



RugFreeCoins Audit



Shibetoshi Token

Smart Contract Security Audit

February 28, 2022

Contents

Audit details	1
Disclaimer	2
Background	3
About the project	4
Target market and the concept	5
Potential to grow with score points	6
Total Points	6
Contract details	7
Contract code function details	8
Contract description table	10
Security issue checking status	20
Audit conclusion	24

Audit details



Audited project

Shibetoshi Token



Contract Address

0x51F11A891110339352988B84057e496FE09E23c4



Client contact

Shibetoshi Team



Blockchain

Binance smart chain



Project website

<https://shibetoshi-token.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 the Shibetoshi Team to perform an audit of the smart contract.

<https://bscscan.com/token/0x51F11A891110339352988B84057e496FE09E23c4>

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

Shibetoshi Token is a token built on the Binance Smart Chain that is with an innovative investment use case the main purpose of which is to help innocent people from Ukraine. Each transaction, purchase and sale incur 12% fee.

Features

- The **shibetoshi rewards** will be distributed among every holder proportional to how many tokens each individual holds in values of **2% when buying and selling**.
- The **sustainability fee of 2% when buying and selling for marketing** is what allows shibetoshi to hold the aforementioned promise. Tokens will be sent to a marketing wallet per transaction. This way, shibetoshi will have enough funds to promote the coin.
- 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.
- **6% Charity fee** per transaction will be sent to a wallet in tokens to help innocent people from Ukraine. This will empower the shibetoshi Token community in the long run.

Tokenomics

12% fee when buying and selling

- 2% of trade goes to holders pockets in shibetoshi tokens.
- 8% of trade goes to the charity & marketing wallet
- 2% of trade goes to the liquidity pool.

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 collecting NFTs or trading NFTs.
- Anyone who's interested in playing anti NFT war game.
- Anyone who's interested in taking part with the future plans of the shibetoshi token.
- Anyone who's interested in making financial transactions with any other party using shibetoshi as the currency.

Core concept

The shibetoshi reward system

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

Sustainable mechanism

The **sustainability fee of 2% when buying and selling for marketing** is what allows shibetoshi to promote the token and use funds to further the development of the platform. Tokens will be sent to a marketing wallet per transaction.

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

6% Charity fee per transaction will be sent to a wallet in tokens to help innocent people from Ukraine. This will empower the shibetoshi Token community in the long run.

Potential to grow with score points

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

Contract details

Token contract details for 28th February 2022

Contract name	Shibetoshi Token
Contract address	0x51F11A891110339352988B84057e496FE09E23c4
Token supply	1,000,000,000,000
Token ticker	SHIBETOSHI
Decimals	9
Token holders	1
Transaction count	1
Marketing wallet	0xf6955a6e1f5ecfa0e07d8d19cec0dc1b2b768cb9
Contract deployer address	0x887221Df47339E589C6E736F018890cf27e28fB0
Contract's current owner address	0x887221df47339e589c6e736f018890cf27e28fb0







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	High security
		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













The 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 their visibility and mutability.












Contract	Type	Bases		
L	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
L	totalSupply	External !		NO!
L	balanceOf	External !		NO!
L	transfer	External !		NO!
L	allowance	External !		NO!
L	approve	External !		NO!
L	transferFrom	External !		NO!
SafeMath	Library			
L	add	Internal 		
L	sub	Internal 		
L	sub	Internal 		












































L	mul	Internal 		
L	div	Internal 		
L	div	Internal 		
L	mod	Internal 		
L	mod	Internal 		
Context	Implementation			
L	_msgSender	Internal 		
L	_msgData	Internal 		
Address	Library			
L	isContract	Internal 		
L	sendValue	Internal 		
L	functionCall	Internal 		
L	functionCall	Internal 		
L	functionCallWithValue	Internal 		
L	functionCallWithValue	Internal 		
L	_functionCallWithValue	Private 		
Ownable	Implementation	Context		





















L		Public 		NO 
L	owner	Public 		NO 
L	renounceOwnership	Public 		onlyOwner
L	transferOwnership	Public 		onlyOwner
L	geUnlockTime	Public 		NO 
L	lock	Public 		onlyOwner
L	unlock	Public 		NO 
IUniswap V2Factory	Interface			
L	feeTo	External 		NO 
L	feeToSetter	External 		NO 
L	getPair	External 		NO 
L	allPairs	External 		NO 
L	allPairsLength	External 		NO 
L	createPair	External 		NO 
L	setFeeTo	External 		NO 
L	setFeeToSetter	External 		NO 
IUniswap V2Pair	Interface			
L	name	External 		NO 





























L	symbol	External !		NO!
L	decimals	External !		NO!
L	totalSupply	External !		NO!
L	balanceOf	External !		NO!
L	allowance	External !		NO!
L	approve	External !		NO!
L	transfer	External !		NO!
L	transferFrom	External !		NO!
L	DOMAIN_SEPARATOR	External !		NO!
L	PERMIT_TYPEHASH	External !		NO!
L	nonces	External !		NO!
L	permit	External !		NO!
L	MINIMUM_LIQUIDITY	External !		NO!
L	factory	External !		NO!
L	token0	External !		NO!
L	token1	External !		NO!
L	getReserves	External !		NO!
L	price0CumulativeLast	External !		NO!

L	price1CumulativeLast	External !		NO!
L	kLast	External !		NO!
L	mint	External !		NO!
L	burn	External !		NO!
L	swap	External !		NO!
L	skim	External !		NO!
L	sync	External !		NO!
L	initialize	External !		NO!
IUniswap V2Router 01	Interface			
L	factory	External !		NO!
L	WETH	External !		NO!
L	addLiquidity	External !		NO!
L	addLiquidityETH	External !		NO!
L	removeLiquidity	External !		NO!
L	removeLiquidityETH	External !		NO!
L	removeLiquidityWithPermit	External !		NO!
L	removeLiquidityETHWithPermit	External !		NO!



L	swapExactTokensForTokens	External !		NO!
L	swapTokensForExactTokens	External !		NO!
L	swapExactETHForTokens	External !		NO!
L	swapTokensForExactETH	External !		NO!
L	swapExactTokensForETH	External !		NO!
L	swapETHForExactTokens	External !		NO!
L	quote	External !		NO!
L	getAmountOut	External !		NO!
L	getAmountIn	External !		NO!
L	getAmountsOut	External !		NO!
L	getAmountsIn	External !		NO!
IUniswap V2Router 02	Interface	IUniswapV2Router01		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External !		NO!
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External !		NO!
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External !		NO!
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External !		NO!
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External !		NO!

shibetoshi	Implementation	Context, IERC20, Ownable		
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	totalFees	Public 		NO 
L	deliver	Public 		NO 
L	reflectionFromToken	Public 		NO 
L	tokenFromReflection	Public 		NO 
L	excludeFromReward	Public 		onlyOwner

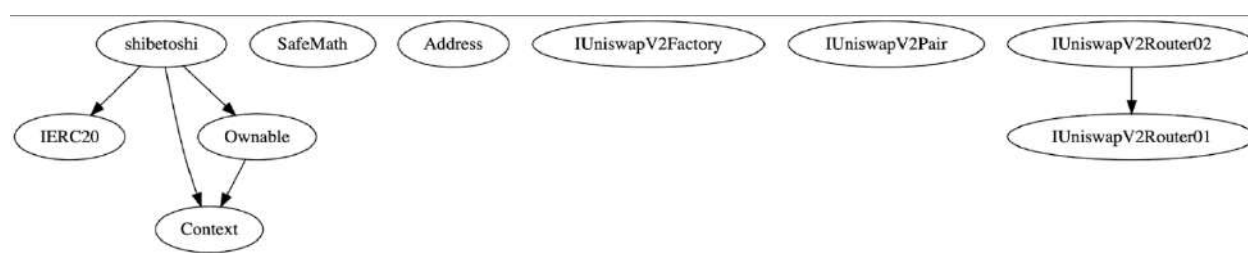
L	includeInReward	External !		onlyOwner
L	_transferBothExcluded	Private 		
L	excludeFromFee	Public !		onlyOwner
L	includeInFee	Public !		onlyOwner
L	setTaxFeePercent	External !		onlyOwner
L	setLiquidityFeePercent	External !		onlyOwner
L	setMaxTxPercent	External !		onlyOwner
L	setMarketingFeePercent	External !		onlyOwner
L	setMarketingWallet	External !		onlyOwner
L	setSwapAndLiquifyEnabled	Public !		onlyOwner
L	changeNumTokensSellToAddToLiquidity	External !		onlyOwner
L		External !		NO!
L	_reflectFee	Private 		
L	_getValues	Private 		
L	_getTValues	Private 		
L	_getRValues	Private 		
L	_getRate	Private 		
L	_getCurrentSupply	Private 		

L	_takeLiquidity	Private 		
L	calculateTaxFee	Private 		
L	calculateLiquidityFee	Private 		
L	removeAllFee	Private 		
L	restoreAllFee	Private 		
L	isExcludedFromFee	Public 		NO 
L	_approve	Private 		
L	_transfer	Private 		
L	swapAndLiquify	Private 		lockTheSw ap
L	swapTokensForEth	Private 		
L	addLiquidity	Private 		
L	_tokenTransfer	Private 		
L	_transferStandard	Private 		
L	_transferToExcluded	Private 		
L	_transferFromExcluded	Private 		

Legend

Symbol	Meaning
	Function can modify state
	Function is payable

Inheritance Hierarchy



Security issue checking status

- **High severity issues**

- ❖ The owner can change all fees without max limit (can set 100%)

```
ftrace | funcSig
function setTaxFeePercent(uint256 taxFee↑) external onlyOwner {
    _taxFee = taxFee↑;
}

ftrace | funcSig
function setLiquidityFeePercent(uint256 liquidityFee↑) external onlyOwner {
    _liquidityFee = liquidityFee↑;
}
```

- ❖ The owner can change max transaction amount without minimum requirement

```
ftrace | funcSig
function setMaxTxPercent(uint256 maxTxPercent↑) external onlyOwner {
    _maxTxAmount = _tTotal.mul(maxTxPercent↑).div(10**2);
}
```

- **Medium severity issues**

No medium severity issues found

- **Low severity issues**

No low severity issues found

Owner privileges

- ❖ The owner can exclude wallets from rewards

```
ftrace | funcSig
function excludeFromReward(address account↑) public onlyOwner {
    // require(account != 0x10ED43C718714eb63d5aA57B78B54704E256024E, 'We can not exclude Uniswap router.');
```

```
    require(!_isExcluded[account↑], "Account is already excluded");
    if (_rOwned[account↑] > 0) {
        _tOwned[account↑] = tokenFromReflection(_rOwned[account↑]);
    }
    _isExcluded[account↑] = true;
    _excluded.push(account↑);
}
```

- ❖ The owner can include wallets from rewards

```
ftrace | funcSig
function includeInReward(address account↑) external onlyOwner {
    require(!_isExcluded[account↑], "Account is already 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 reward fee

```
ftrace | funcSig
function setTaxFeePercent(uint256 taxFee↑) external onlyOwner {
    _taxFee = taxFee↑;
}
ftrace | funcSig
```

- ❖ The owner can change liquidity fee

```
ftrace | funcSig
function setLiquidityFeePercent(uint256 liquidityFee↑) external onlyOwner {
    _liquidityFee = liquidityFee↑;
}
ftrace | funcSig
```

- ❖ The owner can change max transaction amount

```
ftrace | funcSig
function setMaxTxPercent(uint256 maxTxPercent↑) external onlyOwner {
    _maxTxAmount = _tTotal.mul(maxTxPercent↑).div(10**2);
}
ftrace | funcSig
```

- ❖ The owner can change marketing fee

```
ftrace | funcSig
function setMarketingFeePercent(uint256 marketingFee↑) external onlyOwner {
    _MKTshare = marketingFee↑;
}
ftrace | funcSig
```

- ❖ The owner can change marketing wallet

```
ftrace | funcSig
function setMarketingWallet(address _add↑) external onlyOwner {
    _MARKETING_ADDRESS = _add↑;
}
ftrace | funcSig
```

- ❖ The owner can enable/disable swap and change swap point

```
ftrace | funcSig
function setSwapAndLiquifyEnabled(bool _enabled↑) public onlyOwner {
    swapAndLiquifyEnabled = _enabled↑;
    emit SwapAndLiquifyEnabledUpdated(_enabled↑);
}

//write number with 9 zeros because of 9 decimals
ftrace | funcSig
function changeNumTokensSellToAddToLiquidity(
    uint256 _numTokensSellToAddToLiquidity↑
) external onlyOwner {
    numTokensSellToAddToLiquidity = _numTokensSellToAddToLiquidity↑;
}
```

Audit conclusion

While conducting the audit of the shibetoshi smart contract, it was observed that there is nothing alarming with the code in functional wise and it contains two high severity issues since the owner has substantial control within the ecosystem.