



# Audit Report

## **YEET Floki**

February 2022

Type	BEP20
Network	BSC
Address	0x894bfFA1574C621211B0EcBbef971e89C09ddC5a
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>7</b>
Description	7
Recommendation	7
<b>BC - Blacklisted Contracts</b>	<b>8</b>
Description	8
Recommendation	8
<b>Contract Diagnostics</b>	<b>9</b>
<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>L07 - Missing Events Arithmetic</b>	<b>13</b>
Description	13
Recommendation	13

<b>L13 - Divide before Multiply Operation</b>	<b>14</b>
<b>Description</b>	<b>14</b>
<b>Recommendation</b>	<b>14</b>
<b>Contract Functions</b>	<b>15</b>
<b>Contract Flow</b>	<b>18</b>
<b>Domain Info</b>	<b>19</b>
<b>Summary</b>	<b>20</b>
<b>Disclaimer</b>	<b>21</b>
<b>About Coinscope</b>	<b>22</b>

## Contract Review

<b>Contract Name</b>	YeetFlokiToken
<b>Compiler Version</b>	v0.8.9+commit.e5eed63a
<b>Optimization</b>	1500 runs
<b>Licence</b>	None
<b>Explorer</b>	<a href="https://bscscan.com/token/0x894bfFA1574C621211B0EcBbef971e89C09ddC5a">https://bscscan.com/token/0x894bfFA1574C621211B0EcBbef971e89C09ddC5a</a>
<b>Symbol</b>	YEET
<b>Decimals</b>	18
<b>Total Supply</b>	1,000,000,000,000
<b>Source</b>	contract.sol
<b>Domain</b>	yeetfloki.com

## Audit Updates

<b>Initial Audit</b>	20th February 2022
<b>Corrected</b>	

# Contract Analysis

● Critical
 ● Medium
 ● Minor
 ● Pass

Severity	Code	Description
<span style="color: red;">●</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: red;">●</span>	BC	Contract Owner is not able to blacklist wallets from selling

## ST - Stop Transactions

Criticality	critical
Location	contract.sol#L418,420

### 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.

```
require(amount <= maxTxAmount , "Amount is exceeding maxTxAmount");
```

The contract owner has the authority to stop the sales or buys for all users excluding the owner. The owner may take advantage of it by setting the `coolDownTime` to a high value. As a result the users will not be able to trade after their first transaction.

```
if(from != pair && coolDownEnabled){
    uint256 timePassed = block.timestamp - _lastTrade[from];
    require(timePassed > coolDownTime, "You must wait coolDownTime");
    _lastTrade[from] = block.timestamp;
}
if(to != pair && coolDownEnabled){
    uint256 timePassed2 = block.timestamp - _lastTrade[to];
    require(timePassed2 > coolDownTime, "You must wait coolDownTime");
    _lastTrade[to] = block.timestamp;
}
```

### Recommendation

The contract could embody a check for not allowing setting the `_maxTxAmount` less and `coolDownTime` more 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

<b>Criticality</b>	critical
<b>Location</b>	contract.sol#L304

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

```
function setTaxes(uint256 _rfi, uint256 _dev, uint256 _liquidity, uint256 _burn)
public onlyOwner {
    taxes.rfi = _rfi;
    taxes.dev = _dev;
    taxes.liquidity = _liquidity;
    taxes.burn = _burn;
    emit FeesChanged();
}
```

### 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.



## BC - Blacklisted Contracts

<b>Criticality</b>	critical
<b>Location</b>	contract.sol#L1

### Description

The contract owner has the authority to massively stop contracts from transactions. The owner may take advantage of it by calling the `bulkAntiBot` function.

```
require(!_isBot[from] && !_isBot[to], "You are a bot");
```

### Recommendation

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
●	L07	Missing Events Arithmetic
●	L13	Divide before Multiply Operation

## L01 - Public Function could be Declared External

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L55,59,191,194,197,201,210,215,219,224 and 10 more

### Description

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

```
rescueAnyERC20Tokens  
isBot  
setTaxes  
isExcludedFromFee  
includeInFee  
excludeFromFee  
reflectionFromToken  
isExcludedFromReward  
decreaseAllowance  
...
```

### Recommendation

Use the external attribute for functions never called from the contract

## L02 - State Variables could be Declared Constant

**Criticality**

minor

**Location**

contract.sol#L115

### Description

Constant state variables should be declared constant to save gas.

```
_tTotal
```

### Recommendation

Add the constant attribute to state variables that never change.

## L04 - Conformance to Solidity Naming Conventions

**Criticality**

minor

**Location**

contract.sol#L77,148,304,546,550,583,112,122,126,129 and 7 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.

```
_symbol  
_name  
deadAddress  
_lastTrade  
_decimals  
_amount  
_to  
_tokenAddr  
_timeInSeconds  
...
```

### Recommendation

Follow the Solidity naming convention.

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

## L07 - Missing Events Arithmetic

**Criticality**

minor

**Location**

contract.sol#L538,542,550

### 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.

```
coolDownTime = _timeInSeconds * 1
swapTokensAtAmount = amount * 10 ** _decimals
maxTxAmount = amount * 10 ** _decimals
```

### Recommendation

Emit an event for critical parameter changes.

## L13 - Divide before Multiply Operation

<b>Criticality</b>	minor
<b>Location</b>	contract.sol#L473

### Description

Performing divisions before multiplications may cause lose of prediction.

```
unitBalance = deltaBalance / (denominator - taxes.liquidity)
```

### Recommendation

The multiplications should be prior to the divisions.

# Contract Functions

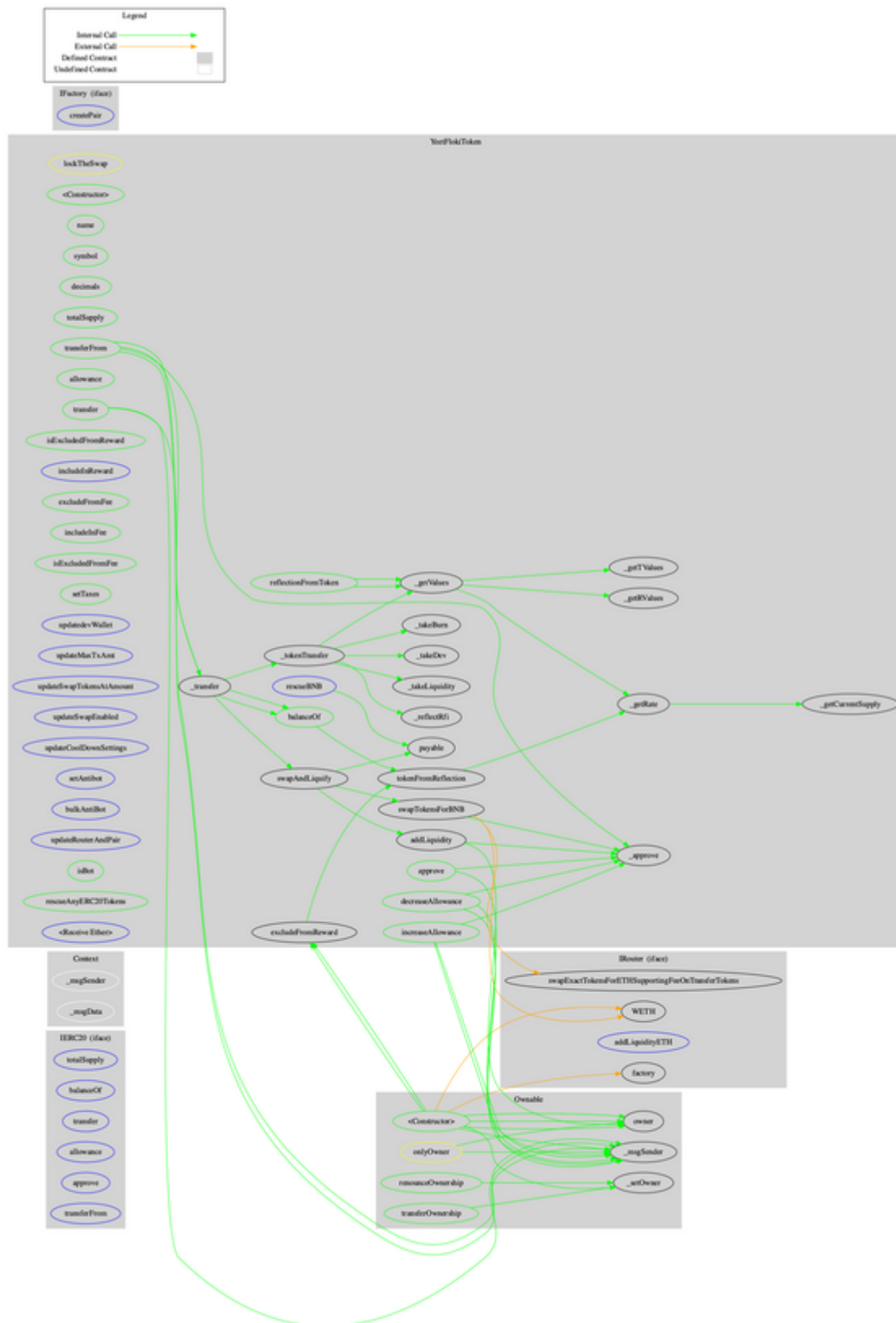
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
<b>IERC20</b>	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
<b>Context</b>	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
<b>Ownable</b>	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
<b>IFactory</b>	Interface			
	createPair	External	✓	-
<b>IRouter</b>	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-



YeetFlokiToken	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	isExcludedFromReward	Public		-
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner
	includeInReward	External	✓	onlyOwner
	excludeFromFee	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	isExcludedFromFee	Public		-
	setTaxes	Public	✓	onlyOwner
	_reflectRfi	Private	✓	
	_takeLiquidity	Private	✓	
	_takeDev	Private	✓	
	_takeBurn	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_approve	Private	✓	
	_transfer	Private	✓	
	_tokenTransfer	Private	✓	

	swapAndLiquify	Private	✓	lockTheSwap
	addLiquidity	Private	✓	
	swapTokensForBNB	Private	✓	
	updatedevWallet	External	✓	onlyOwner
	updateMaxTxAmt	External	✓	onlyOwner
	updateSwapTokensAtAmount	External	✓	onlyOwner
	updateSwapEnabled	External	✓	onlyOwner
	updateCoolDownSettings	External	✓	onlyOwner
	setAntibot	External	✓	onlyOwner
	bulkAntiBot	External	✓	onlyOwner
	updateRouterAndPair	External	✓	onlyOwner
	isBot	Public		-
	rescueBNB	External	✓	onlyOwner
	rescueAnyERC20Tokens	Public	✓	onlyOwner
	<Receive Ether>	External	Payable	-

# Contract Flow



## Domain Info

<b>Domain Name</b>	yeetfloki.com
<b>Registry Domain ID</b>	2673689507_DOMAIN_COM-VRSN
<b>Creation Date</b>	2022-02-08T09:17:44+00:00
<b>Updated Date</b>	2022-02-08T18:24:40+00:00
<b>Registry Expiry Date</b>	2024-02-08T09:17:44+00:00
<b>Registrar WHOIS Server</b>	whois.tucows.co
<b>Registrar URL</b>	<a href="http://tucowsdomains.com">http://tucowsdomains.com</a>
<b>Registrar</b>	TUCOWS, INC.
<b>Registrar IANA ID</b>	69

The domain has been created 12 days before the creation of the audit. It will expire in almost 2 years.

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, blacklisting contracts and stopping transactions. If the contract configuration is abused by the contract owner, then the contract may operate similar to a honeypot. 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>