RugFreeCoins Audit

Eco Tire Token
Smart Contract Security Audit

October 28th, 2022

Contents

Audit details

Audited project
Eco Tire Token

Contract Address
0x2bB05E3d45318DB4eA6D913d05cb92122485531e

Client contact
Eco Tire Token Team

Blockchain
Binance smart chain

Project website
https://ecotirealliance.org/

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 a max transaction limit

✅ The owner can’t pause trading.

✅ The owner can't set fees over 25%.

✅ 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.

Background

Rugfreecoins was commissioned by the Eco Tire Team to perform an audit of the smart contract.

https://bscscan.com/token/0x2bb05e3d45318db4ea6d913d05cb92122485531e

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 smart contract's security posture by remediating the identified issues.

.

Tokenomics

5% when buying & selling

- 1% of trade goes to the liquidity pool.
- 2% of trade goes to the charity wallet in BNB.
- 2% of trade goes to the Foundation wallet in BNB.

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 who’s interested in taking part in the future plans of Eco Tire Token.
- Anyone who’s interested in making financial transactions with any other party using Eco Tire 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 28th of October 2022

Contract name	EcoTire Token
Contract address	0x2bB05E3d45318DB4eA6D913d05cb92122485531e
Token supply	23,000,000,000
Token ticker	ETTOKEN
Decimals	9
Token holders	1
Transaction count	2
Safu Dev Wallet	0xe4e58c3c9c6e9bcc8ada19ba92b91590f61e6af3
Charity wallet address	0xd0c1ad1c21bcf399aaf99241c1ef6c4f1f48ddb7
Foundation wallet address	0x37cef6033f07ad55a1aa31f94dab6248b006b86e
Contract deployer address	0xE4E58C3C9c6E9Bcc8AdA19BA92B91590f61E6af3
Contract's current owner address	0x0c8aedc3eca2985854277d5393d764937ff9ec27

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	
		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
Context	**Implementation**			
L	_msgSender	Internal 🔒		
L	_msgData	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❗
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 🔒		
Ownable	**Implementation**	**Context**		
L		Public ❗	🛑	NO❗
L	owner	Public ❗		NO❗
L	_checkOwner	Internal 🔒		
L	renounceOwnership	External ❗	🛑	onlyOwner
L	transferOwnership	External ❗	🛑	onlyOwner
L	_transferOwnership	Internal 🔒	🛑	
IUniswapV2 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❗
IUniswapV2 Pair	**Interface**			
L	name	External ❗		NO❗
L	symbol	External ❗		NO❗
L	decimals	External ❗		NO❗
L	totalSupply	External ❗		NO❗
L	balanceOf	External ❗		NO❗
L	allowance	External ❗		NO❗
L	approve	External ❗	🛑	NO❗
L	transfer	External ❗	🛑	NO❗
L	transferFrom	External ❗	🛑	NO❗
L	DOMAIN_SEPARATOR	External ❗		NO❗
L	PERMIT_TYPEHASH	External ❗		NO❗
L	nonces	External ❗		NO❗
L	permit	External ❗	🛑	NO❗
L	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❗
└	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**	**IUniswap V2 Router01**		
└	removeLiquidityETHSupportingFeeOn TransferTokens	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermitSupport ingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactTokensForTokensSupportin gFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactETHForTokensSupportingF eeOnTransferTokens	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingF eeOnTransferTokens	External ❗	🛑	NO❗

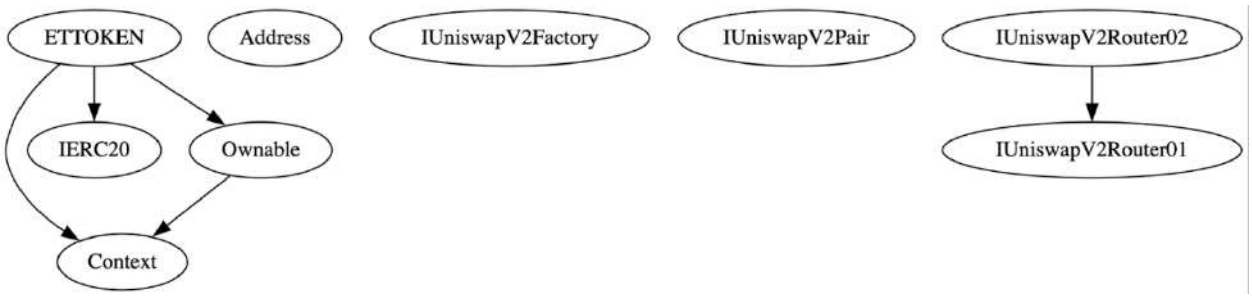
ETTOKEN	Implementation	Context, IERC20, Ownable		
L		Public ❗	🛑	NO❗
L	name	External ❗		NO❗
L	safuDEV	External ❗		NO❗
L	symbol	External ❗		NO❗
L	decimals	External ❗		NO❗
L	version	External ❗		NO❗
L	totalSupply	External ❗		NO❗
L	balanceOf	Public ❗		NO❗
L	transfer	External ❗	🛑	NO❗
L	allowance	External ❗		NO❗
L	approve	External ❗	🛑	NO❗
L	transferFrom	External ❗	🛑	NO❗
L	increaseAllowance	External ❗	🛑	NO❗
L	decreaseAllowance	External ❗	🛑	NO❗
L	minimumTokensBeforeSwapAmount	External ❗		NO❗
L	_approve	Private 🔐	🛑	
L	_transfer	Private 🔐	🛑	
L	swapAndLiquify	Public ❗	🛑	lockTheSwap
L	swapTokensForEth	Private 🔐	🛑	

L	addLiquidity	Private 🔐	🛑	
L	_tokenTransfer	Private 🔐	🛑	
L	countUpFeeShare	Private 🔐	🛑	
L	_transferBothExcluded	Private 🔐	🛑	
L	_getTValues	Private 🔐		
L	_takeLiquidity	Private 🔐	🛑	
L	isExcludedFromFee	External ❗		NO❗
L	excludeFromFee	External ❗	🛑	onlyOwner
L	includeInFee	External ❗	🛑	onlyOwner
L	setNumTokensSellToAddToLiquidity	External ❗	🛑	onlyOwner
L	setFoundationWalletAddress	External ❗	🛑	onlyOwner
L	setCharityWalletAddress	External ❗	🛑	onlyOwner
L	setSwapAndLiquifyEnabled	External ❗	🛑	onlyOwner
L	transferToAddressETH	Private 🔐	🛑	
L		External ❗	💵	NO❗
L	recoverETHfromContract	Public ❗	🛑	onlyOwner
L	recoverTokensFromContract	External ❗	🛑	onlyOwner
L	updateSwapOutput	External ❗	🛑	onlyOwner
L	burn	External ❗	🛑	NO❗
L	_burn	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**
  No low severity issues found

- **Centralization Risk**
  No Centralization Risk found

Owner privileges

- The owner can include/exclude wallets from fees

```
ftrace | funcSig
function excludeFromFee(address account⬆) external onlyOwner {
    _isExcludedFromFee[account⬆] = true;
    emit AuditLog(
        "We have excluded the following walled in fees:",
        account⬆
    );
}

ftrace | funcSig
function includeInFee(address account⬆) external onlyOwner {
    _isExcludedFromFee[account⬆] = false;
    emit AuditLog("We have include the following walled in fees:", account⬆);
}
```

- The owner can change the swap point

```
ftrace | funcSig
function setNumTokensSellToAddToLiquidity(uint256 _minimumTokensBeforeSwap⬆)
    external
    onlyOwner
{
    minimumTokensBeforeSwap = _minimumTokensBeforeSwap⬆;
    emit Log(
        "We have updated minimunTokensBeforeSwap to:",
        minimumTokensBeforeSwap
    );
}
```

❖ The owner can change the foundation wallet and charity wallet

```
ftrace | funcSig
function setFoundationWalletAddress(address _foundationWallet⬆)
    external
    onlyOwner
{
    require(
        _foundationWallet⬆ != address(0),
        "setfoundationWalletAddress: ZERO"
    );
    foundationWalletAddress = payable(_foundationWallet⬆);
    emit AuditLog(
        "We have Updated the foundationWallet:",
        foundationWalletAddress
    );
}

ftrace | funcSig
function setCharityWalletAddress(address _charityWallet⬆)
    external
    onlyOwner
{
    require(_charityWallet⬆ != address(0), "setcharityWalletAddress: ZERO");
    charityWalletAddress = payable(_charityWallet⬆);
    emit AuditLog(
        "We have Updated the charityWallet:",
        charityWalletAddress
    );
}
```

❖ The owner can enable/disable swapping

```
ftrace | funcSig
function setSwapAndLiquifyEnabled(bool _enabled⬆) external onlyOwner {
    require(
        swapAndLiquifyEnabled != _enabled⬆,
        "You can't set it to a value that is already defined."
    );
    swapAndLiquifyEnabled = _enabled⬆;
    emit SwapAndLiquifyEnabledUpdated(_enabled⬆);
}
```

❖ The owner can get the contract BNB balance to the foundation wallet

```
ftrace | funcSig
function recoverETHfromContract() public virtual onlyOwner {
    payable(foundationWalletAddress).transfer(address(this).balance);
    emit AuditLog(
        "We have recover the stock eth from contract.",
        foundationWalletAddress
    );
}
```

❖ The owner can get any BEP20 tokens from the contract (Cannot get native tokens)

```
// Withdraw ERC20 tokens that are potentially stuck in Contract
ftrace | funcSig
function recoverTokensFromContract(address _tokenAddress, uint256 _amount)
    external
    onlyOwner
{
    require(
        _tokenAddress != address(this),
        "Owner can't claim contract's balance of its own tokens"
    );
    IERC20(_tokenAddress).transfer(foundationWalletAddress, _amount);
    emit Log("We have recovered tokens from contract:", _amount);
}
```

❖ The owner can update the swap output (Not recommended to change it)

```
//Update price based on oracle.
ftrace | funcSig
function updateSwapOutput(uint256 _newSwapOutput) external onlyOwner {
    require(
        _newSwapOutput >= 0,
        "New SwapOutput need to be greather than 0"
    );
    require(
        _newSwapOutput != swapOutput,
        "This is the current value, you need to use a different one"
    );
    swapOutput = _newSwapOutput;
}
```

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: **0**

Solidity code functional issue level: **PASS**

Number of owner privileges: **7**

Centralization risk correlated to the active owner: **NONE**

Smart contract active ownership: **YES**