



Cyberscope

# Audit Report

## **Crazy Frog**

June 2022

Type           BEP20

Network       BSC

Address       0x6F88331fcAfa1bb6b6434FeFc528Db304e39347c

Audited by   © cyberscope

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>Contract Review</b>	<b>2</b>
<b>Source Files</b>	<b>2</b>
<b>Audit Updates</b>	<b>2</b>
<b>Contract Analysis</b>	<b>3</b>
<b>ULTW - Unlimited Liquidity to Team Wallet</b>	<b>4</b>
<b>Description</b>	<b>4</b>
<b>Recommendation</b>	<b>4</b>
<b>Contract Diagnostics</b>	<b>5</b>
<b>L01 - Public Function could be Declared External</b>	<b>6</b>
<b>Description</b>	<b>6</b>
<b>Recommendation</b>	<b>6</b>
<b>L02 - State Variables could be Declared Constant</b>	<b>7</b>
<b>Description</b>	<b>7</b>
<b>Recommendation</b>	<b>7</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>L05 - Unused State Variable</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>13</b>
<b>Summary</b>	<b>14</b>
<b>Disclaimer</b>	<b>15</b>
<b>About Cyberscope</b>	<b>16</b>

## Contract Review

<b>Contract Name</b>	CrazyFrog
<b>Compiler Version</b>	v0.8.4+commit.c7e474f2
<b>Optimization</b>	200 runs
<b>Licence</b>	None
<b>Explorer</b>	<a href="https://bscscan.com/token/0x6F88331fcAfa1bb6b6434FeFc528Db304e39347c">https://bscscan.com/token/0x6F88331fcAfa1bb6b6434FeFc528Db304e39347c</a>
<b>Symbol</b>	Cfrog
<b>Decimals</b>	9
<b>Total Supply</b>	100,000,000,000

## Source Files

<b>Filename</b>	<b>SHA256</b>
<b>contract.sol</b>	502650e6df764ace17e9aaa70d5224bc806cd5bfbd749ffc5963e630839787b3

## Audit Updates

<b>Initial Audit</b>	4th June 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

## ULTW - Unlimited Liquidity to Team Wallet

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L381

### 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 calling the `manualswap()` and `manualsend()` methods.

```
function manualswap() external {
    require(_msgSender() == _developmentAddress || _msgSender() ==
    _marketingAddress || _msgSender() == owner());
    uint256 contractBalance = balanceOf(address(this));
    swapTokensForEth(contractBalance);
}

function manualsend() external {
    require(_msgSender() == _developmentAddress || _msgSender() ==
    _marketingAddress || _msgSender() == owner());
    uint256 contractETHBalance = address(this).balance;
    sendETHToFee(contractETHBalance);
}
```

### 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
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L05	Unused State Variable

## L01 - Public Function could be Declared External

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L116,122,190,194,198,202,210,215,219,224,306,312,319,393,404,408

### Description

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

```
excludeMultipleAccountsFromFees
toggleSwap
setFee
setNewMarketingAddress
setNewDevAddress
rescueForeignTokens
transferFrom
approve
allowance
...
```

### Recommendation

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

## L02 - State Variables could be Declared Constant

**Criticality**

minor

**Location**

contract.sol#L99

### Description

Constant state variables should be declared constant to save gas.

```
_previousOwner
```

### Recommendation

Add the constant attribute to state variables that never change.



## L04 - Conformance to Solidity Naming Conventions

**Criticality**

minor

**Location**

contract.sol#L40,305,311,318,306,404,138,151,152,153

### 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.

```
_decimals  
_symbol  
_name  
_tTotal  
_swapEnabled  
_amount  
_to  
_tokenAddr  
marketingAddressUpdated  
...
```

### 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#L99,133

### Description

There are segments that contain unused state variables.

```
_tOwned  
_previousOwner
```

### Recommendation

Remove unused state variables.

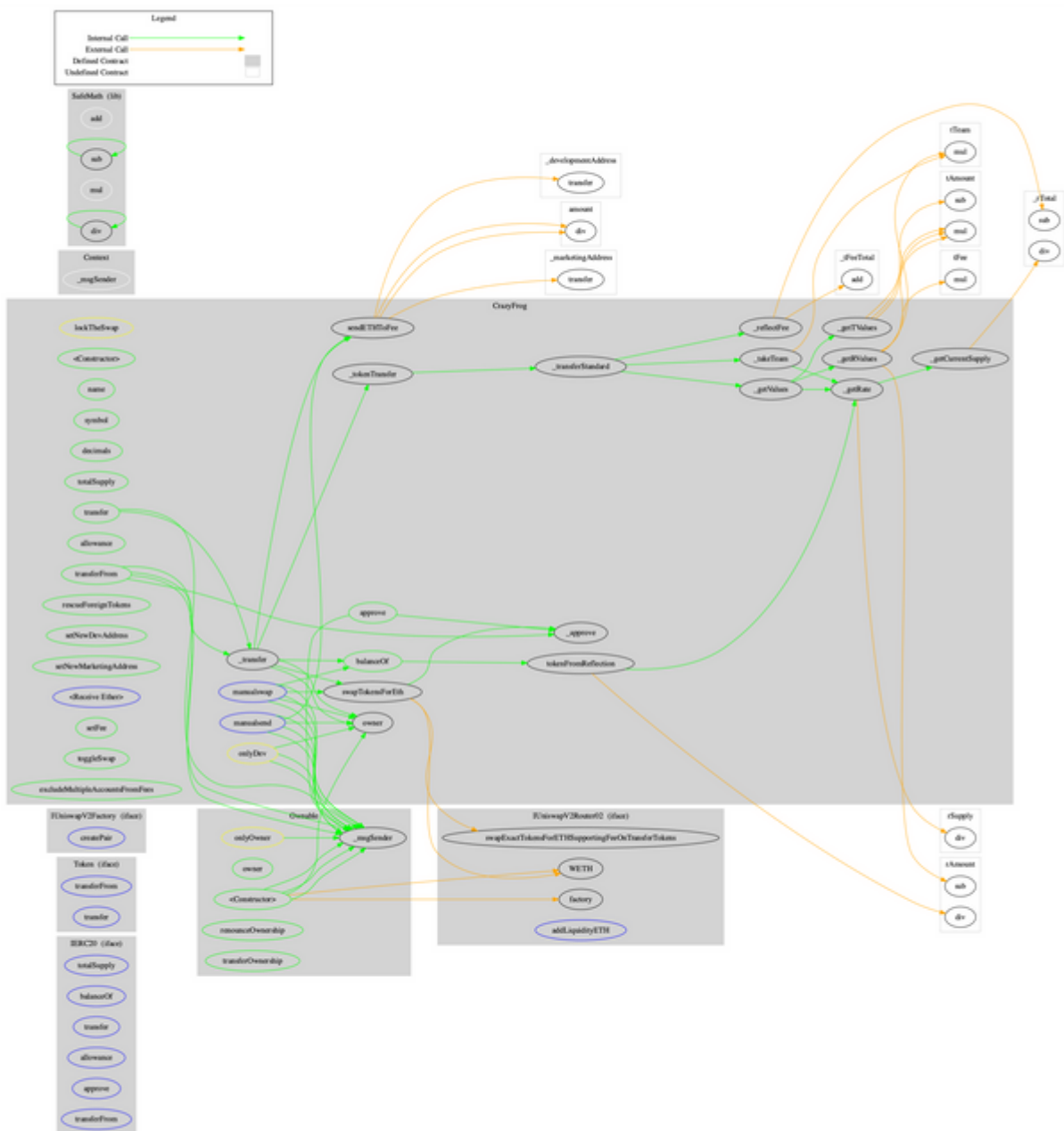
# Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
<b>IERC20</b>	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
<b>Token</b>	Interface			
	transferFrom	External	✓	-
	transfer	External	✓	-
<b>IUniswapV2Factory</b>	Interface			
	createPair	External	✓	-
<b>IUniswapV2Router02</b>	Interface			
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
<b>Context</b>	Implementation			
	_msgSender	Internal		
<b>SafeMath</b>	Library			
	add	Internal		
	sub	Internal		

	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
<b>Ownable</b>	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
<b>CrazyFrog</b>	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	tokenFromReflection	Private		
	_approve	Private	✓	
	_transfer	Private	✓	
	swapTokensForEth	Private	✓	lockTheSwap
	sendETHToFee	Private	✓	
	_tokenTransfer	Private	✓	
	rescueForeignTokens	Public	✓	onlyDev
	setNewDevAddress	Public	✓	onlyDev
	setNewMarketingAddress	Public	✓	onlyDev
	_transferStandard	Private	✓	
	_takeTeam	Private	✓	
	_reflectFee	Private	✓	

	<Receive Ether>	External	Payable	-
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	manualswap	External	✓	-
	manualsend	External	✓	-
	setFee	Public	✓	onlyDev
	toggleSwap	Public	✓	onlyDev
	excludeMultipleAccountsFromFees	Public	✓	onlyOwner

# Contract Flow



## Summary

Crazy Frog is an interesting project that has a friendly and growing community. The Smart Contract analysis reported one minor severity issue. The contract owner has the authority to transfer funds to the team's wallet. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. There is also a limit of max 16% 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.

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>