



RUGFREECOINS



JUDGE TORRES Token

RugfreeCoins Verified on September 27th, 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 change fees without any limit.
- ✗ The owner can 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 can change all fees without any limitation; if the owner changes them to a very high amount, it will pause trading.

```
function setXRPRewardsFee(uint256 value) external onlyOwner {
    XRPRewardsFee = value;
    totalFees = XRPRewardsFee.add(liquidityFee).add(marketingFee);
}

function setLiquiditFee(uint256 value) external onlyOwner {
    liquidityFee = value;
    totalFees = XRPRewardsFee.add(liquidityFee).add(marketingFee);
}

function setMarketingFee(uint256 value) external onlyOwner {
    marketingFee = value;
    totalFees = XRPRewardsFee.add(liquidityFee).add(marketingFee);
}
```

The owner can blacklist any wallet from the contract

```
function blacklistAddress(address account, bool value) external onlyOwner {  
    _isBlacklisted[account] = value;  
}
```

The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.

```
function startTrading() external onlyOwner {  
    tradingEnabled = true;  
}
```

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..... | 21 |
| Security issue checking status..... | 22 |
| Owner privileges..... | 25 |
| Audit conclusion..... | 31 |

Audit details



Audited project

JUDGE TORRES Token



Contract Address

0x23b126c1B1FD64531Fe3e086E9DcD11C392436A



Client contact

JUDGE TORRES Token Team



Blockchain

Binance Smart Chain



Project website

<https://judgetorres.vip/>

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 **JUDGE TORRES Token Team** to perform an audit of the smart contract.

<https://bscscan.com/token/0x23b126c1B1FD64531Fe3e086E9DcD11C392436A>

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

▲ 15 % tax when buying

5% of trade goes to the marketing wallet in XRP

8% of trade is distributed among holders as rewards in XRP.

2% of trade goes to the Liquidity pool.

▲ 16 % tax when selling (Extra 1% sell fee goes to the contract)

5% of trade goes to the marketing wallet in XRP












8% of trade is distributed among holders as rewards in XRP.

2% of trade goes to the Liquidity pool.

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

Potential to grow with score points

| | |
|---|-----------------|
|  Project efficiency | 8 / 10 |
|  Project uniqueness | 7 / 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 | 6 / 10 |
|  Smart contract functionality assessment | 6 / 10 |
|  Total Score | 7.6 / 10 |

Contract details

Token contract details for 27th of September 2023


















| | |
|----------------------------------|---|
| Contract name | JUDGE TORRES |
| Contract address | 0x23b126c1B1FD6b64531Fe3e086E9DcD11C392436A |
| Token supply | 1,000,000,000 |
| Token ticker | JAT |
| Decimals | 18 |
| Token holders | 8 |
| Transaction count | 10 |
| Contract deployer address | 0xf50b880De83aC0cF3B7c092E2Ea16dBcd17B255C |
| Contract's current owner address | 0xf50b880De83aC0cF3B7c092E2Ea16dBcd17B255C |
| Marketing wallet | 0x4b4364E15Bc84801B7994705Ab65fE4100C3595e |
| Dividend Tracker | 0x289A1139Cf63A2C313A77F52B0fc81A1759E2C0C |

Contract code function details

| Nº | Category | Item | Result |
|----|---------------------------------------|--|----------|
| 1 | Coding conventions | ERC20 Token standards | PASS ▾ |
| | | Compile errors | PASS ▾ |
| | | Compiler version security | MEDIUM ▾ |
| | | Visibility specifiers | PASS ▾ |
| | | Gas consumption | LOW ▾ |
| | | 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 |
| | | | | |
| JUDGETORRES | Implementation | ERC20, Ownable | | |
| L | | Public ! |  | ERC20 |
| L | | External ! |  | NO ! |
| L | updateDividendTracker | Public ! |  | onlyOwner |
| L | updateUniswapV2Router | Public ! |  | onlyOwner |
| L | excludeFromFees | Public ! |  | onlyOwner |
| L | excludeMultipleAccountsFromFees | Public ! |  | onlyOwner |
| L | setMarketingWallet | External ! |  | onlyOwner |
| L | setXRPRewardsFee | External ! |  | onlyOwner |
| L | setLiquiditFee | External ! |  | onlyOwner |
| L | setMarketingFee | External ! |  | onlyOwner |
| L | startTrading | External ! |  | onlyOwner |
| L | setAutomatedMarketMakerPair | Public ! |  | onlyOwner |
| L | blacklistAddress | External ! |  | onlyOwner |
| L | _setAutomatedMarketMakerPair | Private  |  | |
| L | updateGasForProcessing | Public ! |  | onlyOwner |
| L | updateClaimWait | External ! |  | onlyOwner |
| L | getClaimWait | External ! | | NO ! |
| L | getTotalDividendsDistributed | External ! | | NO ! |
| L | isExcludedFromFees | Public ! | | NO ! |











| | | | | |
|---|---------------------------------|---|---|-----------------------------|
| L | withdrawableDividendOf | Public ! | | NO ! |
| L | dividendTokenBalanceOf | Public ! | | NO ! |
| L | excludeFromDividends | External ! | ● | onlyOwner |
| L | getAccountDividendsInfo | External ! | | NO ! |
| L | getAccountDividendsInfoAtIndex | External ! | | NO ! |
| L | processDividendTracker | External ! | ● | NO ! |
| L | claim | External ! | ● | NO ! |
| L | getLastProcessedIndex | External ! | | NO ! |
| L | getNumberOfDividendTokenHolders | External ! | | NO ! |
| L | _transfer | Internal 🔒 | ● | |
| L | swapAndSendToFee | Private 🔒 | ● | |
| L | swapAndLiquify | Private 🔒 | ● | |
| L | swapTokensForEth | Private 🔒 | ● | |
| L | swapTokensForXRP | Private 🔒 | ● | |
| L | addLiquidity | Private 🔒 | ● | |
| L | swapAndSendDividends | Private 🔒 | ● | |
| | | | | |
| JUDGETORRES Dividend Tracker | Implementation | Ownable, Dividend Paying Token | | |
| L | | Public ! | ● | Dividend Paying Token |
| L | _transfer | Internal 🔒 | ● | |
| L | withdrawDividend | Public ! | ● | NO ! |
| L | excludeFromDividends | External ! | ● | onlyOwner |
| L | updateClaimWait | 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 |
| | | | | |
| DividendPaying Token | Implementation | ERC20, Ownable, Dividend Paying Token Interface, Dividend Paying Token Optional Interface | | |
| L | | Public ! | ● | ERC20 |
| L | distributeXRPDividends | 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 🔒 | ● | |
| | | | | |
| 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 | _beforeTokenTransfer | 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 ! |
| | | | | |
| IERC20 Metadata | Interface | IERC20 | | |
| L | name | External ! | | NO ! |
| L | symbol | External ! | | NO ! |
| L | decimals | External ! | | NO ! |
| | | | | |
| Context | Implementation | | | |

| | | | | |
|--|------------------|--|---|--|
| L | _msgSender | Internal  | | |
| L | _msgData | Internal  | | |
| | | | | |
| 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  | | |
| | | | | |
| SafeMathUint | Library | | | |
| L | toInt256Safe | Internal  | | |
| | | | | |
| SafeMathInt | Library | | | |
| L | mul | Internal  | | |
| L | div | Internal  | | |
| L | sub | Internal  | | |
| L | add | Internal  | | |
| L | abs | Internal  | | |
| L | toUint256Safe | Internal  | | |
| | | | | |
| DividendPaying TokenInterface | Interface | | | |
| L | dividendOf | External  | | NO  |
| L | withdrawDividend | External  |  | NO  |
| | | | | |

| DividendPaying TokenOptionalInterface | Interface | | | |
|--|------------------------|------------|---|-----------|
| L | withdrawableDividendOf | External ! | | NO ! |
| L | withdrawnDividendOf | External ! | | NO ! |
| L | accumulativeDividendOf | External ! | | NO ! |
| | | | | |
| Ownable | Implementation | Context | | |
| L | | Public ! |  | NO ! |
| L | owner | Public ! | | NO ! |
| L | renounceOwnership | Public ! |  | onlyOwner |
| L | transferOwnership | Public ! |  | onlyOwner |
| | | | | |
| Iterable Mapping | 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 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 ! |
| 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 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 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 ! |

Security issue checking status

❖ High severity issues

The owner can change all fees without any limitation; if the owner changes them to a very high amount, it will pause trading.

```
function setXRPRewardsFee(uint256 value) external onlyOwner {
    XRPRewardsFee = value;
    totalFees = XRPRewardsFee.add(liquidityFee).add(marketingFee);
}

function setLiquidityFee(uint256 value) external onlyOwner {
    liquidityFee = value;
    totalFees = XRPRewardsFee.add(liquidityFee).add(marketingFee);
}

function setMarketingFee(uint256 value) external onlyOwner {
    marketingFee = value;
    totalFees = XRPRewardsFee.add(liquidityFee).add(marketingFee);
}
```

The owner can blacklist any wallet from the contract

```
function blacklistAddress(address account, bool value) external onlyOwner {
    _isBlacklisted[account] = value;
}
```

The owner must enable trade for the holders, if trading remains disabled, no one would be able to buy and sell.

```
function startTrading() external onlyOwner {  
    tradingEnabled = true;  
}
```

❖ Medium severity issues

Utilizing a very old version of Solidity in a token contract can introduce various issues and risks.

```
pragma solidity ^0.6.2;
```

❖ Low severity issues

When excluding multiple wallets from fees, the owner can input a wallet array. However, if the owner provides an exceedingly large wallet array, it may fail due to the gas limit.

```
function excludeMultipleAccountsFromFees(  
    address[] calldata accounts,  
    bool excluded  
) public onlyOwner {  
    for (uint256 i = 0; i < accounts.length; i++) {  
        _isExcludedFromFees[accounts[i]] = excluded;  
    }  
  
    emit ExcludeMultipleAccountsFromFees(accounts, excluded);  
}
```

When sending fees and dividends, both can be swapped to XRP together, rather than swapping them separately, to reduce gas fees.

```
uint256 marketingTokens = contractTokenBalance  
    .mul(marketingFee)  
    .div(totalFees);  
swapAndSendToFee(marketingTokens);  
  
uint256 swapTokens = contractTokenBalance.mul(liquidityFee).div(  
    totalFees  
);  
swapAndLiquify(swapTokens);  
  
uint256 sellTokens = balanceOf(address(this));  
swapAndSendDividends(sellTokens);
```


Owner privileges

- ❖ The owner can update the dividend tracker

```
function updateDividendTracker(address newAddress) public onlyOwner {
    require(
        newAddress != address(dividendTracker),
        "JUDGETORRES: The dividend tracker already has that address"
    );

    JUDGETORRESDividendTracker newDividendTracker = JUDGETORRESDividendTracker(
        payable(newAddress)
    );

    require(
        newDividendTracker.owner() == address(this),
        "JUDGETORRES: The new dividend tracker must be owned by the JUDGETORRES token contract"
    );

    newDividendTracker.excludeFromDividends(address(newDividendTracker));
    newDividendTracker.excludeFromDividends(address(this));
    newDividendTracker.excludeFromDividends(owner());
    newDividendTracker.excludeFromDividends(address(uniswapV2Router));

    emit UpdateDividendTracker(newAddress, address(dividendTracker));

    dividendTracker = newDividendTracker;
}
```

- ❖ The owner can change the router address

```
function updateUniswapV2Router(address newAddress) public onlyOwner {
    require(
        newAddress != address(uniswapV2Router),
        "JUDGE TORRES: The router already has that address"
    );
    emit UpdateUniswapV2Router(newAddress, address(uniswapV2Router));
    uniswapV2Router = IUniswapV2Router02(newAddress);
    address _uniswapV2Pair = IUniswapV2Factory(uniswapV2Router.factory())
        .createPair(address(this), uniswapV2Router.WETH());
    uniswapV2Pair = _uniswapV2Pair;
}
```

- ❖ The owner can include/exclude wallets from fees

```
function excludeFromFees(address account, bool excluded) public onlyOwner {
    require(
        _isExcludedFromFees[account] != excluded,
        "JUDGE TORRES: Account is already the value of 'excluded'"
    );
    _isExcludedFromFees[account] = excluded;

    emit ExcludeFromFees(account, excluded);
}
```

- ❖ The owner can include/exclude multiple wallets from fees

```
function excludeMultipleAccountsFromFees(
    address[] calldata accounts,
    bool excluded
) public onlyOwner {
    for (uint256 i = 0; i < accounts.length; i++) {
        _isExcludedFromFees[accounts[i]] = excluded;
    }

    emit ExcludeMultipleAccountsFromFees(accounts, excluded);
}
```

- ❖ The owner can change marketing wallet

```
function setMarketingWallet(address payable wallet) external onlyOwner {
    _marketingWalletAddress = wallet;
}
```

- ❖ The owner can change all fees

```
function setXRPRewardsFee(uint256 value) external onlyOwner {
    XRPRewardsFee = value;
    totalFees = XRPRewardsFee.add(liquidityFee).add(marketingFee);
}

function setLiquidityFee(uint256 value) external onlyOwner {
    liquidityFee = value;
    totalFees = XRPRewardsFee.add(liquidityFee).add(marketingFee);
}

function setMarketingFee(uint256 value) external onlyOwner {
    marketingFee = value;
    totalFees = XRPRewardsFee.add(liquidityFee).add(marketingFee);
}
```

- ❖ The owner can enable trading, but once enabled can not disable it again

```
function startTrading() external onlyOwner {
    tradingEnabled = true;
}
```

- ❖ The owner can add/remove new lp pairs

```
function setAutomatedMarketMakerPair(
    address pair,
    bool value
) public onlyOwner {
    require(
        pair != uniswapV2Pair,
        "JUDGETORRES: The PancakeSwap pair cannot be removed from
automatedMarketMakerPairs"
    );

    _setAutomatedMarketMakerPair(pair, value);
}
```

- ❖ The owner can blacklist wallets from the contract

```
function blacklistAddress(address account, bool value) external onlyOwner {
    _isBlacklisted[account] = value;
}
```

- ❖ The owner can update the maximum gas limit to process the reward tracker

```
function updateGasForProcessing(uint256 newValue) public onlyOwner {
    require(
        newValue >= 200000 && newValue <= 500000,
        "JUDGETORRES: gasForProcessing must be between 200,000 and 500,000"
    );
    require(
        newValue != gasForProcessing,
        "JUDGETORRES: Cannot update gasForProcessing to same value"
    );
    emit GasForProcessingUpdated(newValue, gasForProcessing);
    gasForProcessing = newValue;
}
```

- ❖ The owner can change the auto reward claim and wait

```
function updateClaimWait(uint256 claimWait) external onlyOwner {  
    dividendTracker.updateClaimWait(claimWait);  
}
```

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: | 6 |
| Solidity code functional issue level: | PASS ▾ |
| Number of owner privileges: | 11 |
| Centralization risk correlated to the active owner: | HIGH ▾ |
| Smart contract active ownership: | ACTIVE ▾ |