



**RUGFREECOINS**



# **BONEX Token**

RugfreeCoins Verified on September 29th, 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.
- ✓ 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 {  
    isTradeEnabled = true;  
}
```

# 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.....	16
Security issue checking status.....	17
Owner privileges.....	18
Audit conclusion.....	22

## Audit details



**Audited project**  
BONEX Token



**Contract Address**  
0x6dfD6668Ee53b414B3B92dD5ED0B3c6e9Facae04



**Client contact**  
BONEX Token Team



**Blockchain**  
Ethereum



**Project website**  
<https://bonex.finance/>

# 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 **BONEX Token Team** to perform an audit of the smart contract.

<https://etherscan.io/address/0x6dfD6668Ee53b414B3B92dD5ED0B3c6e9Fcae04>

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

## ▲ 3% tax when buying & selling

1% of trade goes to the marketing wallet in ETH

1% of trade goes to the Liquidity Pool

1% of trade goes to buyback and burn Pogex tokens












---

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



## Potential to grow with score points

 Project efficiency	9 / 10
 Project uniqueness	10 / 10
 Information quality	9 / 10
 Service quality	10 / 10
 System quality	10 / 10
 Impact on the community	9 / 10
 Impact on the business	10 / 10
 Preparing for the future	9 / 10
 Smart contract security	9 / 10
 Smart contract functionality assessment	10 / 10
 <b>Total Score</b>	<b>9.5/ 10</b>

# Contract details

Token contract details for 29th of September 2023















Contract name	<b>BONEX</b>
Contract address	<b>0x6dfD6668Ee53b414B3B92dD5ED0B3c6e9Facae04</b>
Token supply	<b>1,000,000,000</b>
Token ticker	<b>BONEX</b>
Decimals	<b>18</b>
Token holders	<b>1</b>
Transaction count	<b>1</b>
Contract deployer address	<b>0x9493834A7d32338f68b48756D0693c91CbA0DF4A</b>
Contract's current owner address	<b>0x249338c932d72500f019B3f66cC2758F44B3f9F1</b>
Marketing wallet	<b>0xD9Abcf35C6137954dDa45B91d4Db6c6CF3031E90</b>

# 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	MEDIUM ▾
		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	MEDIUM ▾
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
<b>BONEX</b>	<b>Implementation</b>	<b>IERC20, Ownable</b>		
L		Public !		NO !
L		External !		NO !
L	totalSupply	External !		NO !
L	name	Public !		NO !
L	symbol	Public !		NO !
L	decimals	Public !		NO !
L	balanceOf	Public !		NO !
L	allowance	External !		NO !
L	approve	Public !		NO !
L	_approve	Internal 		
L	approveMax	External !		NO !
L	transfer	External !		NO !
L	transferFrom	External !		NO !
L	_transferFrom	Internal 		
L	shouldDoContractSwap	Internal 		
L	takeFee	Internal 		
L	_basicTransfer	Internal 		
L	shouldTakeFee	Internal 		
L	isFeeExcluded	Public !		NO !

L	doContractSwap	Internal 🔒	🛑	swapping
L	swapTokensForTokens	Private 🔒	🛑	
L	swapAndLiquify	Private 🔒	🛑	
L	swapTokensForEth	Private 🔒	🛑	
L	addLiquidity	Private 🔒	🛑	
L	setIsFeeExempt	External !	🛑	onlyOwner
L	setDoContractSwap	External !	🛑	onlyOwner
L	changeMarketingWallet	External !	🛑	onlyOwner
L	changeBuyFees	External !	🛑	onlyOwner
L	changeSellFees	External !	🛑	onlyOwner
L	enableTrading	External !	🛑	onlyOwner
L	setAuthorizedWallets	External !	🛑	onlyOwner
L	rescueEth	External !	🛑	onlyOwner
L	changeGetFeesOnTransfer	External !	🛑	onlyOwner
<b>Ownable</b>	<b>Implementation</b>	<b>Context</b>		
L		Public !	🛑	NO !
L	owner	Public !		NO !
L	_checkOwner	Internal 🔒		
L	renounceOwnership	Public !	🛑	onlyOwner
L	transferOwnership	Public !	🛑	onlyOwner
L	_transferOwnership	Internal 🔒	🛑	
<b>IERC20</b>	<b>Interface</b>			
L	totalSupply	External !		NO !
L	balanceOf	External !		NO !
L	transfer	External !	🛑	NO !
L	allowance	External !		NO !
L	approve	External !	🛑	NO !

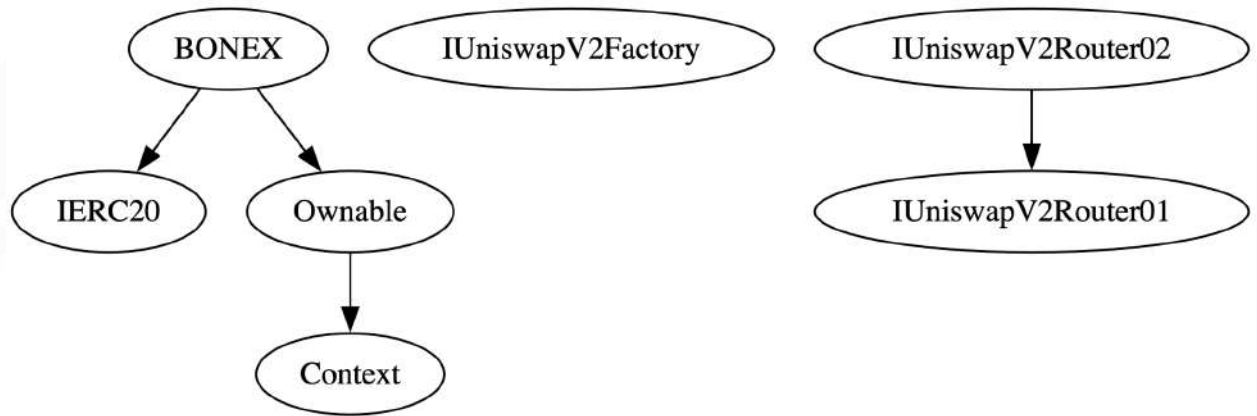
L	transferFrom	External !		NO !
<b>Context</b>	<b>Implementation</b>			
L	_msgSender	Internal 		
L	_msgData	Internal 		
<b>IUniswapV2 Factory</b>	<b>Interface</b>			
L	feeTo	External !		NO !
L	feeToSetter	External !		NO !
L	getPair	External !		NO !
L	allPairs	External !		NO !
L	allPairsLength	External !		NO !
L	createPair	External !		NO !
L	setFeeTo	External !		NO !
L	setFeeToSetter	External !		NO !
<b>IUniswapV2 Router01</b>	<b>Interface</b>			
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 !
<b>IUniswapV2 Router02</b>	<b>Interface</b>	<b>IUniswapV2 Router01</b>		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External !		NO !
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External !		NO !
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForETHSupportingFeeOnTransferTokens	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.

A code editor window with three gray dots in the top-left corner. It contains a function definition in a light gray monospace font. The function name 'enableTrading' is highlighted in green, and the keyword 'true' is highlighted in blue.

```
function enableTrading() external onlyOwner {  
    isTradeEnabled = true;  
}
```

## ❖ Medium severity issues

No medium severity issues found

## ❖ Low severity issues

No low-severity issues found

# Owner privileges

- ❖ Owner can include/exclude wallets from fees

```
function setIsFeeExempt(address holder, bool exempt) external onlyOwner {  
    isFeeExempt[holder] = exempt;  
  
    emit SetIsFeeExempt(holder, exempt);  
}
```

- ❖ Owner can enable/disable swapping

```
function setDoContractSwap(bool _enabled) external onlyOwner {  
    contractSwapEnabled = _enabled;  
  
    emit SetDoContractSwap(_enabled);  
}
```

- ❖ Owner can change marketing wallet address

```
function changeMarketingWallet(address _wallet) external onlyOwner {  
    marketingWallet = _wallet;  
}
```

- ❖ Owner can change all buy and sell fees total fees maximum up to 20%

```
function changeBuyFees(
    uint256 _liquidityFee,
    uint256 _marketingFee,
    uint256 _buyBuyBackFee
) external onlyOwner {
    buyLiquidityFee = _liquidityFee;
    buyMarketingFee = _marketingFee;
    buyBuyBackFee = _buyBuyBackFee;

    buyTotalFee = _liquidityFee + _marketingFee + buyBuyBackFee;

    require(buyTotalFee <= 10, "Total fees can not greater than 10%");
}

function changeSellFees(
    uint256 _liquidityFee,
    uint256 _marketingFee,
    uint256 _sellBuyBackFee
) external onlyOwner {
    sellLiquidityFee = _liquidityFee;
    sellMarketingFee = _marketingFee;
    sellBuyBackFee = _sellBuyBackFee;

    sellTotalFee = _liquidityFee + _marketingFee + _sellBuyBackFee;

    require(sellTotalFee <= 10, "Total fees can not greater than 10%");
}
```

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

```
function enableTrading() external onlyOwner {
    isTradeEnabled = true;
}
```

- ❖ Owner can add/remove authorized wallets ( who can do transactions before enabling tradings)

```
function setAuthorizedWallets(  
    address _wallet,  
    bool _status  
) external onlyOwner {  
    isAuthorized[_wallet] = _status;  
}
```

- ❖ Owner can get contract ETH balance to owner wallet

```
function rescueEth() external onlyOwner {  
    uint256 balance = address(this).balance;  
    require(balance > 0, "No enough ETH to transfer");  
  
    payable(msg.sender).transfer(balance);  
}
```

- ❖ Owner can enable/disable getting fees on the wallet to wallet transactions

```
function changeGetFeesOnTransfer(bool _status) external onlyOwner {  
    getTransferFees = _status;  
}
```

## 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:	1
Solidity code functional issue level:	PASS ▾
Number of owner privileges:	8
Centralization risk correlated to the active owner:	HIGH ▾
Smart contract active ownership:	ACTIVE ▾