#L31

### Description

There are segments that contain unused state variables.

ZERO

### Recommendation

Remove unused state variables.

## L04 - Conformance to Solidity Naming Conventions

<b>Criticality</b>	minor
<b>Location</b>	@uniswap/v2-core/contracts/interfaces/IUniswapV2Pair.sol#L18,19,36  @uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router01.sol#L5  contracts/CMAC.sol#L427,510,511,512,513,514,515,516,517,536 and 32 more

### 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.

```
magnitude
UNISWAPROUTER
TeamWallets
AdvWallets
ResearchWallet
LiquidityWallet
MarketingWallet
PublicSaleWallet
PreSaleWallet
...
```

### Recommendation

Follow the Solidity naming convention.

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

## L09 - Dead Code Elimination

<b>Criticality</b>	minor
<b>Location</b>	@openzeppelin/contracts/utils/Address.sol#L80,90,109,123,169,179,142,152,27,55 and 1 more  contracts/CMAC.sol#L392  @openzeppelin/contracts/utils/Context.sol#L21  @openzeppelin/contracts/token/ERC20/ERC20.sol#L275,252  and 1 more files

### Description

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

```
trySub  
tryMul  
tryMod  
tryDiv  
tryAdd  
sub  
mul  
mod  
div  
...
```

### Recommendation

Remove unused functions.

## L07 - Missing Events Arithmetic

<b>Criticality</b>	minor
<b>Location</b>	contracts/CMAC.sol#L535,546,640,650,663

### 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.

```
maxWalletBPS = bps
maxTxBPS = bps
swapTokensAtAmount = _swapTokensAtAmount
burnFeeOnSellBPS = _burnFee
treasuryFeeBPS = _treasuryFee
```

### Recommendation

Emit an event for critical parameter changes.

## L14 - Uninitialized Variables in Local Scope

<b>Criticality</b>	minor
<b>Location</b>	contracts/CMAC.sol#L443,438,916,971,320

### Description

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

```
takeFee  
info  
success  
swapTokensMarketing  
swapTokensDividends
```

### Recommendation

All the local scoped variables should be initialized.



## L13 - Divide before Multiply Operation

**Criticality**

minor

**Location**

contracts/CMAC.sol#L104,433

### Description

Performing divisions before multiplications may cause lose of prediction.

```
swapTokensDividends = (tokens * dividendFeeBPS) / totalFeeBPS
swapTokensMarketing = (tokens * treasuryFeeBPS) / totalFeeBPS
_mint(AdvWallets[1],100000000 / uint256(3) * (10 ** 18))
_mint(AdvWallets[0],100000000 / uint256(3) * (10 ** 18))
_mint(TeamWallets[1],200000000 / uint256(3) * (10 ** 18))
_mint(TeamWallets[0],200000000 / uint256(3) * (10 ** 18))
```

### Recommendation

The multiplications should be prior to the divisions.

# Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
<b>Ownable</b>	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
<b>ERC20</b>	Implementation	Context, IERC20, IERC20Met adata		
	<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	✓	
	_afterTokenTransfer	Internal	✓	

<b>IERC20Metadata</b>	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
<b>IERC20</b>	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
<b>Address</b>	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		
<b>Context</b>	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
<b>SafeMath</b>	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		
<b>IUniswapV2Factory</b>	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
<b>IUniswapV2Pair</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>IUniswapV2Router01</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		-

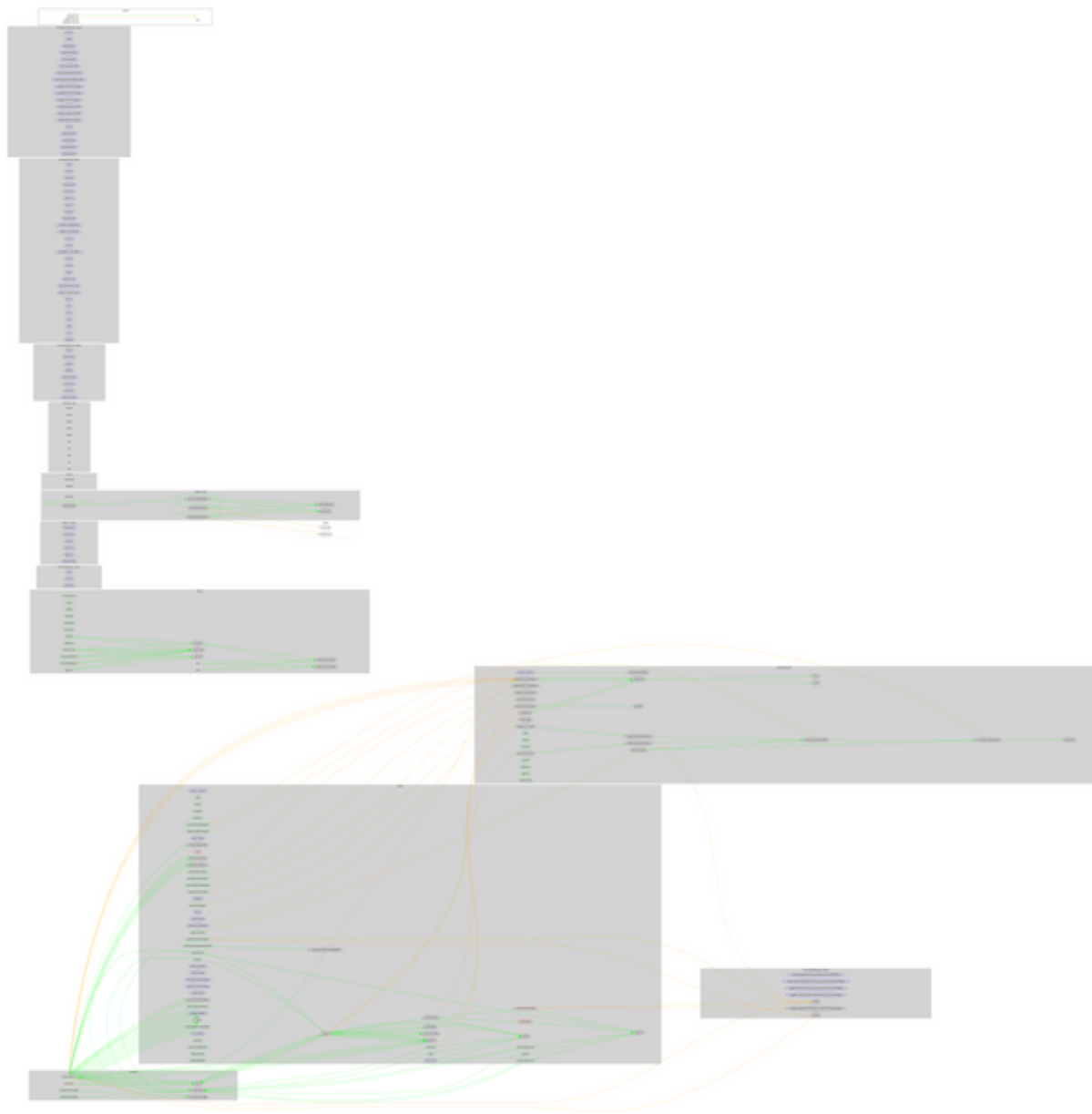
<b>IUniswapV2Router02</b>	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
<b>CMAC</b>	Implementation	Ownable, IERC20		
	<Constructor>	Public	✓	-
	<Receive Ether>	External	Payable	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	allowance	Public		-
	approve	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	transfer	Public	✓	-
	transferFrom	Public	✓	-
	openTrading	External	✓	onlyOwner
	_transfer	Internal	✓	
	_executeTransfer	Private	✓	
	_approve	Private	✓	
	_mint	Private	✓	
	_burn	Private	✓	
	swapTokensForNative	Private	✓	
	addLiquidity	Private	✓	
	includeToWhiteList	Private	✓	

	_executeSwap	Private	✓	
	excludeFromFees	Public	✓	onlyOwner
	isExcludedFromFees	Public		-
	manualSendDividend	External	✓	onlyOwner
	excludeFromDividends	Public	✓	onlyOwner
	isExcludedFromDividends	Public		-
	setWallet	External	✓	onlyOwner
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	setFee	External	✓	onlyOwner
	setFeeOnSell	External	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateUniswapV2Router	Public	✓	onlyOwner
	claim	Public	✓	-
	compound	Public	✓	-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	getAccountInfo	Public		-
	getLastClaimTime	Public		-
	setSwapEnabled	External	✓	onlyOwner
	setTaxEnabled	External	✓	onlyOwner
	setCompoundingEnabled	External	✓	onlyOwner
	updateDividendSettings	External	✓	onlyOwner
	setMaxTxBPS	External	✓	onlyOwner
	excludeFromMaxTx	Public	✓	onlyOwner
	isExcludedFromMaxTx	Public		-
	setMaxWalletBPS	External	✓	onlyOwner
	excludeFromMaxWallet	Public	✓	onlyOwner
	isExcludedFromMaxWallet	Public		-
	rescueToken	External	✓	onlyOwner
	rescueETH	External	✓	onlyOwner
	blackList	Public	✓	onlyOwner
	removeFromBlacklist	Public	✓	onlyOwner
	blackListMany	Public	✓	onlyOwner
	unBlackListMany	Public	✓	onlyOwner

DividendTracker	Implementation	Ownable, IERC20		
	<Constructor>	Public	✓	-
	<Receive Ether>	External	Payable	-
	distributeDividends	Public	Payable	-
	setBalance	External	✓	onlyOwner
	excludeFromDividends	External	✓	onlyOwner
	isExcludedFromDividends	Public		-
	manualSendDividend	External	✓	onlyOwner
	_setBalance	Internal	✓	
	_mint	Private	✓	
	_burn	Private	✓	
	processAccount	Public	✓	onlyOwner
	_withdrawDividendOfUser	Private	✓	
	compoundAccount	Public	✓	onlyOwner
	_compoundDividendOfUser	Private	✓	
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	getAccountInfo	Public		-
	getLastClaimTime	Public		-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public		-
	allowance	Public		-
	approve	Public		-
	transferFrom	Public		-



# Contract Flow



# Summary

Cash Machine (CMAC) is a project with a friendly and growing community. The Smart Contract analysis reported no compiler error and 3 critical issues. There are some functions that can be abused by the owner, like manipulating fees up to 100% and blacklisting users from trading. The contract can also be converted into a honeypot if the admin functions are used in a malicious way. 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.

Coinscope 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 Coinscope 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 Coinscope 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 Coinscope team disclaims any liability for the resulting losses.

## About Coinscope

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.

Coinscope is aiming to make crypto discoverable and efficient globally. It provides all the essential tools to assist users draw their own conclusions.



The Coinscope.co team

<https://www.coinscope.co>