Cyberscope

Audit Report

Nato Doge

March 2022

Type	BEP20
Network	BSC
Address	0x3EE6CC9963aa9E62F3Ac9Bd8d02bdB32B969d10d
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	NATODOGE
Compiler Version	v0.6.12+commit.27d51765
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x3EE6CC9963aa9E62F3Ac9Bd8d02bdB32B969d10d
Symbol	NATODOGE
Decimals	9
Total Supply	1,000,000,000,000
Source	contract.sol
Domain	natodoge.site

Audit Updates

Initial Audit	10th March 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

ULTW - Unlimited Liquidity to Team Wallet

Criticality	minor
Location	contract.sol#L1153

Description

The contract owner has the authority to transfer funds to the team wallet. These funds have been accumulated from fees collected from the contract. The owner may take advantage of it by sequentially calling the `manualUnjamSwap` and the `manualUnjamSend` functions.

```
function manualUnjamSwap() external onlyOwner() {
    uint256 contractBalance = balanceOf(address(this));
    swapTokensForEth(contractBalance);
}

function manualUnjamSend() external onlyOwner() {
    uint256 contractETHBalance = address(this).balance;
    sendETHToMarketingPool(contractETHBalance);
}
```

Recommendation

The contract could embody a check for a reasonable maximum acceptable number of tokens that can be liquified.

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#L929

Description

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

```
function RemoveSniper(address account) external onlyOwner() {
    require(account != 0x10ED43C718714eb63d5aA57B78B54704E256024E, 'We can not blacklist Uniswap router.');
    require(!_isSniper[account], "Account is already blacklisted");
    _isSniper[account] = true;
    _confirmedSnipers.push(account);
}
```

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
	FSA	Fixed Swap Address
	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
	L08	Tautology or Contradiction

FSA - Fixed Swap Address

Criticality	minor
Location	contract.sol#L1

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.

```
here
```

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#L403,412,418,862,866,870,874,878,887,892 and 10 more

Description

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

```
_getETHBalance
isExcludedFromFee
reflectionFromToken
deliver
totalFees
isExcluded
decreaseAllowance
increaseAllowance
transferFrom
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L650,680,648,682,649,644,678,676,677,368 and 1 more

Description

Constant state variables should be declared constant to save gas.

```
_previousOwner
_lockTime
uniswapOnly
tradingOpen
launchTime
_tTotal
_symbol
_numOfTokensToExchangeForMarketingPool
_name
...
```

Recommendation

Add the constant attribute to state variables that never change.

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L367

Description

There are segments that contain unused state variables.

```
_previousOwner
```

Recommendation

Remove unused state variables.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L455,456,473,493,929,1325,1329,1334,1339,1343 and 9 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.

```
_StakePoolWalletAddress
_MarketingPoolWalletAddress
_MarketingPoolFee
StakePoolAddress
_setStakePoolAddress
MarketingPoolWalletAddress
_setMarketingPoolWallet
_setsellMarketingPoolFee
_setbuyMarketingPoolFee
...
```

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#L341,301,311,326,336,248,275,1321,1317

Description

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

```
_getTaxFee
_getMaxTxAmount
sendValue
isContract
functionCallWithValue
functionCall
_functionCallWithValue
```

Recommendation

Remove unused functions.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1329,1334,1339,1343,1348,1353

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.

```
_sellMarketingPoolFee = sellMarketingPoolFee
_buyMarketingPoolFee = buyMarketingPoolFee
_sellstaketaxFee = sellstakeFee
_buystaketaxFee = buystakeFee
_selltaxFee = selltaxFee
_buytaxFee = buytaxFee
```

Recommendation

Emit an event for critical parameter changes.

L08 - Tautology or Contradiction

Criticality	minor
Location	contract.sol#L1329,1334,1339,1343,1348,1353

Description

Detects expressions that are tautologies or contradictions. For instance, an uint variable will always be greater than or equal to zero.

```
require(bool,string)(sellMarketingPoolFee >= 0 && sellMarketingPoolFee <=
9,MarketingPoolFee should be in 0 - 9)
require(bool,string)(buyMarketingPoolFee >= 0 && buyMarketingPoolFee <=
9,MarketingPoolFee should be in 0 - 9)
require(bool,string)(sellstakeFee >= 0 && sellstakeFee <= 5,stakeFee should be
in 0 - 5)
require(bool,string)(buystakeFee >= 0 && buystakeFee <= 5,stakeFee should be in
0 - 5)
require(bool,string)(selltaxFee >= 0 && selltaxFee <= 9,taxFee should be in 0 -
9)
require(bool,string)(buytaxFee >= 0 && buytaxFee <= 9,taxFee should be in 0 - 9)
```

Recommendation

Fix the incorrect comparison by changing the value type or the comparison.

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>	Internal	✓	
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	geUnlockTime	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		-

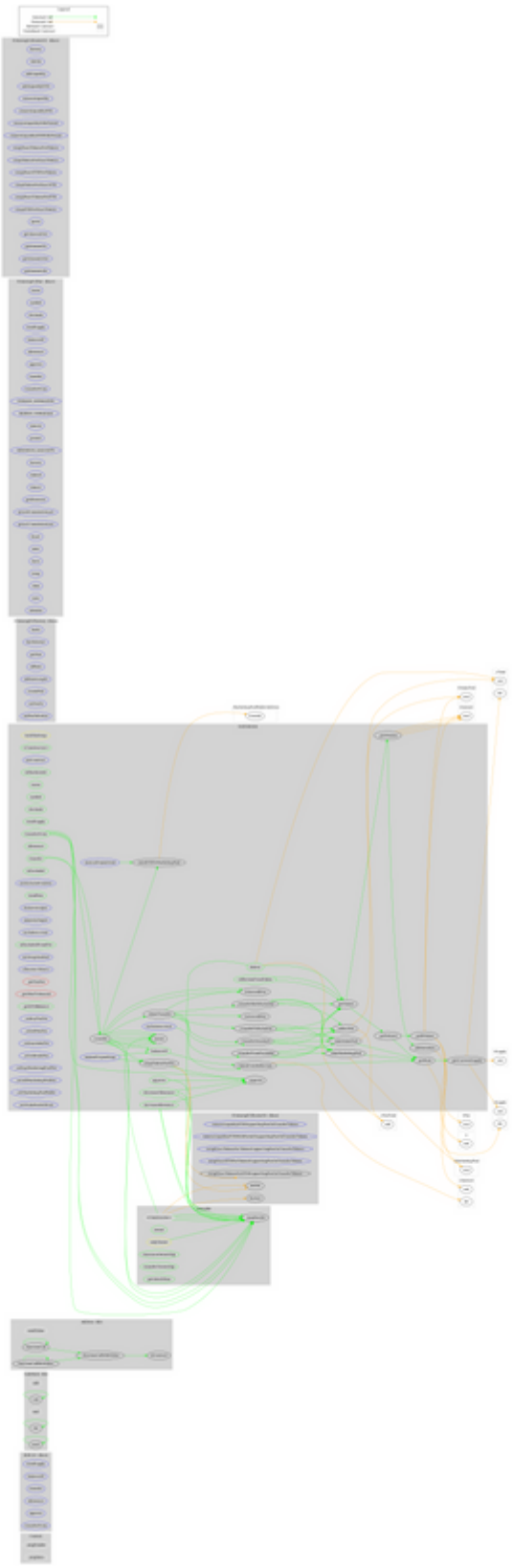
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IUniswapV2Router01	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	✓	-
NATODOGE	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	initContract	External	✓	onlyOwner
	isBlackListed	Public		-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	isExcluded	Public		-
	setExcludeFromFee	External	✓	onlyOwner
	totalFees	Public		-
	RemoveSniper	External	✓	onlyOwner
	amnestySniper	External	✓	onlyOwner
	deliver	Public	✓	-
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-

	excludeAccount	External	✓	onlyOwner
	includeAccount	External	✓	onlyOwner
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	isExcludedFromFee	Public		-
	_approve	Private	✓	
	_transfer	Private	✓	
	swapTokensForEth	Private	✓	lockTheSwap
	sendETHToMarketingPool	Private	✓	
	manualUnjamSwap	External	✓	onlyOwner
	manualUnjamSend	External	✓	onlyOwner
	setSwapEnabled	External	✓	onlyOwner
	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	
	_transferBothExcluded	Private	✓	
	_takeMarketingPool	Private	✓	
	_takeStakePool	Private	✓	
	_reflectFee	Private	✓	
	<Receive Ether>	External	Payable	-
	_getValues	Private		
	_getValues2	Private		
	_getTValues	Private		
	_addonvalues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_getTaxFee	Private		
	_getMaxTxAmount	Private		
	_getETHBalance	Public		-
	_setbuyTaxFee	External	✓	onlyOwner
	_setsellTaxFee	External	✓	onlyOwner
	_setbuystakeFee	External	✓	onlyOwner
	_setsellstakeFee	External	✓	onlyOwner

	_setbuyMarketingPoolFee	External	✓	onlyOwner
	_setsellMarketingPoolFee	External	✓	onlyOwner
	_setMarketingPoolWallet	External	✓	onlyOwner
	_setStakePoolAddress	External	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	natodoge.site
Registry Domain ID	D280387327-CNIC
Creation Date	2022-03-07T16:00:44+00:00
Updated Date	2022-03-07T16:41:17+00:00
Registry Expiry Date	2023-03-07T23:59:59+00:00
Registrar WHOIS Server	whois.godaddy.com
Registrar URL	https://www.godaddy.com/
Registrar	Go Daddy, LLC
Registrar IANA ID	146

The domain has been created 3 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

There are some functions that can be abused by the owner, like blacklisting contracts and transferring funds to the team’s wallet. The maximum fee percentage that can be set is 23% both for buys and sales. 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