Cyberscope

Audit Report

Matic Launchpad

March 2022

Type	BEP20
Network	BSC
Address	0x1E7e0EFb87e609b87F12F1cEa1DAC48569dA2328
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	MaticLaunchpad
Compiler Version	v0.6.12+commit.27d51765
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0x1E7e0EFb87e609b87F12F1cEa1DAC48569dA2328
Symbol	MATICPAD
Decimals	18
Total Supply	5,000,000,000
Source	contract.sol
Domain	

Audit Updates

Initial Audit	13th 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#L1087

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.

```
if(from != owner() && to != owner())
    require(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#L956,960

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 `setTaxFeePercent` or `setLiquidityFeePercent` function with a high percentage value.

```
function setTaxFeePercent(uint256 taxFee) external onlyOwner() {
    _taxFee = taxFee;
}

function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner() {
    _liquidityFee = liquidityFee;
}
```

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.

IF - Incomplete Functionality

Criticality	minor
Location	contract.sol#L845,876

Description

The contract contains a code segment with incomplete functionality. The function accepts payment without tracking the senter's deposit.

```
  function Participate() public payable returns (bool success){
    //address _refer
    //require(sSBlock <= block.number && block.number <= sEBlock);
    //require(sTot < sCap || sCap == 0);
    uint256 _eth = msg.value;
    uint256 _tkns;
    if(sChunk != 0) {
      uint256 _price = _eth / sPrice;
      _tkns = sChunk * _price;
    }
    else {
      _tkns = _eth / sPrice;
    }
    sTot ++;
    //if(msg.sender != _refer && balanceOf(_refer) != 0 && _refer !=
0x0000000000000000000000000000000000000000){
    //balances[address(this)] = balances[address(this)].sub(_tkns / 10);
    //balances[_refer] = balances[_refer].add(_tkns / 10);
    // emit Transfer(address(this), _refer, _tkns / 10);
    //}
    //balances[address(this)] = balances[address(this)].sub(_tkns);
    //balances[msg.sender] = balances[msg.sender].add(_tkns);
    //emit Transfer(address(this), msg.sender, _tkns);
    //emit Transfer(address(this), msg.sender, _tkns);

    return true;
   }

    function startSale(uint _sSUTCDateTime, uint _sEUTCDateTime, uint256
_sChunk, uint256 _sPrice, uint256 _sCap) public onlyOwner() {
    sSUTCDateTime = _sSUTCDateTime;
    sEUTCDateTime = _sEUTCDateTime;
    sChunk = _sChunk;
    sPrice =_sPrice;
```

```
    sCap = _sCap;
    sTot = 0;
  }
```

Recommendation

The contract could either remove the incomplete functionality or implement the entire functionality of participation deposit.

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
	L15	Local Scope Variable Shadowing
	L13	Divide before Multiply Operation

FSA - Fixed Swap Address

Criticality	minor
Location	contract.sol#L758

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.

```
IPancakeswapV2Router02 _pancakeRouter =
IPancakeswapV2Router02(0x05fF2B0DB69458A0750badebc4f9e13aDd608C7F);
// Create a Pancake pair for this new token
pancakeswapV2Pair = IPancakeswapV2Factory(_pancakeRouter.factory())
    .createPair(address(this), _pancakeRouter.WETH());

// set the rest of the contract variables
pancakeswapV2Router = _pancakeRouter;
```

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#L436,445,451,456,464,777,781,785,789,798 and 21 more

Description

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

```
isIncludedFromFee
isExcludedFromFee
releasePrivateSalecoinStatusUpdate
setSwapAndLiquifyEnabled
includeInFee
excludeFromFee
excludeFromReward
reflectionFromToken
deliver
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L713,711,712,737

Description

Constant state variables should be declared constant to save gas.

```
numTokensSellToAddToLiquidity
_symbol
_name
_decimals
```

Recommendation

Add the constant attribute to state variables that never change.

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L691

Description

There are segments that contain unused state variables.

```
balances
```

Recommendation

Remove unused state variables.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L508,509,526,548,741,840,845,872,876,970 and 10 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.

```
_maxTxAmount
_liquidityFee
_taxFee
_amount
_enabled
_sCap
_sPrice
_sChunk
_sEUTCDateTime
...
```

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#L362,322,332,347,357,269,296

Description

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

```
sendValue
isContract
functionCallWithValue
functionCall
_functionCallWithValue
```

Recommendation

Remove unused functions.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L876,956,960,964

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 = _tTotal.mul(maxTxPercent).div(10 ** 2)
_liquidityFee = liquidityFee
_taxFee = taxFee
sSUTCDateTime = _sSUTCDateTime
```

Recommendation

Emit an event for critical parameter changes.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L841

Description

The are variables that are defined in the local scope containing the same name from an upper scope.

```
_owner
```

Recommendation

The local variables should have different names from the upper scoped variables.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L845

Description

Performing divisions before multiplications may cause lose of prediction.

```
_price = _eth / sPrice
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IBEP20	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		
Context	Implementation			
	_msgSender	Internal		
	_msgData	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		-
	lock	Public	✓	onlyOwner
	unlock	Public	✓	-
IPancakeswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IPancakeswapV2Pair	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		-

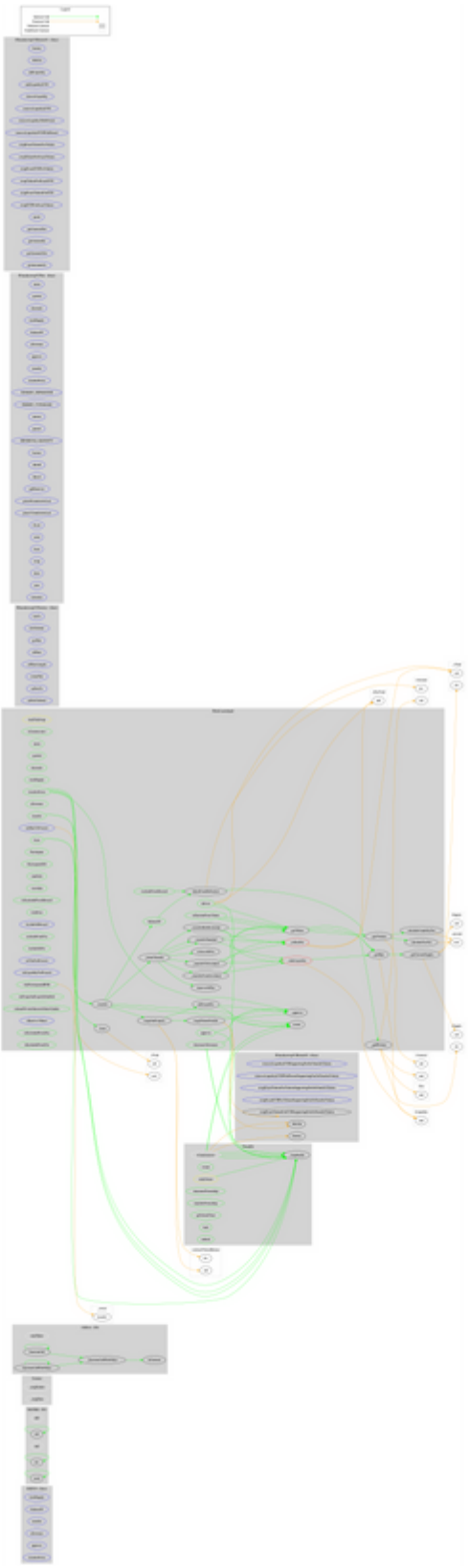
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IPancakeswap V2Router01	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		-
IPancakeswapV2Router02	Interface	IPancakeswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
MaticLaunchpad	Implementation	Context, IBEP20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	decreaseAllowance	Public	✓	-
	burn	Public	✓	-
	GetParticipatedBNB	Public	✓	onlyOwner
	Participate	Public	Payable	-
	ParticipateIDO	Public	Payable	-
	startSale	Public	✓	onlyOwner
	viewSale	Public		-
	isExcludedFromReward	Public		-
	totalFees	Public		-

	deliver	Public	✓	-
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner
	includeInReward	External	✓	onlyOwner
	excludeFromFee	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	setTaxFeePercent	External	✓	onlyOwner
	setLiquidityFeePercent	External	✓	onlyOwner
	setMaxTxPercent	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	releasePrivateSalecoinStatusUpdate	Public	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	calculateTaxFee	Private		
	calculateLiquidityFee	Private		
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	isExcludedFromFee	Public		-
	isIncludedFromFee	Public		-
	_burn	Internal	✓	
	_approve	Private	✓	
	_transfer	Private	✓	
	swapAndLiquify	Private	✓	lockTheSwap
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	_tokenTransfer	Private	✓	
	_transferBothExcluded	Private	✓	
	_transferStandard	Private	✓	

	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	

Contract Flow



Summary

Matic Launchpad is a crypto projects launchpad for Ethereum, Binance Smart Chain and Matic (Polygon) Network. This audit focuses on the Matic Launchpad token contract. There are some functions that can be abused by the owner, like manipulating fees and stopping transactions. 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