



# **RugFreeCoins Audit**



## **Mpad Exchange Token Smart Contract Security Audit October 30, 2021**



# Contents

Audit details	1
Disclaimer	2
Background	3
About the project	4
Target market and the concept	5
Potential to grow with score points	6
Total Points	6
Contract details	7
Contract code function details	8
Contract description table	9
Security issue checking status	19
Owner privileges	21
Audit conclusion	24

# Audit details



## **Audited project**

Mpad Exchange Token



## **Contract Address**

0x0504DCdDeF1bA1F18D44A0149d8B5524c9C5946D



## **Client contact**

Mpad Exchange Team



## **Blockchain**

Binance smart chain



## **Project website**

<https://mpadexchange.com/>

# Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

**DISCLAIMER:** By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Rugfreecoins and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Rugfreecoins) owe no duty of care towards you or any other person, nor does Rugfreecoins make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Rugfreecoins hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Rugfreecoins hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Rugfreecoins, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

# Background

Rugfreecoins was commissioned by Mpad Exchange Token to perform an audit of the smart contract.

**<https://bscscan.com/token/0x0504DCdDeF1bA1F18D44A0149d8B5524c9C5946D>**

The focus of this audit is to verify that the smart contract is secure, resilient and working according to the specifications.

The information in this report should be used to understand the risk exposure of the smart contract, project feasibility, long term sustainability and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

# About the project

Mousepad is a token built on the Binance Smart Chain that aims to develop and exchange platform for trading exclusively for BSC. Each transaction, purchase, and sale incur a 15% fee.

## Features

- ❖ The **token rewards** will be distributed among every holder proportional to how many tokens each individual holds from the fee of **3% when buying and selling**.
- ❖ The **sustainability fee of 5% for Dev and marketing when buying and selling** is what allows Mousepad token to hold the aforementioned promise. Tokens will be swapped into BNB and will be sent to a marketing wallet per transaction. This way, Mousepad token will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.
- ❖ The additional component included under the sustainability section is a **liquidity fee of 2% from buying and selling**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.
- ❖ The **buyback and burn fee of 5% from buying and selling** is used to save from massive dips in order to keep the token market price stable.

## Tokenomics

### 15% fee when buying and selling

- ❖ 3% of trade goes to holders' pockets in tokens.
- ❖ 2% of trade goes to the liquidity pool.
- ❖ 5% of trade goes to the development & marketing wallet.
- ❖ 5% of trade goes to a separate wallet for buyback to burn tokens.

# Target market and the concept

## Target market

- ❖ Anyone who's interested in the Crypto space with long-term investment plans.
- ❖ Anyone who's ready to earn a passive income in tokens by holding tokens.
- ❖ Anyone who's interested in trading tokens.
- ❖ Anyone who's interested in staking and farming and getting rewards.
- ❖ Anyone who's interested in taking part with the future plans of the MousePad token.
- ❖ Anyone who's interested in making financial transactions with any other party using BNB or MousePad as the currency.

## Core concept

### The MousePad reward system

The fee of 3% when buying and selling from each transaction gets sent amongst all holders in tokens. The holders will be eligible to receive tokens, whenever a transaction occurs, and rewards are proportional to how many tokens each individual holds.

### Sustainable mechanism

The sustainability fee of 5% for marketing and development is what allows MousePad to promote the token and use funds to further development of the platform. Tokens will be swapped into BNB and will be sent to a marketing wallet per transaction. This way, Mousepad will have access to the funds without selling tokens as the traditional way, which will enable them to consume funds without hurting the project.

**The liquidity fee of 2%**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

**The controlled buyback collects 5%** tax on each transaction, which is stored inside a private wallet in BNB. It is to save from massive dips in order to keep the token market price stable.

### The future plan

Mpad roadmap includes extensive marketing campaigns, MPad Exchange, investing through Staking/Farming \$Mpad & in soon enough developing our own blockchain and releasing a physical debit card for our users.

# Potential to grow with score points

1.	Project efficiency	9/10
2.	Project uniqueness	8/10
3	Information quality	5/10
4	Service quality	7/10
5	System quality	7/10
6	Impact on the community	8/10
7	Impact on the business	9/10
8	Preparing for the future	8/10
Total Points		<b>7.625/10</b>



# Contract details

## Token contract details for 30<sup>th</sup> October 2021



















<b>Contract name</b>	MousePad
<b>Contract address</b>	0x0504DCdDeF1bA1F18D44A0149d8B5524c9C5946D
<b>Token supply</b>	100,000,000,000
<b>Token ticker</b>	MPAD
<b>Decimals</b>	18
<b>Token holders</b>	1
<b>Transaction count</b>	1
<b>Dev &amp; marketing wallet</b>	Marketing, development, and buyback wallet: 0x9200f0f80038372981fc633ad82c98a061682d54
<b>Contract deployer address</b>	0x2B8373C9150d4b657b92De35a36DC9162A3e3659
<b>Contract's current owner address</b>	0x2b8373c9150d4b657b92de35a36dc9162a3e3659

















# Contract code function details













No	Category	Item	Result
1	Coding conventions	BRC20 Token standards	pass
		compile errors	pass
		Compiler version security	pass
		visibility specifiers	pass
		Gas consumption	low issue
		SafeMath features	pass
		Fallback usage	pass
		tx.origin usage	pass
		deprecated items	pass
		Redundant code	pass
		Overriding variables	pass
2	Function call audit	Authorization of function call	pass
		Low level function (call/delegate call) security	pass
		Returned value security	pass
		Selfdestruct function security	pass
3	Business security	Access control of owners	medium issue
		Business logics	informational
		Business implementations	pass
4	Integer overflow/underflow		pass
5	Reentrancy		pass
6	Exceptional reachable state		pass
7	Transaction ordering dependence		pass
8	Block properties dependence		pass
9	Pseudo random number generator (PRNG)		pass
10	DoS (Denial of Service)		pass
11	Token vesting implementation		pass
12	Fake deposit		pass
13	Event security		pass

# Contract description table







Below table represents the summary of the contracts and methods in the token contract. We scanned the whole contract and listed down all the Interfaces, functions and implementations with its visibility and mutability.

Contract	Type	Bases		
L	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
L	_msgSender	Internal 		
L	_msgData	Internal 		
IBEP20	Interface			
L	totalSupply	External 		NO 
L	balanceOf	External 		NO 
L	transfer	External 		NO 
L	allowance	External 		NO 
L	approve	External 		NO 
L	transferFrom	External 		NO 
SafeMath	Library			
L	tryAdd	Internal 		












L	trySub	Internal 		
L	tryMul	Internal 		
L	tryDiv	Internal 		
L	tryMod	Internal 		
L	add	Internal 		
L	sub	Internal 		
L	mul	Internal 		
L	div	Internal 		
L	mod	Internal 		
L	sub	Internal 		
L	div	Internal 		
L	mod	Internal 		
<b>SafeMathInt</b>	<b>Library</b>			
L	mul	Internal 		
L	div	Internal 		
L	sub	Internal 		
L	add	Internal 		








L	abs	Internal 		
L	toUint256Safe	Internal 		
<b>SafeMathUint</b>	<b>Library</b>			
L	toInt256Safe	Internal 		
Ownable	Implementation	Context		
L		Public 		NO 
L	owner	Public 		NO 
L	renounceOwnership	Public 		onlyOwner
L	transferOwnership	Public 		onlyOwner
<b>IUniswapV2Factory</b>	<b>Interface</b>			
L	feeTo	External 		NO 
L	feeToSetter	External 		NO 
L	getPair	External 		NO 
L	allPairs	External 		NO 
L	allPairsLength	External 		NO 
L	createPair	External 		NO 

















L	setFeeTo	External ¶		NO¶
L	setFeeToSetter	External ¶		NO¶
<b>IUniswapV2Pair</b>	<b>Interface</b>			
L	name	External ¶		NO¶
L	symbol	External ¶		NO¶
L	decimals	External ¶		NO¶
L	totalSupply	External ¶		NO¶
L	balanceOf	External ¶		NO¶
L	allowance	External ¶		NO¶
L	approve	External ¶		NO¶
L	transfer	External ¶		NO¶
L	transferFrom	External ¶		NO¶
L	DOMAIN_SEPARATOR	External ¶		NO¶
L	PERMIT_TYPEHASH	External ¶		NO¶
L	nonces	External ¶		NO¶
L	permit	External ¶		NO¶
L	MINIMUM_LIQUIDITY	External ¶		NO¶


























L	factory	External ¶		NO¶
L	token0	External ¶		NO¶
L	token1	External ¶		NO¶
L	getReserves	External ¶		NO¶
L	price0CumulativeLast	External ¶		NO¶
L	price1CumulativeLast	External ¶		NO¶
L	kLast	External ¶		NO¶
L	mint	External ¶	⦿	NO¶
L	burn	External ¶	⦿	NO¶
L	swap	External ¶	⦿	NO¶
L	skim	External ¶	⦿	NO¶
L	sync	External ¶	⦿	NO¶
L	initialize	External ¶	⦿	NO¶
<b>IUniswapV2Router01</b>	<b>Interface</b>			
L	factory	External ¶		NO¶
L	WETH	External ¶		NO¶
L	addLiquidity	External ¶	⦿	NO¶









L	addLiquidityETH	External ¶		NO¶
L	removeLiquidity	External ¶		NO¶
L	removeLiquidityETH	External ¶		NO¶
L	removeLiquidityWithPermit	External ¶		NO¶
L	removeLiquidityETHWithPermit	External ¶		NO¶
L	swapExactTokensForTokens	External ¶		NO¶
L	swapTokensForExactTokens	External ¶		NO¶
L	swapExactETHForTokens	External ¶		NO¶
L	swapTokensForExactETH	External ¶		NO¶
L	swapExactTokensForETH	External ¶		NO¶
L	swapETHForExactTokens	External ¶		NO¶
L	quote	External ¶		NO¶
L	getAmountOut	External ¶		NO¶
L	getAmountIn	External ¶		NO¶
L	getAmountsOut	External ¶		NO¶
L	getAmountsIn	External ¶		NO¶
<b>IUniswapV2Router02</b>	<b>Interface</b>	<b>IUniswapV2Router01</b>		

L	removeLiquidityETHSupportingFeeOnTransferTokens	External ¶		NO¶
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ¶		NO¶
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ¶		NO¶
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External ¶		NO¶
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External ¶		NO¶
<b>MousePad</b>	<b>Implementation</b>	<b>Context, IBEP20, Ownable</b>		
L		Public ¶		Ownable
L	name	Public ¶		NO¶
L	symbol	Public ¶		NO¶
L	decimals	Public ¶		NO¶
L	totalSupply	Public ¶		NO¶
L	balanceOf	Public ¶		NO¶
L	transfer	Public ¶		NO¶
L	allowance	Public ¶		NO¶



L	approve	Public ¶		NO¶
L	transferFrom	Public ¶		NO¶
L	increaseAllowance	Public ¶		NO¶
L	decreaseAllowance	Public ¶		NO¶
L	isExcludedFromReward	Public ¶		NO¶
L	totalFees	Public ¶		NO¶
L	deliver	Public ¶		NO¶
L	reflectionFromToken	Public ¶		NO¶
L	tokenFromReflection	Public ¶		NO¶
L	excludeFromReward	Public ¶		onlyOwner
L	includeInReward	External ¶		onlyOwner
L	setDevelopmentNbuybackWallet	External ¶		onlyOwner
L	setExcludedFromFee	External ¶		onlyOwner
L	setTaxFeePercent	External ¶		onlyOwner
L	setLiquidityFeePercent	External ¶		onlyOwner
L	setPercentageOfLiquidityForDevelopmentNbuyback	External ¶		onlyOwner
L	setMaxTxPercent	External ¶		onlyOwner
L	setSwapAndLiquifyEnabled	Public ¶		onlyOwner



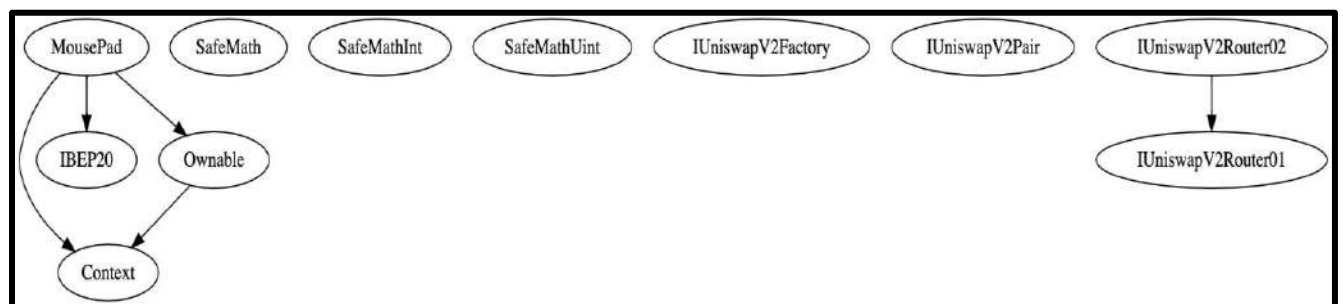
L		External !		NO!
L	setUniswapRouter	External !		onlyOwner
L	setUniswapPair	External !		onlyOwner
L	setExcludedFromAutoLiquidity	External !		onlyOwner
L	_reflectFee	Private 		
L	_getTVValues	Private 		
L	_getRValues	Private 		
L	_getRate	Private 		
L	_getCurrentSupply	Private 		
L	takeTransactionFee	Private 		
L	calculateFee	Private 		
L	isExcludedFromFee	Public !		NO!
L	_approve	Private 		
L	_transfer	Private 		
L	swapAndLiquify	Private 		lockTheSwap
L	swapTokensForBnb	Private 		
L	addLiquidity	Private 		
L	_tokenTransfer	Private 		

L	<code>_transferStandard</code>	Private 		
L	<code>_transferBothExcluded</code>	Private 		
L	<code>_transferToExcluded</code>	Private 		
L	<code>_transferFromExcluded</code>	Private 		

### Legend

Symbol	Meaning
	Function can modify state
	Function is payable

## Inheritance Hierarchy



# Security issue checking status

## ❖ High severity issues

- No high severity issues found.

## ❖ Medium severity issues

- Owner can change all fees without any limitation.

```
ftrace | funcSig
function setTaxFeePercent(uint256 taxFee↑) external onlyOwner {
    _taxFee = taxFee↑;
}

ftrace | funcSig
function setLiquidityFeePercent(uint256 liquidityFee↑) external onlyOwner {
    _liquidityFee = liquidityFee↑;
}

ftrace | funcSig
function setPercentageOfLiquidityForDevelpomentNbuyback(
    uint256 develpomentNbuybackFee↑
) external onlyOwner {
    _percentageOfLiquidityForDevelpomentNbuyback = develpomentNbuybackFee↑;
}

ftrace | funcSig
function setMaxTxPercent(uint256 maxTxPercent↑) external onlyOwner {
    _maxTxAmount = _tTotal.mul(maxTxPercent↑).div(100);
}
```

#### ❖ Low severity issues

- In the includeInReward function, if they use a long wallet list there can be an OUT\_OF\_GAS issue, better to use a small array list at once.

```
ftrace | funcSig
function includeInReward(address account↑) external onlyOwner {
    require(!_isExcluded[account↑], "Account is already excluded");

    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account↑) {
            _excluded[i] = _excluded[_excluded.length - 1];
            tOwned[account↑] = 0;
            _isExcluded[account↑] = false;
            _excluded.pop();
            break;
        }
    }
}
```

#### ❖ Informational

- BuyBack function is not written in the contract.

```
// take developmentNbuyback fee
uint256 developmentNbuybackFee = newBalance
    .mul(_percentageOfLiquidityForDevelopmentNbuyback)
    .div(100);
uint256 bnbForLiquidity = newBalance.sub(developmentNbuybackFee);
if (developmentNbuybackFee > 0) {
    payable(_developmentNbuybackWallet).transfer(
        developmentNbuybackFee
    );
    emit DevelopmentNbuybackFeeSent(
        _developmentNbuybackWallet,
        developmentNbuybackFee
    );
}
```

Buyback function is not written in the contract and buyback amount is getting sent to the dev wallet. Dev should do buyback & burn manually.

# Owner privileges

- ❖ The owner can renounce and transfer ownership.

```
ftrace | funcSig
function renounceOwnership() public virtual onlyOwner {
    emit OwnershipTransferred(_owner, address(0));
    _owner = address(0);
}

ftrace | funcSig
function transferOwnership(address newOwner) public virtual onlyOwner {
    require(
        newOwner != address(0),
        "Ownable: new owner is the zero address"
    );
    emit OwnershipTransferred(_owner, newOwner);
    _owner = newOwner;
}
```

- ❖ The owner can include and exclude wallets from the rewards.

```
ftrace | funcSig
function excludeFromReward(address account) public onlyOwner {
    require(!_isExcluded[account], "Account is already excluded");

    if (_rOwned[account] > 0) {
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}

ftrace | funcSig
function includeInReward(address account) external onlyOwner {
    require(_isExcluded[account], "Account is already excluded");

    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account] = 0;
            _isExcluded[account] = false;
            _excluded.pop();
            break;
        }
    }
}
```



- ❖ The owner can change dev wallet address.

```
ftrace | funcSig
function setDevelopmentNbuybackWallet(address developmentNbuybackWallet↑)
|   external
|   onlyOwner
{
|   _developmentNbuybackWallet = developmentNbuybackWallet↑;
| }
}
```

- ❖ The owner can exclude and include wallets from fees.

```
ftrace | funcSig
function setExcludedFromFee(address account↑, bool e↑) external onlyOwner {
|   _isExcludedFromFee[account↑] = e↑;
| }
}
```

- ❖ The owner can change tax fees,liquidity fees,buyback fees and dev fees.

```
ftrace | funcSig
function setTaxFeePercent(uint256 taxFee↑) external onlyOwner {
|   _taxFee = taxFee↑;
| }

ftrace | funcSig
function setLiquidityFeePercent(uint256 liquidityFee↑) external onlyOwner {
|   _liquidityFee = liquidityFee↑;
| }

ftrace | funcSig
function setPercentageOfLiquidityForDevelopmentNbuyback(
|   uint256 developmentNbuybackFee↑
| ) external onlyOwner {
|   _percentageOfLiquidityForDevelopmentNbuyback = developmentNbuybackFee↑;
| }
}
```

- ❖ The owner can change max transaction precentage.

```
ftrace | funcSig
function setMaxTxPercent(uint256 maxTxPercent↑) external onlyOwner {
|   _maxTxAmount = _tTotal.mul(maxTxPercent↑).div(100);
| }
}
```

- ❖ The owner can enable and disable swap and liquidity add.

```
ftrace | funcSig
function setSwapAndLiquifyEnabled(bool e↑) public onlyOwner {
    _swapAndLiquifyEnabled = e↑;
    emit SwapAndLiquifyEnabledUpdated(e↑);
}
```

- ❖ The owner can change the router address.

```
ftrace | funcSig
function setUniswapRouter(address r↑) external onlyOwner {
    IUniswapV2Router02 uniswapV2Router = IUniswapV2Router02(r↑);
    _uniswapV2Router = uniswapV2Router;
}
```

# Audit conclusion

While conducting the audit of the Mouse Pad smart contract, it was observed that there is a medium severity issue in the owner's access control, low severity issue and an informational concern.