RugFreeCoins Audit

Fantom Libero Financial Freedom Token

Smart Contract Security Audit

March 29, 2022

Contents

Audit details

Audited project
Fantom Libero Financial Freedom Token

Contract Address

0xC3f069D7439baf6D4D6E9478D9Cc77778E62D147

Client contact

Fantom Libero Financial Freedom Token Team

Blockchain

Fantom smart chain

Project website

https://flibero.financial/

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 Fantom Libero Financial Freedom Team to perform an audit of the smart contract.

https://ftmscan.com/token/0xc3f069d7439baf6d4d6e9478d9cc77778e62d147

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

Fantom Libero Financial Freedom is a token built on the Binance Smart Chain that is with an innovative investment use case the main purpose of which is to seek out constant revenue sources, **the first autostaking protocol backed by Defi 3.0 yield farming** on Fantom. FLIBERO will bring an unparallel, fixed APY of **159,058.06%, the highest of its kind** onto the Fantom blockchain, while imposing profound ease, simplicity, and accessibility upon all FLibero token holders. Each transaction, purchase incurs 16% fee, and sale incurs a 25% fee.

Features

- **5% of the buy and 9% sales fees** is directed to the insurance which helps sustain and back the Staking Rewards provided by the Positive Rebase.

- The **sustainability fee of 3% when buying and 5% when selling for treasury, which is allocated for marketing** is what allows FLIBERO Token to hold the aforementioned promise. Tokens will be swapped into FTM and will be sent to a marketing wallet per transaction. This way, Fantom Libero Financial Freedom Token will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.

- The additional component included under the sustainability section is a **liquidity fee of 4% when buying and selling**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

- **2% of all FFLIBERO tokens traded are burnt in the Black Hole**. The more that is traded, the more get put into the fire causing the fire pit to grow in size, larger and larger through self-fulfilling auto-compounding which in return acts to reduce the circulating supply of FLIBERO and keeping the FLIBERO stable.

- **Rewards fee in 2% when buying and 5% when selling** is distributed among holders in USDC.

Roadmap

This is not a roadmap: It's a to-do list

Crypto moves fast, and we move fast too. Pivoting is a way of life. That means that we don't publicly commit to specific timelines, so we can organize our development priorities based on market changes

- [x] Presale on THOREUM ITO Platform
- [x] Pre-Launch Marketing
- [x] Internal Audit
- [ ] Dashboard Stress Test
- [ ] Multi Community Creation
- [ ] Multi Language Website/Docs
- [ ] Multi Language Docs
- [ ] Youtube Marketing Campaign
- [ ] AMA Marketing Campaign
- [ ] Coingecko Listing
- [ ] Coinmarketcap Listing
- [ ] Coin Trackers Listing
- [ ] DappRadar Listing
- [ ] Dashboard V2
- [x] Social Media Marketing
- [x] Expand Core Team
- [x] PR Marketing
- [x] SEO
- [ ] On Ramp Integration
- [ ] Development Mobile Application iOS and Android
- [x] Partnership DeFi
- [ ] Cross-Chain Integration
- [ ] DAO

Tokenomics

16% fee when buying

- 5% of trade goes to insurance fund in FTM
- 3% of trade goes to the treasury in FTM
- 2% trade goes to the black hole
- 4% of trade goes to the liquidity pool.
- 2% of trade goes to the holders in USDC

25% fee when selling

- 9% of trade goes to freedom insurance fund in FTM
- 4.5% of trade goes to the vault in FTM
- 2% trade goes to the blackhole
- 4% of trade goes to the liquidity pool.
- 5% of trade goes to the holders in USDC

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 in receiving automatic staking and compound rewards every 15 minutes.
- Anyone who's interested in receiving fixed interest of 2.02% per day or 159,058.06% per year.
- Anyone who's interested in taking part with FLibero play and earn rewards.
- Anyone who's interested in taking part with the future plans of the Fantom Libero Financial Freedom token.
- Anyone who's interested in making financial transactions with any other party using FLIBERO as the currency.

Core concept

FLIBERO Auto Compound

fLibero enables its holders to extensively compound their investment and returns, as the protocol auto rewards its holders with 2.04% a day with the compounding APY of 159,058.06% is rewarded every 10 minutes, 144 times a day.

fLibero Insurance Treasury

- The FIT uses an algorithm that backs the Rebase Rewards and is supported by a portion of the buy and sell trading fees that accrue in the FIT wallet.

- In simple terms, the staking rewards (rebase rewards) which are distributed every 10 minutes at a rate of 0.04208% are backed by the FIT parameter, thus ensuring a high and stable interest rate to $fLIBERO holders.

- But the funds in this wallet don't just sit there. **fLIBERO will use Defi 3.0 Multichain Farming to increase the FIT exponentially to better support price floor.**

- The funds are bridged to other EVM-compatible blockchains - like Avalanche, Binance, Solana, Metis, Polygon, etc. to farm at the highest APY farms and the profit returned to the FIT fund. **So, the FIT fund will grow exponentially with 100% additional value a year.**

Sustainable mechanism

The sustainability fee of 3% when buying and 5% selling for treasury that allocated for marketing is what allows Fantom Libero Financial Freedom to promote the token and use funds to further the development of the platform. Tokens will be swapped into FTM and will be sent to a marketing wallet per transaction. This way, FLIBERO 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 4% when buying and selling, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

2% of FLIBERO tokens from buying and selling traded are burnt in **Black Hole**. The more that is traded, the more get put into the fire causing the fire pit to grow in size, larger and larger through self-fulfilling Auto-Compounding, reducing the circulating supply and keeping the FLIBERO table.

FLIBERO Bank

By just holding xfLIBERO, holders receive USDC rewards, every day, these amounts are calculated based on your proportion percentage of the overall xfLIBERO in circulation, this percentage gives a proportional access to the funds accumulated from the pool collected from the 7% fLIBERO trading volume, accumulated from the buy & sell tax.

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	10/10
Total Points		**10/10**

Contract details

Token contract details for 29th March 2022

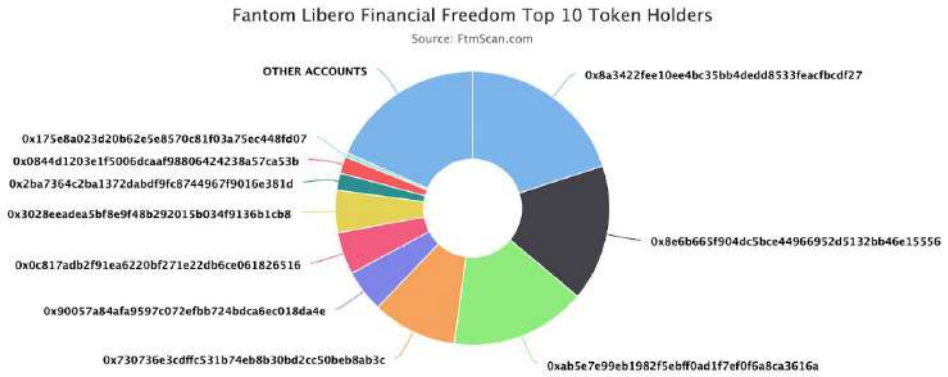
Contract name	Fantom Libero Financial Freedom
Contract address	0xC3f069D7439baf6D4D6E9478D9Cc77778E62D147
Token supply	1,500,000,000
Token ticker	FLIBERO
Decimals	18
Token holders	1,024
Transaction count	1,208
Auto liquidity receiver	0x730736e3cdffc531b74eb8b30bd2cc50beb8ab3c
Risk free value receiver	0x14c02711a4678fc7de388e77e99b07753c856e84
Treasury receiver	0x3ff8970f17d463b83d289827b7b8e5eed61cc3e8
XLibero receiver	0x8689edab5bdb17b11273a5c9412c4bbc8f2ec4f8
Contract deployer address	0x0844d1203E1f5006DCaAf98806424238A57Ca53b

Top token holders

Top 10 Token Holders

The top 10 holders collectively own 81.64%
(1,224,548,343.47 Tokens) of Fantom Libero Financial Freedom

Token Total Supply: 1,500,000,000.00 Token | Total Token Holders: 1,176



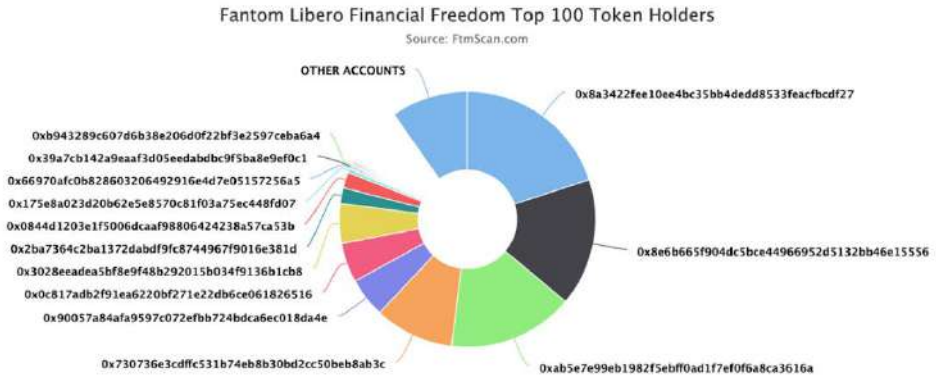
(A total of 1,224,548,343.47 tokens held by the top 10 accounts from the total supply of 1,500,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x8a3422fee10ee4bc35bb4dedd8533feacfbcdf27	300,000,000	20.0000%
2	0x8e6b665f904dc5bce44966952d5132bb46e15556	241,736,377.896	16.1158%
3	0xab5e7e99eb1982f5ebff0ad1f7ef0f6a8ca3616a	240,391,747.006015995707491261	16.0261%
4	0x730736e3cdffc531b74eb8b30bd2cc50beb8ab3c	150,000,000	10.0000%
5	0x90057a84afa9597c072efbb724bdca6ec018da4e	75,000,000	5.0000%
6	0x0c817adb2f91ea6220bf271e22db6ce061826516	75,000,000	5.0000%
7	0x3028eeadea5bf8e9f48b292015b034f9136b1cb8	75,000,000	5.0000%
8	0x2ba7364c2ba1372dabdf9fc8744967f9016e381d	30,000,000	2.0000%
9	0x0844d1203e1f5006dcaaf98806424238a57ca53b	29,999,967	2.0000%
10	0x175e8a023d20b62e5e8570c81f03a75ec448fd07	7,420,251.564	0.4947%

Top 100 token holders

The top 100 holders collectively own 90.30%
(1,354,537,075.40 Tokens) of Fantom Libero Financial Freedom

Token Total Supply: 1,500,000,000.00 Token | Total Token Holders: 1,181



(A total of 1,354,537,075.40 tokens held by the top 100 accounts from the total supply of 1,500,000,000.00 token)

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
		Selfdestruct function security	pass
3	Business security	Access control of owners	Centralised 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
Context	**Implementation**			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Auth	**Implementation**	**Context**		
└		Public ❗	🛑	NO❗
└	authorize	Public ❗	🛑	onlyOwner
└	unauthorize	Public ❗	🛑	onlyOwner
└	isOwner	Public ❗		NO❗
└	isAuthorized	Public ❗		NO❗
└	transferOwnership	Public ❗	🛑	onlyOwner
SafeMathInt	**Library**			
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	sub	Internal 🔒		

L	add	Internal 🔒		
L	abs	Internal 🔒		
SafeMath	**Library**			
L	add	Internal 🔒		
L	sub	Internal 🔒		
L	sub	Internal 🔒		
L	mul	Internal 🔒		
L	div	Internal 🔒		
L	div	Internal 🔒		
L	mod	Internal 🔒		
IERC20	**Interface**			
L	totalSupply	External ❗		NO❗
L	balanceOf	External ❗		NO❗
L	allowance	External ❗		NO❗
L	transfer	External ❗	🛑	NO❗
L	approve	External ❗	🛑	NO❗
L	transferFrom	External ❗	🛑	NO❗
InterfaceLP	**Interface**			
L	sync	External ❗	🛑	NO❗

Roles	**Library**			
L	add	Internal 🔒	🛑	
L	remove	Internal 🔒	🛑	
L	has	Internal 🔒		
ERC20Detailed	**Implementation**	**IERC20**		
L		Public ❗	🛑	NO❗
L	name	Public ❗		NO❗
L	symbol	Public ❗		NO❗
L	decimals	Public ❗		NO❗
IDEXRouter	**Interface**			
L	factory	External ❗		NO❗
L	WETH	External ❗		NO❗
L	addLiquidity	External ❗	🛑	NO❗
L	addLiquidityETH	External ❗	💵	NO❗
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
IDEXFactory	**Interface**			

└	createPair	External ❗	🛑	NO❗
FLiberoToken	**Implementation**	**ERC20Detailed, Auth**		
└		Public ❗	🛑	ERC20Detailed Auth
└		External ❗	💵	NO❗
└	totalSupply	External ❗		NO❗
└	allowance	External ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	checkFeeExempt	External ❗		NO❗
└	shouldRebase	Internal 🔒		
└	shouldTakeFee	Internal 🔒		
└	shouldSwapBack	Internal 🔒		
└	getCirculatingSupply	Public ❗		NO❗
└	manualSync	Public ❗	🛑	NO❗
└	transfer	External ❗	🛑	validRecipient
└	_basicTransfer	Internal 🔒	🛑	
└	_transferFrom	Internal 🔒	🛑	
└	transferFrom	External ❗	🛑	validRecipient
└	_swapAndLiquify	Private 🔐	🛑	
└	_addLiquidity	Private 🔐	🛑	

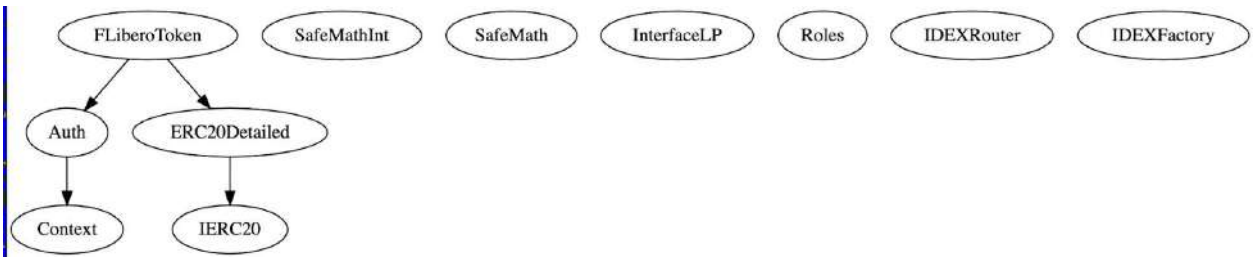
└	addLiquidityUsdc	Private 🔐	🛑	
└	_swapTokensForFTM	Private 🔐	🛑	
└	_swapTokensForUsdc	Private 🔐	🛑	
└	swapBack	Internal 🔒	🛑	swapping
└	takeFee	Internal 🔒	🛑	
└	decreaseAllowance	External ❗	🛑	NO❗
└	increaseAllowance	External ❗	🛑	NO❗
└	approve	External ❗	🛑	NO❗
└	_rebase	Private 🔐	🛑	
└	coreRebase	Private 🔐	🛑	
└	manualRebase	External ❗	🛑	authorized
└	setAutomatedMarketMakerPair	Public ❗	🛑	onlyOwner
└	setInitialDistributionFinished	External ❗	🛑	onlyOwner
└	setFeeExempt	External ❗	🛑	onlyOwner
└	setSwapThreshold	External ❗	🛑	onlyOwner
└	setFeeReceivers	External ❗	🛑	onlyOwner
└	setFees	External ❗	🛑	onlyOwner
└	clearStuckBalance	External ❗	🛑	onlyOwner
└	rescueToken	External ❗	🛑	onlyOwner
└	setAutoRebase	External ❗	🛑	onlyOwner
└	setRebaseFrequency	External ❗	🛑	onlyOwner

L	setRewardYield	External ❗	🛑	onlyOwner
L	setFeesOnNormalTransfers	External ❗	🛑	onlyOwner
L	setIsLiquidityEnabled	External ❗	🛑	onlyOwner
L	setIsLiquidityInFtm	External ❗	🛑	onlyOwner
L	setIsRfvInFtm	External ❗	🛑	onlyOwner
L	setNextRebase	External ❗	🛑	onlyOwner
L	setMaxSellTransaction	External ❗	🛑	onlyOwner

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

Inheritance Hierarchy



Security issue checking status

- **High severity issues**

No medium severity issues found.

- **Medium severity issues**

No medium severity issues found

- **Low severity issues**

No low severity issues found

- **Centralization risk**

❖ The owner can disable trading anytime

```
ftrace | funcSig
function setInitialDistributionFinished(bool _value) external onlyOwner {
    require(initialDistributionFinished != _value, "Not changed");
    initialDistributionFinished = _value;
}
```

Owner privileges

- The owner can manually rebase

```
ftrace | funcSig
function manualRebase() external authorized {
    require(!inSwap, "Try again");
    require(nextRebase <= block.timestamp, "Not in time");

    int256 supplyDelta = int256(
        _totalSupply.mul(rewardYield).div(REWARD_YIELD_DENOMINATOR)
    );

    coreRebase(supplyDelta);
    manualSync();
}
```

- The owner can add a new LP address

```
ftrace | funcSig
function setAutomatedMarketMakerPair(address _pair⬆, bool _value⬆)
    public
    onlyOwner
{
    require(
        automatedMarketMakerPairs[_pair⬆] != _value⬆,
        "Value already set"
    );

    automatedMarketMakerPairs[_pair⬆] = _value⬆;

    if (_value⬆) {
        _makerPairs.push(_pair⬆);
    } else {
        require(_makerPairs.length > 1, "Required 1 pair");
        for (uint256 i = 0; i < _makerPairs.length; i++) {
            if (_makerPairs[i] == _pair⬆) {
                _makerPairs[i] = _makerPairs[_makerPairs.length - 1];
                _makerPairs.pop();
                break;
            }
        }
    }

    emit SetAutomatedMarketMakerPair(_pair⬆, _value⬆);
}
```

❖ The owner can enable/disable trading

```
ftrace | funcSig
function setInitialDistributionFinished(bool _value) external onlyOwner {
    require(initialDistributionFinished != _value, "Not changed");
    initialDistributionFinished = _value;
}
```

❖ The owner can exclude wallets from fees

```
ftrace | funcSig
function setFeeExempt(address _addr, bool _value) external onlyOwner {
    require(_isFeeExempt[_addr] != _value, "Not changed");
    _isFeeExempt[_addr] = _value;
}
```

❖ The owner can change swap point

```
ftrace | funcSig
function setSwapThreshold(uint256 _value) external onlyOwner {
    require(swapThreshold != _value, "Not changed");
    swapThreshold = _value;
}
```

❖ The owner can change all fee receivers

```
ftrace | funcSig
function setFeeReceivers(
    address _liquidityReceiver,
    address _treasuryReceiver,
    address _riskFreeValueReceiver,
    address _xLiberoReceiver
) external onlyOwner {
    require(_liquidityReceiver != address(0), "Invalid _liquidityReceiver");
    require(_treasuryReceiver != address(0), "Invalid _treasuryReceiver");
    require(
        _riskFreeValueReceiver != address(0),
        "Invalid _riskFreeValueReceiver"
    );
    require(_xLiberoReceiver != address(0), "Invalid _xLiberoReceiver");

    liquidityReceiver = _liquidityReceiver;
    treasuryReceiver = _treasuryReceiver;
    riskFreeValueReceiver = _riskFreeValueReceiver;
    xLiberoReceiver = _xLiberoReceiver;
}
```

- The owner can change all fees, buy fees maximum upto 25% and sell fees maximum upto 50%

```
ftrace | funcSig
function setFees(
    uint256 _liquidityFee,
    uint256 _riskFreeValue,
    uint256 _treasuryFee,
    uint256 _burnFee,
    uint256 _xLiberoFee,
    uint256 _sellFeeTreasuryAdded,
    uint256 _sellFeeRFVAdded,
    uint256 _sellBurnFeeAdded,
    uint256 _sellxLiberoFeeAdded
) external onlyOwner {
    liquidityFee = _liquidityFee;
    buyFeeRFV = _riskFreeValue;
    treasuryFee = _treasuryFee;
    buyBurnFee = _burnFee;
    buyxLiberoFee = _xLiberoFee;
    sellFeeTreasuryAdded = _sellFeeTreasuryAdded;
    sellFeeRFVAdded = _sellFeeRFVAdded;
    sellBurnFeeAdded = _sellBurnFeeAdded;
    sellxLiberoFeeAdded = _sellxLiberoFeeAdded;

    totalBuyFee = liquidityFee
        .add(treasuryFee)
        .add(buyFeeRFV)
        .add(buyBurnFee)
        .add(buyxLiberoFee);
    totalSellFee = totalBuyFee
        .add(sellFeeTreasuryAdded)
        .add(sellFeeRFVAdded)
        .add(sellBurnFeeAdded)
        .add(sellxLiberoFeeAdded);

    require(totalBuyFee < MAX_TOTAL_BUY_FEE_RATE, "Total buy fee too high");
    require(
        totalSellFee < MAX_TOTAL_SELL_FEE_RATE,
        "Total sell fee too high"
    );
}
```

❖ The owner can get bnb and other tokens in contract to the owner wallet

```
ftrace | funcSig
function clearStuckBalance(address _receiver) external onlyOwner {
    uint256 balance = address(this).balance;
    payable(_receiver).transfer(balance);
}

ftrace | funcSig
function rescueToken(address tokenAddress)
    external
    onlyOwner
    returns (bool success)
{
    require(tokenAddress != address(this), "Not allow recuse Libero");
    uint256 amount = ERC20Detailed(tokenAddress).balanceOf(address(this));
    return ERC20Detailed(tokenAddress).transfer(msg.sender, amount);
}
```

❖ The owner can enable/disable auto rebase and can change rebase frequency

```
ftrace | funcSig
function setAutoRebase(bool _autoRebase) external onlyOwner {
    require(autoRebase != _autoRebase, "Not changed");
    autoRebase = _autoRebase;
}

ftrace | funcSig
function setRebaseFrequency(uint256 _rebaseFrequency) external onlyOwner {
    require(_rebaseFrequency <= MAX_REBASE_FREQUENCY, "Too high");
    rebaseFrequency = _rebaseFrequency;
}
```

❖ The owner can change the reward percentage

```
ftrace | funcSig
function setRewardYield(uint256 _rewardYield) external onlyOwner {
    require(rewardYield != _rewardYield, "Not changed");
    rewardYield = _rewardYield;
}
```

❖ The owner can enable/disable fees on wallet-to-wallet transactions

```
ftrace | funcSig
function setFeesOnNormalTransfers(bool _enabled) external onlyOwner {
    require(feesOnNormalTransfers != _enabled, "Not changed");
    feesOnNormalTransfers = _enabled;
}
```

❖ The owner can enable/disable adding liquidity

```
ftrace | funcSig
function setIsLiquidityEnabled(bool _value) external onlyOwner {
    require(isLiquidityEnabled != _value, "Not changed");
    isLiquidityEnabled = _value;
}
```

❖ The owner can enable/disable add lp from FTM, if disable liquidity add with USDC

```
ftrace | funcSig
function setIsLiquidityInFtm(bool _value) external onlyOwner {
    require(isLiquidityInFtm != _value, "Not changed");
    isLiquidityInFtm = _value;
}
```

❖ The owner can enable/disable take risk fee value from FTM or not

```
ftrace | funcSig
function setIsRfvInFtm(bool _value) external onlyOwner {
    require(isRfvInFtm != _value, "Not changed");
    isRfvInFtm = _value;
}
```

❖ The owner can manually change next rebase time

```
ftrace | funcSig
function setNextRebase(uint256 _nextRebase) external onlyOwner {
    nextRebase = _nextRebase;
}
```

❖ The owner can change max sell transaction amount minimum up to 100 tokens

```
ftrace | funcSig
function setMaxSellTransaction(uint256 _maxTxn) external onlyOwner {
    require(_maxTxn >= MIN_MAX_SELL_AMOUNT, "Too small");
    maxSellTransactionAmount = _maxTxn;
}
```

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: **0**

Solidity code functional issue level: **PASSED**

Number of owner privileges: **16**

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

Smart contract active ownership: **YES**