RugFreeCoins Audit

Staking Pro Token
Smart Contract Security Audit

September 16 2022

Contents

Audit details

Audited project
Stack Pro Token

Contract Address
0x1395a49930954CBcC07d2DC9Dc4ADEf8B4D1c0D1

Client contact
Stack Pro Team

Blockchain
Binance smart chain

Project website
https://www.stackpro.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.

Overview

✅ No mint function found; the owner cannot mint tokens after initial deployment.

✅ There are no max tx limits or wallet limits in the contract.

✅ The owner can’t pause trading.

✅ The owner can't set fees over 20%.

✅ Owner can't blacklist wallets.

✅ The owner can’t claim the contract’s balance of its own token.

Background

Rugfreecoins was commissioned by the Stack Pro Team to perform an audit of the smart contract.

https://bscscan.com/token/0x1395a49930954CBcC07d2DC9Dc4ADEf8B4D1c0D1

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

.

Roadmap

Q2 2022

- Protocol Launch
- Swap Platform Launch
- NFT Marketplace Launch
- SAFU Contract
- KYC Badge
- Social Media Campaign
- Influencer Campaign
- LaunchPad Campaign

Q3 2022

- Private Sale Campaign
- PreSale Campaign
- Public Sale
- Influencer Campaign
- Social Media Marketing
- Pancake Listing
- CoinGecko Listing
- CoinMarketCap Listing
-

Q4 2022

- Swap on AVAX Chain
- NFT Marketplace on AVAX Chain
- Influencer Campaign
- Social Media Marketing
- Merge Partnerships
- Team Expansion

Q1 2023

- SWAP on ETH Chain
- NFT Marketplace on AVAX Chain
- Strategic Marketing
- Team Expansion

Q2 2023

- Development Mobile Application iOS and Android
- Launch Mobile Application iOS and Android

.

Tokenomics

9% when buying & selling

- 2.5% of trade goes to the Liquidity pool.
- 2.5% of trade goes to the SPRO Insurance Fund in BUSD
- 2.5% of trade goes to the Treasury wallet in BUSD
- 1% of trade goes to the burn wallet.
- 1% of trade goes to a max buyer in every hour

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 interested in receiving fixed interest in 60 minutes and 200,000% per year.
- Anyone who’s interested in taking part in the future plans of the Stack Pro.
- Anyone who’s interested in making financial transactions with any other party using Stack Pro Token as the currency.

Potential to grow with score points

1.	Project efficiency	8/10
2.	Project uniqueness	9/10
3	Information quality	9/10
4	Service quality	9/10
5	System quality	8/10
6	Impact on the community	8/10
7	Impact on the business	9/10
8	Preparing for the future	8/10
9	Smart contract security	10/10
10	Smart contract functionality assessment	10/10
Total Points		**8.8/10**

Contract details

Token contract details for 16th of September 2022

Contract name	Stack Pro
Contract address	0x1395a49930954CBcC07d2DC9Dc4ADEf8B4D1c0D1
Token supply	500,000
Token ticker	SPRO
Decimals	18
Token holders	1
Transaction count	1
Insurance wallet	0x0738aee3c16736b993477227025c861cfaf48a0f
Treasury wallet	0x94557cbad5b7f6a964fa058864ac122707fa56d0
Contract deployer address	0x8e024e9a2818d2b7079b2e839d502f34354197f6
Contract's current owner address	0x8e024e9a2818d2b7079b2e839d502f34354197f6

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	
		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**		
L	Function Name	Visibility	Mutability	Modifiers
SPRO	**Implementation**	**IERC20, Ownable**		
L		Public ❗	🛑	NO❗
L	name	Public ❗		NO❗
L	symbol	Public ❗		NO❗
L	decimals	Public ❗		NO❗
L	rebase	Internal 🔒	🛑	
L	transfer	External ❗	🛑	validRecipient
L	transferFrom	External ❗	🛑	validRecipient
L	_basicTransfer	Internal 🔒	🛑	
L	_transferFrom	Internal 🔒	🛑	
L	takeFee	Internal 🔒	🛑	
L	swapAndLiquify	Private 🔐	🛑	
L	swapTokensForEth	Private 🔐	🛑	
L	swapTokensForTokens	Private 🔐	🛑	
L	addLiquidity	Private 🔐	🛑	

L	swapBack	Internal 🔒	🛑	swapping
L	shouldTakeFee	Internal 🔒		
L	shouldRebase	Internal 🔒		
L	shouldSwapBack	Internal 🔒		
L	enableSwap	External ❗	🛑	onlyOwner
L	setAutoRebase	External ❗	🛑	onlyOwner
L	setAutoAddLiquidity	External ❗	🛑	onlyOwner
L	allowance	External ❗		NO❗
L	enableTrading	Public ❗	🛑	onlyOwner
L	decreaseAllowance	External ❗	🛑	NO❗
L	increaseAllowance	External ❗	🛑	NO❗
L	approve	External ❗	🛑	NO❗
L	checkFeeExempt	External ❗		NO❗
L	updateBuyFees	Public ❗	🛑	onlyOwner
L	updateSellFees	Public ❗	🛑	onlyOwner
L	updateSwapPercentages	Public ❗	🛑	onlyOwner
L	getCirculatingSupply	Public ❗		NO❗
L	isNotInSwap	External ❗		NO❗
L	manualSync	External ❗	🛑	NO❗
L	setFeeReceivers	External ❗	🛑	onlyOwner
L	getLiquidityBacking	Public ❗		NO❗

L	setWhitelist	External ❗	🛑	onlyOwner
L	setLP	External ❗	🛑	onlyOwner
L	totalSupply	External ❗		NO❗
L	balanceOf	Public ❗		NO❗
L	isContract	Internal 🔒		
L	payBiggestBuyerOutside	External ❗	🛑	onlyOwner
L	_checkAndPayBiggestBuyer	Private 🔐	🛑	
L	_transferBNBToWallet	Private 🔐	🛑	
L	getPeriod	Public ❗		NO❗
L		External ❗	💵	NO❗
Ownable	**Implementation**	**Context**		
L		Public ❗	🛑	NO❗
L	owner	Public ❗		NO❗
L	_checkOwner	Internal 🔒		
L	renounceOwnership	Public ❗	🛑	onlyOwner
L	transferOwnership	Public ❗	🛑	onlyOwner
L	_transferOwnership	Internal 🔒	🛑	
IERC20	**Interface**			
L	totalSupply	External ❗		NO❗
L	balanceOf	External ❗		NO❗

L	transfer	External ❗	🛑	NO❗
L	allowance	External ❗		NO❗
L	approve	External ❗	🛑	NO❗
L	transferFrom	External ❗	🛑	NO❗
SignedSafe Math	**Library**			
L	mul	Internal 🔒		
L	div	Internal 🔒		
L	sub	Internal 🔒		
L	add	Internal 🔒		
SafeMath	**Library**			
L	tryAdd	Internal 🔒		
L	trySub	Internal 🔒		
L	tryMul	Internal 🔒		
L	tryDiv	Internal 🔒		
L	tryMod	Internal 🔒		
L	add	Internal 🔒		
L	sub	Internal 🔒		
L	mul	Internal 🔒		
L	div	Internal 🔒		

└	mod	Internal 🔒		
└	sub	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
IUniswapV2 Router02	**Interface**	**IUniswapV 2Router01**		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
IUniswapV2 Pair	**Interface**			
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	allowance	External ❗		NO❗
└	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❗

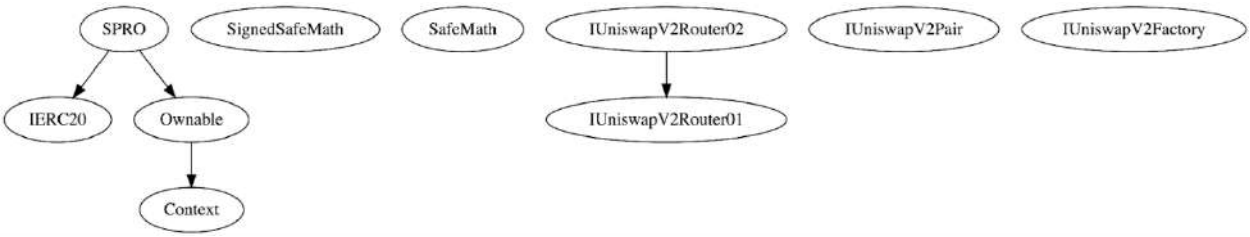
IUniswapV2 Factory	Interface			
L	feeTo	External ❗		NO❗
L	feeToSetter	External ❗		NO❗
L	getPair	External ❗		NO❗
L	allPairs	External ❗		NO❗
L	allPairsLength	External ❗		NO❗
L	createPair	External ❗	🛑	NO❗
L	setFeeTo	External ❗	🛑	NO❗
L	setFeeToSetter	External ❗	🛑	NO❗
Context	**Implementation**			
L	_msgSender	Internal 🔒		
L	_msgData	Internal 🔒		
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❗

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**
  No Centralization Risk found

Owner privileges

- The owner can enable trading/disable swapping

```
function enableSwap(bool status⬆) external onlyOwner {
    swapEnabled = status⬆;
}
```

- The owner can enable/disable rebase

```
ftrace | funcSig
function setAutoRebase(bool _flag⬆) external onlyOwner {
    if (_flag⬆) {
        _autoRebase = _flag⬆;
        _lastRebasedTime = block.timestamp;
    } else {
        _autoRebase = _flag⬆;
    }
}
```

- The owner can enable/disable adding liquidity

```
ftrace | funcSig
function setAutoAddLiquidity(bool _flag⬆) external onlyOwner {
    if (_flag⬆) {
        _autoAddLiquidity = _flag⬆;
        _lastAddLiquidityTime = block.timestamp;
    } else {
        _autoAddLiquidity = _flag⬆;
    }
}
```

❖ The owner can enable trading, once enabled cannot disable again

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

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

```
ftrace | funcSig
function updateBuyFees(
    uint256 burn,
    uint256 insurance,
    uint256 liquidity,
    uint256 maxBuyer,
    uint256 treasury
) public onlyOwner {
    buyAutoBurnFee = burn;
    buyInsuranceFee = insurance;
    buyLiquidityFee = liquidity;
    buyMaxBuyerFee = maxBuyer;
    buyTreasuryFee = treasury;

    buyTotalFee = burn.add(insurance).add(liquidity).add(maxBuyer).add(
        treasury
    );

    require(
        buyTotalFee.add(sellTotalFee) <= 200,
        "Fees can not be greater than 20%"
    );
}
```

❖ The owner can change all sell fees, total fees maximum up to 20%

```
ftrace | funcSig
function updateSellFees(
    uint256 burn,
    uint256 insurance,
    uint256 liquidity,
    uint256 maxBuyer,
    uint256 treasury
) public onlyOwner {
    sellAutoBurnFee = burn;
    sellInsuranceFee = insurance;
    sellLiquidityFee = liquidity;
    sellMaxBuyerFee = maxBuyer;
    sellTreasuryFee = treasury;

    sellTotalFee = burn.add(insurance).add(liquidity).add(maxBuyer).add(
        treasury
    );

    require(
        buyTotalFee.add(sellTotalFee) <= 200,
        "Fees can not be greater than 20%"
    );
}
```

❖ Owner can change all swap percentages

```
ftrace | funcSig
function updateSwapPercentages(
    uint256 insurance,
    uint256 liquidity,
    uint256 maxBuyer,
    uint256 treasury
) public onlyOwner {
    insuranceSwap = insurance;
    liquiditySwap = liquidity;
    maxBuyerSwap = maxBuyer;
    treasurySwap = treasury;

    totalSwap = insurance.add(liquidity).add(maxBuyer).add(treasury);
}
```

❖ The owner can change all fee receiver address

```
ftrace | funcSig
function setFeeReceivers(
    address _treasuryFeeWallet⬆,
    address _insuranceFeeWallet⬆
) external onlyOwner {
    treasuryFeeWallet = _treasuryFeeWallet⬆;
    insuranceFeeWallet = _insuranceFeeWallet⬆;
}
```

❖ The owner can exclude wallets from the fees

```
ftrace | funcSig
function setWhitelist(address _addr⬆) external onlyOwner {
    _isFeeExempt[_addr⬆] = true;
}
```

❖ The owner can change lp address

```
ftrace | funcSig
function setLP(address _address⬆) external onlyOwner {
    pairContract = IUniswapV2Pair(_address⬆);
}
```

❖ The owner can manually send reward to biggest buyer

```
ftrace | funcSig
function payBiggestBuyerOutside(uint256 _hour⬆) external onlyOwner {
    _checkAndPayBiggestBuyer(_hour⬆);
}
```

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: **PASS**

Number of risk issues: **0**

Solidity code functional issue level: **PASS**

Number of owner privileges: **11**

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

Smart contract active ownership: **YES**