Cyberscope

Audit Report

Croba Inu

April 2022

NetWork CRO

Address 0xf0DF6a66432dAa565607D24E10C0d29984E073fe

Audited by © cyberscope

Table of Contents

Contract Review

Contract Name	CrobaInu
Compiler Version	v0.8.9+commit.e5eed63a
Optimization	200 runs
Licence	Unlicense
Explorer	https://cronoscan.com//address/0xf0DF6a66432dAa565607D24E10C0d29984E073fe
Symbol	CROBA
Decimals	2
Total Supply	1,000,000,000
Domain	crobainu.com

Source Files

Filename	**SHA256**
contract.sol	531b0697765265392f759b1896fca9144e62a84cc5dc5f116df7b410a2ad9a79

Audit Updates

Initial Audit	7th April 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	medium
Location	contract.sol#L571

Description

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 `tradingOpen` to false.

```
if (!authorizations[sender] && !authorizations[recipient]) {
  require(tradingOpen, "Trading is not enabled");
```

Recommendation

The contract could prevent the toggle of tradingOpen once it is enabled.

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#L574

Description

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

```
if (lockingEnabled) {
  require(!isLocked[sender] && !isLocked[recipient], "Wallet is locked");
}
```

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
	L04	Conformance to Solidity Naming Conventions
	L07	Missing Events Arithmetic
	L13	Divide before Multiply Operation
	L14	Uninitialized Variables in Local Scope

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L106,110,123,960

Description

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

```
tradingStatus
transferOwnership
unauthorize
authorize
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L380,206

Description

Constant state variables should be declared constant to save gas.

```
dividendsPerShareAccuracyFactor
_totalSupply
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L141,188,870,882,886,927,928,929,936,960,377,378,379,380,382,383,385,386,418,429,437

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.

```
_feeWallets
_multipliers
_fees
_maxHoldings
_maxTxAmount
_allowances
_balances
_totalSupply
_decimals
...
```

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#L230,260,780,791,799,807,922,926,936,960

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.

```
deadBlocks = _deadBlocks
targetLiquidity = _target
swapThreshold = _swapThreshold
distributorGas = gas
_maxTxAmount = amount
_maxTxAmount = (_totalSupply * percent) / base
_maxHoldings = amount
_maxHoldings = (_totalSupply * percent) / base
dividendsPerShare =
dividendsPerShare.add(dividendsPerShareAccuracyFactor.mul(msg.value).div(totalSh
ares))
...
```

Recommendation

Emit an event for critical parameter changes.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L842

Description

Performing divisions before multiplications may cause lose of prediction.

```
fees =
_fees.total.mul(100).div(_fees.divisor).mul(_multipliers.sellMultiplier).div(_mu
ltipliers.divisor)
fees =
_fees.total.mul(100).div(_fees.divisor).mul(_multipliers.buyMultiplier).div(_mul
tipliers.divisor)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L766,639

Description

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

```
feeAmount
i
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

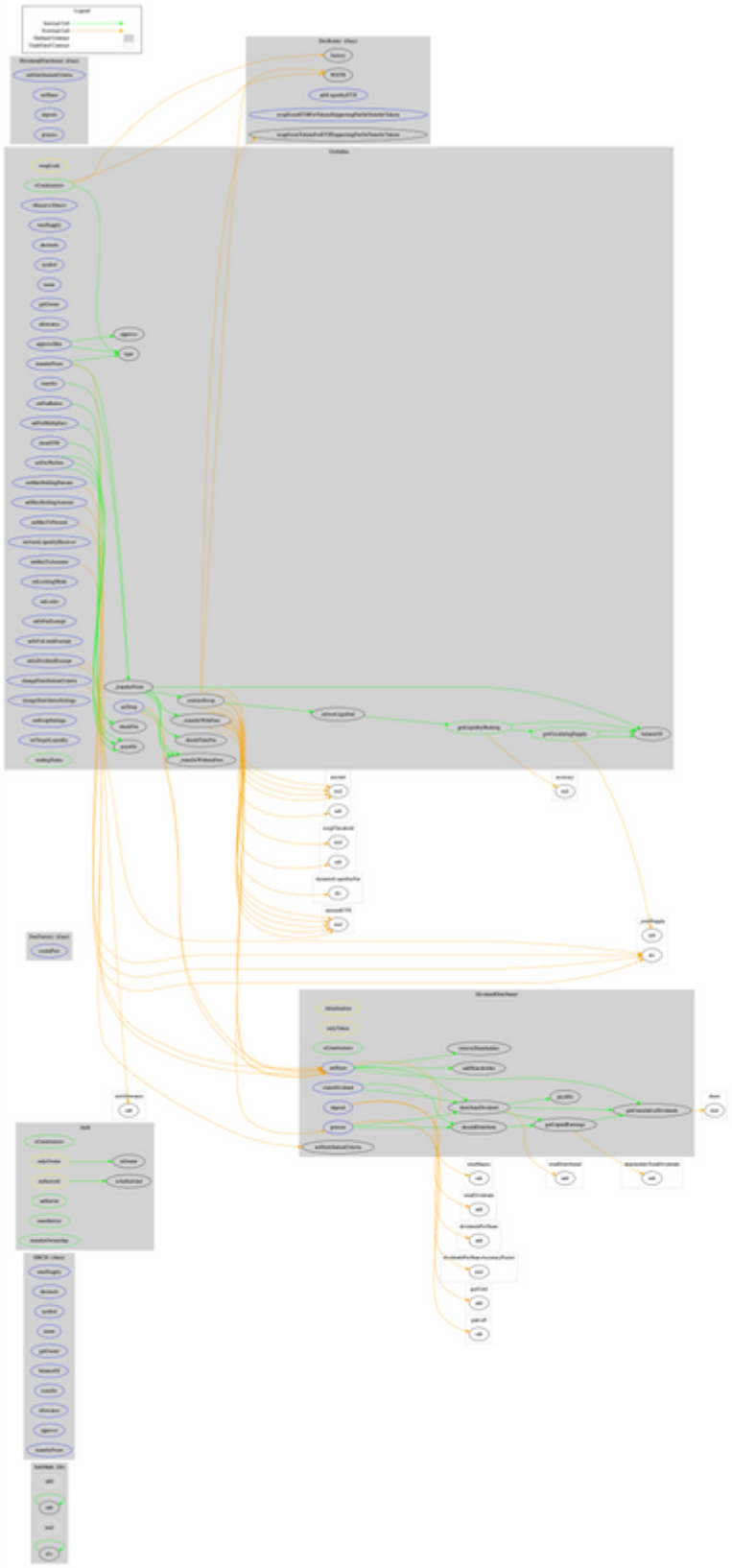
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
ERC20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Auth	Implementation			
	<Constructor>	Public	✓	-
	authorize	Public	✓	onlyOwner
	unauthorize	Public	✓	onlyOwner
	isOwner	Public		-
	isAuthorized	Public		-
	transferOwnership	Public	✓	onlyOwner
DexFactory	Interface			

	createPair	External	✓	-
DexRouter	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IDividendDistributor	Interface			
	setDistributionCriteria	External	✓	-
	setShare	External	✓	-
	deposit	External	Payable	-
	process	External	✓	-
DividendDistributor	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	✓	
CrobaInu	Implementation	ERC20, Auth		
	<Constructor>	Public	✓	Auth
	<Receive Ether>	External	Payable	-

	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	Public		-
	allowance	External		-
	approve	Public	✓	-
	approveMax	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	_transferFrom	Internal	✓	
	shouldTakeFee	Internal		
	_transferWithoutFees	Internal	✓	
	_transferWithFees	Internal	✓	
	contractSwap	Internal	✓	swapLock
	airDrop	External	✓	onlyOwner
	setMaxHoldingPercent	External	✓	onlyOwner
	setMaxHoldingAmount	External	✓	onlyOwner
	setMaxTxPercent	External	✓	onlyOwner
	setMaxTxAmount	External	✓	onlyOwner
	setFeeMultipliers	External	✓	onlyOwner
	setFeeRatios	External	✓	onlyOwner
	checkFee	Internal		
	setFeeWallets	External	✓	onlyOwner
	setAutoLiquidityReceiver	External	✓	onlyOwner
	clearETH	External	✓	authorized
	setLockingMode	External	✓	onlyOwner
	setLocks	External	✓	onlyOwner
	setIsFeeExempt	External	✓	onlyOwner
	setIsTxLimitExempt	External	✓	onlyOwner
	setIsDividendExempt	External	✓	authorized
	changeDistributionCriteria	External	✓	authorized
	changeDistributorSettings	External	✓	authorized
	setSwapSettings	External	✓	onlyOwner

	setTargetLiquidity	External	✓	onlyOwner
	getCirculatingSupply	Public		-
	getLiquidityBacking	Public		-
	isOverLiquified	Public		-
	tradingStatus	Public	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	crobainu.com
Registry Domain ID	2683117909_DOMAIN_COM-VRSN
Creation Date	2022-03-20T22:08:24Z
Updated Date	2022-03-20T22:08:58Z
Registry Expiry Date	2023-03-20T22:08:24Z
Registrar WHOIS Server	whois.porkbun.com
Registrar URL	http://porkbun.com
Registrar	Porkbun LLC
Registrar IANA ID	1861

The domain has been created 18 days before the creation of the audit. It will expire in 12 months.

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

Summary

Token is an interesting project that has a friendly and growing community. There are some functions that can be abused by the owner, like stopping transactions and blacklisting wallets. The maximum fee percentage that can be set in sales and buys is 25%. The contract owner can set 100% fees in the transfers from wallet to 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.

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