



RugFreeCoins Audit



**Bet Your Bens Token
Smart Contract Security Audit**

October 12th, 2022

Contents

Audit details	1
Disclaimer	2
Overview	3
Background	4
Target market and the concept	6
Potential to grow with score points	7
Total Points	7
Contract details	8
Contract code function details	9
Contract description table	11
Security issue checking status	19
Owner privileges	21
Audit conclusion	24

Audit details



Audited project
Bet Your Beans Token



Contract Address
0xac02c166330029db5Fb90B889c8E5390cb855609



Client contact
Bet Your Beans Team



Blockchain
Binance smart chain



Project website
<https://www.betyourbeans.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.

Overview

- ✅ No mint function found; the owner cannot mint tokens after initial deployment.
- ✅ The owner can't set fees over 25%.
- ✅ Owner can't blacklist wallets.
- ❌ The Owner can limit trading to a very low amount.
- ❌ The owner can pause trading.
- ❌ The owner can claim the contract's balance of its own token.
- ❌ Auto LP Goes to reachable address (The owner can access them).

Background

Rugfreecoins was commissioned by the Bet Your Beans Team to perform an audit of the smart contract.

<https://bscscan.com/token/0xac02c166330029db5Fb90B889c8E5390cb855609>

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, and long-term sustainability, and as a guide to improving the security posture of the smart contract by remediating the issues that were identified.

Tokenomics

12% when buying & selling

- 5% of trade goes to the Development wallet in BNB
- 4% of trade goes to the Treasury wallet in BNB.
- 3% 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 by holding tokens.
- Anyone who's interested in trading tokens.
- Anyone who's ready to staking and receive rewards.
- Anyone interested in participating in the future plans of the BYB token.
- Anyone who's interested in making financial transactions with any other party using BYB Token as the currency.

Potential to grow with score points

1.	Project efficiency	9/10
2.	Project uniqueness	9/10
3	Information quality	9/10
4	Service quality	9/10
5	System quality	9/10
6	Impact on the community	9/10
7	Impact on the business	9/10
8	Preparing for the future	9/10
9	Smart contract security	10/10
10	Smart contract functionality assessment	10/10
Total Points		9.2/10

Contract details

Token contract details for 12th of October 2022

Contract name	Bet Your Beans
Contract address	0xac02c166330029db5Fb90B889c8E5390cb855609
Token supply	1,000,000,000
Token ticker	BYB
Decimals	18
Token holders	1
Transaction count	2
Development wallet	0x2bf86e56aaa53ff640958ef2aa4376c4721d3d22
Treasure wallet	0x6dd860199a6416cb29108e85c2ed73775e479fde
Contract deployer address	0x89352214a56bA80547A2842bbE21AEdD315722Ca
Contract's current owner address	0x2bf86e56aaa53ff640958ef2aa4376c4721d3d22

















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
		Self-destruct function security	pass
3	Business security	Access control of owners	High Centralized issues
		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
IPink AntiBot	Interface			
L	setTokenOwner	External !		NO !
L	onPreTransferCheck	External !		NO !
BetYour Beans	Implementation	ERC20, Ownable		
L		Public !		ERC20
L	_transfer	Internal 		
L	_swapAndLiquify	Internal 		
L	_swapTokensForBNB	Internal 		
L	_addLiquidity	Private 		
L	getTokensInStuck	External !		onlyOwner
L	_isTrading	Internal 		
L	setMinimumTokensBeforeSwap	External !		onlyOwner
L	excludeFromFee	External !		onlyOwner
L	includeInFee	External !		onlyOwner












L	setLiquidityProvider	External !		onlyOwner
L	setTradingEnabled	External !		onlyOwner
L	setTreasuryWallet	External !		onlyOwner
L	setLiquidityOwner	External !		onlyOwner
L	setDevWallet	External !		onlyOwner
L	setTax	External !		onlyOwner
L	setMaxTradeLimit	External !		onlyOwner
L	setUniswapRouter	External !		onlyOwner
L	setSwapAndLiquify	External !		onlyOwner
L		External !		NO !
















ERC20	Implementation	Context, IERC20, IERC20 Metadata		
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	_transfer	Internal 		
L	_mint	Internal 		
L	_burn	Internal 		
L	_approve	Internal 		
L	_spendAllowance	Internal 		
L	_beforeTokenTransfer	Internal 		
L	_afterTokenTransfer	Internal 		
Ownable	Implementation	Context		
L		Public !		NO !
L	owner	Public !		NO !
L	renounceOwnership	Public !		onlyOwner
L	transferOwnership	Public !		onlyOwner
L	_transferOwnership	Internal 		
Context	Implementation			
L	_msgSender	Internal 		
L	_msgData	Internal 		

SafeERC20	Library			
L	safeTransfer	Internal 		
L	safeTransferFrom	Internal 		
L	safeApprove	Internal 		
L	safeIncreaseAllowance	Internal 		
L	safeDecreaseAllowance	Internal 		
L	_callOptionalReturn	Private 		
SafeMath	Library			
L	tryAdd	Internal 		
L	trySub	Internal 		
L	tryMul	Internal 		
L	tryDiv	Internal 		
L	tryMod	Internal 		
L	add	Internal 		
L	sub	Internal 		
L	mul	Internal 		
L	div	Internal 		
L	mod	Internal 		
L	sub	Internal 		
L	div	Internal 		



L	mod	Internal 		
IUniswap V2 Factory	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 V2 Router01	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 V2 Router02	Interface	IUniswap V2 Router01		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External !		NO !
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External !		NO !
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External !		NO !

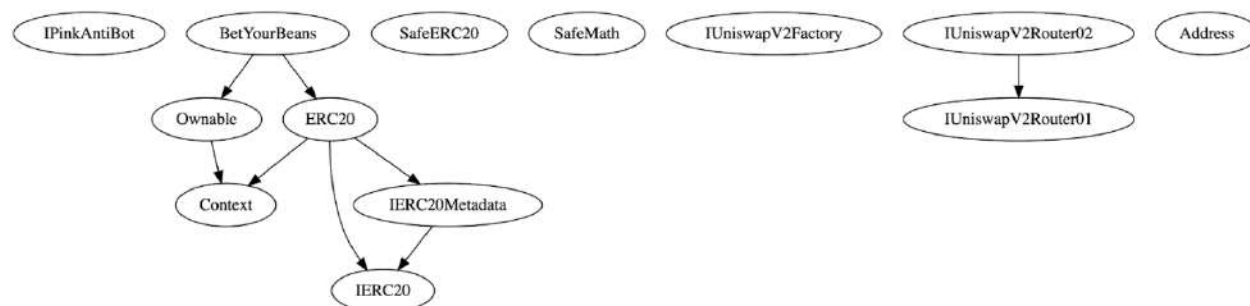
IERC20	Interface			
L	totalSupply	External !		NO !
L	balanceOf	External !		NO !
L	transfer	External !		NO !
L	allowance	External !		NO !
L	approve	External !		NO !
L	transferFrom	External !		NO !
IERC20 Metadata	Interface	IERC20		
L	name	External !		NO !
L	symbol	External !		NO !
L	decimals	External !		NO !
Address	Library			
L	isContract	Internal 		
L	sendValue	Internal 		
L	functionCall	Internal 		
L	functionCall	Internal 		
L	functionCallWithValue	Internal 		
L	functionCallWithValue	Internal 		
L	functionStaticCall	Internal 		

L	functionStaticCall	Internal 		
L	functionDelegateCall	Internal 		
L	functionDelegateCall	Internal 		
L	verifyCallResult	Internal 		

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 (Informed and Fixed)

Return value of low-level calls not used

Returns value of payable call function is not using

```
uint _amountBNB = address(this).balance;
if (treasuryFee > 0) payable(treasuryWallet).call{
    value: _amountBNB.mul(treasuryFee).div(_totalFee),
    gas: 30000
}("");

if (devFee > 0) payable(devWallet).call{
    value: _amountBNB.mul(devFee).div(_totalFee),
    gas: 30000
}("");
```

❖ Centralization issues

Auto LP tokens go to the owner wallet (it should go to an unreachable address)

```
function _addLiquidity(uint256 tokenAmount, uint256 ethAmount) private {
    // Approve token transfer to cover all possible scenarios
    _approve(address(this), address(uniswapV2Router), tokenAmount);

    // Add the liquidity
    uniswapV2Router.addLiquidityETH{value: ethAmount}(
        address(this),
        tokenAmount,
        0, // Slippage is unavoidable
        0, // Slippage is unavoidable
        liquidityOwner,
        block.timestamp
    );
}
```

Owner can withdraw Native tokens from the contract

```
function getTokensInStuck() external onlyOwner {
    if (balanceOf(address(this)) > 0) {
        super._transfer(address(this), msg.sender, balanceOf(address(this)));
    }

    if (address(this).balance > 0) {
        payable(msg.sender).call{value: address(this).balance, gas: 3000}("");
    }
}
```

Owner can enable/disable trading any time

```
function setTradingEnabled(bool _flag) external onlyOwner {
    tradingEnabled = _flag;
}
```

Owner can change the max trade limit to a very low amount (Can set it to 1 token too)

```
function setMaxTradeLimit(uint _limit) external onlyOwner {
    require(_limit > 0, "invalid limit");
    maxTradeLimit = _limit;

    emit UpdatedMaxTradeLimit(_limit);
}
```

Owner privileges

- ❖ The owner can enable/disable swapping

```
function setSwapAndLiquify(bool _flag) external onlyOwner {
    swapAndLiquifyEnabled = _flag;
}
```

- ❖ The owner can change router address

```
function setUniswapRouter(address _router) external onlyOwner {
    require (_router != address(uniswapV2Router), 'already settled');
    uniswapV2Router = IUniswapV2Router02(_router);

    address pair = IUniswapV2Factory(uniswapV2Router.factory()).getPair(address(this), uniswapV2Router.WETH());
    if (pair != address(0)) return;

    uniswapPair = IUniswapV2Factory(uniswapV2Router.factory())
        .createPair(address(this), uniswapV2Router.WETH());

    emit UpdatedRouter(_router);
}
```

- ❖ The owner can change max trade limit

```
function setMaxTradeLimit(uint _limit) external onlyOwner {
    require(_limit > 0, "invalid limit");
    maxTradeLimit = _limit;

    emit UpdatedMaxTradeLimit(_limit);
}
```

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

```
function setTax(uint _liquidityFee, uint _treasuryFee, uint _devFee) external onlyOwner {
    require (_liquidityFee <= FEE_LIMIT, "!available sell tax");
    require (_treasuryFee <= FEE_LIMIT, "!available sell tax");
    require (_devFee <= FEE_LIMIT, "!available sell tax");

    liquidityFee = _liquidityFee;
    treasuryFee = _treasuryFee;
    devFee = _devFee;
    totalFee = liquidityFee.add(treasuryFee).add(devFee);

    emit UpdatedTax(liquidityFee, treasuryFee, devFee);
}
```

- ❖ The owner can change all fee receiver addresses

```
function setTreasuryWallet(address _wallet) external onlyOwner {
    treasuryWallet = _wallet;

    emit UpdatedTreasuryWallet(_wallet);
}

function setLiquidityOwner(address _wallet) external onlyOwner {
    liquidityOwner = _wallet;

    emit UpdatedLiquidityLocker(_wallet);
}

function setDevWallet(address _wallet) external onlyOwner {
    devWallet = _wallet;

    emit UpdatedDevWallet(_wallet);
}
```

- ❖ The owner can enable/disable trading any time

```
function setTradingEnabled(bool _flag) external onlyOwner {
    tradingEnabled = _flag;
}
```

- ❖ The owner can add/remove LP providers (these wallets can do transactions when trading is disabled)

```
function setLiquidityProvider(address _wallet, bool _flag) external onlyOwner{
    _isLiquidityProvider[_wallet] = _flag;
}
```

- ❖ The owner can include/exclude wallets from fees

```
function excludeFromFee(address account) external onlyOwner {
    _isExcludedFromFee[account] = true;

    emit UpdatedWhiteList(account, true);
}

function includeInFee(address account) external onlyOwner {
    _isExcludedFromFee[account] = false;

    emit UpdatedWhiteList(account, false);
}
```

- ❖ The owner can change the swap point

```
function setMinimumTokensBeforeSwap(uint256 _minimumTokensBeforeSwap) external onlyOwner {
    minimumTokensBeforeSwap = _minimumTokensBeforeSwap;

    emit UpdatedSwapAmount(_minimumTokensBeforeSwap);
}
```

- ❖ The owner can get all native tokens and BNB in the contract to owner wallet

```
function getTokensInStuck() external onlyOwner {
    if (balanceOf(address(this)) > 0) {
        super._transfer(address(this), msg.sender, balanceOf(address(this)));
    }

    if (address(this).balance > 0) {
        payable(msg.sender).call{value: address(this).balance, gas: 3000}("");
    }
}
```

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: **PASS**

Number of risk issues: **4**

Solidity code functional issue level: **PASS**

Number of owner privileges: **10**

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

Smart contract active ownership: **YES**