



# Audit Report

## **Commander DeFi**

February 2022

Type	BEP20
Network	BSC
Address	0x3cDF630d16B8709c3174784d113B78ee052D95f4
Audited by	© coinscope

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>Contract Review</b>	<b>2</b>
<b>Audit Updates</b>	<b>2</b>
<b>Contract Analysis</b>	<b>3</b>
<b>ST - Stop Transactions</b>	<b>4</b>
<b>Description</b>	<b>4</b>
<b>Recommendation</b>	<b>4</b>
<b>Contract Diagnostics</b>	<b>5</b>
<b>CR - Code Repetition</b>	<b>6</b>
<b>Description</b>	<b>6</b>
<b>Recommendation</b>	<b>6</b>
<b>L04 - Conformance to Solidity Naming Conventions</b>	<b>8</b>
<b>Description</b>	<b>8</b>
<b>Recommendation</b>	<b>8</b>
<b>L15 - Local Scope Variable Shadowing</b>	<b>9</b>
<b>Description</b>	<b>9</b>
<b>Recommendation</b>	<b>9</b>
<b>Contract Functions</b>	<b>10</b>
<b>Contract Flow</b>	<b>14</b>
<b>Domain Info</b>	<b>15</b>
<b>Summary</b>	<b>16</b>
<b>Disclaimer</b>	<b>17</b>
<b>About Coinscope</b>	<b>18</b>

## Contract Review

<b>Contract Name</b>	CommanderDefi
<b>Compiler Version</b>	v0.8.3+commit.8d00100c
<b>Optimization</b>	200 runs
<b>Licence</b>	MIT
<b>Explorer</b>	<a href="https://bscscan.com/token/0x3cDF630d16B8709c3174784d113B78ee052D95f4">https://bscscan.com/token/0x3cDF630d16B8709c3174784d113B78ee052D95f4</a>
<b>Symbol</b>	COMFI
<b>Decimals</b>	18
<b>Total Supply</b>	100,000,000,000,000
<b>Source</b>	contract.sol
<b>Domain</b>	commanderdefi.io

## Audit Updates

<b>Initial Audit</b>	14th 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: blue;">●</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#L442

### 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(from != owner() && to != owner())  
    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.

# Contract Diagnostics

● Critical    ● Medium    ● Minor

Severity	Code	Description
●	CR	Code Repetition
●	L04	Conformance to Solidity Naming Conventions
●	L15	Local Scope Variable Shadowing

## CR - Code Repetition

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L485,498,511,525

### Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily. The following four function calls are called sequentially in four different functions.

```
_takeLiquidity(tLiquidity);  
_takeCommunity(tCommunity);  
_takeMaintenance(tMaintenance);  
_takeLiquidityWallet(tLiquidityWallet);
```

```
function _takeLiquidity(uint256 tLiquidity) private {  
    uint256 currentRate = _getRate();  
    uint256 rLiquidity = tLiquidity.mul(currentRate);  
    _rOwned[address(this)] = _rOwned[address(this)].add(rLiquidity);  
    if(!_isExcluded[address(this)])  
        _tOwned[address(this)] = _tOwned[address(this)].add(tLiquidity);  
}  
  
function _takeCommunity(uint256 tCommunity) private {  
    uint256 currentRate = _getRate();  
    uint256 rCommunity = tCommunity.mul(currentRate);  
    _rOwned[community()] = _rOwned[community()].add(rCommunity);  
    if(!_isExcluded[community()])  
        _tOwned[community()] = _tOwned[community()].add(tCommunity);  
}  
...
```

### Recommendation

There could be one function that is responsible for calling the same function with different fee and address. For instance:

```
_takeFee(address(this), tLiquidity);
```

```
_takeCommunity(community(), tCommunity);  
_takeMaintenance(maintenance(), tMaintenance);  
_takeLiquidityWallet(liquidityWallet(), tLiquidityWallet);
```



## L04 - Conformance to Solidity Naming Conventions

<b>Criticality</b>	minor
<b>Location</b>	Commander DeFi.sol#L364,370,376,382,53,57,58,59,62,65 and 3 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.

```
_maxTxAmount  
_liquidityWalletFee  
_maintenanceFee  
_communityFee  
_taxFee  
_decimals  
_symbol  
_name  
_tTotal  
...
```

### Recommendation

Follow the Solidity naming convention.

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

## L15 - Local Scope Variable Shadowing

**Criticality**

minor

**Location**

Commander DeFi.sol#L141,413

### Description

There are variables that are defined in the local scope containing the same name from an upper scope.

```
owner
```

### Recommendation

The local variables should have different names from the upper scoped variables.

# Contract Functions

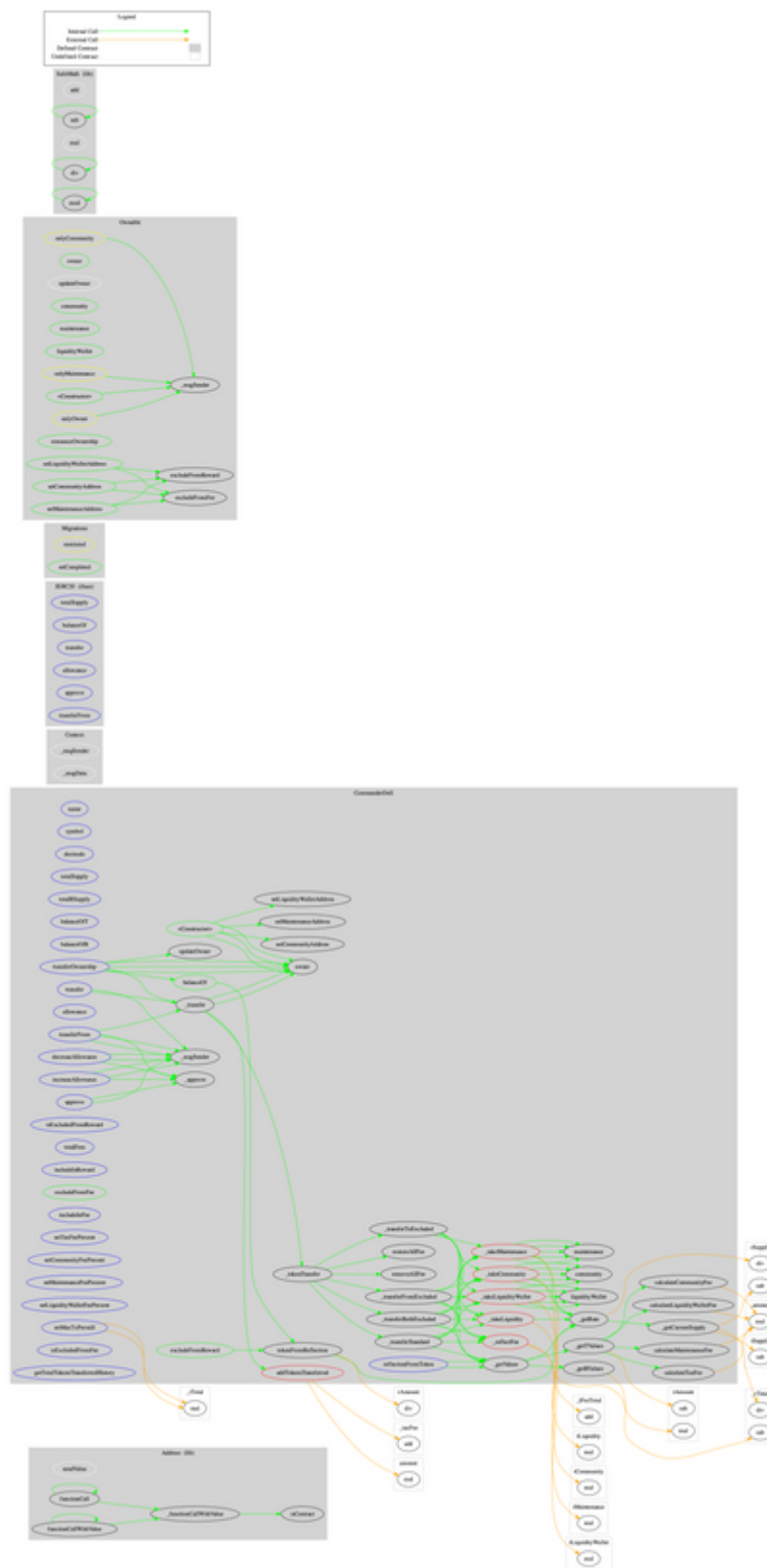
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
<b>Address</b>	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	_functionCallWithValue	Private	✓	
<b>CommanderDeFi</b>	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	totalRSupply	External		onlyOwner
	balanceOf	Public		-
	balanceOfT	External		onlyOwner
	balanceOfR	External		onlyOwner
	transfer	External	✓	-
	transferOwnership	External	✓	onlyOwner
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
	increaseAllowance	External	✓	-
	decreaseAllowance	External	✓	-
	isExcludedFromReward	External		-

	totalFees	External		-
	reflectionFromToken	External		-
	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner
	includeInReward	External	✓	onlyOwner
	excludeFromFee	Public	✓	onlyOwner
	includeInFee	External	✓	onlyOwner
	setTaxFeePercent	External	✓	onlyOwner
	setCommunityFeePercent	External	✓	onlyOwner
	setMaintenanceFeePercent	External	✓	onlyOwner
	setLiquidityWalletFeePercent	External	✓	onlyOwner
	setMaxTxPermill	External	✓	onlyOwner
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	_takeCommunity	Private	✓	
	_takeMaintenance	Private	✓	
	_takeLiquidityWallet	Private	✓	
	calculateTaxFee	Private		
	calculateCommunityFee	Private		
	calculateMaintenanceFee	Private		
	calculateLiquidityWalletFee	Private		
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	isExcludedFromFee	External		-
	_approve	Private	✓	
	addTokensTransferred	Private	✓	
	getTotalTokensTransferredHistory	External		-
	_transfer	Private	✓	
	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	

	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	
	_transferBothExcluded	Private	✓	
<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>Migrations</b>	Implementation			
	setCompleted	Public	✓	restricted
<b>Ownable</b>	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	updateOwner	Internal	✓	onlyOwner
	community	Public		-
	maintenance	Public		-
	liquidityWallet	Public		-
	renounceOwnership	Public	✓	onlyOwner
	excludeFromReward	Public	✓	onlyOwner
	excludeFromFee	Public	✓	onlyOwner
	setCommunityAddress	Public	✓	onlyOwner
	setMaintenanceAddress	Public	✓	onlyOwner
	setLiquidityWalletAddress	Public	✓	onlyOwner
<b>SafeMath</b>	Library			
	add	Internal		

	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		

# Contract Flow



## Domain Info

<b>Domain Name</b>	commanderdefi.io
<b>Registry Domain ID</b>	d3e6976e6a5749a3b938a6aeaf024378-DONUTS
<b>Creation Date</b>	2021-11-22T13:43:56Z
<b>Updated Date</b>	2022-01-05T21:18:35Z
<b>Registry Expiry Date</b>	2022-11-22T13:43:56Z
<b>Registrar WHOIS Server</b>	whois.godaddy.com/
<b>Registrar URL</b>	<a href="http://www.godaddy.com/domains/search.aspx?ci=8990">http://www.godaddy.com/domains/search.aspx?ci=8990</a>
<b>Registrar</b>	GoDaddy.com, LLC
<b>Registrar IANA ID</b>	146

The domain has been created 3 months before the creation of the audit. It will expire in 9 months.

There is no public billing information, the creator is protected by the privacy settings.



## Summary

The Smart Contract analysis reported one medium severity issue. The contract owner has the ability to stop the transactions. Apart from that, the contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. There is also a limit of max 10% fees.

# 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>