



# **RugFreeCoins Audit**



## **Shibokami Token**

### **Smart Contract Security Audit**

**February 20, 2022**



# Contents

Audit details	1
Disclaimer	2
Background	3
About the project	4
Target market and the concept	6
Potential to grow with score points	7
Total Points	7
Contract details	8
Contract code function details	11
Contract description table	12
Security issue checking status	18
Owner privileges	19
Audit conclusion	22

# Audit details



**Audited project**  
Shibokami Token



**Contract Address**  
0x2010A1F1e07E3C5A5e5aEC0d4Bc8a3594F54177b



**Client contact**  
Shibokami Token Team



**Blockchain**  
Binance smart chain



**Project website**  
<https://shibokami.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 Shibokami Team to perform an audit of the smart contract.

**<https://bscscan.com/token/0x2010a1f1e07e3c5a5e5aec0d4bc8a3594f54177b>**

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 improving the security posture of the smart contract by remediating the issues that were identified.

# About the project

Shibokami is a token built on the Binance Smart Chain that is with an innovative investment use case the main purpose of which is to seek out constant revenue sources of rewards, burning and strengthening liquidity. Each transaction, purchase incurs a 12% fee, and sale incurs a 14% fee.

## Features

The **sustainability fee of 4% when buying and 6% when selling for marketing** is what allows Shibokami Token to hold the aforementioned promise. Tokens will be swapped into BNB and will be sent to the marketing wallet. This way, Shibokami Token will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.

Shibokami Token has the burn strategy that a **2% fee in each transaction when buying and selling** is getting charged that benefits and rewards those who invest long-term. This feature slowly reduces supply making each Shibokami Token more and more valuable.

The **Shibokami Token rewards** will be distributed among every holder proportional to how many tokens each individual holds in values of **2% when buying selling**.

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.

## Tokenomics

### 12% fee when buying

- 2% of trade goes to holders pockets in tokens
- 2% of trade goes to the burn wallet.
- 4% of trade goes to the marketing wallet.
- 4% of trade goes to the liquidity pool.

### 14% fee when selling

- 2% of trade goes to holders pockets in tokens
- 2% of trade goes to the burn wallet.
- 6% of trade goes to the marketing wallet.
- 4% of trade goes to the liquidity pool.

# Roadmap



## PHASE 1

WEBSITE CREATION  
SOCIAL MEDIA MARKETING  
WHITEPAPER V1  
2500 HOLDERS  
COINGECKO LISTING

## PHASE 2

APPLY FOR COINMARKETCAP  
COMMUNITY CONTESTS &  
GIVEAWAYS  
AUDIT  
5000 HOLDERS



## PHASE 3



INSTAGRAM & TWITTER CREATOR PARTNERSHIPS  
FACEBOOK/GOOGLE ADS  
INFLUENCER PARTNERSHIPS  
CEX LISTING  
WHITEPAPER RELEASE V2.  
10000 HOLDERS

# 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 the Crypto space with long-term investment plans.
- Anyone who's interested in taking part with the future plans of the Shibokami token.
- Anyone who's interested in making financial transactions with any other party using Shibokami as the currency.

## Core concept

### The Shibokami reward system

2% of each transaction when buying and selling is split amongst all holders in Shibokami tokens. Holders will be eligible to receive tokens in every tx and rewards are proportional to how many tokens each individual holds.

### Sustainable mechanism

**The sustainability fee of 4% when buying and 6% when selling for marketing wallet** is what allows Shibokami to promote the token and use funds to further the development of the platform. Tokens will be swapped into BNB and getting sent to the marketing wallet. This way, Shibokami token will have access to the funds without selling tokens as the traditional way, which will enable them to consume funds without hurting the project.

Shibokami token has the burn strategy that **2% fee in each transaction when selling** is getting charged that benefits and rewards those who invest long-term. This feature slowly reduces supply making each Shibokami more and more valuable.

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



# Potential to grow with score points

1.	Project efficiency	9/10
2.	Project uniqueness	7/10
3	Information quality	7/10
4	Service quality	9/10
5	System quality	9/10
6	Impact on the community	8/10
7	Impact on the business	9/10
8	Preparing for the future	7/10
Total Points		<b>8.125/10</b>

# Contract details

## Token contract details for 20<sup>th</sup> February 2022

Contract name	Shibokami
Contract address	0x2010A1F1e07E3C5A5e5aEC0d4Bc8a3594F54177b
Token supply	420,000,000,000,000,000
Token ticker	Shibokami
Decimals	9
Token holders	1,175
Transaction count	10,492
Top 100 holders dominance	0xb6bddf12fa5bde762a4dca43ef9e9794cbfb6f86
Marketing wallet	0xf755f4b39766b5a80125345c979d3d8c56539dac
Contract deployer address	0x11a930e80C624bbA86ed69e6475e35f817EC3304
Contract's current owner address	0x11a930e80c624bba86ed69e6475e35f817ec3304

# 1. Top token holders

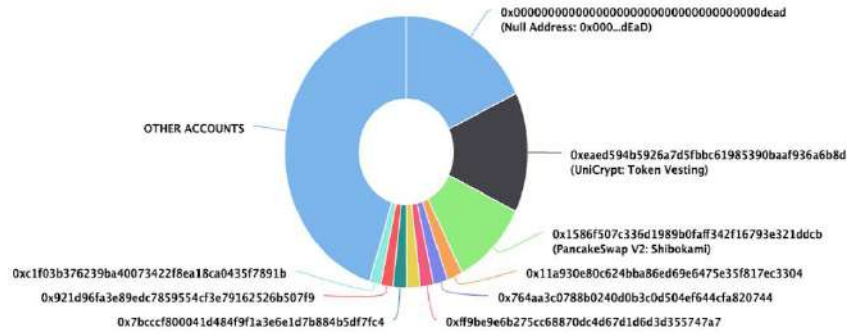
## Top 10 Token Holders

The top 10 holders collectively own 54.90% (230,563,295,148,397,000.00 Tokens) of Shibokami

Token Total Supply: 420,000,000,000,000.00 Token | Total Token Holders: 1,175

Shibokami Top 10 Token Holders

Source: BscScan.com



(A total of 230,563,295,148,397,000.00 tokens held by the top 10 accounts from the total supply of 420,000,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	Null Address: 0x000...dEaD	75,383,341,490,064,000.916441472	17.9484%
2	UniCrypt: Token Vesting	59,969,234,430,262,000.96367435	14.2784%
3	PancakeSwap V2: Shibokami	42,931,563,986,162,600.464971654	10.2218%
4	0x11a930e80c624bba86ed69e6475e35f817ec3304	8,574,339,624,724,530.60363691	2.0415%
5	0x764aa3c0788b0240d0b3c0d504ef644cfa820744	7,892,301,790,638,410.198073555	1.8791%
6	0xff9be9e6b275cc68870dc4d67d1d6d3d355747a7	7,646,424,811,033,620.259389205	1.8206%
7	0xe5a0bd61711bf7cd7f62d1e5813db70d235a7fc	7,498,024,855,363,930.604907059	1.7852%
8	0x7bccc800041d484f9f1a3e6e1d7b884b5df7fc4	7,458,474,550,548,570.923225668	1.7754%
9	0x921d96fa3e89edc7859554cf3e79162526b507f9	6,895,324,288,206,930.147758694	1.6417%
10	0xc1f03b376239ba40073422f8ea18ca0435f7891b	6,316,265,321,393,480.972129398	1.5039%

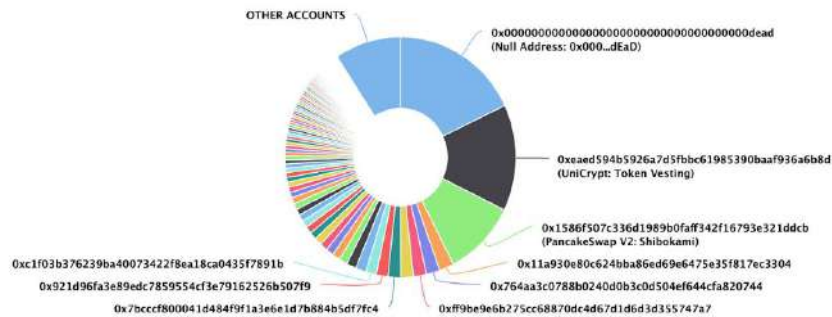
## Top 100 Token Holders

The top 100 holders collectively own 90.74% (381,087,699,564,333,000.00 Tokens) of Shibokami

Token Total Supply: 420,000,000,000,000.00 Token | Total Token Holders: 1,175

### Shibokami Top 100 Token Holders

Source: BscScan.com



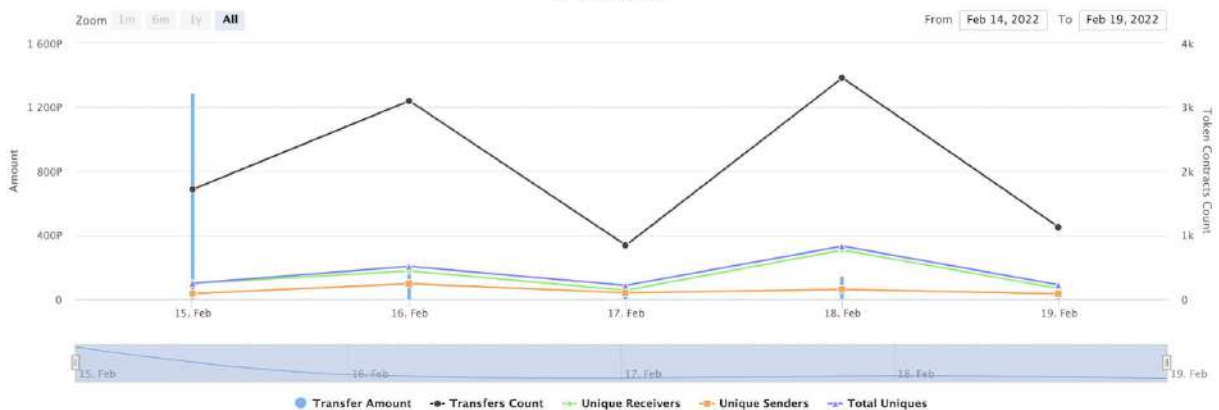
(A total of 381,087,699,564,333,000.00 tokens held by the top 100 accounts from the total supply of 420,000,000,000,000.00 token)

## Contract interaction details

Time Series: Token Contract Overview

Tue 15, Feb 2022 - Sat 19, Feb 2022

Token Contract 0x2010a1f1e07e3c5a5e5aec0d4bc8a3594f54177b (Shibokami)  
Source: BscScan.com






















# 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	pass
		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	pass
		Business logics	pass
		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
<b>IERC20</b>	<b>Interface</b>			
L	totalSupply	External 		NO 
L	balanceOf	External 		NO 
L	transfer	External 		NO 
L	allowance	External 		NO 
L	approve	External 		NO 
L	transferFrom	External 		NO 
<b>Address</b>	<b>Library</b>			
L	sendValue	Internal 		
<b>Context</b>	<b>Implementation</b>			
L	_msgSender	Internal 		
L	_msgData	Internal 		



<b>Ownable</b>	<b>Implementation</b>	<b>Context</b>		
L		Public ⓘ		NO⚡
L	owner	Public ⓘ		NO⚡
L	renounceOwnership	Public ⓘ		onlyOwner
L	transferOwnership	Public ⓘ		onlyOwner
L	_setOwner	Private 		
<b>IFactory</b>	<b>Interface</b>			
L	createPair	External ⓘ		NO⚡
IRouter	Interface			
L	factory	External ⓘ		NO⚡
L	WETH	External ⓘ		NO⚡
L	addLiquidityETH	External ⓘ		NO⚡
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External ⓘ		NO⚡
<b>Shibokami</b>	<b>Implementation</b>	<b>Context, IERC20, Ownable</b>		
L		Public ⓘ		NO⚡
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	reflectionFromToken	Public 		NO 
L	tokenFromReflection	Public 		NO 
L	excludeFromReward	Public 		onlyOwner
L	includeInReward	External 		onlyOwner
L	excludeFromFee	Public 		onlyOwner
L	includeInFee	Public 		onlyOwner
L	isExcludedFromFee	Public 		NO 
L	setSellTaxes	External 		onlyOwner
L	setBuyTaxes	External 		onlyOwner

L	_reflectRfi	Private 		
L	_takeLiquidity	Private 		
L	_takeMarketing	Private 		
L	_takeBurn	Private 		
L	_getValues	Private 		
L	_getTValues	Private 		
L	_getRValues	Private 		
L	_getRate	Private 		
L	_getCurrentSupply	Private 		
L	_approve	Private 		
L	_transfer	Private 		
L	_tokenTransfer	Private 		
L	swapAndLiquify	Private 		lockTheSw ap
L	addLiquidity	Private 		
L	swapTokensForBNB	Private 		
L	airdropTokens	External 		onlyOwner

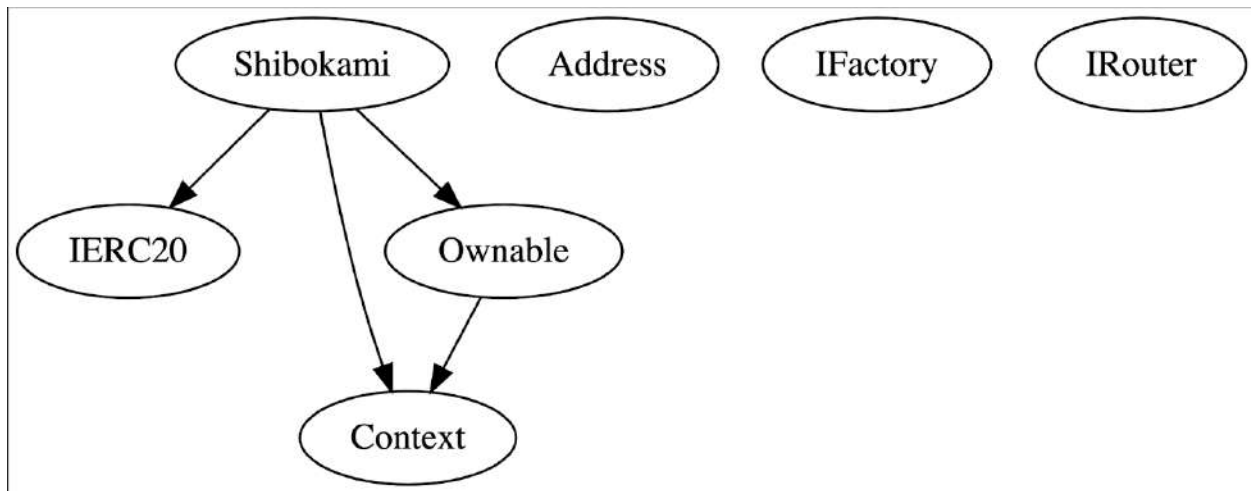
L	updateWallet	External !		onlyOwner
L	updateSwapTokensAtAmount	External !		onlyOwner
L	setMaxWallet	External !		onlyOwner
L	updateSwapEnabled	External !		onlyOwner
L	updateRouterAndPair	External !		onlyOwner
L	rescueBNB	External !		onlyOwner
L	rescueAnyBEP20Tokens	Public !		onlyOwner
L		External !		NO!

### 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**

No medium severity issues found.

- ❖ **Low severity issues**

No low severity issues found.

# Owner privileges

- ❖ The owner can include/exclude wallets from 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 not 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 include/exclude wallets from fees

```
ftrace | funcSig
function excludeFromFee(address account↑) public onlyOwner {
    _isExcludedFromFee[account↑] = true;
}

ftrace | funcSig
function includeInFee(address account↑) public onlyOwner {
    _isExcludedFromFee[account↑] = false;
}
```

- ❖ The owner can change all buy and sell fees maximum up to 22%

```
fttrace | funcSig
function setSellTaxes(uint256 _rfi↑, uint256 _marketing↑, uint256 _liquidity↑, uint256 _burn↑) external onlyOwner {
    require(_rfi↑ + _marketing↑ + _liquidity↑ + _burn↑ <= 22, "Max Fee is 22%");
    sellTaxes = Taxes(_rfi↑, _marketing↑, _liquidity↑, _burn↑);
}

fttrace | funcSig
function setBuyTaxes(uint256 _rfi↑, uint256 _marketing↑, uint256 _liquidity↑, uint256 _burn↑) external onlyOwner {
    require(_rfi↑ + _marketing↑ + _liquidity↑ + _burn↑ <= 22, "Max Fee is 22%");
    buyTaxes = Taxes(_rfi↑, _marketing↑, _liquidity↑, _burn↑);
}
```

- ❖ The owner can airdrop tokens

```
fttrace | funcSig
function airdropTokens(address[] memory recipients↑, uint256[] memory amounts↑) external onlyOwner{
    require(recipients↑.length == amounts↑.length, "Invalid size");
    address sender = msg.sender;
    for(uint256 i; i<recipients↑.length; i++){
        address recipient = recipients↑[i];
        uint256 rAmount = amounts↑[i]*_getRate();
        rOwned[sender] = rOwned[sender]- rAmount;
        rOwned[recipient] = rOwned[recipient] + rAmount;
        emit Transfer(sender, recipient, amounts↑[i]);
    }
}
```

- ❖ The owner can update marketing wallet

```
fttrace | funcSig
function updateWallet(address newMarketing↑) external onlyOwner{
    marketingWallet = newMarketing↑;
}
```

- ❖ The owner can change token swap point

```
fttrace | funcSig
function updateSwapTokensAtAmount(uint256 amount↑) external onlyOwner{
    swapTokensAtAmount = amount↑ * 10**_decimals;
}
```

- ❖ The owner can change max wallet limit

```
ftrace | funcSig
function setMaxWallet(uint256 amount↑) external onlyOwner{
    maxWalletBalance = amount↑ * 10**_decimals;
}
```

- ❖ The owner can enable/disable swapping

```
ftrace | funcSig
function updateSwapEnabled(bool _enabled↑) external onlyOwner{
    swapEnabled = _enabled↑;
}
```

- ❖ The owner can update router address and pair address

```
ftrace | funcSig
function updateRouterAndPair(address newRouter↑, address newPair↑) external onlyOwner{
    router = IRouter(newRouter↑);
    pair = newPair↑;
}
```

- ❖ The owner can get BNB in contract to owner wallet

```
//Use this in case BNB are sent to the contract by mistake
ftrace | funcSig
function rescueBNB(uint256 weiAmount↑) external onlyOwner{
    require(address(this).balance >= weiAmount↑, "insufficient BNB balance");
    payable(msg.sender).transfer(weiAmount↑);
}
```

- ❖ The owner can get all other tokens inside the contract to the owner wallet

```
ftrace | funcSig
function rescueAnyBEP20Tokens(address _tokenAddr↑, address _to↑, uint _amount↑) public onlyOwner {
    IERC20(_tokenAddr↑).transfer(_to↑, _amount↑);
}
```



# Audit conclusion

While conducting the audit of the Shibokami smart contract, it was observed that there is nothing alarming with the code.