Cyberscope

Audit Report

Digitalatto

August 2022

Type	BEP20
Network	BSC
Address	0x0a96ee8b3d59aea26b4cc31342747e176e711fdd
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	DigitalattoCoin
Compiler Version	v0.8.15+commit.e14f2714
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x0a96ee8b3d59aea26b4cc31342747e176e711fdd
Symbol	DGTL
Decimals	9
Total Supply	100,000,000,000
Domain	https://digitalatto.io

Source Files

Filename	SHA256
contract.sol	364d69ede40e01cafbf191ac3560e4c544cb5bd4fccb37e309d2843c890d7de3

Audit Updates

Initial Audit	16th August 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 Diagnostics

🔴 Critical 🟠 Medium ⚪ Minor

Severity	Code	Description
🔴	ZD	Zero Division
⚪	FSA	Fixed Swap Address
⚪	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
⚪	L09	Dead Code Elimination
⚪	L11	Unnecessary Boolean equality

ZD - Zero Division

Criticality	critical
Location	contract.sol#L638

Description

The contract is using variables that may be set to zero as denominators. As a result, the transactions will revert.

The variable `totalFee` can be set to zero.

```
function swapBack() internal swapping {
        uint256 dynamicLiquidityFee = isOverLiquified(targetLiquidity, targetLiquidityDenominator) ?
0 : liquidityFee;
        uint256 amountToLiquify = swapThreshold.mul(dynamicLiquidityFee).div(totalFee).div(2);
```

Recommendation

The contract should prevent those variables to be set to zero or should not allow to execute the corresponding statements.

FSA - Fixed Swap Address

Criticality	minor
Location	contract.sol#L506

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.

```
constructor (
        address _dexRouter
    ) Auth(msg.sender) {
        router = IDEXRouter(_dexRouter);
        pair = IDEXFactory(router.factory()).createPair(WBNB, address(this));
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L186,198,736,182

Description

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

```
authorize
launch
transferOwnership
unauthorize
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L439,440,436,438,291,446,278

Description

Constant state variables should be declared constant to save gas.

```
WBNB
_totalSupply
dividendsPerShareAccuracyFactor
DEAD
BUSD
DEAD_NON_CHECKSUM
ZERO
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L716,765,213,775,780,437,790,450,444,277,442,785,316,440,447,446,449,438,269,278,436,439,443

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.

```
_symbol
_target
_minPeriod
ZERO
_amount
BUSD
WBNB
_token
DEAD
...
```

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#L436,440

Description

There are segments that contain unused state variables.

```
DEAD_NON_CHECKSUM
BUSD
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L316,780,742,765,725,785,716

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.

```
autoBuybackCap = _cap
targetLiquidity = _target
buybackMultiplierNumerator = numerator
liquidityFee = _liquidityFee
_maxTxAmount = amount
swapThreshold = _amount
minPeriod = _minPeriod
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L732

Description

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

```
launched
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L531

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
require(bool,string)(buyBacker[msg.sender] == true,)
```

Recommendation

Remove the equality to the boolean constant.

Contract Functions

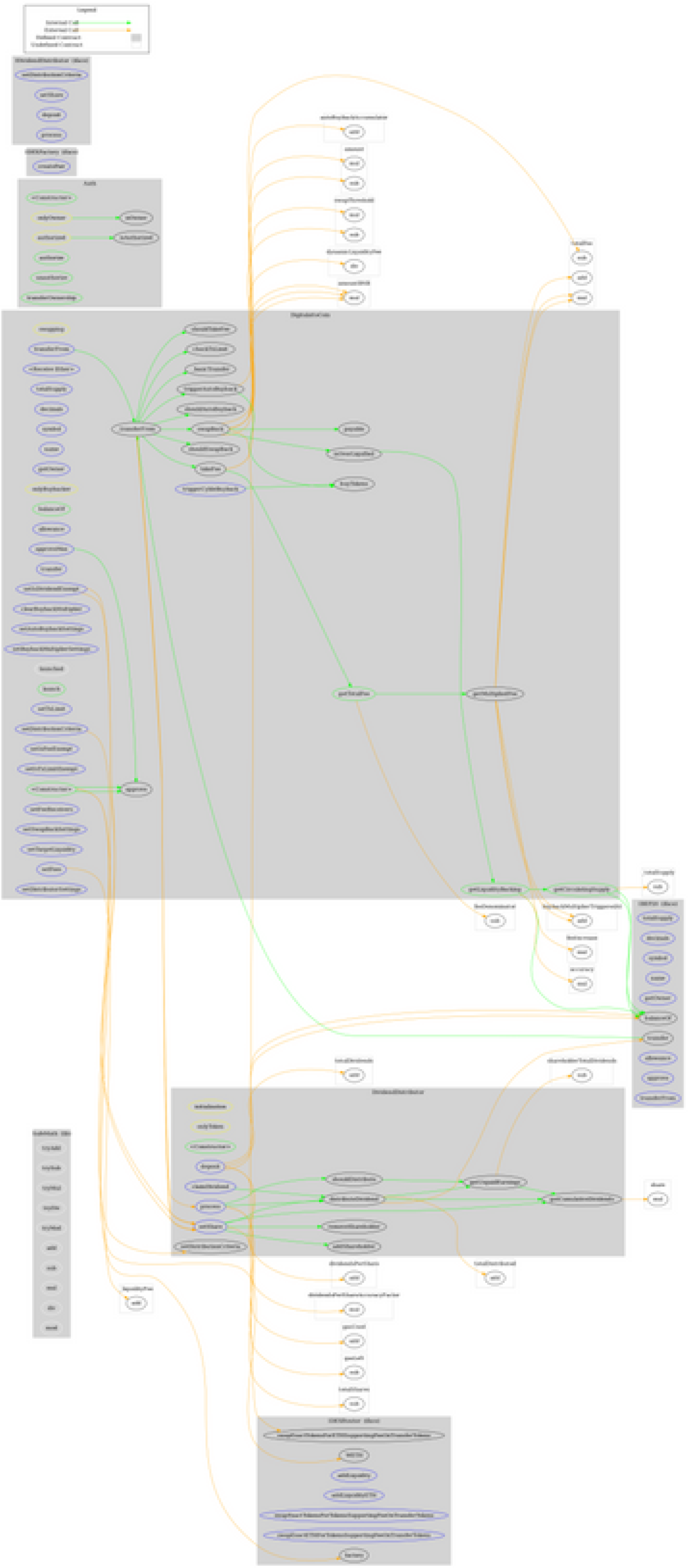
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
SafeMath	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	Internal		
IBEP20	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
IDEXFactory	Interface			
	createPair	External	✓	-
IDEXRouter	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	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	✓	
DigitalattoCoin	Implementation	IBEP20, 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	✓	
	_basicTransfer	Internal	✓	
	checkTxLimit	Internal		
	shouldTakeFee	Internal		
	getTotalFee	Public		-
	getMultipliedFee	Public		-
	takeFee	Internal	✓	
	shouldSwapBack	Internal		
	swapBack	Internal	✓	swapping
	shouldAutoBuyback	Internal		
	triggerCybleBuyback	External	✓	authorized
	clearBuybackMultiplier	External	✓	authorized
	triggerAutoBuyback	Internal	✓	
	buyTokens	Internal	✓	swapping

	setAutoBuybackSettings	External	✓	authorized
	setBuybackMultiplierSettings	External	✓	authorized
	launched	Internal		
	launch	Public	✓	authorized
	setTxLimit	External	✓	authorized
	setIsDividendExempt	External	✓	authorized
	setIsFeeExempt	External	✓	authorized
	setIsTxLimitExempt	External	✓	authorized
	setFees	External	✓	authorized
	setFeeReceivers	External	✓	authorized
	setSwapBackSettings	External	✓	authorized
	setTargetLiquidity	External	✓	authorized
	setDistributionCriteria	External	✓	authorized
	setDistributorSettings	External	✓	authorized
	getCirculatingSupply	Public		-
	getLiquidityBacking	Public		-
	isOverLiquified	Public		-

Contract Flow



Domain Info

Domain Name	digitalatto.io
Registry Domain ID	9de4403757bf42acbc6506bcc6fe2883-DONUTS
Creation Date	2021-10-09T09:12:15Z
Updated Date	2021-11-23T14:13:05Z
Registry Expiry Date	2022-10-09T09:12:15Z
Registrar WHOIS Server	whois.godaddy.com/
Registrar URL	http://www.godaddy.com/domains/search.aspx?ci=8990
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain has been created in about 2 months before the creation of the audit.

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

Summary

Digitalatto Token is an interesting project that has a friendly and growing community.

The Smart Contract analysis reported no compiler error or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users’ transactions.

The buy fees are 14%. The sales fees are normally 14% but may vary because of the auto-buyback feature.

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