Cyberscope

Audit Report

Leave Earth Plan

March 2022

Type	BEP20
Network	BSC
Address	0x8eebe881c846dfd259560cbcf28e84a7529bada8
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	LEP
Compiler Version	v0.6.12+commit.27d51765
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0x8eebe881c846dfd259560cbcf28e84a7529bada8
Symbol	LEP
Decimals	9
Total Supply	18,888
Source	contract.sol
Domain	lep-token.com

Audit Updates

Initial Audit	1st 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	medium
Location	contract.sol#L863

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 `maxTxAmount` to zero or by setting the openBool to false.

```
      if(!openBool && from!=owner() && to!=owner() && !openUser[from] &&
!openUser[to]){
            require(false,"open is not");
        }

      if(!swappingTwo&& from != owner() && to != owner()&&from==uniswapV2Pair&&
!isbig[to]){
             require(balanceOf(to).add(amount) <= _maxTxAmount, "Transfer amount
exceeds the maxTxAmount.");
```

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.

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L1062,1057,1067

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 `setMarketingFee` function with a high percentage value.

```
    function setMarketingFee(uint256 value) external onlyOwner{
        marketingFee = value;
        totalFees = liquidityFee.add(marketingFee).add(backFee);
    }
```

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#L953

Description

The contract owner has the authority to stop contracts from transactions. This happens automatically after a user makes a transaction, he gets added into a book list that can no longer sell his tokens anymore. This is the behaviour of a honey pot and is a very malicious functionality.

```
 function checkAdd(address sender,address recipient,uint256 amount) internal{
                  if(amount>balanceOf(sender).div(1000)){
            require(!isbook[sender] && !isbook[recipient], '_listed address');
          }
          if(openBool){
                                  if(launchedAt + lunchTime >= block.number){
    if(recipient!=uniswapV2Pair){
         isbook[recipient] = true;
    }
}}}
```

Recommendation

The contract owner could set the lunchTime to a very small value and then renounce ownership. By doing that, the if condition will never be met and the malicious auto-add to blacklist functionality will not be triggered anymore.

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
	L09	Dead Code Elimination
	L07	Missing Events Arithmetic

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L158,167,563,571,576,583,602,610,621,639 and 15 more

Description

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

```
BigSellAmount
copyMapData
getAmountBNB
getBNB
seteddToken
setswapTokensAtAmount
setisSellAddress
setsellBnbAmount
setsellCount
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L815,826,812,843,838

Description

Constant state variables should be declared constant to save gas.

```
owe
ererer
deadWallet
backFee
USDT
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L356,357,373,395,535,539,901,1140,815

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.

```
USDT
BigSellAmount
USDCRewardsFee
_totalSupply
_balances
WETH
MUM_LIQUIDITY
PERMIT_TYPEHASH
DOMAIN_SEPARATOR
```

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#L29,967,297,313

Description

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

```
mod
launched
_msgData
```

Recommendation

Remove unused functions.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L875,901,929,1057,1062,1067

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.

```
marketingFee = value
liquidityFee = value
swapTokensAtAmount = amount
txone = value
lunchTime = amount
```

Recommendation

Emit an event for critical parameter changes.

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	✓	-
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		

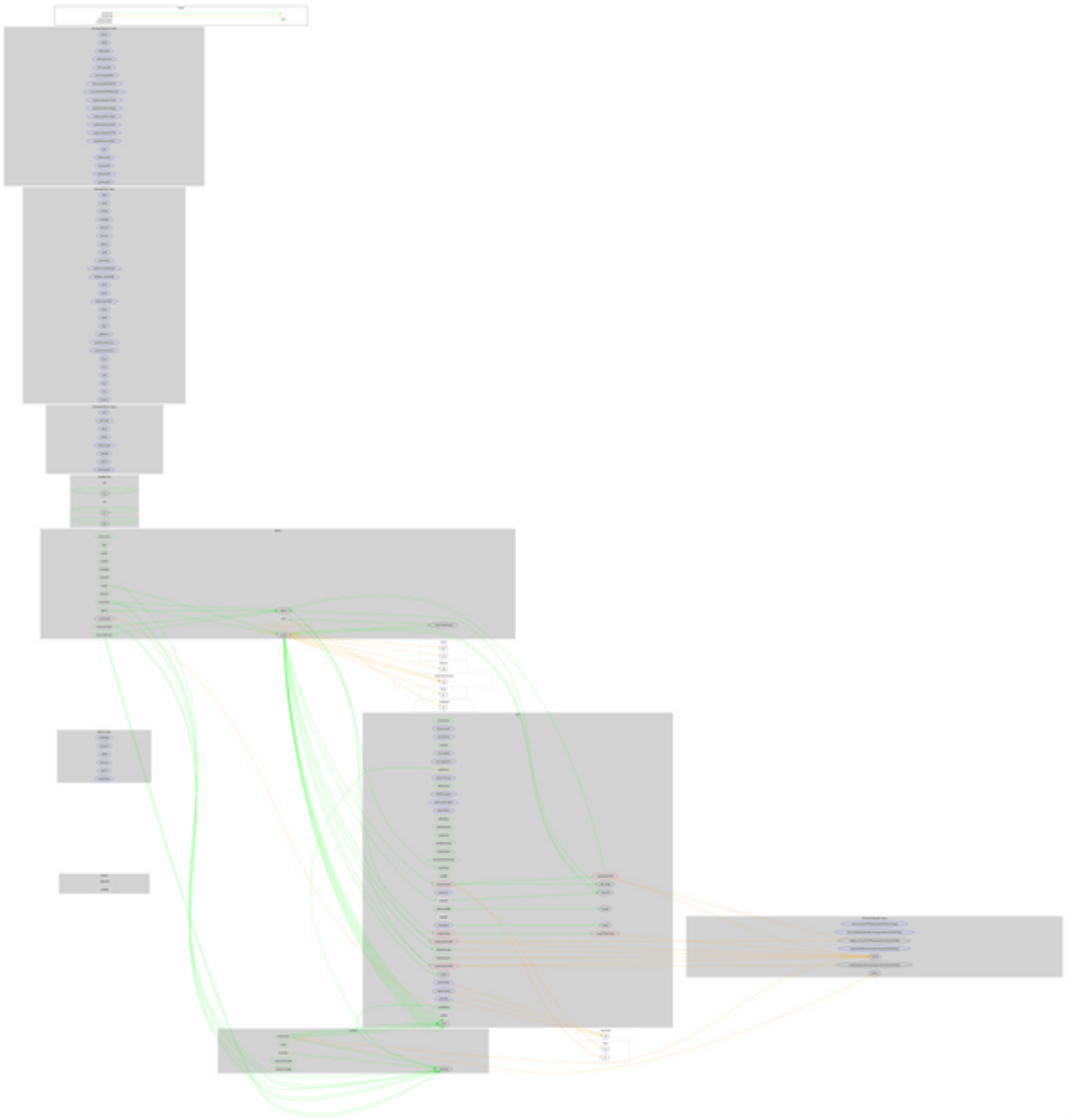
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	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	✓	-
ERC20	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	✓	-
	_transferfather	Internal	✓	
	_transfer	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
LEP	Implementation	ERC20		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	setLunchTime	External	✓	onlyOwner
	setmarket	Public	✓	onlyOwner
	setLswapping	External	✓	onlyOwner
	setLswappingTwo	External	✓	onlyOwner
	setMaxTxPercent	External	✓	pauseOwner
	setBigNumber	Public	✓	onlyOwner
	USDCRewardsFee	External	✓	onlyOwner
	setExcludedFromFees	External	✓	onlyOwner
	isbookAddress	External	✓	onlyOwner

	setBackBool	Public	✓	pauseOwner
	setisBoolNumber	Public	✓	pauseOwner
	setsellCount	Public	✓	pauseOwner
	setsellBnbAmount	Public	✓	pauseOwner
	setisSellAddress	Public	✓	pauseOwner
	setswapTokensAtAmount	Public	✓	onlyOwner
	seteddToken	Public	✓	onlyOwner
	getBNB	Public	Payable	pauseOwner
	getAmountBNB	Public	Payable	pauseOwner
	setopenUser	External	✓	onlyOwner
	setopenBool	External	✓	onlyOwner
	checkAdd	Internal	✓	
	launched	Internal		
	launch	Internal	✓	
	copyMapData	Public	✓	-
	swapTokensForUSDC	Private	✓	
	swapTokensForUSDD	Private	✓	
	swapTokensForEth	Private	✓	
	setLiquiditFee	External	✓	onlyOwner
	setMarketingFee	External	✓	onlyOwner
	setBackFee	External	✓	onlyOwner
	swapAndLiquify	Private	✓	
	addLiquidity	Private	✓	
	swapETHForTokens	Private	✓	
	tryBool	Public		-
	BigSellAmount	Public		-
	isBigSellAmount	Public		-
	buyBackTokens	Private	✓	
	_transfer	Internal	✓	

Contract Flow



Domain Info

Domain Name	lep-token.com
Registry Domain ID	2677586538_DOMAIN_COM-VRSN
Creation Date	2022-02-25T10:05:24Z
Updated Date	2022-02-25T10:05:24Z
Registry Expiry Date	2023-02-25T10:05:24Z
Registrar WHOIS Server	whois.godaddy.com
Registrar URL	http://www.godaddy.com
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain has been created 4 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

The Smart Contract analysis reported no compiler errors but 3 very severe issues. There are some functions that can be abused by the owner, like manipulating fees up to 100% and stopping the trades for everyone else apart from the owner. Most importantly the contract is automatically operating as a honeypot, every time a user buys he gets added into a blacklist that he can no longer sell his tokens. The contract is unsafe for the users and should be fixed before any liquidity is added.

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