



RUGFREECOINS



Bazooka Tools Token

RugfreeCoins Verified on December 20th, 2023

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 over 20%. (Up to 60%)
- ✓ 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

Owner can change buy and sell fees each up-to 30% (total 60%)

```
function setBuyTaxes(uint256 _marketingTax) external onlyOwner {
    buyTaxes.marketingTax = _marketingTax;
    require(_marketingTax <= 30, "Can not set buy fees higher than 30%");
    emit BuyFeesUpdated(_marketingTax);
}

function setSellTaxes(uint256 _marketingTax) external onlyOwner {
    sellTaxes.marketingTax = _marketingTax;
    require(_marketingTax <= 30, "Can not set buy fees higher than 30%");
    emit SellFeesUpdated(_marketingTax);
}
```

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(!tradingEnabled, "Trading is already enabled");  
    tradingEnabled = true;  
    launchTax=true;  
    startTradingBlock = block.number;  
}
```

Contents

Overview.....	2
Contents.....	4
Audit details.....	5
Disclaimer.....	6
Background.....	7
Tokenomics.....	8
Target market and the concept.....	9
Potential to grow with score points.....	10
Contract details.....	11
Contract code function details.....	12
Contract description table.....	13
Inheritance Hierarchy.....	16
Security issue checking status.....	17
Owner privileges.....	19
Audit conclusion.....	22

Audit details



Audited project

Bazooka Tools Token



Contract Address

0x08C937FB973226751C13c7CC862c819C09EA5dba



Client contact

Bazooka Tools Token Team



Blockchain

Ethereum



Project website

<https://bazooka.club/>

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 **Bazooka Tools Token Team** to perform an audit of the smart contract.

<https://etherscan.io/token/0x08c937fb973226751c13c7cc862c819c09ea5dba>

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

▲ 2% tax when buying & selling: Standard tax

2% of trade goes to the marketing wallet in ETH

▲ First 4 blocks when buying & selling

40% of trade goes to the marketing wallet in WTH

▲ First 25 blocks when buying & selling

15% of trade goes to the marketing wallet in ETH when buying

30% of trade goes to the marketing wallet in ETH when selling

▲ First 50 blocks when buying & selling

15% of trade goes to the marketing wallet in ETH when buying

30% of trade goes to the marketing wallet in ETH when selling

▲ First 75 blocks when buying & selling

5% of trade goes to the marketing wallet in ETH when buying

20% of trade goes to the marketing wallet in ETH when selling

▲ First 100 blocks when buying & selling












2% of trade goes to the marketing wallet in ETH when buying

8% of trade goes to the marketing wallet in ETH when selling

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 Bazooka Tools token ecosystem.
- ▶ Anyone who's interested in taking part in the future plans of Bazooka Tools Token.
- ▶ Anyone who's interested in making financial transactions with any other party using Bazooka Tools 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	8 / 10
 Smart contract functionality assessment	9 / 10
 Total Score	8.2 / 10

Contract details

Token contract details for 20th of December 2023






























Contract name	Bazooka Tools
Contract address	0x08C937FB973226751C13c7CC862c819C09EA5dba
Token supply	10,000,000
Token ticker	BAZOOKA
Decimals	18
Token holders	2
Transaction count	2
Contract deployer address	0xda7Adb506fAd184fb014967d3b22DaBa2AF80693
Contract's current owner address	0xda7Adb506fAd184fb014967d3b22DaBa2AF80693
Marketing wallet	0xb0A5a045123B57F9176156a80FaFD9605840f44E









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 ▾
3	Business security & centralisation	Self destruct function security	PASS ▾
		Access control of owners	HIGH ▾
		Business logics	PASS ▾
4	Integer overflow/underflow	Business implementation	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
Context	Implementation			
L	_msgSender	Internal 		
L	_msgData	Internal 		
Ownable	Implementation	Context		
L		Public 		NO 
L	owner	Public 		NO 
L	_checkOwner	Internal 		
L	renounceOwnership	Public 		onlyOwner
L	transferOwnership	Public 		onlyOwner
L	_transferOwnership	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 

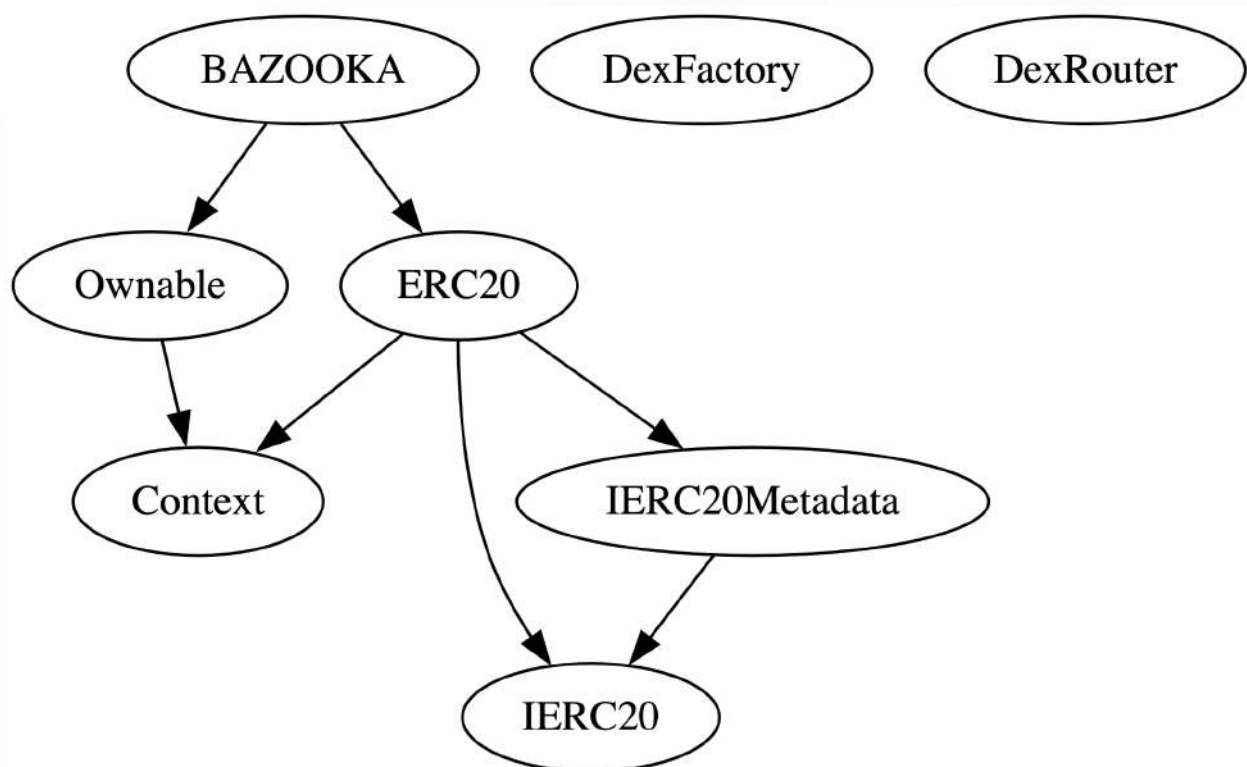
IERC20Meta data	Interface	IERC20		
L	name	External !		NO !
L	symbol	External !		NO !
L	decimals	External !		NO !
ERC20	Implementation	Context, IERC20, IERC20 Metadata		
L		Public !		NO !
L	name	Public !		NO !
L	symbol	Public !		NO !
L	decimals	Public !		NO !
L	totalSupply	Public !		NO !
L	balanceOf	Public !		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 		
DexFactory	Interface			

L	createPair	External !		NO !
DexRouter	Interface			
L	factory	External !		NO !
L	WETH	External !		NO !
L	addLiquidityETH	External !		NO !
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External !		NO !
BAZOOKA	Implementation	ERC20, Ownable		
L		Public !		ERC20
L	setmarketingWallet	External !		onlyOwner
L	enableTrading	External !		onlyOwner
L	setBuyTaxes	External !		onlyOwner
L	setSellTaxes	External !		onlyOwner
L	setSwapTokensAtAmount	External !		onlyOwner
L	toggleSwapping	External !		onlyOwner
L	setWhitelistStatus	External !		onlyOwner
L	removeLimits	Internal 		
L	NormalizeTaxes	External !		onlyOwner
L	_takeTax	Internal 		
L	_transfer	Internal 		
L	internalSwap	Internal 		
L	swapToETH	Internal 		
L	checkWhitelist	External !		NO !
L	setMaxWalletPercentage	External !		onlyOwner
L		External !		NO !

Legend

Symbol	Meaning
🛑	Function can modify state
💰	Function is payable

Inheritance Hierarchy



Security issue checking status

❖ High severity issues

Owner can change buy and sell fees each up-to 30% (total 60%)

```
function setBuyTaxes(uint256 _marketingTax) external onlyOwner {
    buyTaxes.marketingTax = _marketingTax;
    require(_marketingTax <= 30, "Can not set buy fees higher than 30%");
    emit BuyFeesUpdated(_marketingTax);
}

function setSellTaxes(uint256 _marketingTax) external onlyOwner {
    sellTaxes.marketingTax = _marketingTax;
    require(_marketingTax <= 30, "Can not set buy fees higher than 30%");
    emit SellFeesUpdated(_marketingTax);
}
```

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(!tradingEnabled, "Trading is already enabled");
    tradingEnabled = true;
    launchTax=true;
    startTradingBlock = block.number;
}
```

❖ Medium severity issues

If the contract fail to send eth to marketing wallet one time then the swap will never trigger again

```
function internalSwap() internal {
    isSwapping = true;
    uint256 taxAmount = balanceOf(address(this));
    if (taxAmount == 0) {
        return;
    }
    swapToETH(balanceOf(address(this)));
    (bool success, ) = marketingWallet.call{value: address(this).balance}("");
    if (!success) {
        // Log the failure rather than reverting the transaction
        emit TransferFailed(marketingWallet, address(this).balance);
    } else {
        // Only set isSwapping to false if the transfer succeeded
        isSwapping = false;
    }
}
```

❖ Low severity issues

No low severity issues found

Owner privileges

- ❖ Owner can change max wallet limit minimum up-to 1%

```
function setMaxWalletPercentage(uint256 _percentage) external onlyOwner {  
    require(_percentage > 1, "Percentage must be greater than 1%");  
    require(_percentage <= 100, "Percentage must be less than or equal to 100");  
    maxWalletPercentage = _percentage;  
    emit MaxWChanged(_percentage);  
}
```

- ❖ Owner can enable/disable swapping

```
function toggleSwapping() external onlyOwner {  
    swapAndLiquifyEnabled = (swapAndLiquifyEnabled) ? false : true;  
}
```

- ❖ Owner can include/exclude wallets from fees

```
function setWhitelistStatus(  
    address _wallet,  
    bool _status  
) external onlyOwner {  
    whitelisted[_wallet] = _status;  
    emit Whitelist(_wallet, _status);  
}
```

- ❖ Owner can disable launch taxed

```
function NormalizeTaxes() external onlyOwner{
    launchTax=false;
}
```

- ❖ Owner can change swap limit up-to 0.5%

```
function setSwapTokensAtAmount(uint256 _newAmount) external onlyOwner {
    require(
        _newAmount > 0 && _newAmount <= (_totalSupply * 5) / 1000,
        "Minimum swap amount must be greater than 0 and less than 0.5% of total supply!"
    );
    swapTokensAtAmount = _newAmount;
    emit SwapThresholdUpdated(swapTokensAtAmount);
}
```

- ❖ Owner can change buy and sell fees each up-to 30%

```
function setBuyTaxes(uint256 _marketingTax) external onlyOwner {
    buyTaxes.marketingTax = _marketingTax;
    require(_marketingTax <= 30, "Can not set buy fees higher than 30%");
    emit BuyFeesUpdated(_marketingTax);
}

function setSellTaxes(uint256 _marketingTax) external onlyOwner {
    sellTaxes.marketingTax = _marketingTax;
    require(_marketingTax <= 30, "Can not set buy fees higher than 30%");
    emit SellFeesUpdated(_marketingTax);
}
```

- ❖ Owner can enable trading,once enabled can not disable again

```
function enableTrading() external onlyOwner {  
    require(!tradingEnabled, "Trading is already enabled");  
    tradingEnabled = true;  
    launchTax=true;  
    startTradingBlock = block.number;  
}
```

- ❖ Owner can change marketing wallet address

```
function setmarketingWallet(address _newmarketing) external onlyOwner {  
    require(  
        _newmarketing != address(0),  
        "can not set marketing to dead wallet"  
    );  
    marketingWallet = _newmarketing;  
    emit marketingWalletChanged(_newmarketing);  
}
```

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:	3
Solidity code functional issue level:	PASS ▾
Number of owner privileges:	8
Centralization risk correlated to the active owner:	HIGH ▾
Smart contract active ownership:	ACTIVE ▾