



# Audit Report

## **Minions Metaverse**

February 2022

Type	BEP20
Network	BSC
Address	0xCf81dEa4621936FbE69b3A371DE0cf1D5b5B6B7
Audited by	© coinscope

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>Contract Review</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>6</b>
Description	6
Recommendation	6
<b>Contract Diagnostics</b>	<b>7</b>
<b>CO - Code Optimization</b>	<b>8</b>
Description	8
Recommendation	8
<b>CR - Code Repetition</b>	<b>9</b>
Description	9
Recommendation	9
<b>L01 - Public Function could be Declared External</b>	<b>10</b>
Description	10
Recommendation	10
<b>L02 - State Variables could be Declared Constant</b>	<b>11</b>
Description	11
Recommendation	11
<b>L04 - Conformance to Solidity Naming Conventions</b>	<b>12</b>
Description	12
Recommendation	12

<b>L12 - Using Variables before Declaration</b>	<b>13</b>
Description	13
Recommendation	13
<b>L07 - Missing Events Arithmetic</b>	<b>14</b>
Description	14
Recommendation	14
<b>L14 - Uninitialized Variables in Local Scope</b>	<b>15</b>
Description	15
Recommendation	15
<b>Contract Functions</b>	<b>16</b>
<b>Contract Flow</b>	<b>23</b>
<b>Domain Info</b>	<b>24</b>
<b>Summary</b>	<b>25</b>
<b>Disclaimer</b>	<b>26</b>
<b>About Coinscope</b>	<b>27</b>

## Contract Review

<b>Contract Name</b>	Mini
<b>Compiler Version</b>	v0.6.12+commit.27d51765
<b>Optimization</b>	200 runs
<b>Licence</b>	MIT
<b>Explorer</b>	<a href="https://bscscan.com/token/0xaCf81dEa4621936FbE69b3A371DE0cf1D5b5B6B7">https://bscscan.com/token/0xaCf81dEa4621936FbE69b3A371DE0cf1D5b5B6B7</a>
<b>Symbol</b>	Mini
<b>Decimals</b>	18
<b>Total Supply</b>	99,999,999,999
<b>Source</b>	contract.sol
<b>Domain</b>	minions.world

## Audit Updates

<b>Initial Audit</b>	10th February 2022
<b>Corrected</b>	

# Contract Analysis

● Critical
 ● Medium
 ● Minor
 ● Pass

Severity	Code	Description
<span style="color: gold;">●</span>	ST	Contract Owner is not able to stop or pause transactions
<span style="color: blue;">●</span>	OCTD	Contract Owner is not able to transfer tokens from specific address
<span style="color: blue;">●</span>	OTUT	Owner Transfer User's Tokens
<span style="color: red;">●</span>	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
<span style="color: blue;">●</span>	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
<span style="color: blue;">●</span>	MT	Contract Owner is not able to mint new tokens
<span style="color: blue;">●</span>	BT	Contract Owner is not able to burn tokens from specific wallet
<span style="color: blue;">●</span>	BC	Contract Owner is not able to blacklist wallets from selling

## ST - Stop Transactions

Criticality	medium
Location	contract.sol#L293

### 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(!_isExcludedFromFees[from] || !_isExcludedFromFees[to]) {  
    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#L166

### 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 `setFees` function with a high percentage value.

```
function setFees(uint256 _rewardFee, uint256 _liquidityFee, uint256
 _marketingFee, uint256 _ecoSystemFee, uint256 _burnFee) external onlyOwner{
    BUSDRewardsFee = _rewardFee;
    liquidityFee = _liquidityFee;
    marketingFee = _marketingFee;
    ecoSystemFee = _ecoSystemFee;
    burnFee = _burnFee;
    totalFees =
    BUSDRewardsFee.add(liquidityFee).add(marketingFee).add(ecoSystemFee).add(burnFee
);
}
```

### 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.

# Contract Diagnostics

● Critical    ● Medium    ● Minor

Severity	Code	Description
●	CO	Code Optimization
●	CR	Code Repetition
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L12	Using Variables before Declaration
●	L07	Missing Events Arithmetic
●	L14	Uninitialized Variables in Local Scope



## CO - Code Optimization

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L311

### Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations.

The contract swaps tokens for BUSD three times in the same execution thread.

```
uint256 marketingTokens =
contractTokenBalance.mul(marketingFee).div(totalFees.sub(burnFee));
swapAndSendToFee(marketingWalletAddress, marketingTokens);

uint256 ecoSystemTokens =
contractTokenBalance.mul(ecoSystemFee).div(totalFees.sub(burnFee));
swapAndSendToFee(ecoSystemWalletAddress, ecoSystemTokens);

...

uint256 sellTokens =
contractTokenBalance.mul(BUSDRewardsFee).div(totalFees.sub(burnFee));
swapAndSendDividends(sellTokens);
```

### Recommendation

There could be issued one call for swap tokens with the total value and distribute the proportional value to each recipient.

## CR - Code Repetition

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L311

### Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily.

The expression `totalFees.sub(burnFee)` is calculated four times in the same execution thread. The result of this expression will be the same in all of the calculations.

```
uint256 marketingTokens =
contractTokenBalance.mul(marketingFee).div(totalFees.sub(burnFee));
swapAndSendToFee(marketingWalletAddress, marketingTokens);

uint256 ecoSystemTokens =
contractTokenBalance.mul(ecoSystemFee).div(totalFees.sub(burnFee));
swapAndSendToFee(ecoSystemWalletAddress, ecoSystemTokens);

uint256 swapTokens =
contractTokenBalance.mul(liquidityFee).div(totalFees.sub(burnFee));
swapAndLiquify(swapTokens);

uint256 sellTokens =
contractTokenBalance.mul(BUSDRewardsFee).div(totalFees.sub(burnFee));
swapAndSendDividends(sellTokens);
```

### Recommendation

The contract could calculate the expression once in a variable and use the variable on the four occurrences.

## L01 - Public Function could be Declared External

**Criticality**

minor

**Location**

Mini.sol#L125,142,184,201,220,224 and 3 more

### Description

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

```
process  
getAccountAtIndex  
dividendTokenBalanceOf  
...
```

### Recommendation

Use the external attribute for functions never called from the contract

## L02 - State Variables could be Declared Constant

<b>Criticality</b>	minor
<b>Location</b>	Mini.sol#L24

### Description

Constant state variables should be declared constant to save gas.

```
deadWallet
```

### Recommendation

Add the constant attribute to state variables that never change.

## L04 - Conformance to Solidity Naming Conventions

<b>Criticality</b>	minor
<b>Location</b>	Mini.sol#L166,367,26,29,31,529 and 4 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.

```
_account  
BUSDRewardsFee  
_maxTxAmount  
...
```

### Recommendation

Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>

## L12 - Using Variables before Declaration

**Criticality**

minor

**Location**

Mini.sol#L358

### Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
lastProcessedIndex  
iterations  
claims
```

### Recommendation

The variables should be declared before any usage of them.

## L07 - Missing Events Arithmetic

**Criticality**

minor

**Location**

Mini.sol#L166,175,179

### 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 = value * (10 ** 18)
swapTokensAtAmount = value * (10 ** 18)
BUSDRewardsFee = _rewardFee
```

### Recommendation

Emit an event for critical parameter changes.

## L14 - Uninitialized Variables in Local Scope

<b>Criticality</b>	minor
<b>Location</b>	Mini.sol#L358

### Description

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

```
lastProcessedIndex  
claims  
iterations
```

### Recommendation

All the local scoped variables should be initialized.



# Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
<b>Context</b>	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
<b>DividendPayingToken</b>	Implementation	ERC20, Ownable, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface		
	<Constructor>	Public	✓	ERC20
	distributeBUSDDividends	Public	✓	onlyOwner
	withdrawDividend	Public	✓	-
	_withdrawDividendOfUser	Internal	✓	
	dividendOf	Public		-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	
<b>DividendPayingTokenInterface</b>	Interface			
	dividendOf	External		-
	withdrawDividend	External	✓	-

<b>DividendPayingTokenOptionallInterface</b>	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-
<b>ERC20</b>	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
<b>IERC20</b>	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-

<b>IERC20Metadata</b>	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
<b>IterableMapping</b>	Library			
	get	Public		-
	getIndexOfKey	Public		-
	getKeyAtIndex	Public		-
	size	Public		-
	set	Public	✓	-
	remove	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		-
	mint	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		-
<b>IUniswapV2Router02</b>	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
<b>Mini</b>	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	updateDividendTracker	Public	✓	onlyOwner
	updateUniswapV2Router	Public	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	setMarketingWallet	External	✓	onlyOwner
	setEcoSystemWallet	External	✓	onlyOwner
	setFees	External	✓	onlyOwner
	setSwapTokensAmount	External	✓	onlyOwner
	setMaxTxAmount	External	✓	onlyOwner
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateGasForProcessing	Public	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getClaimWait	External		-
	getTotalDividendsDistributed	External		-
	isExcludedFromFees	Public		-
	withdrawableDividendOf	Public		-

	dividendTokenBalanceOf	Public		-
	excludeFromDividends	External	✓	onlyOwner
	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-
	processDividendTracker	External	✓	-
	claim	External	✓	-
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	_transfer	Internal	✓	
	swapAndSendToFee	Private	✓	
	swapAndLiquify	Private	✓	
	swapTokensForEth	Private	✓	
	swapTokensForBUSD	Private	✓	
	addLiquidity	Private	✓	
	swapAndSendDividends	Private	✓	
<b>MiniDividendTracker</b>	Implementation	Ownable, DividendPayingToken		
	<Constructor>	Public	✓	DividendPayingToken
	_transfer	Internal	✓	
	withdrawDividend	Public	✓	-
	excludeFromDividends	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getAccount	Public		-
	getAccountAtIndex	Public		-
	canAutoClaim	Private		
	setBalance	External	✓	onlyOwner
	process	Public	✓	-
	processAccount	Public	✓	onlyOwner
<b>Ownable</b>	Implementation	Context		
	<Constructor>	Public	✓	-

	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
<b>SafeMath</b>	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
<b>SafeMathInt</b>	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
	toUint256Safe	Internal		
<b>SafeMathUint</b>	Library			
	toInt256Safe	Internal		

# Contract Flow



## Domain Info

<b>Domain Name</b>	minions.world
<b>Registry Domain ID</b>	306edba660df468cbdd2b520a6236052-DONUTS
<b>Creation Date</b>	2021-12-16T11:15:36Z
<b>Updated Date</b>	2022-02-08T21:10:59Z
<b>Registry Expiry Date</b>	2022-12-16T11:15:36Z
<b>Registrar WHOIS Server</b>	whois.namecheap.com
<b>Registrar URL</b>	<a href="https://www.namecheap.com/">https://www.namecheap.com/</a>
<b>Registrar</b>	NameCheap, Inc.
<b>Registrar IANA ID</b>	1068

The domain has been created about 2 months before the creation of the audit. It will expire in 10 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 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.

Coinscope 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 Coinscope 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 Coinscope 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 Coinscope team disclaims any liability for the resulting losses.

## About Coinscope

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.

CoinScope is aiming to make crypto discoverable and efficient globally. It provides all the essential tools to assist users draw their own conclusions.



The Coinscope.co team

<https://www.coinscope.co>