RugFreeCoins Audit

Sonic Bytes Token

Smart Contract Security Audit

April 13, 2022

Contents

Audit details

Audited project
Sonic Bytes

Contract Address
0xFE50f64993b80B5b9CCA77F81F50B93f6330221f

Client contact
Sonic Bytes Team

Blockchain
Binance smart chain

Project website
https://sonicbytes.net

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 Sonic Bytes Team to perform an audit of the smart contract.

https://bscscan.com/address/0xFE50f64993b80B5b9CCA77F81F50B93f6330221f#code

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

Sonic Bytes 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 seek out constant revenue sources, and heading towards building even greater Community, with safe tokenomics and play to earn game. Each transaction, purchase and sale incur 10% fee.

Features

- The **Sonic Bytes reflections** will be distributed in tokens among every holder proportional to how many tokens each individual holds in values of **5% when buying and selling.**

- The additional component included under the sustainability section is a **liquidity fee of 3% when buying and selling**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.
- **Sonic Bytes** has a burn strategy that benefits and rewards those who invest long-term. This feature slowly reduces supply making each Sonic Bytes more and more valuable. 2% fee goes to the burn wallet buying and selling.

Roadmap

Jan - March 2022

- Create our Advanced Smart Contract
- Test, Check, and Security Scan SONIC Bytes Contract
- Gnosis Safe Multisig Wallets for Charity & Developer wallet
- Deploy on Binance Smart Chain
- Verify Contract
- SSL Website and Social Media Sites
- Whitepaper
- Presale
- Pancakeswap
- Advertising / Marketing

April - June 2022

- Play to Earn Tokens on Mobile App
- Play to Earn Tokens Online Game
- Advertisement System to integrate into Games.
- Mobile Wallet
- Certik Audit
- Gate.io Exchange, MEXC Exchange and more
- Coin Market Cap, Coin Gecko and more listings

June - August 2022

- Continued Development of our Games
- Integrating of features and upgrades
- Upgrades of our Advertisement System
- Webpage updates
- Exchange Listings

Tokenomics

10% fee when buying and buying

- 5% of trade goes to holders’ pockets in tokens.
- 3% of trade goes to the liquidity pool.
- 2% of trade goes to the burn wallet.

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 by holding tokens.
- Anyone who's interested in trading tokens.
- Anyone who's interested in playing and earning tokens through P2E games.
- Anyone who's interested in taking part with the future plans of the Sonic Bytes.
- Anyone who's interested in making financial transactions with any other party using Sonic Bytes as the currency.

Core concept

The Sonic Bytes reward system

5% of each transaction when buying and selling is split amongst all holders. Holders will be eligible to receive tokens in each transaction and rewards are proportional to how many tokens each individual holds.

Sustainable mechanism

The liquidity fee of 3% when buying and selling, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

Sonic Bytes has the burn strategy that benefits and rewards those who invest long-term. This feature slowly reduces supply making Sonic Bytes price more and more valuable.

Potential to grow with score points

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

Contract details

Token contract details for 13th April 2022

Contract name	SonicBytes
Contract address	0xFE50f64993b80B5b9CCA77F81F50B93f6330221f
Token supply	1,000,000,000,000,000
Token ticker	BYTES
Decimals	18
Token holders	1
Transaction count	1
Contract deployer address	0x1e8B9da44Cb33b57cbCCBAc74BE8f058c84C92a1
Contract's current owner address	0x1e8B9da44Cb33b57cbCCBAc74BE8f058c84C92a1

Token Distribution

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

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**		
└	Function Name	Visibility	Mutability	Modifiers
IPinkAntiBot	**Interface**			
└	setTokenOwner	External ❗	🛑	NO❗
└	onPreTransferCheck	External ❗	🛑	NO❗
IERC20	**Interface**			
└	totalSupply	External ❗		NO❗
└	decimals	External ❗		NO❗
└	symbol	External ❗		NO❗
└	name	External ❗		NO❗
└	getOwner	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	🛑	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗

SafeMath	**Library**			
└	tryAdd	Internal 🔒		
└	trySub	Internal 🔒		
└	tryMul	Internal 🔒		
└	tryDiv	Internal 🔒		
└	tryMod	Internal 🔒		
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	sub	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
Context	**Implementation**			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Enumerable Set	**Library**			
└	_add	Private 🔐	🛑	

L	_remove	Private		
L	_contains	Private		
L	_length	Private		
L	_at	Private		
L	add	Internal		
L	remove	Internal		
L	contains	Internal		
L	length	Internal		
L	at	Internal		
L	add	Internal		
L	remove	Internal		
L	contains	Internal		
L	length	Internal		
L	at	Internal		
L	add	Internal		
L	remove	Internal		
L	contains	Internal		
L	length	Internal		
L	at	Internal		
Address	**Library**			

└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	_functionCallWithValue	Private 🔐	🛑	
Ownable	**Implementation**	**Context**		
└		Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
IUniswapV2 Factory	**Interface**			
└	feeTo	External ❗		NO❗
└	feeToSetter	External ❗		NO❗
└	getPair	External ❗		NO❗
└	allPairs	External ❗		NO❗
└	allPairsLength	External ❗		NO❗
└	createPair	External ❗	🛑	NO❗

└	setFeeTo	External ❗	🛑	NO❗
└	setFeeToSetter	External ❗	🛑	NO❗
IUniswapV2 Pair	**Interface**			
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transfer	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
└	DOMAIN_SEPARATOR	External ❗		NO❗
└	PERMIT_TYPEHASH	External ❗		NO❗
└	nonces	External ❗		NO❗
└	permit	External ❗	🛑	NO❗
└	MINIMUM_LIQUIDITY	External ❗		NO❗
└	factory	External ❗		NO❗
└	token0	External ❗		NO❗
└	token1	External ❗		NO❗

└	getReserves	External ❗		NO❗
└	price0CumulativeLast	External ❗		NO❗
└	price1CumulativeLast	External ❗		NO❗
└	kLast	External ❗		NO❗
└	mint	External ❗	🛑	NO❗
└	burn	External ❗	🛑	NO❗
└	swap	External ❗	🛑	NO❗
└	skim	External ❗	🛑	NO❗
└	sync	External ❗	🛑	NO❗
└	initialize	External ❗	🛑	NO❗
IUniswapV2 Router01	**Interface**			
└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
└	addLiquidity	External ❗	🛑	NO❗
└	addLiquidityETH	External ❗	💵	NO❗
└	removeLiquidity	External ❗	🛑	NO❗
└	removeLiquidityETH	External ❗	🛑	NO❗
└	removeLiquidityWithPermit	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermit	External ❗	🛑	NO❗
└	swapExactTokensForTokens	External ❗	🛑	NO❗

└	swapTokensForExactTokens	External ❗	🛑	NO❗
└	swapExactETHForTokens	External ❗	💵	NO❗
└	swapTokensForExactETH	External ❗	🛑	NO❗
└	swapExactTokensForETH	External ❗	🛑	NO❗
└	swapETHForExactTokens	External ❗	💵	NO❗
└	quote	External ❗		NO❗
└	getAmountOut	External ❗		NO❗
└	getAmountIn	External ❗		NO❗
└	getAmountsOut	External ❗		NO❗
└	getAmountsIn	External ❗		NO❗
IUniswapV2 Router02	**Interface**	**IUniswapV 2Router01**		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
Sonic_Bytes	**Implementation**	**Context, IERC20, Ownable**		
└		Public ❗	🛑	NO❗

└	setEnableAntiBot	External ❗	🛑	onlyOwner
└	symbol	External ❗		NO❗
└	name	External ❗		NO❗
└	decimals	External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	getOwner	External ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	transfer	Public ❗	🛑	NO❗
└	allowance	Public ❗		NO❗
└	approve	Public ❗	🛑	NO❗
└	transferFrom	Public ❗	🛑	NO❗
└	increaseAllowance	Public ❗	🛑	NO❗
└	decreaseAllowance	Public ❗	🛑	NO❗
└	isExcludedFromReward	External ❗		NO❗
└	totalFees	External ❗		NO❗
└	setNewTaxes	External ❗	🛑	onlyOwner
└	deliver	External ❗	🛑	NO❗
└	reflectionFromToken	External ❗		NO❗
└	tokenFromReflection	Public ❗		NO❗
└	setRouterAddress	External ❗	🛑	onlyOwner
└	excludeFromReward	External ❗	🛑	onlyOwner

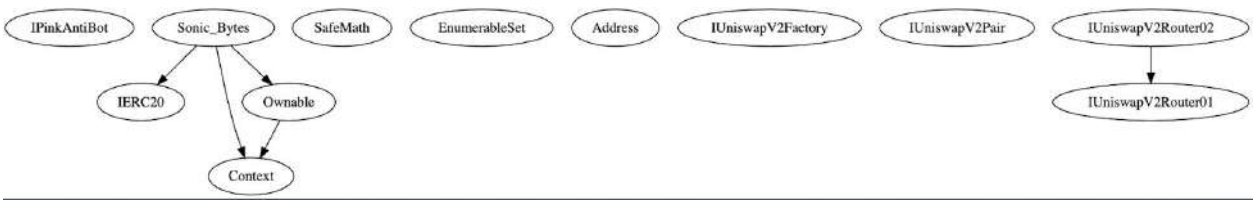
└	includeInReward	External ❗	🛑	onlyOwner
└	_transferBothExcluded	Private 🔐	🛑	
└	excludeFromFee	External ❗	🛑	onlyOwner
└	includeInFee	External ❗	🛑	onlyOwner
└	setNumTokensSellToAddToLiquidity	External ❗	🛑	onlyOwner
└	setSwapAndLiquifyEnabled	External ❗	🛑	onlyOwner
└		External ❗	💵	NO❗
└	_reflectFee	Private 🔐	🛑	
└	_getValues	Private 🔐		
└	_getTValues	Private 🔐		
└	_getRValues	Private 🔐		
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	_takeLiquidity	Private 🔐	🛑	
└	_takeBurn	Private 🔐	🛑	
└	removeAllFee	Private 🔐	🛑	
└	restoreAllFee	Private 🔐	🛑	
└	isExcludedFromFee	External ❗		NO❗
└	_approve	Private 🔐	🛑	
└	_transfer	Private 🔐	🛑	
└	swapAndLiquify	Private 🔐	🛑	lockTheSwap

└	swapTokensForEth	Private		
└	addLiquidity	Private		
└	_tokenTransfer	Private		
└	enableDisableBot	External		onlyOwner
└	_transferStandard	Private		
└	_transferToExcluded	Private		
└	_transferFromExcluded	Private		

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

- **Informational**

❖ After swapping and adding liquidity if the contract has remaining BNB of more than 0.01, that amount will transfer to the dev wallet.

- **Centralization risk**

No Centralization issues found

Owner privileges

- The owner can enable/disable pink anti bot

```
ftrace | funcSig
function setEnableAntiBot(bool _enable⬆) external onlyOwner {
    antiBotEnabled = _enable⬆;
}
```

- The owner can change all fees maximum up to 15%

```
ftrace | funcSig
function setNewTaxes(
    uint256 _newtaxfee⬆,
    uint256 _newburnFee⬆,
    uint256 _newlpfee⬆
) external onlyOwner {
    uint256 subtotal = _newtaxfee⬆.add(_newburnFee⬆).add(_newlpfee⬆);
    require(subtotal <= MAX_TAX_FEE, "Tax fee exceeded from 15%");

    _previousTaxFee = _taxFee;
    _taxFee = _newtaxfee⬆;

    _previousBurnFee = _burnFee;
    _burnFee = _newburnFee⬆;

    _previousLiquidityFee = _liquidityFee;
    _liquidityFee = _newlpfee⬆;
}
```

❖ The owner can change the router address

```
ftrace | funcSig
function setRouterAddress(address newRouterAddress⬆) external onlyOwner {
    require(
        newRouterAddress⬆ != address(uniswapV2Router),
        "This router is already used."
    );
    uniswapV2Router = IUniswapV2Router02(newRouterAddress⬆);
    uniswapV2Pair = IUniswapV2Factory(uniswapV2Router.factory()).getPair(
        address(this),
        uniswapV2Router.WETH()
    );
    if (uniswapV2Pair == address(0)) {
        uniswapV2Pair = IUniswapV2Factory(uniswapV2Router.factory())
            .createPair(address(this), uniswapV2Router.WETH());
    }
}
```

❖ The owner can include/exclude wallets from rewards

```
ftrace | funcSig
function excludeFromReward(address account⬆) external onlyOwner {
    require(
        account⬆ != address(uniswapV2Router),
        "We can not exclude uniswap router."
    );
    require(!_isExcluded[account⬆], "Account was already excluded");
    if (_rOwned[account⬆] > 0) {
        _tOwned[account⬆] = tokenFromReflection(_rOwned[account⬆]);
    }
    _isExcluded[account⬆] = true;
    _excluded.add(account⬆);
    emit ExcludedFromReward(account⬆);
}

ftrace | funcSig
function includeInReward(address account⬆) external onlyOwner {
    require(_isExcluded[account⬆], "Account was already included");
    if (_excluded.contains(account⬆)) {
        _excluded.remove(account⬆);
        _tOwned[account⬆] = 0;
        _isExcluded[account⬆] = false;
    }
    emit IncludedToReward(account⬆);
}
```

❖ The owner can include/exclude wallets from fees

```
ftrace | funcSig
function excludeFromFee(address account⬆) external onlyOwner {
    _isExcludedFromFee[account⬆] = true;
    emit ExcludedFromFee(account⬆);
}

ftrace | funcSig
function includeInFee(address account⬆) external onlyOwner {
    _isExcludedFromFee[account⬆] = false;
    emit IncludedToFee(account⬆);
}
```

❖ The owner can change max token amount sell to add liquidity

```
ftrace | funcSig
function setNumTokensSellToAddToLiquidity(uint256 amount⬆)
    external
    onlyOwner
{
    numTokensSellToAddToLiquidity = amount⬆;
    emit UpdateNumtokensSellToAddToLiquidity(amount⬆);
}
```

❖ The owner can enable/disable swap

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

Audit conclusion

RugFreeCoins team has performed in-depth testings, line by line manual code review, and automated audit of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, manipulations, and hacks. According to the smart contract audit.

Smart contract functional Status: **PASSED**

Number of risk issues: **1**

Solidity code functional issue level: **PASSED**

Number of owner privileges: **7**

Centralization risk correlated to the active owner: **LOW**

Smart contract active ownership: **YES**