RugFreeCoins Audit

BTC Vantage Token

Smart Contract Security Audit

April 24, 2022

Contents

Audit details

Audited project
BTC Vantage Token

Contract Address
0x41d569B08314087e220b0fD8e3A7E3CcF8c4cC0b

Client contact
BTC Vantage Team

Blockchain
Binance smart chain

Project website
https://btcvantage.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 BTC Vantage Team to perform an audit of the smart contract.

https://bscscan.com/token/0x41d569B08314087e220b0fD8e3A7E3CcF8c4cC0b

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, long-term sustainability, and as a guide to improving the security posture of the smart contract by remediating the issues that were identified.

.

About the project

BTC Vantage (Vantage) is a group of investors inspired by the dividend-paying tokenomics of blockchain technology. Vantage is an innovative smart chain contract that pays dividends to its investors in the form of Binance Pegged Bitcoin (BTCB) while also supporting a native burn mechanism – making it a deflationary token by nature. Vantage will also offer a weekly lottery pool with Binance Pegged Ethereum (BETH) as the jackpot for lucky long-term holders.

Our smart contract is the next generation in state-of-the-art smart chain contracts because of its tax flexibility. Vantage's smart contract allows for the accumulation and pay-out of all available tokens on the Binance Smart Chain (BSC) while also offering the flexibility to code zero value to any tax.

Features

- The **rewards** will be distributed in BTC among every holder proportional to how many tokens each individual holds in values of **6% when buying and 9% when selling.**

- The **sustainability fee of 3% when buying and 4% when selling for marketing** is what allows BTC Vantage to hold the aforementioned promise. Tokens will be swapped into ETH and will be sent to a marketing wallet. This way, BTC Vantage Token will have enough funds to promote the coin and spend for future development and marketing without selling tokens as the traditional way.

- The additional component included under the sustainability section is a **liquidity fee of 2% when buying and selling**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

- **The lottery fee is 2% when buying and selling** is what allows BTC Vantage Token to become the most commonly known and recognized lottery token in the crypto sphere. In order for a cryptocurrency to grow and gain traction, especially in the Altcoin market, it must have a 'use-case', which only usually comes with the promise of a better future. The BTC Vantage Token team is motivated by the idea that the coin will have a use-case from day one! The gambling industry has been around for centuries, and there will be an ever-growing crowd of 'gamblers' and players in the crypto sphere, as cryptocurrencies slowly become the staple in terms of money transactions around the world. In order to support this transition, BTC Vantage Token wishes to establish itself as the most competitive and well-known lottery token in the industry, all the while progressively growing a following.

With BTC Vantage Token, the chances of winning are relative to how many tokens you hold, which means that all holders are incentivized to buy more tokens in the long term if they wish to increase their chances of winning the lottery.

ROADMAP

Q1 2022

- Smart Contract Development and Deployment on BEP-20
- Smart Contract Audit Completion
- Website and Social Media Unveiling
- Telegram Community Building
- Private and Pre-sale
- Pancake Swap Launch
- Coin Market Cap and Coin Gecko Listing
- Lottery D-app
- Reward Dashboard and Calculator

Q2 2022

- Growing Telegram Community
- Influencers Marketing Campaigns
- Partnership with other Projects
- Staking pool
- Start P2E development
- CEX Listing

Q3 2022

- P2E Launch
- Additional CEX
- NFT Marketplace
- NFT Staking

Tokenomics

12% fee when buying

- 6% of trade goes to holders pockets in BTC rewards.
- 3% of trade goes to the marketing wallet in ETH
- 2% of trade goes to the liquidity pool
- 1% of trade goes to the lottery pool in ETH.

15% fee when selling

- 9% of trade goes to holders pockets in BTC rewards.
- 4% of trade goes to the marketing wallet in ETH
- 2% of trade goes to the liquidity pool
- 2% of trade goes to the lottery pool in ETH.

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 in BTC by holding tokens.
- Anyone who’s interested in trading tokens.
- Anyone who is ready to hold and be eligible to win in the weekly lottery
- Anyone who is ready to hold a large portion of tokens and be eligible to get a high chance of winning in the weekly lottery.
- Anyone who’s interested in taking part in the future plans of the BTC Vantage token.
- Anyone who’s interested in making financial transactions with any other party using BTC Vantage token as the currency.

Core concept

The BTC reward system

6% of each transaction when buying and 9% when selling gets converted to BTC and is split amongst all holders. Holders will be eligible to receive tokens in each transaction and rewards are proportional to how many tokens each individual holds.

Sustainable mechanism

The sustainability fee of 3% when buying and 4% when selling for dev and marketing is what allows BTC Vantage token to promote the token and use funds to further the development of the platform. Tokens will be swapped into ETH and will be sent to a marketing wallet. This way, BTC Vantage will have access to the funds without selling tokens as the traditional way, which will enable them to consume funds without hurting the project.

The liquidity fee of 2% when buying and selling, is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

Lottery pool & lottery platform

The concept encourages,

- Investors, to hold the token for a long time, which makes them believe in the project and keep their hopes high of expecting to win a huge prize at once.
- Investors buy more and more since the chance of winning is higher.
- The concept is revolutionary and certainly can get the attraction of new investors as the project progresses along.
- Project market price and market cap can keep stable if everything goes according to the plan since keeping tokens will seem more profitable than selling.

Weekly Lottery drawing

BTC Vantage lottery is with a strong use case specifically targeting the gambling industry aiming for any long-term believers and holders to give a chance to be eligible for the weekly lottery and win. The most unique core part of the BTC Vantage is that the chances of winning are relative to how many tokens investors hold, which means that all holders are incentivized to buy more tokens in the long term if they wish to increase their chances of winning the lottery.

How Chances of Winning are Calculated

Chances of winning will be calculated in indirect proportion to how many tokens each holder has. This means that having more tokens does increase your chances of winning, but not in a linear fashion. Instead, a logarithmic function will be used to convert the proportion of holdings that each investor has, and calculate their chances of winning accordingly. This will lower the discrepancy in the probability of winning between a whale and a small investor while keeping our largest investors at an advantage.

No. of tokens	% chance of winning	Log transformation	% of winning (log transformation)
15	0.50	1.17609	0.36784
07	0.23	0.84510	0.26432
05	0.17	0.69897	0.21861
03	0.10	0.477120	0.14923
Total		**3.19728**	**1**

Lottery Drawing Dapp

The lottery platform will be visible in an interface, where all contract holders are visible with their wallet IDs and the number of tokens they hold. Holders can connect wallets and check the probability of winning against the rest of the holders.

Winners will be chosen on a random draw, live on video chat. The winners will be populated on the web with wallet IDs and the amounts they won.

Potential to grow with score points

1.	Project efficiency	10/10
2.	Project uniqueness	10/10
3	Information quality	10/10
4	Service quality	10/10
5	System quality	10/10
6	Impact on the community	10/10
7	Impact on the business	10/10
8	Preparing for the future	9/10
Total Points		**9.875/10**

Contract details

Token contract details for 24th April 2022

Contract name	BTC Vantage
Contract address	0x41d569B08314087e220b0fD8e3A7E3CcF8c4cC0b
Token supply	100,000,000,000
Token ticker	$VANTAGE
Decimals	9
Token holders	3
Transaction count	3
Dividend tracker	0x7b4ee882d2f762652e9b791c894faf2bdafd5200
Lottery wallet	0xcfc73237fe626cfe1b9644c53114d82b82fd3c3b
Marketing wallet	0x47086d2337449babafa40bde64bf90803ff91051
Game wallet	0x2b2616bda09905c8f138b0cdfbc6230b4eb674a8
Stake pool address	0xd30af5899af9acc0a91876390ed161e96e0a304c
Contract deployer address	0xdD58AF119E66F48f2DcA19f4d63236c799edc4A0
Contract's current owner address	0x8c1015414aaf182737116fee4e102c03ba55a16b

Tokens are distributed as follows:

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	High Centralization risk
		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**		
└	Function Name	Visibility	Mutability	Modifiers
VANTAGE	**Implementation**	**IERC20**, **Ownable**		
└		Public ❗	🛑	NO❗
└		External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	name	Public ❗		NO❗
└	symbol	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	getHolderDetails	Public ❗		NO❗
└	getLastProcessedIndex	Public ❗		NO❗
└	getNumberOfTokenHolders	Public ❗		NO❗
└	totalDistributedRewards	Public ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	Public ❗	🛑	NO❗
└	_approve	Internal 🔒	🛑	

└	approveMax	External ❗	🛑	NO❗
└	transfer	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
└	_transferFrom	Internal 🔒	🛑	
└	_basicTransfer	Internal 🔒	🛑	
└	shouldTakeFee	Internal 🔒		
└	takeFee	Internal 🔒	🛑	
└	shouldSwapBack	Internal 🔒		
└	clearStuckBalance	External ❗	🛑	onlyOwner
└	getBep20Tokens	External ❗	🛑	onlyOwner
└	updateBuyFees	Public ❗	🛑	onlyOwner
└	updateSellFees	Public ❗	🛑	onlyOwner
└	updateSwapPercentages	Public ❗	🛑	onlyOwner
└	enableTrading	Public ❗	🛑	onlyOwner
└	whitelistPreSale	Public ❗	🛑	onlyOwner
└	___claimRewards	Public ❗	🛑	NO❗
└	claimProcess	Public ❗	🛑	NO❗
└	blackListWallets	Public ❗	🛑	onlyOwner
└	isBlacklisted	Public ❗		NO❗
└	isRewardExclude	Public ❗		NO❗

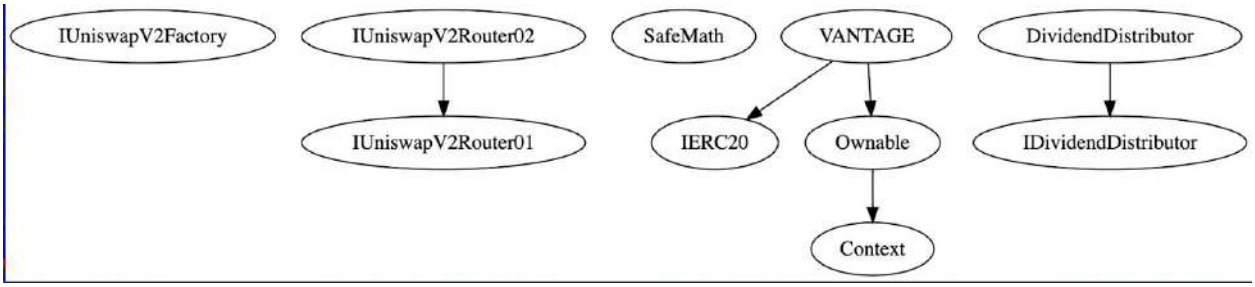
└	isFeeExclude	Public ❗		NO❗
└	isMaxWalletExclude	Public ❗		NO❗
└	isMaxTxExcluded	Public ❗		NO❗
└	setIsMaxTxExempt	External ❗	🛑	onlyOwner
└	setMaxTxAmount	External ❗	🛑	onlyOwner
└	isExemptTimeLock	Public ❗		NO❗
└	changeSellCoolDownTime	Public ❗	🛑	onlyOwner
└	enableSellCollDown	Public ❗	🛑	onlyOwner
└	exemptTimeLock	Public ❗	🛑	onlyOwner
└	swapBackInBnb	Internal 🔒	🛑	swapping
└	swapAndLiquify	Private 🔐	🛑	
└	swapTokensForEth	Private 🔐	🛑	
└	swapTokensForTokens	Private 🔐	🛑	
└	addLiquidity	Private 🔐	🛑	
└	setIsDividendExempt	External ❗	🛑	onlyOwner
└	setIsFeeExempt	External ❗	🛑	onlyOwner
└	setIsMaxWalletExempt	External ❗	🛑	onlyOwner
└	addAuthorizedWallets	External ❗	🛑	onlyOwner
└	setMarketingWallet	External ❗	🛑	onlyOwner

└	setLotteryWallet	External ❗	🛑	onlyOwner
└	setGamePoolAddress	External ❗	🛑	onlyOwner
└	setStakePoolAddress	External ❗	🛑	onlyOwner
└	changeLotteryAndMarketingToken	External ❗	🛑	onlyOwner
└	setMaxWalletToken	External ❗	🛑	onlyOwner
└	changeSellFeeMultiplier	External ❗	🛑	onlyOwner
└	setSwapBackSettings	External ❗	🛑	onlyOwner
└	setDistributionCriteria	External ❗	🛑	onlyOwner
└	setDistributorSettings	External ❗	🛑	onlyOwner
└	purgeBeforeSwitch	Public ❗	🛑	onlyOwner
└	includeMeinRewards	Public ❗	🛑	NO❗
└	switchToken	Public ❗	🛑	onlyOwner

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**

- The owner can enable/disable trading any time

```
ftrace | funcSig
function enableTrading(bool _status) public onlyOwner {
    tradingOpen = _status;
}
```

- The owner can change the max transaction limit without any minimum limit.

```
ftrace | funcSig
function setMaxTxAmount(uint256 amount) external onlyOwner {
    maxTxAmount = amount * (10**9);
}
```

Owner privileges

- The owner can get BNB and other bep20 tokens in the contract to the owner wallet

```
ftrace | funcSig
function clearStuckBalance(uint256 amountPercentage) external onlyOwner {
    uint256 amountBNB = address(this).balance;
    payable(msg.sender).transfer((amountBNB * amountPercentage) / 100);
}

ftrace | funcSig
function getBep20Tokens(address _tokenAddress, uint256 amount)
    external
    onlyOwner
{
    require(
        IERC20(_tokenAddress).balanceOf(address(this)) >= amount,
        "No Enough Tokens"
    );
    IERC20(_tokenAddress).transfer(msg.sender, amount);
}
```

- The owner can update all buy fees maximum up to 20%

```
ftrace | funcSig
function updateBuyFees(
    uint256 reward,
    uint256 marketing,
    uint256 liquidity,
    uint256 burn,
    uint256 staking,
    uint256 gamePool,
    uint256 lottery
) public onlyOwner {
    buyRewardFee = reward;
    buyMarketingFee = marketing;
    buyLiquidityFee = liquidity;
    buyBurnFee = burn;
    buyStakePoolFee = staking;
    buyGameFee = gamePool;
    buyLotteryFee = lottery;
    buyTotalFees = reward.add(marketing).add(liquidity).add(burn).add(
        staking
    );
    buyTotalFees = buyTotalFees.add(gamePool).add(lottery);

    require(buyTotalFees <= 20, "Fees can not be greater than 20%");
}
```

❖ The owner can update all sell fees maximum up to 20%

```
function updateSellFees(
    uint256 reward,
    uint256 marketing,
    uint256 liquidity,
    uint256 burn,
    uint256 staking,
    uint256 gamePool,
    uint256 lottery
) public onlyOwner {
    sellRewardFee = reward;
    sellMarketingFee = marketing;
    sellLiquidityFee = liquidity;
    sellBurnFee = burn;
    sellStakePoolFee = staking;
    sellGameFee = gamePool;
    sellLotteryFee = lottery;
    sellTotalFees = reward.add(marketing).add(liquidity).add(burn).add(
        staking
    );
    sellTotalFees = sellTotalFees.add(gamePool).add(lottery);

    require(sellTotalFees <= 20, "Fees can not be greater than 20%");
}
```

❖ The owner can change all swap percentages

```
function updateSwapPercentages(
    uint256 reward,
    uint256 marketing,
    uint256 liquidity,
    uint256 lottery
) public onlyOwner {
    rewardSwap = reward;
    marketingSwap = marketing;
    liquiditySwap = liquidity;
    lotterySwap = lottery;
    totalSwap = reward.add(marketing).add(liquidity).add(lottery);
}
```

❖ The owner can enable/disable trading

```
ftrace | funcSig
function enableTrading(bool _status⬆) public onlyOwner {
    tradingOpen = _status⬆;
}
```

❖ The owner can whitelist pre-sale address

```
ftrace | funcSig
function whitelistPreSale(address _preSale⬆) public onlyOwner {
    isFeeExempt[_preSale⬆] = true;
    isDividendExempt[_preSale⬆] = true;
    isAuthorized[_preSale⬆] = true;
    isMaxWalletExempt[_preSale⬆] = true;
}
```

❖ The owner can block/unblock wallets from the contract

```
ftrace | funcSig
function blackListWallets(address wallet⬆, bool _status⬆) public onlyOwner {
    isBlacklist[wallet⬆] = _status⬆;
}
```

❖ The owner can include/exclude wallets from max transaction

```
ftrace | funcSig
function setIsMaxTxExempt(address holder⬆, bool exempt⬆) external onlyOwner {
    isMaxTxExempt[holder⬆] = exempt⬆;
}
```

❖ The owner can change max transaction amount

```
ftrace | funcSig
function setMaxTxAmount(uint256 amount⬆) external onlyOwner {
    maxTxAmount = amount⬆ * (10**9);
}
```

❖ The owner can enable/disable sell cool down and can change sell cool down time

```
ftrace | funcSig
function changeSellCoolDownTime(uint256 _time⬆) public onlyOwner {
    cooldownTimerInterval = _time⬆;
}

ftrace | funcSig
function enableSellCollDown(bool _status⬆) public onlyOwner {
    coolDownEnabled = _status⬆;
}
```

❖ The owner can include/exclude wallets from selling cool down time.

```
ftrace | funcSig
function exemptTimeLock(address wallet⬆, bool _status⬆) public onlyOwner {
    isTimelockExempt[wallet⬆] = _status⬆;
}
```

❖ The Owner can include/exclude wallets from dividend

```
ftrace | funcSig
function setIsDividendExempt(address holder⬆, bool exempt⬆)
    external
    onlyOwner
{
    require(holder⬆ != address(this) && holder⬆ != pair);
    isDividendExempt[holder⬆] = exempt⬆;
    if (exempt⬆) {
        dividendTracker.setShare(holder⬆, 0);
    } else {
        dividendTracker.setShare(holder⬆, _balances[holder⬆]);
    }
}
```

❖ The owner can include/exclude wallets from fee

```
ftrace | funcSig
function setIsFeeExempt(address holder, bool exempt) external onlyOwner {
    isFeeExempt[holder] = exempt;
}
```

❖ The owner can include/exclude wallets from max wallet

```
ftrace | funcSig
function setIsMaxWalletExempt(address holder, bool exempt)
    external
    onlyOwner
{
    isMaxWalletExempt[holder] = exempt;
}
```

❖ The owner can change lottery, game pool and stake pool wallet address

```
ftrace | funcSig
function setLotteryWallet(address _lotteryFeeReceiver) external onlyOwner {
    lotteryFeeReceiver = _lotteryFeeReceiver;
}

ftrace | funcSig
function setGamePoolAddress(address _gameWallet) external onlyOwner {
    gameWallet = _gameWallet;
}

ftrace | funcSig
function setStakePoolAddress(address _stakePool) external onlyOwner {
    stakePoolAddress = _stakePool;
}
```

❖ The owner can change lottery and marketing fee receiving token address

```
ftrace | funcSig
function changeLotteryAndMarketingToken(address _tokenAddress)
    external
    onlyOwner
{
    LOTTERY = _tokenAddress;
}
```

❖ The owner can change max wallet token amount.

```
ftrace | funcSig
function setMaxWalletToken(uint256 amount⬆) external onlyOwner {
    maxWalletTokens = amount⬆ * (10**9);
}
```

❖ The owner can change sell fee multiplier

```
ftrace | funcSig
function changeSellFeeMultiplier(uint256 amount⬆) external onlyOwner {
    sellTaxMultiplier = amount⬆;
}
```

❖ The owner can enable/disable swapping and can change swap point.

```
ftrace | funcSig
function setSwapBackSettings(bool _enabled⬆, uint256 _amount⬆)
    external
    onlyOwner
{
    swapEnabled = _enabled⬆;
    swapThreshold = _amount⬆;
}
```

❖ The owner can change the max gas limit for distributing dividend.

```
ftrace | funcSig
function setDistributorSettings(uint256 gas⬆) external onlyOwner {
    require(gas⬆ < 750000);
    distributorGas = gas⬆;
}
```

- The owner can get dividend tracker token amount to the owner wallet (This function should use before changing the reward token)

```
ftrace | funcSig
function purgeBeforeSwitch() public onlyOwner {
    dividendTracker.purge(msg.sender);
}
```

- The owner can change reward token address

```
ftrace | funcSig
function switchToken(address rewardToken, bool isIncludeHolders)
    public
    onlyOwner
{
    require(rewardToken != WBNB, "Can not reward BNB in this tracker");
    REWARD = rewardToken;
    // get current shareholders list
    address[] memory currentHolders = dividendTracker.getShareHoldersList();
    dividendTracker = new DividendDistributor(address(router), rewardToken);
    if (isIncludeHolders) {
        // add old share holders to new tracker
        for (uint256 i = 0; i < currentHolders.length; i++) {
            try
                dividendTracker.setShare(
                    currentHolders[i],
                    _balances[currentHolders[i]]
                )
            {} catch {}
        }
    }

    emit ChangeRewardTracker(rewardToken);
}
```

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: **PASSED**

Number of risk issues: **2**

Solidity code functional issue level: **PASSED**

Number of owner privileges: **22**

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

Smart contract active ownership: **YES**