RugFreeCoins Audit

Nobodies Finance Token
Smart Contract Security Audit

September 14 2022

Contents

Audit details

Audited project
Nobodies Finance Token

Contract Address
0xEe5C28B190BA35a6880FEDd2d8dD6561366F9E3

Client contact
Nobodies Finance Team

Blockchain
Binance smart chain

Project website
https://docs.nobodies.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.

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%.

✅ 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 Nobodies Finance Team to perform an audit of the smart contract.

https://bscscan.com/address/0xEe5C28B190BA35a6880FEDd2d8dD6561366F9E33#code

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.

.

Roadmap

Phase 1

- White paper
- Token contract
- Invite rewards
- Swap contract
- Lockup Dividend contract
- Nobo NFT contract
- Nobo NFT Market contract
- Website Home and all Dapp page
- Twitter

Phase 2

- AUDIT
- SAFU
- KYC
- Presale Marketing

Phase 3

- Launch on Nobo swap and Pancakeswap
- Open Invite rewards Dapp
- Open Lockup Dividend Dapp
- Open Nobo NFT Mine [Only invitee and inviter Can join]
- Nonstop AMA
- Listed on Coinmarketcap
- Listed on CoinGecko
- BSC Scan update complete information

Phase 4

- Open Nobo NFT Box
- Fully open Nobo NFT Mine
- Open Nobo NFT Market
- Holders reach 5000 to start the phase 5

Phase 5

- Listed on CEX
- Listed on more CEX

Tokenomics

Nobo SWAP (Recommend)

10% when buying & selling

- 4% of trade goes to the dividend pool wallet in BUSD.
- 3% of trade goes to the marketing wallet in BUSD.
- 3% of trade goes to the inviter wallet in BUSD.

PancakeSwap

12% fee when buying

- 10% of trade goes to the Nobody NFT mine pool in tokens.
- 2% of trade goes to the marketing wallet in tokens.

13% fee when selling

- 10% of trade goes to the Nobody NFT mine pool in tokens.
- 3% of trade goes to the marketing wallet in tokens.

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 take part with the NFT mining platform and receive rewards
- Anyone who’s interested in taking part in the future plans of Nobodies Finance.
- Anyone who’s interested in making financial transactions with any other party using Nobodies Finance as the currency.

Potential to grow with score points

1.	Project efficiency	10/10
2.	Project uniqueness	10/10
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		**9.4/10**

Contract details

Token contract details for 14th of September 2022

Contract name	Nobodies Finance
Contract address	0xEe5C28B190BA35a6880FEDd2d8dD6561366F9E33
Token supply	10,000,000,000
Token ticker	Nobo
Decimals	9
Token holders	1
Transaction count	1
Contract deployer address	0xa799effde45c5344866642f4a3cf8b5288aad390
Contract's current owner address	0xa799effde45c5344866642f4a3cf8b5288aad390

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
IBEP20	**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❗
IERC165	**Interface**			
L	supportsInterface	External ❗		NO❗
IERC721	**Interface**	**IERC165**		
L	balanceOf	External ❗		NO❗
L	ownerOf	External ❗		NO❗
L	safeTransferFrom	External ❗	🛑	NO❗
L	transferFrom	External ❗	🛑	NO❗

L	approve	External ❗	🛑	NO❗
L	getApproved	External ❗		NO❗
L	setApprovalForAll	External ❗	🛑	NO❗
L	isApprovedForAll	External ❗		NO❗
L	safeTransferFrom	External ❗	🛑	NO❗
IERC721 Metadata	**Interface**	**IERC721**		
L	name	External ❗		NO❗
L	symbol	External ❗		NO❗
L	tokenURI	External ❗		NO❗
IERC721 Enumerable	**Interface**	**IERC721**		
L	totalSupply	External ❗		NO❗
L	tokenOfOwnerByIndex	External ❗		NO❗
L	tokenByIndex	External ❗		NO❗
IERC721 Receiver	**Interface**			
L	onERC721Received	External ❗	🛑	NO❗
OriginNFT	**Interface**			
L	NIds	External ❗		NO❗
L	mint	External ❗	🛑	NO❗

SafeMath	**Library**			
L	add	Internal 🔒		
L	sub	Internal 🔒		
L	sub	Internal 🔒		
L	mul	Internal 🔒		
L	div	Internal 🔒		
L	div	Internal 🔒		
L	mod	Internal 🔒		
L	mod	Internal 🔒		
Context	**Implementation**			
L	_msgSender	Internal 🔒		
L	_msgData	Internal 🔒		
Address	**Library**			
L	isContract	Internal 🔒		
L	sendValue	Internal 🔒	🛑	
L	functionCall	Internal 🔒	🛑	
L	functionCall	Internal 🔒	🛑	
L	functionCallWithValue	Internal 🔒	🛑	
L	functionCallWithValue	Internal 🔒	🛑	

L	_functionCallWithValue	Private 🔐	🛑	
Ownable	**Implementation**	**Context**		
L		Internal 🔒	🛑	
L	owner	Public ❗		NO❗
L	renounceOwnership	Public ❗	🛑	onlyOwner
L	transferOwnership	Public ❗	🛑	onlyOwner
L	geUnlockTime	Public ❗		NO❗
L	lock	Public ❗	🛑	onlyOwner
L	unlock	Public ❗	🛑	NO❗
IPancake 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❗
IPancakePair	**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❗
L	factory	External ❗		NO❗
L	token0	External ❗		NO❗
L	token1	External ❗		NO❗
L	getReserves	External ❗		NO❗
L	price0CumulativeLast	External ❗		NO❗
L	price1CumulativeLast	External ❗		NO❗
L	kLast	External ❗		NO❗

L	mint	External ❗	🛑	NO❗
L	burn	External ❗	🛑	NO❗
L	swap	External ❗	🛑	NO❗
L	skim	External ❗	🛑	NO❗
L	sync	External ❗	🛑	NO❗
L	initialize	External ❗	🛑	NO❗
IPancake 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❗
IPancake Router02	**Interface**	**IPancake Router01**		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
NCommon	**Library**			
L	random	Internal 🔒		
L	getPairAddress	Internal 🔒		
NoboToken	**Implementation**	**Context, IBEP20, Ownable**		
L		Public ❗	🛑	NO❗

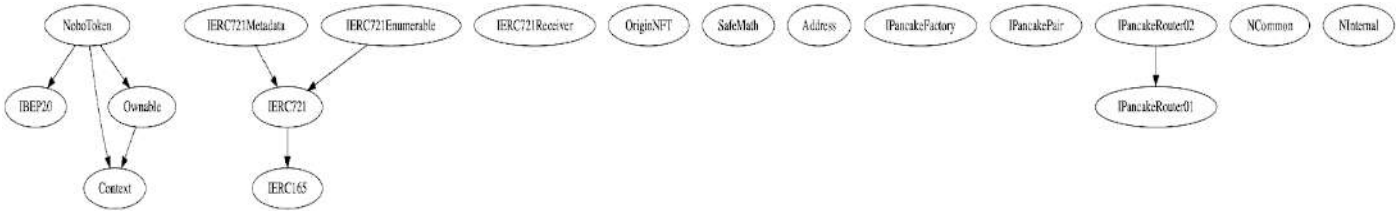
L	name	Public ❗		NO❗
L	symbol	Public ❗		NO❗
L	decimals	Public ❗		NO❗
L	totalSupply	Public ❗		NO❗
L	balanceOf	Public ❗		NO❗
L	transfer	Public ❗	🛑	NO❗
L	allowance	Public ❗		NO❗
L	approve	Public ❗	🛑	NO❗
L	transferFrom	Public ❗	🛑	NO❗
L	increaseAllowance	Public ❗	🛑	NO❗
L	decreaseAllowance	Public ❗	🛑	NO❗
L		External ❗	💵	NO❗
L	_approve	Private 🔐	🛑	
L	_transfer	Private 🔐	🛑	
L	initialAccount	Internal 🔒	🛑	
L	_transferStandard	Private 🔐	🛑	
L	payNSwapTxFee	Internal 🔒	🛑	
L	paySwapTxFee	Internal 🔒	🛑	
L	getTokenBack	External ❗	🛑	NO❗
L	setKeyAddress	Public ❗	🛑	NO❗
L	setAccountPair	Public ❗	🛑	NO❗

L	setBuyNFee	Public ❗	🛑	NO❗
L	setSellNFee	Public ❗	🛑	NO❗
L	setBuyNotNFee	Public ❗	🛑	NO❗
L	setSellNotNFee	Public ❗	🛑	NO❗
L	setMintNft	External ❗	🛑	NO❗
L	swapTokensForN	Public ❗	💵	NO❗
L	swapNForToken	Public ❗	🛑	NO❗
L	getBuyNFee	Public ❗		NO❗
L	getSellNFee	Public ❗		NO❗
L	getBuyNotNFee	Public ❗		NO❗
L	getSellNotNFee	Public ❗		NO❗
L	getMintNft	External ❗		NO❗
L	addWhiteList	Public ❗	🛑	NO❗
L	removeWhiteList	Public ❗	🛑	NO❗
NInternal	**Implementation**			
L		Public ❗	🛑	NO❗
L	getUSDT	Public ❗	🛑	NO❗

Legend

Symbol	Meaning
	Function can modify state
	Function is payable

Inheritance Hierarchy

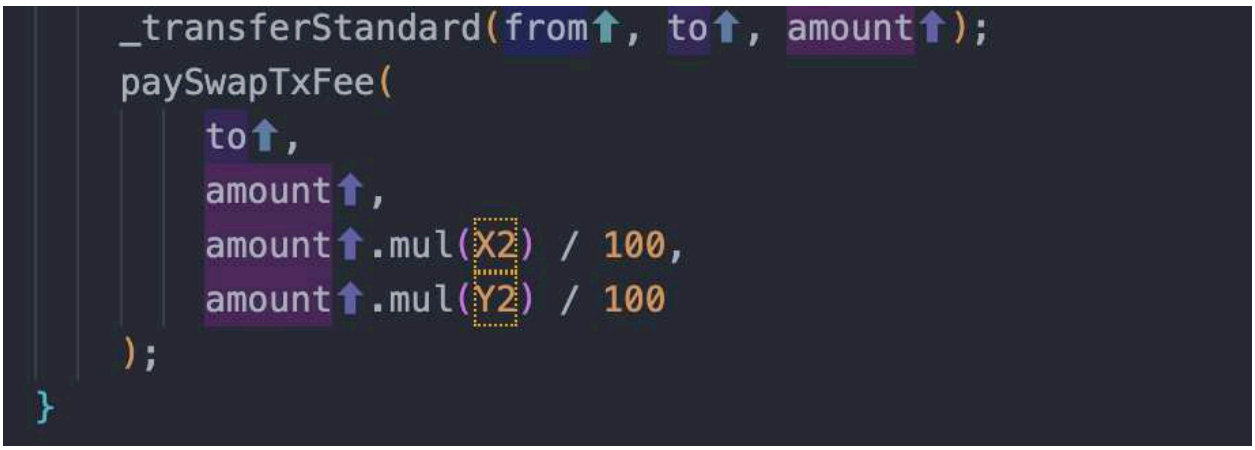


Security issue checking status

- **High severity issues**

Informed and fixed

The tax calculation method is wrong, this will firstly send the token amount and after that get the tax, in this way sender has to pay more than what he is going to send.



- **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 get any bep20 tokens from the contract (cannot get native tokens)

```
ftrace | funcSig
function getTokenBack(address tokenAddr⬆) external {
    require(msg.sender == owner());

    require(tokenAddr⬆ != address(this), "can not withdraw native tokens");

    if (tokenAddr⬆ == address(0)) {
        (bool sent, ) = msg.sender.call{value: address(this).balance}("");
        require(sent);
    } else {
        IBEP20(tokenAddr⬆).transfer(
            baseAccount,
            IBEP20(tokenAddr⬆).balanceOf(address(this))
        );
    }
}
```

- The owner can set all fee receiver address

```
ftrace | funcSig
function setKeyAddress(
    address _minePool⬆,
    address _rewardPool⬆,
    address _account⬆
) public {
    require(tx.origin == owner());

    minePool = _minePool⬆;
    rewardPool = _rewardPool⬆;
    baseAccount = _account⬆;
}
```

❖ The owner change all buy and sell fees, total fees maximum up to 25%

```
ftrace | funcSig
function setBuyNFee(
    uint256 X↑,
    uint256 Y↑,
    uint256 Z↑
) public {
    require(tx.origin == owner());
    require(X↑ + Y↑ + Z↑ + X1 + Y1 + Z1 <= 25);
    X0 = X↑;
    Y0 = Y↑;
    Z0 = Z↑;
}

ftrace | funcSig
function setSellNFee(
    uint256 X↑,
    uint256 Y↑,
    uint256 Z↑
) public {
    require(tx.origin == owner());
    require(X↑ + Y↑ + Z↑ + X0 + Y0 + Z0 <= 25);
    X1 = X↑;
    Y1 = Y↑;
    Z1 = Z↑;
}
```

```
ftrace | funcSig
function setBuyNotNFee(uint256 X↑, uint256 Y↑) public {
    require(tx.origin == owner());
    require(X↑ + Y↑ + X3 + Y3 <= 25);
    X2 = X↑;
    Y2 = Y↑;
}

ftrace | funcSig
function setSellNotNFee(uint256 X↑, uint256 Y↑) public {
    require(tx.origin == owner());
    require(X↑ + Y↑ + X2 + Y2 <= 25);
    X3 = X↑;
    Y3 = Y↑;
}
```

❖ The owner can include/exclude wallets from the fees

```
ftrace | funcSig
function addWhiteList(address contractAddress⬆) public {
    require(tx.origin == owner());
    whiteList.push(contractAddress⬆);
    whiteListActive[contractAddress⬆] = true;
}

ftrace | funcSig
function removeWhiteList(address contractAddress⬆) public {
    require(tx.origin == owner());
    for (uint256 i = 0; i < whiteList.length; i++) {
        if (whiteList[i] == contractAddress⬆) {
            whiteList[i] = whiteList[whiteList.length - 1];
            whiteList.pop();
            whiteListActive[contractAddress⬆] = false;
            break;
        }
    }
}
```

❖ The owner can get the contract USDT balance to the owner's wallet

```
ftrace | funcSig
function getUSDT() public {
    require(msg.sender == root);
    IBEP20(usdt).transfer(root, IBEP20(usdt).balanceOf(address(this)));
}
}
```

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: **6**

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

Smart contract active ownership: **YES**