



Cyberscope

# Audit Report

## **MarvinPePe**

May 2022

Type           BEP20

Network       BSC

Address       0x9E4EDb2f186097Dc23cebB78E856Dc7A7C31a554

Audited by   © cyberscope

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>Contract Review</b>	<b>3</b>
<b>Source Files</b>	<b>3</b>
<b>Audit Updates</b>	<b>3</b>
<b>Contract Analysis</b>	<b>4</b>
<b>ST - Stop Transactions</b>	<b>5</b>
Description	5
Recommendation	5
<b>ELFM - Exceed Limit Fees Manipulation</b>	<b>7</b>
Description	7
Recommendation	7
<b>ULTW - Unlimited Liquidity to Team Wallet</b>	<b>8</b>
Description	8
Recommendation	8
<b>Contract Diagnostics</b>	<b>10</b>
<b>MC - Missing Check</b>	<b>11</b>
Description	11
Recommendation	11
<b>L01 - Public Function could be Declared External</b>	<b>12</b>
Description	12
Recommendation	12
<b>L02 - State Variables could be Declared Constant</b>	<b>13</b>
Description	13
Recommendation	13
<b>L04 - Conformance to Solidity Naming Conventions</b>	<b>14</b>
Description	14

<b>Recommendation</b>	<b>14</b>
<b>L05 - Unused State Variable</b>	<b>15</b>
<b>Description</b>	<b>15</b>
<b>Recommendation</b>	<b>15</b>
<b>L07 - Missing Events Arithmetic</b>	<b>16</b>
<b>Description</b>	<b>16</b>
<b>Recommendation</b>	<b>16</b>
<b>L09 - Dead Code Elimination</b>	<b>17</b>
<b>Description</b>	<b>17</b>
<b>Recommendation</b>	<b>17</b>
<b>L13 - Divide before Multiply Operation</b>	<b>18</b>
<b>Description</b>	<b>18</b>
<b>Recommendation</b>	<b>18</b>
<b>Contract Functions</b>	<b>19</b>
<b>Contract Flow</b>	<b>24</b>
<b>Domain Info</b>	<b>25</b>
<b>Summary</b>	<b>26</b>
<b>Disclaimer</b>	<b>27</b>
<b>About Cyberscope</b>	<b>28</b>

## Contract Review

<b>Contract Name</b>	Meta
<b>Compiler Version</b>	v0.8.13+commit.abaa5c0e
<b>Optimization</b>	200 runs
<b>Licence</b>	None
<b>Explorer</b>	<a href="https://bscscan.com/token/0x9E4EDb2f186097Dc23cebB78E856Dc7A7C31a554">https://bscscan.com/token/0x9E4EDb2f186097Dc23cebB78E856Dc7A7C31a554</a>
<b>Symbol</b>	MVP
<b>Decimals</b>	18
<b>Total Supply</b>	8,400,000,000
<b>Domain</b>	marvinpepe.com

## Source Files

<b>Filename</b>	<b>SHA256</b>
<b>contract.sol</b>	75e238539235c1bbbc5cdebcad852c1f987cafbbc0f5238a5a4427569826f233

## Audit Updates

<b>Initial Audit</b>	3rd May 2022
<b>Corrected</b>	

# 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#L670, 689, 772

### Description

The contract owner has the authority to stop transactions for all users excluding the owner. Owner can do that in multiple ways. The owner may take advantage of it by setting the `_maxTxAmount` and/or `_walletMax` to zero.

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

```
if(checkWalletLimit && !isWalletLimitExempt[recipient])  
    require(balanceOf(recipient).add(finalAmount) <= _walletMax);
```

The contract owner can also convert the contract into a honeypot and prevent the users from selling by increasing the `_totalTaxIfSelling`.

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

### Recommendation

The contract could embody a check for not allowing setting the `_maxTxAmount` and `_walletMax` 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 contract could also not allow the `_totalTaxIfSelling` to be higher than a reasonable amount.

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

<b>Criticality</b>	critical
<b>Location</b>	contract.sol#L557, 565

### 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 `setSelTaxes` 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 setSelTaxes(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.



## ULTW - Unlimited Liquidity to Team Wallet

Criticality	minor
Location	contract.sol#L597, 674, 706

### 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 setting a high fee to the `setNumTokensBeforeSwap` variable.

```
function setNumTokensBeforeSwap(uint256 newLimit) external onlyOwner() {  
    minimumTokensBeforeSwap = newLimit;  
}
```

```
uint256 contractTokenBalance = balanceOf(address(this));  
    bool overMinimumTokenBalance = contractTokenBalance >=  
minimumTokensBeforeSwap;
```

```
function swapAndLiquify(uint256 tAmount) private lockTheSwap {  
  
    uint256 tokensForLP =  
tAmount.mul(_liquidityShare).div(_totalDistributionShares).div(2);  
    uint256 tokensForSwap = tAmount.sub(tokensForLP);  
  
    swapTokensForEth(tokensForSwap);
```

### 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
●	MC	Missing Check
●	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
●	L13	Divide before Multiply Operation

## MC - Missing Check

**Criticality**

minor

**Location**

contract.sol#L573, 708

### Description

The contract is processing variables that have not properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues.

```
function setDistributionSettings(uint256 newLiquidityShare, uint256
newMarketingShare, uint256 newTeamShare) external onlyOwner() {
    _liquidityShare = newLiquidityShare;
    _marketingShare = newMarketingShare;
    _teamShare = newTeamShare;

    _totalDistributionShares =
    _liquidityShare.add(_marketingShare).add(_teamShare);
}
```

```
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);
```

### Recommendation

The contract should properly check the variables according to the required specifications

## L01 - Public Function could be Declared External

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L160,165,171,494,498,502,506,514,518,523,528,532,545,553,609,614,618,626,648,653

### Description

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

```
transferFrom
transfer
changeRouterVersion
getCirculatingSupply
setSwapAndLiquifyByLimitOnly
setSwapAndLiquifyEnabled
setIsExcludedFromFee
setMarketPairStatus
approve
...
```

### Recommendation

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

## L02 - State Variables could be Declared Constant

**Criticality**

minor

**Location**

contract.sol#L141,140

### Description

Constant state variables should be declared constant to save gas.

```
asdasd
_lockTime
```

### Recommendation

Add the constant attribute to state variables that never change.

## L04 - Conformance to Solidity Naming Conventions

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L208,209,225,244,609,390,398,399,400,402,403,404,406,407,408,410,411,412,415,416

### 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.

```
_walletMax  
_maxTxAmount  
_totalDistributionShares  
_totalTaxIfSelling  
_totalTaxIfBuying  
_teamShare  
_marketingShare  
_liquidityShare  
_sellTeamFee  
...
```

### 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#L140,141

### Description

There are segments that contain unused state variables.

```
_lockTime  
asdasd
```

### Recommendation

Remove unused state variables.



## L07 - Missing Events Arithmetic

**Criticality**

minor

**Location**

contract.sol#L557,565,573,581,593,597

### 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.

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

### Recommendation

Emit an event for critical parameter changes.

## L09 - Dead Code Elimination

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L118,101,105,109,113,82,93

### 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.

## L13 - Divide before Multiply Operation

**Criticality**

minor

**Location**

contract.sol#L450

### Description

Performing divisions before multiplications may cause lose of prediction.

```
minimumTokensBeforeSwap = totalSupply_.mul(2).div(10 ** 6) * 10 ** _decimals
```

### Recommendation

The multiplications should be prior to the divisions.

# Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
<b>Context</b>	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
<b>IERC20</b>	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
<b>SafeMath</b>	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
<b>Address</b>	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	

	_functionCallWithValue	Private	✓	
<b>Ownable</b>	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	waiveOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	getTime	Public		-
<b>IUniswapV2Factory</b>	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
<b>IUniswapV2Pair</b>	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		-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
<b>IUniswapV2Router01</b>	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	✓	-
Meta	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	✓	
	setMarketPairStatus	Public	✓	onlyOwner
	setIsTxLimitExempt	External	✓	onlyOwner
	setIsExcludedFromFee	Public	✓	onlyOwner
	setBuyTaxes	External	✓	onlyOwner
	setSellTaxes	External	✓	onlyOwner
	setDistributionSettings	External	✓	onlyOwner
	setMaxTxAmount	External	✓	onlyOwner
	enableDisableWalletLimit	External	✓	onlyOwner
	setIsWalletLimitExempt	External	✓	onlyOwner
	setWalletLimit	External	✓	onlyOwner

	setNumTokensBeforeSwap	External	✓	onlyOwner
	setMarketingWalletAddress	External	✓	onlyOwner
	setTeamWalletAddress	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	setSwapAndLiquifyByLimitOnly	Public	✓	onlyOwner
	getCirculatingSupply	Public		-
	transferToAddressETH	Private	✓	
	changeRouterVersion	Public	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	transfer	Public	✓	-
	transferFrom	Public	✓	-
	_transfer	Private	✓	
	_basicTransfer	Internal	✓	
	swapAndLiquify	Private	✓	lockTheSwap
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	takeFee	Internal	✓	



# Contract Flow



## Domain Info

<b>Domain Name</b>	marvinpepe.com
<b>Registry Domain ID</b>	2692751078_DOMAIN_COM-VRSN
<b>Creation Date</b>	2022-04-29T17:52:02.0Z
<b>Updated Date</b>	2022-05-02T13:21:36.0Z
<b>Registry Expiry Date</b>	2023-04-29T17:52:02.0Z
<b>Registrar WHOIS Server</b>	whois.dynadot.com
<b>Registrar URL</b>	<a href="http://www.dynadot.com">http://www.dynadot.com</a>
<b>Registrar</b>	DYNADOT LLC
<b>Registrar IANA ID</b>	472

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

There are some functions that can be abused by the owner like stopping transactions, manipulating fees and transferring funds to the team's wallet. There is no check for the maximum fee percentage that can be set. 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>