



RUGFREECOINS



Bonk Burn Token

RugfreeCoins Verified on February 15th, 2024

Overview

- ✓ No mint function found, the owner cannot mint tokens after initial deployment.
- ✓ The owner can't set a max transaction limit
- ✓ The owner can't pause trading once it's enabled
- ✗ The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.
- ✓ The owner can't change fees.
- ✓ The owner can't blacklist wallets.
- ✓ The owner can't set a max wallet limit
- ✓ The owner can't claim the contract's balance of its own token.

! HIGH SEVERITY ISSUES

The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.

```
function enableTrading() external onlyOwner {  
    require(!isTradingEnabled, "BonkBurn: Trading already enabled");  
    isTradingEnabled = true;  
  
    emit EnableTrading();  
}
```

Contents

Overview.....	2
Contents.....	3
Audit details.....	4
Disclaimer.....	5
Background.....	6
Tokenomics.....	7
Target market and the concept.....	8
Potential to grow with score points.....	9
Contract details.....	10
Contract code function details.....	11
Contract description table.....	12
Inheritance Hierarchy.....	15
Security issue checking status.....	16
Owner privileges.....	17
Audit conclusion.....	19

Audit details



Audited project
Bonk Burn Token



Contract Address
0x4484117BC8bE10BFf7B7BE2c2c37dd1969Cd3B87



Client contact
Bonk Burn Token Team



Blockchain
Binance Smart chain



Project website
<https://bonkburncoin.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 **Bonk Burn Token Team** to perform an audit of the smart contract.

<https://bscscan.com/token/0x4484117BC8bE10BFf7B7BE2c2c37dd1969Cd3B87>

This audit focuses on verifying 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 smart contract's security posture by remediating the identified issues.

Tokenomics

▲ 5% tax when buying & selling

3% of trade goes to two dev wallets in BNB












1% of trade goes to the Liquidity Pool

1% of trade goes to the burn wallet

Target market and the concept

- ▶ 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 taking part in the Bonk Burn token ecosystem.
- ▶ Anyone who's interested in taking part in the future plans of Bonk Burn Token.
- ▶ Anyone who's interested in making financial transactions with any other party Bonk Burn Token as the currency.

Potential to grow with score points

 Project efficiency	8 / 10
 Project uniqueness	8 / 10
 Information quality	8 / 10
 Service quality	8 / 10
 System quality	8 / 10
 Impact on the community	8 / 10
 Impact on the business	9 / 10
 Preparing for the future	8 / 10
 Smart contract security	9 / 10
 Smart contract functionality assessment	10 / 10
 Total Score	8.4 / 10

Contract details

Token contract details for 15th of February 2024































Contract name	Bonk Burn
Contract address	0x4484117BC8bE10BFf7B7BE2c2c37dd1969Cd3B87
Token supply	1,000,000,000
Token ticker	BonkBurn
Decimals	9
Token holders	1
Transaction count	1
Contract deployer address	0xc2403080490CAa0A1410938C7cf834C724ED15dF
Contract's current owner address	0xc2403080490CAa0A1410938C7cf834C724ED15dF







Contract code function details

Nº	Category	Item	Result
1	Coding conventions	ERC20 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 ▾
2	Function call audit	Overriding variables	PASS ▾
		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 & centralisation	Access control of owners	HIGH ▾
		Business logics	PASS ▾
		Business implementation	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
Address	Library			
L	sendValue	Internal 		
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			
L	name	External 		NO 
L	symbol	External 		NO 
L	decimals	External 		NO 
IRouter01	Interface			
L	factory	External 		NO 
L	WETH	External 		NO 
L	addLiquidityETH	External 		NO 

L	addLiquidity	External !		NO !
L	swapExactETHForTokens	External !		NO !
L	getAmountsOut	External !		NO !
L	getAmountsIn	External !		NO !
IRouter02	Interface	IRouter01		
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External !		NO !
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForTokens	External !		NO !
IFactoryV2	Interface			
L	getPair	External !		NO !
L	createPair	External !		NO !
Context	Implementation			
L		Public !		NO !
L	_msgSender	Internal 		
L	_msgData	Internal 		
Ownable	Implementation	Context		
L		Public !		NO !
L	owner	Public !		NO !
L	renounceOwnership	Public !		onlyOwner
L	transferOwnership	Public !		onlyOwner
L	_setOwner	Private 		

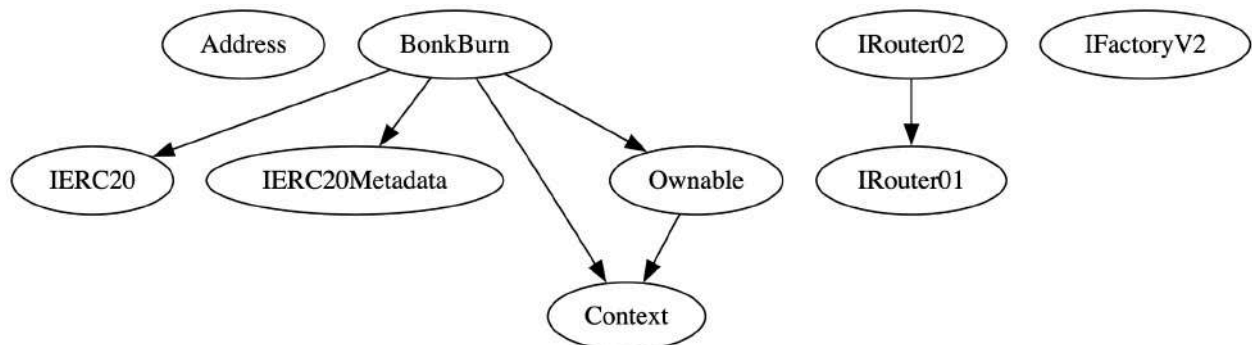
BonkBurn	Implementation	Context, Ownable, IERC20, IERC20 Metadata		
L		Public !	●	NO !
L	isNoFeeWallet	External !		NO !
L	setNoFeeWallet	Public !	●	onlyOwner
L	isLimitedAddress	Internal 🔒		
L	is_buy	Internal 🔒		
L	is_sell	Internal 🔒		
L	canSwap	Internal 🔒		
L	toggleCanSwapFees	External !	●	onlyOwner
L	_transfer	Internal 🔒	●	
L	takeTaxes	Internal 🔒	●	
L	swapAndLiquify	Internal 🔒	●	inSwapFlag
L	swapAndSendMarketingAndDevelopment	Internal 🔒	●	inSwapFlag
L	changeWallets	External !	●	onlyOwner
L	setPresaleAddress	External !	●	onlyOwner
L	enableTrading	External !	●	onlyOwner
L	name	External !		NO !
L	symbol	External !		NO !
L	decimals	External !		NO !
L	totalSupply	External !		NO !
L	balanceOf	Public !		NO !
L	allowance	External !		NO !
L	transfer	Public !	●	NO !
L	transferFrom	Public !	●	NO !
L	approve	Public !	●	NO !
L	_approve	Internal 🔒	●	

L	increaseAllowance	Public !	🛑	NO !
L	decreaseAllowance	Public !	🛑	NO !
L	_burn	Internal 🔒	🛑	
L	_update	Internal 🔒	🛑	
L		External !	💰	NO !

Legend

Symbol	Meaning
🛑	Function can modify state
💰	Function is payable

Inheritance Hierarchy



Security issue checking status

❖ High severity issues

The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.

```
function enableTrading() external onlyOwner {  
    require(!isTradingEnabled, "BonkBurn: Trading already enabled");  
    isTradingEnabled = true;  
  
    emit EnableTrading();  
}
```

❖ Medium severity issues

No medium severity issues found

❖ Low severity issues

No high severity issues found

Owner privileges

- ❖ The owner can include/exclude wallets from fees

```
function setNoFeeWallet(address account, bool enabled) public onlyOwner {  
    noFee[account] = enabled;  
    emit SetNoFeeWallet(account, enabled);  
}
```

- ❖ The owner can enable/disable swapping

```
function toggleCanSwapFees(bool value) external onlyOwner {  
    require(canSwapFees != value, "BonkBurn: Cannot set same value");  
    canSwapFees = value;  
    emit ToggleCanSwapFees(value);  
}
```

- ❖ The owner can change dev wallets and marketing wallets

```
function changeWallets(address newMarketing, address newDev1, address newDev2)  
external onlyOwner {  
    require(newMarketing != address(0), "BonkBurn: Address Zero");  
    require(newDev1 != address(0), "BonkBurn: Address Zero");  
    require(newDev2 != address(0), "BonkBurn: Address Zero");  
  
    marketingWallet = payable(newMarketing);  
    dev1Wallet = payable(newDev1);  
    dev2Wallet = payable(newDev2);  
  
    emit ChangeWallets(newMarketing, newDev1, newDev2);  
}
```

- ❖ Owner can exclude pre-sale contract

```
function setPresaleAddress(address presale, bool value) external onlyOwner {  
    require(isPresaleAddress[presale] != value, "BonkBurn: Cannot set same  
value");  
    isPresaleAddress[presale] = value;  
    noFee[presale] = value;  
    liquidityAdd[presale] = value;  
  
    emit SetPresaleAddress(presale, value);  
}
```

- ❖ The owner can enable trading, once enabled can not disable it again

```
function enableTrading() external onlyOwner {  
    require(!isTradingEnabled, "BonkBurn: Trading already enabled");  
    isTradingEnabled = true;  
  
    emit EnableTrading();  
}
```

Audit conclusion

RugFreeCoins team has performed in-depth testing, 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 ▾
Smart contract security Status:	HIGH ISSUES ▾
Number of risk issues:	01
Solidity code functional issue level:	PASS ▾
Number of owner privileges:	05
Centralization risk correlated to the active owner:	HIGH ▾
Smart contract active ownership:	ACTIVE ▾