Cyberscope

Audit Report

ElonDogeInu

March 2022

Type	BEP20
Network	BSC
Address	0x45930a2919B9d814b8DB0fC91ebE4F5d9B9985b3
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	ElonDogeInu
Compiler Version	v0.7.4+commit.3f05b770
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0x45930a2919B9d814b8DB0fC91ebE4F5d9B9985b3
Symbol	EDInu
Decimals	4
Total Supply	1,000,000,000,000,000
Source	contract.sol
Domain	elondogeinu.org

Audit Updates

Initial Audit	23rd 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

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L861,883,966

Description

The contract owner has the authority to stop the sales for all users excluding the owner. The owner may take advantage of it by setting the `cooldownTimerInterval` or sellMultiplier or cooldownTimerInterval to a high value. These can cause the contract to operate as a honeypot.

```
uint256 feeAmount =
  rAmount.div(feeDenominator * 100).mul(totalFee).mul(multiplier);

if (!isSell && (launchedAt + deadBlocks) > block.number) {
  feeAmount = rAmount.div(100).mul(99);
}
```

```
if (sender == pair && buyCooldownEnabled && !isTimelockExempt[recipient]) {
  require(
    cooldownTimer[recipient] < block.timestamp,
    "buy Cooldown exists"
  );
  cooldownTimer[recipient] = block.timestamp + cooldownTimerInterval;
}
```

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 not open yet");
}
```

Recommendation

The contract could embody a check for not allowing setting the _maxTxAmount less than a reasonable amount. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply.

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.

OTUT - Owner Transfer User's Tokens

Criticality	critical
Location	contract.sol#L1259,1296

Description

The contract owner has the authority to transfer the balance of a user's contract to other contracts. The owner may take advantage of it by calling the `multiTransfer_fixed` function.

```
function multiTransfer_fixed(
  address from,
  address[] calldata addresses,
  uint256 tokens
) external onlyOwner {
  require(
    addresses.length < 2001,
    "GAS Error: max airdrop limit is 2000 addresses"
  ); // to prevent overflow

  uint256 SCCC = tokens * addresses.length;

  require(balanceOf(from) >= SCCC, "Not enough tokens in wallet");

  for (uint256 i = 0; i < addresses.length; i++) {
    _basicTransfer(from, addresses[i], tokens);
    if (!isDividendExempt[addresses[i]]) {
      try
        distributor.setShare(addresses[i], balanceOf(addresses[i]))
      {} catch {}
    }
  }

  // Dividend tracker
  if (!isDividendExempt[from]) {
    try distributor.setShare(from, balanceOf(from)) {} catch {}
  }
}
```

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.

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L1001

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 `set_sell_multiplier` function with a high percentage value.

```
function set_sell_multiplier(uint256 Multiplier) external onlyOwner {
  sellMultiplier = Multiplier;
}
```

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#L892

Description

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

```
if (blacklistMode) {
  require(
    !isBlacklisted[sender] && !isBlacklisted[recipient],
    "Blacklisted"
  );
}
```

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
Medium	MTS	Manipulate Total Supply
Minor	L01	Public Function could be Declared External
Minor	L02	State Variables could be Declared Constant
Minor	L04	Conformance to Solidity Naming Conventions
Minor	L05	Unused State Variable
Minor	L06	Missing Events Access Control
Minor	L07	Missing Events Arithmetic
Minor	L09	Dead Code Elimination
Minor	L13	Divide before Multiply Operation
Minor	L14	Uninitialized Variables in Local Scope

MTS - Manipulate Total Supply

Criticality	medium
Location	contract.sol#L635,686

Description

Owner is able to manipulate total supply. This change will have a direct impact on the token price and Market Cap.

The contract should also sync the pair contract every time that the total supply is manipulated.

```
if (supplyDelta < 0) {
  _totalSupply = _totalSupply.sub(uint256(-supplyDelta));
} else {
  _totalSupply = _totalSupply.add(uint256(supplyDelta));
}

if (_totalSupply > MAX_SUPPLY) {
  _totalSupply = MAX_SUPPLY;
}

rate = rSupply.div(_totalSupply);
```

Recommendation

The contract owner should carefully manage the adjustment of the circulating supply (increases or decreases), according to the token’s price fluctuations.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L144,148,160,547,558,588,593,841,1011,1019 and 6 more

Description

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

```
isOverLiquified
rescueBNB
rescueToken
cooldownEnabled
manage_blacklist
enable_blacklist
launchStatus
tradingStatus
setBeforeRebase
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L264,277,462,461,463

Description

Constant state variables should be declared constant to save gas.

```
ZERO
WBNB
DEAD
dividendsPerShareAccuracyFactor
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L182,303,255,263,264,547,558,588,593,686 and 54 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.

```
_maxWalletToken
_maxTxAmount
rSupply
_totalSupply
rebase_count
LPStatus
_isBot
_allowances
_rBalance
...
```

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#L56

Description

There are segments that contain unused state variables.

```
MAX_INT256
```

Recommendation

Remove unused state variables.

L06 - Missing Events Access Control

Criticality	minor
Location	contract.sol#L1181

Description

Detected missing events for critical access control parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
master = _master
```

Recommendation

Emit an event for critical parameter changes.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L303,1001,1005,1011,1019,1128,1156,1243,1250

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.

```
_maxTxAmount = rSupply.div(1000).mul(maxTXPercentage_base1000)
_maxWalletToken = rSupply.div(1000).mul(maxWallPercent_base1000)
swapThreshold = rSupply.div(10000).mul(_percentage_base10000)
liquidityFee = _liquidityFee
launchedAt = _launchblock
deadBlocks = _deadBlocks
swapMultiplier = Multiplier
sellMultiplier = Multiplier
minPeriod = _minPeriod
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L84

Description

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

```
abs
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L547,558,955,1041,1156,1243,1250,457 and 1 more

Description

Performing divisions before multiplications may cause lose of prediction.

```
_maxWalletToken = rSupply.div(1000).mul(50)
_maxTxAmount = rSupply.div(1000).mul(10)
_maxTxAmount = rSupply.div(1000).mul(maxTXPercentage_base1000)
_maxWalletToken = rSupply.div(1000).mul(maxWallPercent_base1000)
swapThreshold = rSupply.div(10000).mul(_percentage_base10000)
amountToLiquify = tokensToSell.div(totalFee).mul(dynamicLiquidityFee).div(2)
feeAmount = rAmount.div(100).mul(99)
feeAmount = rAmount.div(feeDenominator * 100).mul(totalFee).mul(multiplier)
newSupply =
rebase1000(0,int256(_totalSupply.div(1000).mul(_percentage_base1000)),coinAmount
)
...
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L1031

Description

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

```
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		
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	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	✓	-

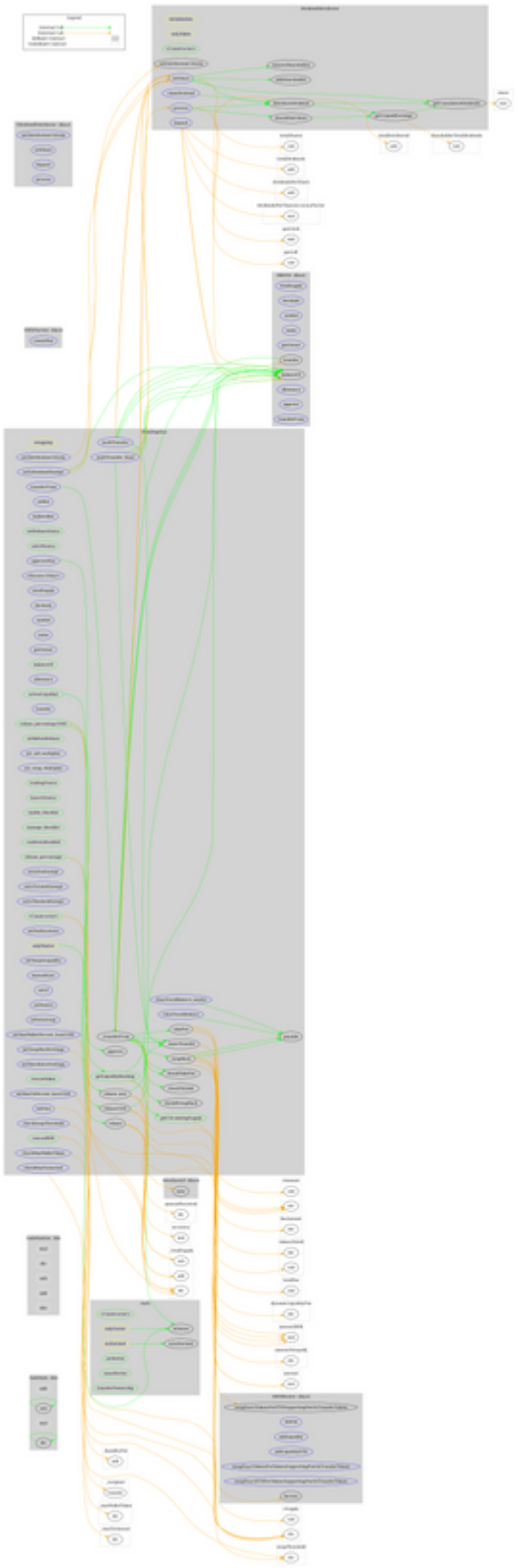
	authorize	Public	✓	onlyOwner
	unauthorize	Public	✓	onlyOwner
	isOwner	Public		-
	isAuthorized	Public		-
	transferOwnership	Public	✓	onlyOwner
IDEXFactory	Interface			
	createPair	External	✓	-
InterfaceLP	Interface			
	sync	External	✓	-
IDEXRouter	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	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	✓	
ElonDogeInu	Implementation	IBEP20, Auth		
	rebase_percentage	Public	✓	onlyMaster
	rebase_percentage1000	Public	Payable	onlyMaster
	setBot	External	✓	onlyOwner
	bulkSetBot	External	✓	onlyOwner
	setRebaseStatus	Public	✓	onlyOwner
	setLPStatus	Public	✓	onlyOwner
	rebase	Public	✓	onlyMaster
	rebase1000	Public	Payable	onlyMaster
	rebase_new	Public	Payable	onlyMaster
	<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	✓	-
	setBeforeRebase	Public	✓	onlyOwner
	_transferFrom	Internal	✓	
	_basicTransfer	Internal	✓	

	checkTxLimit	Internal		
	shouldTakeFee	Internal		
	takeFee	Internal	✓	
	shouldSwapBack	Internal		
	clearStuckBalance	External	✓	authorized
	clearStuckBalance_sender	External	✓	onlyOwner
	set_sell_multiplier	External	✓	onlyOwner
	set_swap_Multiplier	External	✓	onlyOwner
	tradingStatus	Public	✓	onlyOwner
	launchStatus	Public	✓	onlyOwner
	enable_blacklist	Public	✓	onlyOwner
	manage_blacklist	Public	✓	onlyOwner
	cooldownEnabled	Public	✓	onlyOwner
	swapBack	Internal	✓	swapping
	setIsDividendExempt	External	✓	authorized
	setIsFeeExempt	External	✓	authorized
	setIsTxLimitExempt	External	✓	authorized
	setIsTimelockExempt	External	✓	authorized
	setFees	External	✓	authorized
	setFeeReceivers	External	✓	authorized
	setSwapBackSettings	External	✓	authorized
	setTargetLiquidity	External	✓	authorized
	manualSync	External	✓	-
	setLP	External	✓	onlyOwner
	setMaster	External	✓	onlyOwner
	isNotInSwap	External		-
	checkSwapThreshold	External		-
	setDistributionCriteria	External	✓	authorized
	setDistributorSettings	External	✓	authorized
	rescueToken	Public	✓	onlyOwner
	rescueBNB	Public	✓	onlyOwner
	getCirculatingSupply	Public		-
	getLiquidityBacking	Public		-
	isOverLiquified	Public		-
	checkMaxWalletToken	External		-

	checkMaxTxAmount	External		-
	setMaxWalletPercent_base1000	External	✓	onlyOwner
	setMaxTxPercent_base1000	External	✓	onlyOwner
	multiTransfer	External	✓	onlyOwner
	multiTransfer_fixed	External	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	elondogeinu.org
Registry Domain ID	D402200000019230419-LROR
Creation Date	2022-03-03T15:09:08Z
Updated Date	2022-03-03T15:09:09Z
Registry Expiry Date	2023-03-03T15:09:08Z
Registrar WHOIS Server	whois.tucows.com
Registrar URL	http://www.tucows.com
Registrar	Tucows Domains Inc.
Registrar IANA ID	69

The domain has been created 20 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 manipulating fees, stopping transactions, blacklisting addresses and transferring user’s tokens. The contract can be converted into a honeypot and prevent users from selling if the owner abuses the admin functions. 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