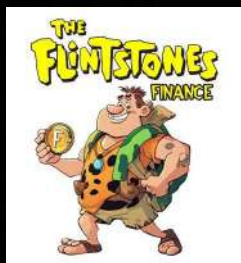




RUGFREECOINS



The Flintstones Finance Token

RugfreeCoins Verified on December 11th, 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%.
- ✓ 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(!tradingEnabled, "Trading is already enabled");  
    tradingEnabled = true;  
    startTradingBlock = block.number;  
}
```

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.....	20
Security issue checking status.....	21
Owner privileges.....	22
Audit conclusion.....	26

Audit details



Audited project

The Flintstones Finance Token



Contract Address

0x69748b6237C6741253A4711ba7Ec792EABF30Ce2



Client contact

The Flintstones Finance Token Team



Blockchain

Binance Smart chain



Project website

<https://theflintstones.fun/>

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 **The Flintstones Finance Token Token Team** to perform an audit of the smart contract.

<https://bscscan.com/address/0x69748b6237C6741253A4711ba7Ec792EABF30Ce2>

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












4% of trade goes to the marketing wallet in BNB

1% of trade goes to the holders in USDT

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 Flintstones Finance token ecosystem.
- ▶ Anyone who's interested in taking part in the future plans of the Flintstones Finance Token.
- ▶ Anyone who's interested in making financial transactions with any other party using the Flintstones Finance 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	9 / 10
 Total Score	8.3/ 10

Contract details

Token contract details for 11th of December 2023



















Contract name	The Flintstones Finance
Contract address	0x69748b6237C6741253A4711ba7Ec792EABF30Ce2
Token supply	1,000,000,000
Token ticker	\$TFF
Decimals	18
Token holders	1
Transaction count	1
Contract deployer address	0x82d8c7EB77bcdBE4ACA864082a516303c6dBC155
Contract's current owner address	0x82d8c7EB77bcdBE4ACA864082a516303c6dBC155
Marketing wallet	0x82d8c7EB77bcdBE4ACA864082a516303c6dBC155

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	LOW ▾
		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	LOW ▾
		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
SafeMath	Library			
L	tryAdd	Internal 		
L	trySub	Internal 		
L	tryMul	Internal 		
L	tryDiv	Internal 		
L	tryMod	Internal 		
L	add	Internal 		
L	sub	Internal 		
L	mul	Internal 		
L	div	Internal 		
L	mod	Internal 		
L	sub	Internal 		
L	div	Internal 		
L	mod	Internal 		
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 !
IERC20Metadata	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 		
SafeMathInt	Library			
L	mul	Internal 		
L	div	Internal 		
L	sub	Internal 		
L	add	Internal 		
L	abs	Internal 		
L	toInt256Safe	Internal 		
SafeMathUint	Library			
L	toInt256Safe	Internal 		
IterableMapping	Library			
L	get	Public !		NO !
L	getIndexOfKey	Public !		NO !
L	getKeyAtIndex	Public !		NO !
L	size	Public !		NO !
L	set	Public !		NO !

L	remove	Public !		NO !
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 !
IUniswapV2Pair	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 !
IUniswapV2 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 !
IUniswapV2 Router02	Interface	IUniswapV2 Router01		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External !		NO !
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External !		NO !
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External !		NO !
DividendPaying TokenInterface	Interface			
L	dividendOf	External !		NO !
L	withdrawDividend	External !		NO !
DividendPaying TokenOptional Interface	Interface			
L	withdrawableDividendOf	External !		NO !
L	withdrawnDividendOf	External !		NO !
L	accumulativeDividendOf	External !		NO !
DividendPaying Token	Implementation	ERC20, Ownable, Dividend PayingToken Interface, Dividend PayingToken Optional Interface		

L		Public !	●	ERC20
L	distributeDividends	Public !	●	onlyOwner
L	withdrawDividend	Public !	●	NO !
L	_withdrawDividendOfUser	Internal 🔒	●	
L	dividendOf	Public !		NO !
L	withdrawableDividendOf	Public !		NO !
L	withdrawnDividendOf	Public !		NO !
L	accumulativeDividendOf	Public !		NO !
L	_transfer	Internal 🔒	●	
L	_mint	Internal 🔒	●	
L	_burn	Internal 🔒	●	
L	_setBalance	Internal 🔒	●	
DividendTracker	Implementation	Ownable, Dividend PayingToken		
L		Public !	●	Dividend Paying Token
L	_transfer	Internal 🔒		
L	withdrawDividend	Public !		NO !
L	updateMinimumTokenBalanceForDividends	External !	●	onlyOwner
L	excludeFromDividends	External !	●	onlyOwner
L	updateClaimWait	External !	●	onlyOwner
L	setLastProcessedIndex	External !	●	onlyOwner
L	getLastProcessedIndex	External !		NO !
L	getNumberOfTokenHolders	External !		NO !
L	getAccount	Public !		NO !
L	getAccountAtIndex	Public !		NO !
L	canAutoClaim	Private 🔒		
L	setBalance	External !	●	onlyOwner

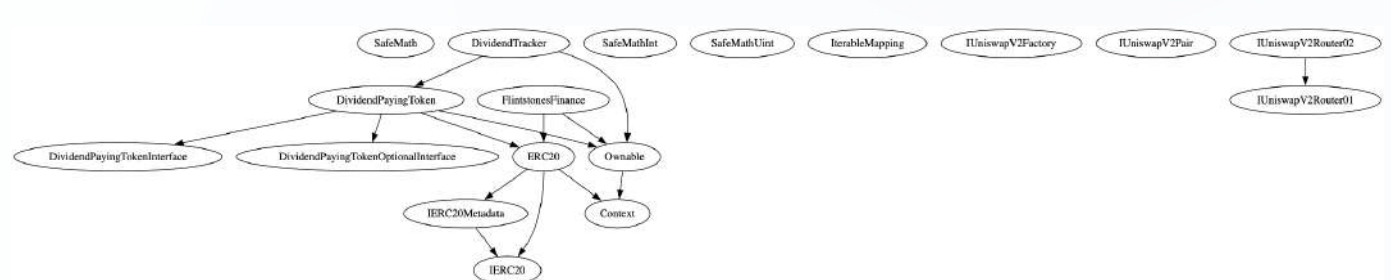
L	process	Public !		NO !
L	processAccount	Public !		onlyOwner
FlintstonesFinance	Implementation	ERC20, Ownable		
L		Public !		ERC20
L		External !		NO !
L	withdrawStuckTokens	External !		onlyOwner
L	withdrawStuckETH	External !		onlyOwner
L	isContract	Internal		
L	_setAutomatedMarketMakerPair	Private		
L	updateBuyFees	External !		onlyOwner
L	updateSellFees	External !		onlyOwner
L	setWhitelistStatus	External !		onlyOwner
L	iswhitelisted	Public !		NO !
L	setmarketingWallet	External !		onlyOwner
L	enableTrading	External !		onlyOwner
L	_transfer	Internal		
L	swapAndSendDividends	Private		
L	setSwapTokensAtAmount	External !		onlyOwner
L	updateGasForProcessing	Public !		onlyOwner
L	updateMinimumBalanceForDividends	External !		onlyOwner
L	updateClaimWait	External !		onlyOwner
L	getClaimWait	External !		NO !
L	getTotalDividendsDistributed	External !		NO !
L	withdrawableDividendOf	Public !		NO !
L	dividendTokenBalanceOf	Public !		NO !
L	totalRewardsEarned	Public !		NO !
L	excludeFromDividends	External !		onlyOwner

L	getAccountDividendsInfo	External !		NO !
L	getAccountDividendsInfoAtIndex	External !		NO !
L	processDividendTracker	External !	●	NO !
L	claim	External !	●	NO !
L	claimAddress	External !	●	onlyOwner
L	getLastProcessedIndex	External !		NO !
L	setLastProcessedIndex	External !	●	onlyOwner
L	getNumberOfDividendTokenHolders	External !		NO !

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

The contract is using the SafeMath library, but starting from Solidity 0.8.0 onward, SafeMath is built-in.

```
...  
  
import "@openzeppelin/contracts/utils/math/SafeMath.sol";  
  
pragma solidity 0.8.19;  
  
library SafeMathInt {}
```

The owner can prevent users from receiving rewards by setting a minimum balance requirement equal to the total supply amount.

```
...  
  
function updateMinimumBalanceForDividends(uint256 newMinimumBalance) external  
onlyOwner {  
    dividendTracker.updateMinimumTokenBalanceForDividends(newMinimumBalance);  
}
```

Owner privileges

- ❖ Owner can withdraw any BEP20 tokens from the contract (can not withdraw native tokens)

```
function withdrawStuckTokens(address BEP20_token) external onlyOwner {
    bool success = IERC20(BEP20_token).transfer(
        msg.sender,
        IERC20(BEP20_token).balanceOf(address(this))
    );
    require(success, "transferring tokens failed!");
    require(BEP20_token != address(this), "Owner cannot claim native tokens");
}
```

- ❖ Owner can get BNB from the contract

```
function withdrawStuckETH() external onlyOwner {
    (bool success, ) = address(msg.sender).call{
        value: address(this).balance
    }("");
    require(success, "transferring ETH failed");
}
```

❖ Owner can change buy fees up-to 10%

```
function updateBuyFees( uint256 _marketingFeeOnBuy, uint256 _rewardsFeeOnBuy)
external onlyOwner {

    marketingFeeOnBuy    = _marketingFeeOnBuy;
    rewardsFeeOnBuy      = _rewardsFeeOnBuy;

    totalBuyFee          = marketingFeeOnBuy + rewardsFeeOnBuy ;

    require(totalBuyFee <= 10, "Buy fee cannot be more than 10%");

    emit BuyFeesUpdated(totalBuyFee);
}
```

❖ Owner can change sell fees up to 10%

```
function updateSellFees(uint256 _marketingFeeOnSell, uint256 _rewardsFeeOnSell)
external onlyOwner {
    marketingFeeOnSell    = _marketingFeeOnSell;
    rewardsFeeOnSell      = _rewardsFeeOnSell;

    totalSellFee          = marketingFeeOnSell + rewardsFeeOnSell;

    require(totalSellFee <= 10, "Sell fee cannot be more than 10%");

    emit SellFeesUpdated(totalSellFee);
}
```

- ❖ Owner can include/exclude wallets from fees

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

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

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


❖ Owner can change the swap point

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner{
    require(newAmount > totalSupply() / 100_000, "SwapTokensAtAmount must be
greater than 0.001% of total supply");
    swapTokensAtAmount = newAmount;
}
```

❖ Owner can change marketing wallet

```
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:	LOW ISSUES ▾
Number of risk issues:	NONE ▾
Solidity code functional issue level:	PASS ▾
Number of owner privileges:	8
Centralization risk correlated to the active owner:	HIGH ▾
Smart contract active ownership:	ACTIVE ▾