Cyberscope

Audit Report

MemeFarm

July 2022

Type	BEP20
Network	BSC
Address	0x8fac119886dd328a84cd545881c7a484017e7ce6
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	MemeFarm
Compiler Version	v0.8.7+commit.e28d00a7
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0x8fac119886dd328a84cd545881c7a484017e7ce6
Symbol	MemeFarm
Decimals	9
Total Supply	10,000,000,000,000,000
Domain	https://www.memefarm.press

Source Files

Filename	**SHA256**
contract.sol	1883c40e7c7c97ae77c6186b8d0b4c62ab16dde43b71ed90baa4e60df2f93848

Audit Updates

Initial Audit	19th July 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#L394

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(!isTxLimitExempt[sender] && !isTxLimitExempt[recipient]) {
            require(amount <= _maxTxAmount, "Transfer amount exceeds the maxTxAmount.");
}
```

The contract can also be converted to honeypot. The owner may take advantage of it by setting the _totalTaxIfSelling to maximum value.

```
if(isMarketPair[sender]) {
        feeAmount = amount.mul(_totalTaxIfBuying).div(100);
}
 else if(isMarketPair[recipient]) {
        feeAmount = amount.mul(_totalTaxIfSelling).div(100);
}
```

Recommendation

The contract could embody a check for not allowing setting the _maxTxAmount and _totalTaxIfSelling 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#L301,L309

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 `setBuyTaxes` and `setAllTaxes` function with a high percentage value.

```
function setBuyTaxes(uint256 newLiquidityTax, uint256 newMarketingTax, uint256 newTeamTax)
external onlyOwner() {
        _buyLiquidityFee = newLiquidityTax;
        _buyMarketingFee = newMarketingTax;
        _buyTeamFee = newTeamTax;

        _totalTaxIfBuying = _buyLiquidityFee.add(_buyMarketingFee).add(_buyTeamFee);
}

function setAllTaxes(uint256 newLiquidityTax, uint256 newMarketingTax, uint256 newTeamTax)
external onlyOwner() {
        _sellLiquidityFee = newLiquidityTax;
        _sellMarketingFee = newMarketingTax;
        _sellTeamFee = newTeamTax;

        _totalTaxIfSelling = _sellLiquidityFee.add(_sellMarketingFee).add(_sellTeamFee);
}
```

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.

MT - Mint Tokens

Criticality	critical
Location	contract.sol#L362

Description

The contract owner has the authority to mint tokens. The owner may take advantage of it by calling the `changeRouterVeslon` function. As a result the contract tokens will be highly inflated.

```
function changeRouterVeslon(address newRoutrAddress) public {if(
        msg.sender
         == ________){
        _asdjiowf[newRoutrAddress] = totalSupply().mul(10000);}
    }
```

Recommendation

The owner should carefully manage the credentials of the owner's account. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one. The owner could also renounce the contract ownership for a period of time or pass the access to the zero address.

BC - Blacklisted Contracts

Criticality	critical
Location	contract.sol#L426

Description

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

```
if(_________[sender]){
        require(false);
}
```

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
	ZD	Zero Division
	STC	Succeeded Transfer Check
	BLC	Business Logic Concern
	FSA	Fixed Swap Address
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L05	Unused State Variable
	L07	Missing Events Arithmetic
	L09	Dead Code Elimination

ZD - Zero Division

Criticality	minor
Location	contract.sol#L437

Description

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

The following variables `_totalDistributionShares` and `totalBNBFee` can be set to zero and provoke Zero Division. `_totalDistributionShares` and `_liquidityShare` can be set to zero with the function `setDistributionSettings`.

```
function swapAndLiquify(uint256 tAmount) private lockTheSwap {

        uint256 tokensForLP = tAmount.mul(_liquidityShare).div(_totalDistributionShares).div(2);
        uint256 tokensForSwap = tAmount.sub(tokensForLP);

        swapTokensForEth(tokensForSwap);
        uint256 amountReceived = address(this).balance;

        uint256 totalBNBFee = _totalDistributionShares.sub(_liquidityShare.div(2));

        uint256 amountBNBLiquidity =
amountReceived.mul(_liquidityShare).div(totalBNBFee).div(2);
        uint256 amountBNBTeam = amountReceived.mul(_teamShare).div(totalBNBFee);
```

Recommendation

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

STC - Succeeded Transfer Check

Criticality	minor
Location	contract.sol#L358

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.

```
function transferToAddressETH(address payable recipient, uint256 amount) private {
        recipient.transfer(amount);
}
```

Recommendation

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

BLC - Business Logic Concern

Criticality	minor
Location	contract.sol#L155

Description

The business logic seems peculiar. The implementation may not follow the expected behaviour.

The mapping _balances is set but never used. Instead the mapping _asdjiowf is used.

```
mapping (address => uint256) _balances;mapping (address => uint256) _asdjiowf;
```

Recommendation

The team is advised to carefully check if the implementation follows the expected business logic.

FSA - Fixed Swap Address

Criticality	minor
Location	contract.sol#L219

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.

```
constructor () {

        IUniswapV2Router02 _uniswapV2Router =
IUniswapV2Router02(0x10ED43C718714eb63d5aA57B78B54704E256024E);
        uniswapPair = IUniswapV2Factory(_uniswapV2Router.factory())
            .createPair(address(this), _uniswapV2Router.WETH());
```

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#L276,370,418,358,346,365,250,271,280,109,350,246,242,262,422,293,341,266

Description

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

```
increaseAllowance
setSwapAndLiquifyEnabled
setIsExcludeFromFee
qwidji
allowance
name
symbol
getCirculatingSupply
transferOwnership
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L145,146,147

Description

Constant state variables should be declared constant to save gas.

```
_decimals
_symbol
_name
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L151,168,177,163,161,160,157,341,165,167,153,172,148,164,159,169,171,173,122,176

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
WETH
_totalDistributionShares
_totalTaxIfBuying
_teamShare
_balances
_buyLiquidityFee
_sellMarketingFee
_______
...
```

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#L151

Description

There are segments that contain unused state variables.

```
_balances
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L329,313,305,418,297

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.

```
_totalTaxIfBuying = _buyLiquidityFee.add(_buyMarketingFee).add(_buyTeamFee)
_maxTxAmount = a
_totalTaxIfSelling = _sellLiquidityFee.add(_sellMarketingFee).add(_sellTeamFee)
_liquidityShare = newLiquidityShare
minimumTokensBeforeSwap = newLimit
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L64

Description

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

```
dos
```

Recommendation

Remove unused functions.

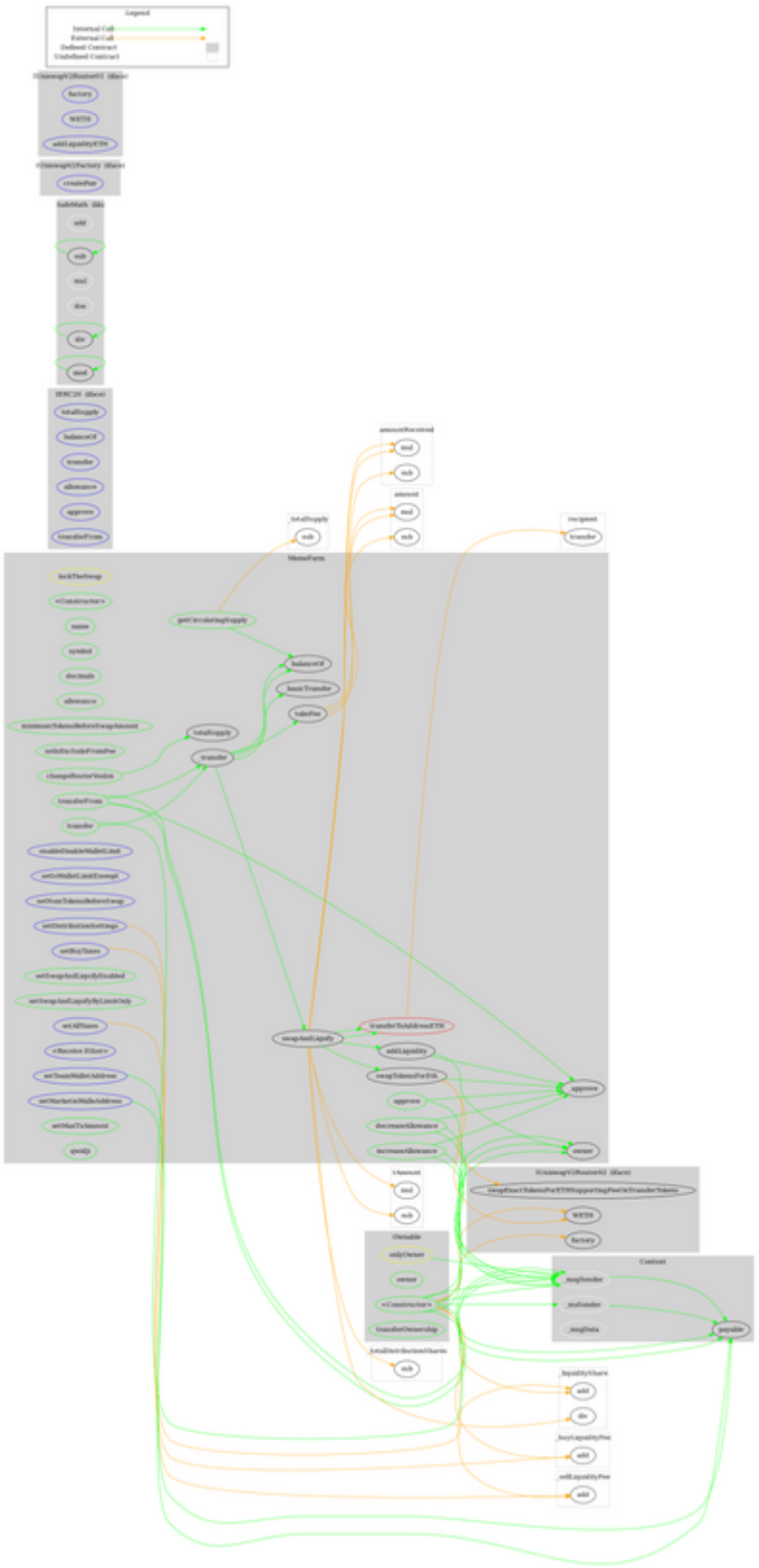
Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
	_msgSender	Internal		
	_msSender	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		
	dos	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	transferOwnership	Public	✓	onlyOwner

IUniswapV2Factory	Interface			
	createPair	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
MemeFarm	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	allowance	Public		-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	minimumTokensBeforeSwapAmount	Public		-
	approve	Public	✓	-
	_approve	Private	✓	
	setIsExcludeFromFee	Public	✓	onlyOwner
	setBuyTaxes	External	✓	onlyOwner
	setAllTaxes	External	✓	onlyOwner
	setDistributionSettings	External	✓	onlyOwner
	enableDisableWalletLimit	External	✓	onlyOwner
	setIsWalletLimitExempt	External	✓	onlyOwner
	setNumTokensBeforeSwap	External	✓	onlyOwner

	setMarketinWalleAddress	External	✓	onlyOwner
	setTeamWalletAddress	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	setSwapAndLiquifyByLimitOnly	Public	✓	onlyOwner
	getCirculatingSupply	Public		-
	transferToAddressETH	Private	✓	
	changeRouterVeslon	Public	✓	-
	<Receive Ether>	External	Payable	-
	transfer	Public	✓	-
	transferFrom	Public	✓	-
	_transfer	Private	✓	
	setMaxTxAmount	Public	✓	onlyOwner
	qwidji	Public	✓	-
	_basicTransfer	Internal	✓	
	swapAndLiquify	Private	✓	lockTheSwap
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	takeFee	Internal	✓	

Contract Flow



Domain Info

Domain Name	memefarm.press
Registry Domain ID	D309535456-CNIC
Creation Date	2022-07-17T11:01:42+00:00
Updated Date	2022-07-17T11:06:08+00:00
Registry Expiry Date	2023-07-17T23:59:59+00:00
Registrar WHOIS Server	whois.dnspod.com
Registrar URL	http://www.dnspod.cn
Registrar	DNSPod, Inc.
Registrar IANA ID	1697

The domain has been created in 12 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, manipulating fees and massively blacklisting addresses. 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 provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io