



RugFreeCoins Audit



Libero Financial Token

Smart Contract Security Audit

January 29, 2022

Contents

Audit details	1
Disclaimer	2
Background	3
About the project	4
Target market and the concept	6
Potential to grow with score points	9
Total Points	9
Contract details	10
Contract code function details	11
Contract description table	12
Security issue checking status	19
Owner privileges	21
Audit conclusion	26

Audit details



Audited project

Libero Financial Token



Contract Address

0x0DFCb45EAE071B3b846E220560Bbcdd958414d78



Client contact

Libero Financial Team



Blockchain

Binance smart chain



Project website

<https://libero.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 Libero Financial Token to perform an audit of the smart contract.

<https://bscscan.com/token/0x0DFCb45EAE071B3b846E220560Bbcdd958414d78>

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 improve the security posture of the smart contract by remediating the issues that were identified.

About the project

Libero Financial 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 BSC. LIBERO will bring an unparalleled, fixed APY of 158,893.59%, the highest of its kind onto the BSC blockchain, while imposing profound ease, simplicity, and accessibility upon all Libero token holders. Each transaction, purchase incurs 12% fee, and sale incurs a 12% fee.

Features

- ❖ **5%** of the buy and **10%** of the sales fees is directed to the RFV 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 Libero Financial to hold the aforementioned promise. Tokens will be swapped into BNB and will be sent to a marketing wallet per transaction. This way, Libero Financial 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 5% when buying and selling**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

Tokenomics

13% fee when buying

- ❖ 5% of trade goes to the liquidity pool.
- ❖ 3% of trade goes to the treasury for marketing in BNB.
- ❖ 5% of trade goes to the treasury for staking rewards in BUSD.

20% fee when selling

- ❖ 5% of trade goes to the liquidity pool.
- ❖ 5% of trade goes to the treasury for marketing in BNB.
- ❖ 10% of trade goes to the treasury for staking rewards in BUSD

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 chan

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

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 30 minutes.
- ❖ Anyone who's interested in receiving fixed interest of 2.02% per day or 158,893.59% per year.
- ❖ Anyone who's interested in taking part with Libero play and earn rewards.
- ❖ Anyone who's interested in taking part with the future plans of the Libero Financial token.
- ❖ Anyone who's interested in making financial transactions with any other party using Libero Financial as the currency.

Core concept

Reward mechanism

5% of all buy & 10% of all sales go to the RFV farming/buyback wallet. But the funds in this wallet don't just sit there.

The funds are bridged to other EVM-compatible blockchains - like Avalanche, Fantom, Solana, Metis, Polygon, etc. to farm at the highest APY farms and the profit returned to the RFV fund. **So, the RFV fund will grow exponentially with at least 50% additional value a year.**

Buybacks & Burn function will use this RFV fund to support the LIBERO's price. Previous buybacks can be viewed on the "Buyback History" panel in the dashboard. During a buyback, \$LIBERO tokens are bought back, paired with \$BNB, and "burnt" to the \$LIBERO/\$BNB PancakeSwap V2 liquidity pool.

This not only increases the token price, but it also increases the pair's liquidity.

There are two major accomplishments here:

1. Holders no longer need to always keep up with the latest and greatest farm across all these different blockchains. They can just buy the tokens, relax, hold, and reap the earnings via 2.02% daily reward or a compounding 158,893.59% APY and buybacks from farming rewards.
2. When the price falls after a sell, 10% of the sale will be invested in aggressive yield farms that will be used for a future and powerful **compounded buyback**.

Sustainable mechanism

The **sustainability fee of 3% when buying and 5% selling for treasury that allocated for marketing** is what allows Libero Financial to promote the token and use funds to further the development of the platform. Tokens will be swapped into BNB and will be sent to a marketing wallet per transaction. This way, Libero Financial 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.

The use cases

With 4% burn of total supply every week, LIBERO's total supply will constantly be deflating against your balance, while your balance is constantly increasing against LIBERO's total supply. This built-in mechanism creates a true supply/demand metric to the LIBERO token as it becomes ever scarcer against your balance with time.

In simple words, if you just hold LIBERO, your share of total market cap will ever-increasing. Even if market cap is not grow, the USD value of tokens in your wallet will still be growing. So just by holding LIBERO in your wallet, you will reap the exponential rewards of triple effect:

1. Your token amount grows 2.02% / day & compounding to 158,893.59% a year.
2. Your total token value in USD will grows as your shares/total Market cap increase because of 4% supply burned every week.
3. LIBERO's Market cap grow bigger when new investors come, but your shares/total Market cap is not declining but ever growing, your total token value in USD will further increase.

Libero Play

Participants deposit their **\$LIBERO** tokens into Libero P.L.A.Y. for a chance to win a prize of more \$LIBERO using the Libero V2 app on the website. Simply press the Libero P.L.A.Y. button and follow the easy directions. Your tokens never leave your wallet, and you can enter and leave a contest at any time.

Although the foundations of PLAY and PLSA are the same, Libero P.L.A.Y. improves on the PLSA model, because everyone who participates continue to earn \$LIBERO staking rewards with a different but still very high ratio. Those who enter more \$LIBERO have an increased odds of winning the big prize, and once you enter you are automatically entered in every new prize pool until you withdraw your tokens. There is an EARLY EXIT FEE of 10% decreasing for the first 10 days you stay entered. So if you deposit and withdraw the same day, you will pay 10% of the Libero tokens you deposited to enter the contest. This decreases by 1% over the next ten days.

The amount of the prize pool per contest is determined by the amount of \$LIBERO deposited. More tokens deposited into Libero P.L.A.Y. means a higher grand prize!

Winners are determined by random draw with those depositing more \$LIBERO having a greater odd of winning

Potential to grow with score points

1.	Project efficiency	9/10
2.	Project uniqueness	9/10
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		9.75/10

Contract details

Token contract details for 29th January 2022





















Contract name	Libero Financial
Contract address	0x0DFCb45EAE071B3b846E220560Bbccdd958414d78
Token supply	5,000,000,000
Token ticker	LIBERO
Decimals	18
Token holders	1
Transaction count	1
Risk free value receiver	0x94dc0b13e66aba9450b3cc44c2643bbb4c264bc7
Treasury receiver	0x9cce932283183f637e4870a63bdf1e6c348dbb64
Contract deployer address	0xCc2fE3615a45fCaCC3534d53Be41C6543a0A312d
Contract's current owner address	0xcc2fe3615a45fcacc3534d53be41c6543a0a312d





















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	pass
		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

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 its visibility and mutability.




















Contract	Type	Bases		
L	Function Name	Visibility	Mutability	Modifiers
SafeMathInt	Library			
L	mul	Internal 		
L	div	Internal 		
L	sub	Internal 		
L	add	Internal 		
L	abs	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 





SafeMath	Library			
L	add	Internal 		
L	sub	Internal 		
L	sub	Internal 		
L	mul	Internal 		
L	div	Internal 		
L	div	Internal 		
L	mod	Internal 		
InterfaceLP	Interface			
L	sync	External 		NO 
Roles	Library			
L	add	Internal 		
L	remove	Internal 		
L	has	Internal 		
MinterRole	Implementation			
L		Public 		NO 
L	isMinter	Public 		NO 

L	renounceMinter	Public !		NO!
L	_addMinter	Internal 		
L	_removeMinter	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			
L	createPair	External !		NO!
Ownable	Implementation			
L		Public !		NO!
L	owner	Public !		NO!
L	isOwner	Private 		
L	renounceOwnership	Public !		onlyOwner
L	transferOwnership	Public !		onlyOwner
L	_transferOwnership	Internal 		
LiberoToken	Implementation	ERC20Detailed, Ownable, MinterRole		
L		Public !		ERC20Detailed
L		External !		NO!
L	totalSupply	External !		NO!
L	allowance	External !		NO!
L	balanceOf	Public !		NO!
L	checkFeeExempt	External !		NO!
L	checkSwapThreshold	External !		NO!

L	shouldRebase	Internal 🔒		
L	shouldTakeFee	Internal 🔒		
L	shouldSwapBack	Internal 🔒		
L	getCirculatingSupply	Public !		NO !
L	getLiquidityBacking	Public !		NO !
L	isOverLiquified	Public !		NO !
L	manualSync	Public !	🛑	NO !
L	transfer	External !	🛑	validRecipient
L	_basicTransfer	Internal 🔒	🛑	
L	_transferFrom	Internal 🔒	🛑	
L	transferFrom	External !	🛑	validRecipient
L	_swapAndLiquify	Private 🗝️	🛑	
L	_addLiquidity	Private 🗝️	🛑	
L	_addLiquidityBusd	Private 🗝️	🛑	
L	_swapTokensForBNB	Private 🗝️	🛑	
L	_swapTokensForBusd	Private 🗝️	🛑	
L	swapBack	Internal 🔒	🛑	swapping
L	takeFee	Internal 🔒	🛑	
L	decreaseAllowance	External !	🛑	NO !

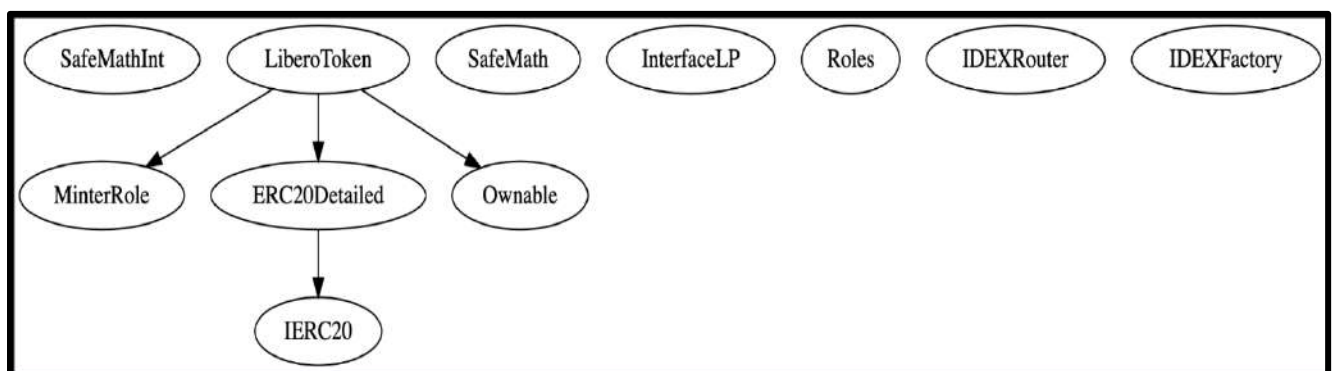
L	increaseAllowance	External !		NO!
L	approve	External !		NO!
L	_rebase	Private 🦒		
L	coreRebase	Private 🦒		
L	externalRebase	External !		onlyMinter
L	addMinter	Public !		onlyOwner
L	removeMinter	Public !		onlyOwner
L	setAutomatedMarketMakerPair	Public !		onlyOwner
L	setInitialDistributionFinished	External !		onlyOwner
L	setFeeExempt	External !		onlyOwner
L	setTargetLiquidity	External !		onlyOwner
L	setSwapBackSettings	External !		onlyOwner
L	setFeeReceivers	External !		onlyOwner
L	setFees	External !		onlyOwner
L	clearStuckBalance	External !		onlyOwner
L	rescueToken	External !		onlyOwner
L	setAutoRebase	External !		onlyOwner
L	setRebaseFrequency	External !		onlyOwner
L	setRewardYield	External !		onlyOwner

L	setFeesOnNormalTransfers	External !		onlyOwner
L	setIsLiquidityInBnb	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 high severity issues found.

❖ Medium severity issues

No medium severity issues found.

❖ Low severity issues

No low severity issues found.

❖ Informational

- Owner can change max sell amount without any limitation

```
ftrace | funcSig
function setMaxSellTransaction(uint256 _maxTxn↑) external onlyOwner {
    maxSellTransactionAmount = _maxTxn↑;
}
```

- Owner can set initialDistribution value any time

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

- Owner can withdraw all BNB and BEP20 tokens in the contract.

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

ftrace | funcSig
function rescueToken(address tokenAddress↑, uint256 tokens↑)
    external
    onlyOwner
    returns (bool success↑)
{
    return ERC20Detailed(tokenAddress↑).transfer(msg.sender, tokens↑);
}

ftrace | funcSig
```

Owner privileges

- ❖ Owner can call rebase and mint a number of tokens maximum up to 340,282,366,920,938,463,374,607,431,768,211,455. But this number will be distributed to all holders, owner cannot mint to any wallet they want.

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

    uint256 circulatingSupply = getCirculatingSupply();
    int256 supplyDelta = int256(
        circulatingSupply.mul(rewardYield).div(rewardYieldDenominator)
    );

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

- ❖ Owner can enable/disable initial distribution.

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

- ❖ Owner can exclude wallets from fees. (fees excluded wallets will also exclude from max sell limit and they can trade when trading is disabled).

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


- ❖ Owner can change liquidity adding point (by default this value is set to 50%, means liquidity will add only if liquidity value is less than 50% of circulating supply).

```
ftrace | funcSig
function setTargetLiquidity(uint256 target↑, uint256 accuracy↑)
    external
    onlyOwner
{
    targetLiquidity = target↑;
    targetLiquidityDenominator = accuracy↑;
}
```

- ❖ Owner can enable/disable swapping and can change swap threshold (buy default this value set to 0.001% from the total supply).

```
ftrace | funcSig
function setSwapBackSettings(
    bool _enabled↑,
    uint256 _num↑,
    uint256 _denom↑
) external onlyOwner {
    swapEnabled = _enabled↑;
    gonSwapThreshold = TOTAL_GONS.div(_denom↑).mul(_num↑);
}
```

- ❖ Owner can change all fee receivers.

```
ftrace | funcSig
function setFeeReceivers(
    address _liquidityReceiver↑,
    address _treasuryReceiver↑,
    address _riskFreeValueReceiver↑
) external onlyOwner {
    liquidityReceiver = _liquidityReceiver↑;
    treasuryReceiver = _treasuryReceiver↑;
    riskFreeValueReceiver = _riskFreeValueReceiver↑;
}
```

- ❖ Owner can change all fees, each fee maximum up to 20% and total buy fees maximum up to 25% total sell fees can set maximum up to 65%.

```

ftrace | funcSig
function setFees(
    uint256 _liquidityFee↑,
    uint256 _riskFreeValue↑,
    uint256 _treasuryFee↑,
    uint256 _sellFeeTreasuryAdded↑,
    uint256 _sellFeeRFVAdded↑,
    uint256 _feeDenominator↑
) external onlyOwner {
    require(
        _liquidityFee↑ <= MAX_FEE_RATE &&
        _riskFreeValue↑ <= MAX_FEE_RATE &&
        _treasuryFee↑ <= MAX_FEE_RATE &&
        _sellFeeTreasuryAdded↑ <= MAX_FEE_RATE &&
        _sellFeeRFVAdded↑ <= MAX_FEE_RATE,
        "wrong"
    );

    liquidityFee = _liquidityFee↑;
    buyFeeRFV = _riskFreeValue↑;
    treasuryFee = _treasuryFee↑;
    sellFeeTreasuryAdded = _sellFeeTreasuryAdded↑;
    sellFeeRFVAdded = _sellFeeRFVAdded↑;
    totalBuyFee = liquidityFee.add(treasuryFee).add(buyFeeRFV);
    totalSellFee = totalBuyFee.add(sellFeeTreasuryAdded).add(
        sellFeeRFVAdded
    );
    feeDenominator = _feeDenominator↑;
    require(totalBuyFee < feeDenominator / 4);
}

```

- ❖ Owner can change max sell transaction without any limitation.

```

ftrace | funcSig
function setMaxSellTransaction(uint256 _maxTxn↑) external onlyOwner {
    maxSellTransactionAmount = _maxTxn↑;
}

```

- ❖ Owner can withdraw all BNB and bep20 tokens in the contract.

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

ftrace | funcSig
function rescueToken(address tokenAddress↑, uint256 tokens↑)
    external
    onlyOwner
    returns (bool success↑)
{
    return ERC20Detailed(tokenAddress↑).transfer(msg.sender, tokens↑);
}

ftrace | funcSig
```

- ❖ Owner can enable/disable auto rebase (by default this is disable)

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

ftrace | funcSig
```

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

ftrace | funcSig
```

- ❖ Owner can change rebase frequency maximum up to 30 min.

- ❖ Owner can change rewardYield value (when auto Rebase enable new tokens will mint according to these value)

```
ftrace | funcSig
function setRewardYield(
    uint256 _rewardYield↑,
    uint256 _rewardYieldDenominator↑
) external onlyOwner {
    rewardYield = _rewardYield↑;
    rewardYieldDenominator = _rewardYieldDenominator↑;
}
```

- ❖ Owner can enable/disable fees on normal transfers.

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

- ❖ Owner can change IP pairs if this value is true liquidity will add in BNB and if false IP will add in BUSD.

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

- ❖ Owner can manually change next auto rebase time.

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

Audit conclusion

While conducting the audit of the Libero Financial smart contract, it was observed that there is nothing alarming with the code and it contains informational concerns since the owner has substantial control within the ecosystem.