Cyberscope

Audit Report

MoonBox Finance

July 2022

Type	BEP20
Network	BSC
Address	0xebacc6644bd0d190fc0c8185c0dd9e0620303ec0
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	MoonBox
Compiler Version	v0.7.5+commit.eb77ed08
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xebacc6644bd0d190fc0c8185c0dd9e0620303ec0
Symbol	MoonBox
Decimals	5
Total Supply	5,000,000
Domain	https://moonbox.finance/

Audit Updates

Initial Audit	7th July 2022
Corrected	

Source Files

Filename	SHA256
ERC20Detailed.sol	9c0bc75dee17430b0027df70fb32be155e24490fc3bbc066254f14fa551802ad
IERC20.sol	54f4c52272e4e35c41f66b21f90e64aa67e77f0ae61f62019607b8cd869cfc9b
IMoonBoxAffiliate.sol	9284213d6a0e6cfafe357a5677a53f4b9a1b2cd4b0b6adcc86ec93274ea30012
IMoonBoxLottery.sol	b1388c322df15afd200f5add2fe646fc91ee06ff6fe534aa04d13b3da63463f1
IPancakeSwapFactory.sol	7d8d67fe3c94b0747e2c5a2264e8a264bad642d89de04fec97c8b6cf06a37097
IPancakeSwapPair.sol	6de50e02ae38df087338e281ec9734be35d7b2153a4f1b6502509e14651807a3
IPancakeSwapRouter.sol	98372ae1eb65975838d002eb1d2755e60b1883f982834d8662358eb245ffa82c
MoonBox.sol	6c643e075d9094c6a301d1393d3493daa8db1b8b94bdf9250f298bc7571a3549
Ownable.sol	8fb2c0ca3ae16d4534a73d66ac988d6e3a03e839b297e969b33141e6c87a4ddf
ReentrancyGuard.sol	de025fa53d77f5f4d0ef9de56827bbc580583cd95a51e2b3259857a7f7d57e4f
SafeMath.sol	a6127a196f59d76417535051229ef9bccdc25500868a6aa30f0cb4df572d4f9c
SafeMathInt.sol	2d4f584c0eecd0091a391f277953f68560a9efb26e6762e24ed0d2c04de324bb

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	contracts/MoonBox.sol#L254,L265

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 maxSellTransactionAmount to zero.

```
require(amount <= maxSellTransactionAmount, "Error amount");
```

The contract owner has the authority to stop the sales for all users excluding the owner. The owner may also take advantage of it by setting the limitSellRate, limitSellRateBonusPr, limitSellRateBonus and limitSellRateReduce to zero to make limitAmount zero. As a result the contract will operate as a honeypot .

```
uint256 txLimitRate = getSellLimitRate(sender);

uint256 limitAmount = balanceOf(sender).mul(txLimitRate).div(
                feeDenominator
 );

require(
    amount <= limitAmount,
    "ERR: Can't sell more than limit rate"
);
```

Recommendation

The contract could embody a check for not allowing setting the maxSellTransactionAmountlimitSellRate, limitSellRateBonusPr, limitSellRateBonus and limitSellRateReduce less than a reasonable amount and creator from zero. 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.

ULTW - Unlimited Liquidity to Team Wallet

Criticality	minor
Location	contracts/MoonBox.sol#L437,L724

Description

The contract owner has the authority to transfer funds without limit to the team wallet. These funds have been accumulated from fees collected from the contract. The owner may take advantage of it by calling the `withdrawAllToTreasury` and `toTreasure` methods.

```
function withdrawAllToTreasury() external swapping onlyOwner {
    uint256 amountToSwap = _gonBalances[address(this)].div(
        _gonsPerFragment
    );
    require(
        amountToSwap > 0,
        "There is no token deposited in token contract"
    );
    swapTokenToBnb(amountToSwap, treasuryReceiver);
}

function toTreasure() external onlyOwner {
    address payable sender = payable(msg.sender);
    sender.transfer(address(this).balance);
}
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may volatile the token's price.

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
	ZD	Zero Division
	US	Untrusted Source
	STC	Succeeded Transfer Check
	MTS	Manipulate Total Supply
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L07	Missing Events Arithmetic
	L13	Divide before Multiply Operation
	L14	Uninitialized Variables in Local Scope

ZD - Zero Division

Criticality	minor
Location	contracts/MoonBox.sol#L627

Description

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

The total can be set to zero from totalTresuryFee and totalInsuranceFee.

```
function updateRateFee() internal {
    uint256 totalTresuryFee = treasuryFeeB.add(treasuryFeeS);
    uint256 totalInsuranceFee = insuranceFundFeeB.add(insuranceFundFeeS);
    uint256 total = totalTresuryFee.add(totalInsuranceFee);
    treasuryFeeRate = totalTresuryFee.mul(feeDenominator).div(total);
    insuranceFundFeeRate = feeDenominator.sub(treasuryFeeRate);
}
```

Recommendation

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

US - Untrusted Source

Criticality	minor
Location	contracts/MoonBox.sol#L392

Description

The contract uses an external contract in order to determine the transaction's flow. The external contract is untrusted. As a result it may produce security issues and harm the transactions.

```
_mgmFee = affiliate.getTotalAffiliateBuy();
```

Recommendation

The contract should use a trusted external source. A trusted source could be either a commonly recognized or an audited contract. The pointing addresses should not be able to change after the initialization.

STC - Succeeded Transfer Check

Criticality	minor
Location	contracts/MoonBox.sol#L726

Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
sender.transfer(address(this).balance);
```

Recommendation

The contract should check if the result of the transfer methods is successful.

MTS - Manipulate Total Supply

Criticality	minor
Location	contracts/MoonBox.sol#L187

Description

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

```
for (uint256 i = 0; i < times; i++) {
    _totalSupply = _totalSupply
        .mul((10**RATE_DECIMALS).add(rebaseRate))
        .div(10**RATE_DECIMALS);
```

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	contracts/MoonBox.sol#L560,643

Description

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

```
setPairAddress
getCirculatingSupply
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contracts/MoonBox.sol#L20,79,21,22,89,49,88

Description

Constant state variables should be declared constant to save gas.

```
DEAD
limitSellRateBonusPr
ZERO
_decimals
_symbol
feeDenominator
_name
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contracts/MoonBox.sol#L612,601,110,22,89,26,25,643,633,576,335,577,21,647,662,592,20,669,651,27,493,677,109,108,111,88,583,655,556,632

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.

```
_addrs
_addr
_limitSellRateBonus
_insuranceFundFeeS
DEAD
_totalSupply
_autoRebase
_initRebaseStartTime
_maxTxn
...
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contracts/MoonBox.sol#L662,651,655,601,612

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.

```
treasuryFeeB = _treasuryFeeB
treasuryFeeS = _treasuryFeeS
limitSellRateBonus = _limitSellRateBonus
limitSellRate = _limitSellRate
limitSellRateReduce = _limitSellRateReduce
```

Recommendation

Emit an event for critical parameter changes.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contracts/MoonBox.sol#L168

Description

Performing divisions before multiplications may cause lose of prediction.

```
times = deltaTime.div(900)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contracts/MoonBox.sol#L170

Description

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

```
rebaseRate
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
ERC20Detailed	Implementation	IERC20		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
IERC20	Interface			
	getDecimal	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	transfer	External	✓	-
	approve	External	✓	-
	transferFrom	External	✓	-
IMoonBoxAffiliate	Interface			
	setAffiliateRevenue	External	✓	-
	claim	External	✓	-
	getAffiliate	External		-
	getF0	External		-
	getTotalAffiliateBuy	External		-
	getTotalAffiliateSell	External		-
IMoonBoxLottery	Interface			
	donate	External	✓	-

IPancakeSwap Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IPancakeSwap Pair	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	✓	-

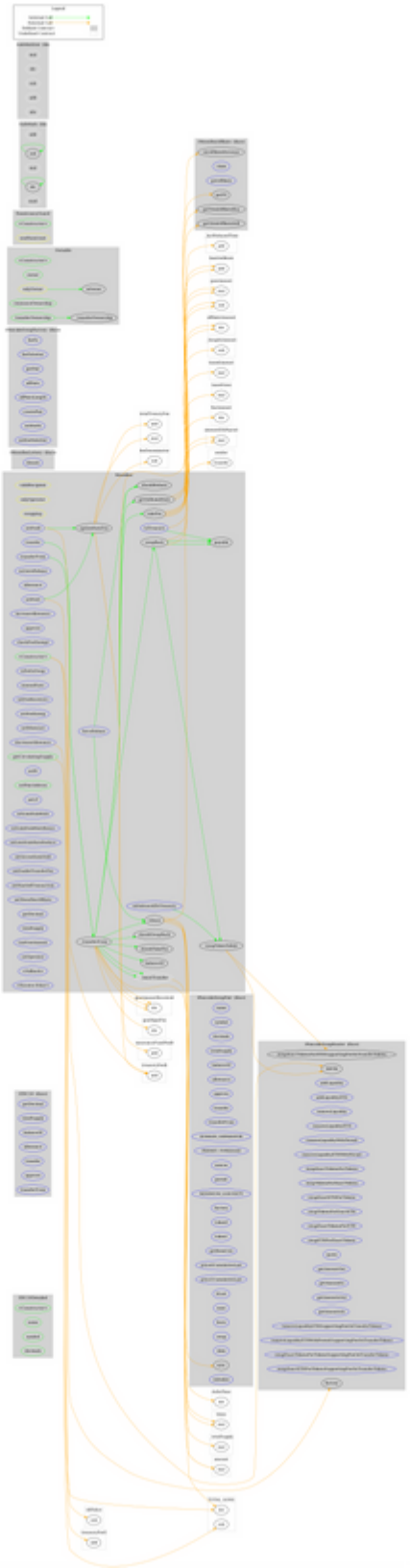
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IPancakeSwap Router	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	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		-
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-

MoonBox	Implementation	ERC20Detailed, Ownable		
	<Constructor>	Public	✓	ERC20Detailed Ownable
	forceRebase	External	✓	onlyOperator
	rebase	Internal	✓	
	transfer	External	✓	validRecipient
	transferFrom	External	✓	validRecipient
	_basicTransfer	Internal	✓	
	_transferFrom	Internal	✓	
	getSellLimitRate	Public		-
	takeFee	Internal	✓	
	swapBack	Internal	✓	swapping
	withdrawAllToTreasury	External	✓	swapping onlyOwner
	swapTokenToBnb	Internal	✓	
	shouldTakeFee	Internal		
	shouldRebase	Internal		
	shouldSwapBack	Internal		
	setAutoRebase	External	✓	onlyOwner
	allowance	External		-
	decreaseAllowance	External	✓	-
	increaseAllowance	External	✓	-
	approve	External	✓	-
	checkFeeExempt	External		-
	getCirculatingSupply	Public		-
	isNotInSwap	External		-
	manualSync	External	✓	-
	setFeeReceivers	External	✓	onlyOwner
	setFeeExemp	External	✓	onlyOwner
	setWhiteList	External	✓	onlyOwner
	setFeeS	External	✓	onlyOwner
	setFeeB	External	✓	onlyOwner
	updateRateFee	Internal	✓	
	setPr	External	✓	onlyOwner

	setPairAddress	Public	✓	onlyOwner
	setLP	External	✓	onlyOwner
	setLimitSaleRate	External	✓	onlyOwner
	setLimitSaleRateBonus	External	✓	onlyOwner
	setLimitSaleRateReduce	External	✓	onlyOwner
	setSecondLimitSell	External	✓	onlyOwner
	setEnableTransferFee	External	✓	onlyOwner
	setMaxSellTransaction	External	✓	onlyOwner
	setMoonBoxAffiliate	External	✓	onlyOwner
	getDecimal	External		-
	totalSupply	External		-
	balanceOf	Public		-
	taxFreeAmount	External		-
	setOperator	External	✓	onlyOwner
	toTreasure	External	✓	onlyOwner
	<Fallback>	External	Payable	-
	<Receive Ether>	External	Payable	-
Ownable	Implementation			
	<Constructor>	Public	✓	-
	owner	Public		-
	isOwner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
ReentrancyGuard	Implementation			
	<Constructor>	Public	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		

	div	Internal		
	div	Internal		
	mod	Internal		
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		

Contract Flow



Domain Info

Domain Name	moonbox.finance
Registry Domain ID	5e3493374d2246918c97d1186cbaa3b8-DONUTS
Creation Date	2022-05-23T07:56:22Z
Updated Date	2022-05-30T15:22:23Z
Registry Expiry Date	2023-05-23T07:56:22Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created in 11 months before the creation of the audit.

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 stopping transactions and transferring funds to the team’s wallet. The contract can be converted into a honeypot and prevent users from selling if the owner abuses the admin functions. In addition, the affiliate contract is out of the audit’s scope. 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 provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io