Cyberscope

Audit Report

Z OutBreak

February 2022

Type	BEP20
Network	BSC
Address	0x342C58829c25Cb9a55E7d19330E6978352EaD53d
Audited by	© cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Audit Updates 3
Contract Analysis 4
Contract Inconsistency 5
ST - Stop Transactions 6
Description 6
Recommendation 6
ELFM - Exceed Limit Fees Manipulation 7
Description 7
Recommendation 7
BC - Blacklisted Contracts 8
Description 8
Recommendation 8
Contract Diagnostics 9
L01 - Public Function could be Declared External 10
Description 10
Recommendation 10
L02 - State Variables could be Declared Constant 11
Description 11
Recommendation 11
L03 - Redundant Statements 12
Description 12
Recommendation 12
L04 - Conformance to Solidity Naming Conventions 13
Description 13

Recommendation 13
L09 - Dead Code Elimination 14
Description 14
Recommendation 14
L13 - Divide before Multiply Operation 15
Description 15
Recommendation 15
Contract Functions 16
Contract Flow 23
Domain Info 24
Summary 25
Disclaimer 26
About Cyberscope 27

Contract Review

Contract Name	ZOUTBREAK
Compiler Version	v0.8.8+commit.dddeac2f
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x342C58829c25Cb9a55E7d19330E6978352EaD53d
Symbol	ZOB
Decimals	7
Total Supply	10,000,000
Source	contract.sol
Domain	zoutbreak.com

Audit Updates

Initial Audit	25th February 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

Contract Inconsistency

Criticality	critical

The contract contains a lot of segments that create inconsistency and cannot guarantee a clear business logic. The contract could contain better naming conventions and clean up the business logic functionality. The following are some code excerpts that are creating inconsistency.

The expression does not imply that the Chief_Security_Officer is not allowed to trade.

```
require(from != Chief_Security_Officer || to != Chief_Security_Officer,
        "Security personnel is not allowed to trade");
```

The expression does not imply that the addresses in the Security_Manager list are not allowed to trade.

```
require(!Security_Manager[from] || !Security_Manager[to],
        "Security personnel is not allowed to trade");
```

Pure declaration of variables that have been defined as functions.

```
_tokenTransfer(from,to,amount,takeAllFees);
restoreAllFees;
```

Missing naming conventions that is making the code unreadable.

```
uint256 private pi1;
uint256 private pi2;
uint256 private bpf;
uint256 private brf;
uint256 private spfiinu;
```

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L637,648

Description

The contract owner has the authority to stop the sales for all users excluding the owner. The owner may take advantage of it in many ways. For instance, by adding the address to the boe or by increasing one of the variants that increase the SAT timer.

```
} else if (to == uniswapV2Pair) {
        require(!boe[from]);
        if (nwtbs > 0 || wttsai2 > 0) {
            require(block.timestamp > SAT[from]);
        }
```

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 `Public_Trading_Enabled` to false.

```
require(Public_Trading_Enabled || ieff[from] || ieff[to],
        "Public Trading has not been enabled yet.");
```

Recommendation

The contract could embody a check for not allowing setting the Public_Trading_Enabled to false after the initial toggle. The contract should not allow to timelock the addresses more than a reasonable time period.

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#L1188

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by setting the fees a high percentage value. There are a lot of variables that affect the fee variant like the spfai1, spfai2, boepf etc.

```
function F12__Set_Fees_For_Sells_With_Price_Impact1(
    uint256 project_fee_under_impact1,
    uint256 project_fee_above_impact1,
    uint256 reflections_fee_under_impact1,
    uint256 reflections_fee_above_impact1
    ) external onlySecurityManager {
    //The fees are a percentage number
    require(pi1 != 0, "Cannot set price impact1 fees when impact1 is 0");
    require(project_fee_under_impact1 <= project_fee_above_impact1,
    "The project fee under price impact1 cannot be larger than the fee above
price impact1");
    require(reflections_fee_under_impact1 <= reflections_fee_above_impact1,
    "The reflections fee under price impact1 cannot be larger than the fee above
price impact1");
    ...
    spfui1 = project_fee_under_impact1;
    spfai1 = project_fee_above_impact1;
    srfui1 = reflections_fee_under_impact1;
    srfai1 = reflections_fee_above_impact1;
}
```

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	medium
Location	contract.sol#L631

Description

The contract owner has the authority to stop contracts from transactions. The owner may take advantage of it by calling the `F01_Blacklist_Malicious_Account` function.

```
require(!isBlacklisted[from]);
require(!isBlacklisted[to]);
```

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
	L03	Redundant Statements
	L04	Conformance to Solidity Naming Conventions
	L09	Dead Code Elimination
	L13	Divide before Multiply Operation

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L561,564,567,570,577,581,584,588,593,597 and 5 more

Description

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

```
F27__CSO_Change__By_Chief_Security_Officer
F26__CEO_Change__By_Chief_Security_Officer
A14__CSO_Address
A13__CEO_Address
reflectionFromToken
decreaseAllowance
increaseAllowance
transferFrom
approve
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L431,438,439

Description

Constant state variables should be declared constant to save gas.

```
_symbol
_name
_decimals
```

Recommendation

Add the constant attribute to state variables that never change.

L03 - Redundant Statements

Criticality	minor
Location	contract.sol#L717

Description

Detect the usage of redundant statements that have no effect.

```
ZOUTBREAK
```

Recommendation

Remove redundant statements if they congest code but offer no value.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L148,149,150,151,141,143,144,146,246,247 and 106 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.

```
ExceptAccounts
minAmountTokens_ProjectFundingSwap
Project_Funding_Swap_Mode
Public_Trading_Enabled
Blockchain_Managers_Count
Marketing_Managers_Count
Blockchain_Manager
Marketing_Manager
SAT
...
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L116,103,106,109,112,86,96,20,75,78 and 1 more

Description

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

```
addLiquidity
mod
_msgData
sendValue
isContract
functionCallWithValue
functionCall
_functionCallWithValue
...
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L623,741

Description

Performing divisions before multiplications may cause lose of prediction.

```
blockchainSupportBNB = BalanceBNB.div(100).mul(bsf)
marketingBNB = BalanceBNB.div(100).mul(mf)
productDevelopmentBNB = BalanceBNB.div(100).mul(pdf)
amount < balanceOf(uniswapV2Pair).div(10000).mul(pi2)
amount < balanceOf(uniswapV2Pair).div(10000).mul(pi1)
```

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	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-

	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-

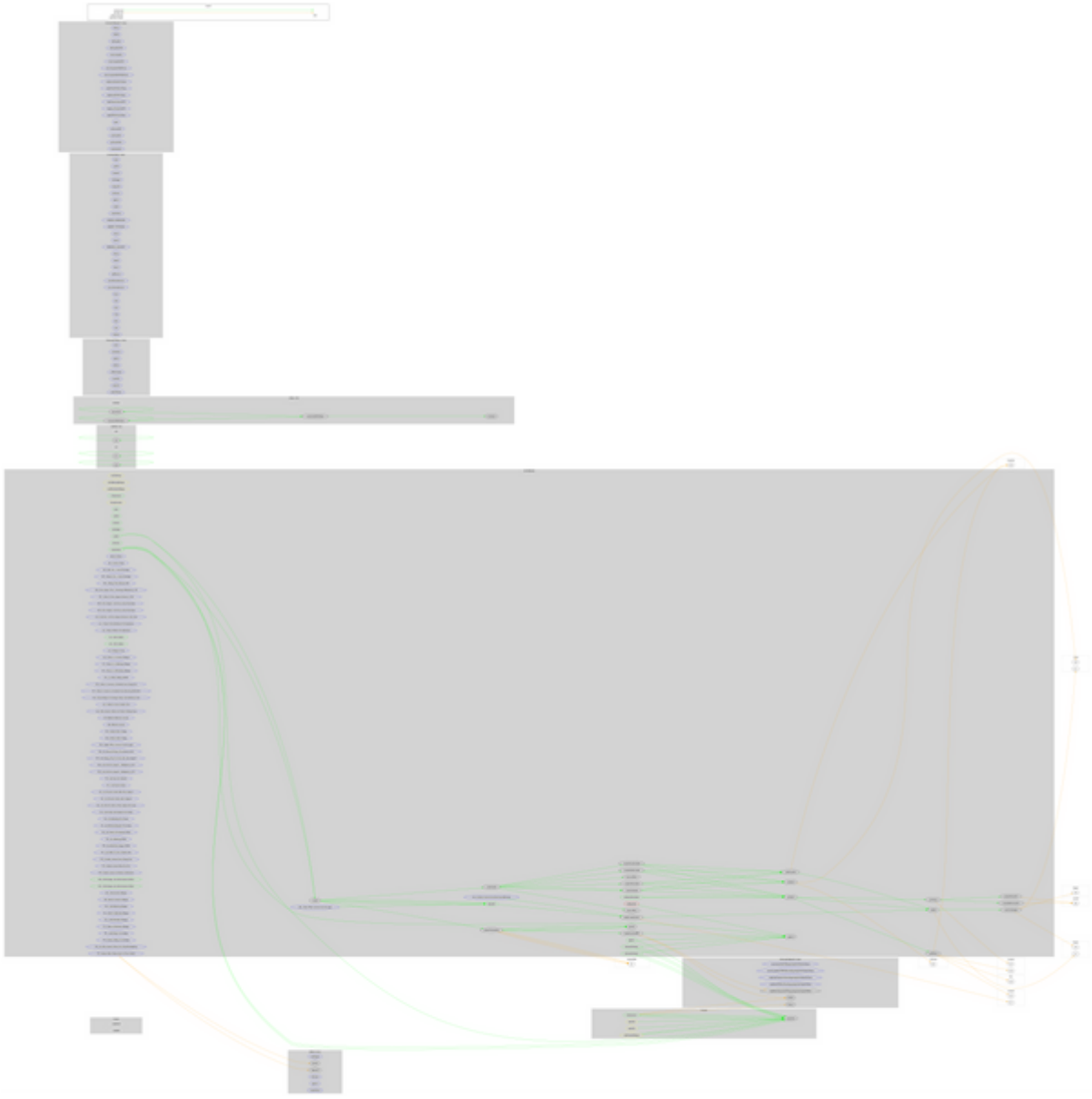
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
ZOUTBREAK	Implementation	Context, 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	✓	-
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	_approve	Private	✓	
	_transfer	Private	✓	
	projectFundingSwap	Private	✓	lockTheSwap
	swapTokensForBNB	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		
	_takeProjectFee	Private	✓	
	calculateReflectionsFee	Private		
	calculateProjectFee	Private		
	removeAllFees	Private	✓	
	restoreAllFees	Private	✓	
	<Receive Ether>	External	Payable	-
	A01__Security_Check	External		-
	A02__Buy_Fees__Actual_Percentage	External		-
	A03__Transfer_Fees__Actual_Percentage	External		-
	A04__Waiting_Time_Between_Sells	External		-
	A05__Check_When_Account_Can_Sell_Again	External		-
	A06__Price_Impact_Tiers__Percentage_Multiplied_by_100	External		-
	A07__Check_If_Price_Impacts_Feature_Is_Used	External		-
	A08__Price_Impact1__Sell_Fees_Actual_Percentage	External		-
	A09__Price_Impact2__Sell_Fees_Actual_Percentage	External		-
	A10__Sell_Fees__If_Price_Impacts_Feature_Is_Not_Used	External		-
	A11__Project_Fee_Distribution_Per_Department	External		-
	A12__Project_Wallets_Per_Department	External		-
	A13__CEO_Address	Public		-
	A14__CSO_Address	Public		-
	A15__Managers_Count	External		-
	A16__Check_if_is_Security_Manager	External		-
	A17__Check_if_is_Marketing_Manager	External		-
	A18__Check_if_is_Blockchain_Manager	External		-

	A19__Is_Public_Trading_Enabled	External		-
	A20__Check_if_Account_is_Excluded_from_Paying_Fees	External		-
	A21__Check_if_Account_is_Excluded_from_Receiving_Reflections	External		-
	A22__Check_Bridge_Or_Exchange_Project_And_Reflection_Fees	External		-
	A23__Wallet_To_Save_Transfer_Fees	External		-
	A24__Min_Amount_Tokens_for_Project_Funding_Swap	External		-
	F01_Blacklist_Malicious_Account	External	✓	ExceptAccounts
	F02_Whitelist_Account	External	✓	onlySecurityManager
	F03__Enable_Public_Trading	External	✓	onlySecurityManager
	F04__Disable_Public_Trading	External	✓	onlySecurityManager
	F05__Update_When_Account_Can_Sell_Again	External	✓	onlySecurityManager
	F06__Set_Normal_Waiting_Time_Between_Sells	External	✓	onlySecurityManager
	F07__Set_Waiting_Time_For_Next_Sell_After_Impact2	External	✓	onlySecurityManager
	F08__Set_Sell_Price_Impact1__Multiplied_by_100	External	✓	onlySecurityManager
	F09__Set_Sell_Price_Impact2__Multiplied_by_100	External	✓	onlySecurityManager
	F10__Set_Fees_For_Transfers	External	✓	onlySecurityManager
	F11__Set_Fees_For_Buys	External	✓	onlySecurityManager
	F12__Set_Fees_For_Sells_With_Price_Impact1	External	✓	onlySecurityManager
	F13__Set_Fees_For_Sells_Above_Impact2	External	✓	onlySecurityManager
	F14__Set_Fees_For_Sells_if_Price_Impacts_Not_Used	External	✓	onlySecurityManager
	F15__Set_Product_Development_Fee_Portion	External	✓	onlyCEO
	F16__Set_Marketing_Fee_Portion	External	✓	onlyMarketingManager

	F17__Set_BlockchainSupport_Fee_Portion	External	✓	onlyBlockchainManager
	F18__Set_Product_Development_Wallet	External	✓	onlyCEO
	F19__Set_Marketing_Wallet	External	✓	onlyMarketingManager
	F20__Set_Blockchain_Support_Wallet	External	✓	onlyBlockchainManager
	F21__Set_Wallet_To_Save_Transfer_Fees	External	✓	onlyCEO
	F22__Exclude_Account_From_Paying_Fees	External	✓	onlySecurityManager
	F23__Enable_Account_Must_Pay_Fees	External	✓	onlySecurityManager
	F24__Exclude_Account_from_Receiving_Reflections	External	✓	onlySecurityManager
	F25__Enable_Account_will_Receive_Reflections	External	✓	onlySecurityManager
	F26__CEO_Change__By_Chief_Security_Officer	Public	✓	onlyCSO
	F27__CSO_Change__By_Chief_Security_Officer	Public	✓	onlyCSO
	F28__Add_Security_Manager	External	✓	onlySecurityManager
	F29__Remove_Security_Manager	External	✓	onlySecurityManager
	F30__Add_Marketing_Manager	External	✓	-
	F31__Remove_Marketing_Manager	External	✓	onlyMarketingManager
	F32__Add_Blockchain_Manager	External	✓	-
	F33__Remove_Blockchain_Manager	External	✓	onlyBlockchainManager
	F34__Add_Bridge_Or_Exchange	External	✓	ExceptAccounts onlySecurityManager
	F35__Remove_Bridge_Or_Exchange	External	✓	onlySecurityManager
	F36__Set_Min_Amount_Tokens_For_ProjectFundingSwap	External	✓	onlySecurityManager
	F37__Rescue_Other_Tokens_Sent_To_This_Contract	External	✓	-

Contract Flow



Domain Info

Domain Name	zoutbreak.com
Registry Domain ID	2660455236_DOMAIN_COM-VRSN
Creation Date	2021-12-09T04:53:16
Updated Date	2021-12-09T05:06:12
Registry Expiry Date	
Registrar WHOIS Server	whois.tucows.com
Registrar URL	http://tucowsdomains.com
Registrar	TUCOWS, INC.
Registrar IANA ID	69

The domain has been created 3 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 manipulating fees, blacklisting contracts and stopping transactions. If the contract configuration is abused by the contract owners, then the contract could operate as a honeypot.

The contract has naming and coding conventions issues. Hence, the functionality of the business logic cannot be guaranteed.

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 provides all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io