



RugFreeCoins Audit



SPACE PEPE Token Smart Contract Security Audit

May 11th ,2023

Contents

Audit details	1
Disclaimer	2
Overview	3
Background	4
Roadmap	5
Target market and the concept	7
Potential to grow with score points	8
Total Points	8
Contract details	9
Contract code function details	10
Contract description table	12
Security issue checking status	18
Owner privileges	20
Audit conclusion	24

Audit details



Audited project
SPACE PEPE Token



Contract Address
0xc637EF75ab6772Bb6872B5a8Cb65F96a461Da267



Client contact
SPACE PEPE Team



Blockchain
Binance smart chain



Project website
<https://spacepepe.id/>

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.
- ✓ The owner can't set a max transaction limit
- ✓ The owner can't pause trading.
- ✓ The owner can't set fees over 25%.
- ✓ The owner can't blacklist wallets.
- ✓ The owner can't set a max wallet limit
- ✗ The owner can claim the contract's balance of its own token.

Background

Rugfreecoins was commissioned by the SPACE PEPE Team to perform an audit of the smart contract.

<https://bscscan.com/token/0xc637EF75ab6772Bb6872B5a8Cb65F96a461Da267>

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

Phase 01

- Website and Whitepaper
- Community Building
- KYC & Audit
- Global Marketing Campaign
- Social Media Ads Campaign
- Prelaunch Marketing
- Airdrop claimed

Phase 02

- Fair Launch on Pinksale
- Pancakeswap Listing
- CG Listing
- CMC Listing
- Viral Twitter Marketing Campaign
- 50K+ holder
- NFT realeased

Phase 03

- Project Team Partnership
- Build 10+ Local Community
- 100k+ Holders
- Top Tier CEX Listing
- All features released

Tokenomics

3% tax when buying & selling

3% of trade goes in BNB to the marketing wallet

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 taking part of the SPACE PEPE to earn ecosystem.
- Anyone who's interested in taking part in the future plans of SPACE PEPE Token.
- Anyone who's interested in making financial transactions with any other party using SPACE PEPE Token as the currency.

Potential to grow with score points

1.	Project efficiency	9/10
2.	Project uniqueness	9/10
3	Information quality	9/10
4	Service quality	9/10
5	System quality	9/10
6	Impact on the community	9/10
7	Impact on the business	9/10
8	Preparing for the future	8/10
9	Smart contract security	9/10
10	Smart contract functionality assessment	9/10
Total Points		8.9/10

Contract details

Token contract details for 11th of May 2023

Contract name	SPACE PEPE
Contract address	0xc637EF75ab6772Bb6872B5a8Cb65F96a461Da267
Token supply	210,000,000,000,000,000
Token ticker	\$PEPE
Decimals	6
Token holders	13,362
Transaction count	23,564
Contract deployer address	0x11D7dd648E430bC0c0A018C4EacE78821d5E03cD
Contract's current owner address	0x11d7dd648e430bc0c0a018c4eace78821d5e03cd
Marketing wallet	0x054428914C1C4703933680AF26e43d1644b94563













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	Low issue
		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	
		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		Low issue












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
SafeMath	Library			
L	add	Internal 		
L	sub	Internal 		
L	sub	Internal 		
L	mul	Internal 		
L	div	Internal 		
L	div	Internal 		
IERC20	Interface			
L	totalSupply	External 		NO 
L	decimals	External 		NO 
L	symbol	External 		NO 

L	name	External !		NO !
L	getOwner	External !		NO !
L	balanceOf	External !		NO !
L	transfer	External !		NO !
L	allowance	External !		NO !
L	approve	External !		NO !
L	transferFrom	External !		NO !
IFactoryV2	Interface			
L	getPair	External !		NO !
L	createPair	External !		NO !
IV2Pair	Interface			
L	factory	External !		NO !
L	getReserves	External !		NO !
L	sync	External !		NO !
IRouter01	Interface			

L	factory	External !		NO !
L	WETH	External !		NO !
L	addLiquidityETH	External !		NO !
L	addLiquidity	External !		NO !
L	swapExactETHForTokens	External !		NO !
L	getAmountsOut	External !		NO !
L	getAmountsIn	External !		NO !
IRouter02	Interface	IRouter01		
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External !		NO !
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForTokens	External !		NO !
SPEPE	Implementation	IERC20		
L		Public !		NO !
L		External !		NO !

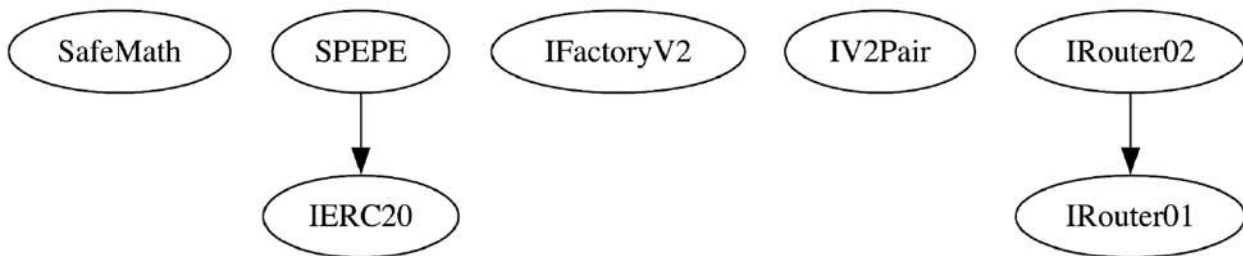
L	totalSupply	External !		NO !
L	decimals	External !		NO !
L	symbol	External !		NO !
L	name	External !		NO !
L	getOwner	External !		NO !
L	allowance	Public !		NO !
L	balanceOf	Public !		NO !
L	transfer	Public !		NO !
L	approve	External !		NO !
L	_approve	Internal 		
L	approveContractContingency	External !		onlyOwner
L	transferFrom	External !		NO !
L	isExcludedFromProtection	External !		NO !
L	_hasLimits	Internal 		
L	_transfer	Internal 		
L	finalizeTransfer	Internal 		

L	_checkLiquidityAdd	Internal 		
L	transferOwner	External !		onlyOwner
L	renounceOwnership	External !		onlyOwner
L	setExcludedFromProtection	External !		onlyOwner
L	excludePresaleAddresses	External !		onlyOwner
L	setOperator	External !		onlyOwner
L	setDevAddress	External !		onlyOwner
L	excludeMultipleAccountsFromFees	Public !		onlyOwner
L	enableTrading	Public !		onlyOwner
L	setFee	Public !		onlyOwner
L	setLPPair	External !		onlyOwner
L	withdrawToken	External !		onlyOwner
L	withdrawStuckETH	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 emitting a transfer event for fees transfer to devAddress, its better to emit the transfer event for it to populate actual data in block explorer

```
_balances[to↑] = _balances[to↑].add(transferAmount);  
_balances[devAddress] = _balances[devAddress].add(fee);  
  
emit Transfer(from↑, to↑, amount↑);
```

Operator address is not using anywhere in the contract, it's recommended to remove it to save gas fee.

```
ftrace | funcSig  
function setOperator(address _newOperator↑) external onlyOwner {  
    address oldOperator = operator;  
    if (oldOperator != address(0)) {  
        _liquidityHolders[oldOperator] = false;  
    }  
    operator = _newOperator↑;  
    _liquidityHolders[_newOperator↑] = true;  
}
```


❖ Centralization Risk

hasLiquidityAdded value will set to true if the owner does any transaction event, it's not getting added liquidity

```
if (!hasLiqBeenAdded) {  
    _checkLiquidityAdd(from↑, to↑);  
    if (  
        !hasLiqBeenAdded &&  
        _hasLimits(from↑, to↑) &&  
        !_isExcludedFromProtection[from↑] &&  
        !_isExcludedFromProtection[to↑] &&  
        !other  
    ) {  
        revert("Pre-liquidity transfer protection.");  
    }  
}
```

Owner privileges

- ❖ Owner can transfer ownership

```
ftrace | funcSig
function transferOwner(address _newOwner↑) external onlyOwner {
    require(
        _newOwner↑ != address(0),
        "Call renounceOwnership to transfer owner to the zero address."
    );
    require(
        _newOwner↑ != DEAD,
        "Call renounceOwnership to transfer owner to the zero address."
    );
    if (balanceOf(_owner) > 0) {
        finalizeTransfer(_owner, _newOwner↑, balanceOf(_owner));
    }

    address oldOwner = _owner;
    _owner = _newOwner↑;
    _isExcludedFromFee[_owner] = true;

    emit OwnershipTransferred(oldOwner, _newOwner↑);
}
```

- ❖ Owner can renounce the ownership

```
ftrace | funcSig
function renounceOwnership() external onlyOwner {
    address oldOwner = _owner;
    _owner = address(0);
    emit OwnershipTransferred(oldOwner, address(0));
}
```

- ❖ Owner can include/exclude wallets from protection (included wallets can do transactions before adding lp)

```
ftrace | funcSig
function setExcludedFromProtection(
    address _account↑,
    bool _enabled↑
) external onlyOwner {
    _isExcludedFromProtection[_account↑] = _enabled↑;
}
```

- ❖ Owner can exclude pre sale address

```
ftrace | funcSig
function excludePresaleAddresses(
    address _router↑,
    address _presale↑
) external onlyOwner {
    require(allowedPresaleExclusion);
    require(
        _router↑ != address(this) &&
        _presale↑ != address(this) &&
        lpPair != _router↑ &&
        lpPair != _presale↑,
        "Just don't."
    );
    if (_router↑ == _presale↑) {
        liquidityHolders[_presale↑] = true;
    } else {
        liquidityHolders[_router↑] = true;
        liquidityHolders[_presale↑] = true;
    }
}
```

- ❖ Owner can change dev address

```
ftrace | funcSig
function setDevAddress(address _newAddress↑) external onlyOwner {
    require(devAddress != address(0), "address cannot be 0");
    devAddress = payable(_newAddress↑);
}
```

- ❖ Owner can include/exclude multiple wallets from fees

```
ftrace | funcSig
function excludeMultipleAccountsFromFees(
    address[] calldata _accounts↑,
    bool _excluded↑
) public onlyOwner {
    for (uint256 i = 0; i < _accounts↑.length; i++) {
        isExcludedFromFee[_accounts↑[i]] = _excluded↑;
    }
}
```

- ❖ Owner can enable trading, once enabled can not disable again

```
ftrace | funcSig
function enableTrading() public onlyOwner {
    require(!tradingEnabled, "Trading already enabled!");
    require(hasLiqBeenAdded, "Liquidity must be added.");
    tradingEnabled = true;
    allowedPresaleExclusion = false;
}

ftrace | funcSig
function setFee(
```

- ❖ Owner can change buy and sell tax upto 4%

```
ftrace | funcSig
function setFee(
    uint256 _taxFeeOnBuy↑,
    uint256 _taxFeeOnSell↑
) public onlyOwner {
    require(_taxFeeOnBuy↑ < 40, "Tax cannot be more than 3.");
    require(_taxFeeOnSell↑ < 40, "Tax cannot be more than 3.");
    taxFeeOnBuy = _taxFeeOnBuy↑;
    taxFeeOnSell = _taxFeeOnSell↑;
}

ftrace | funcSig
```

- ❖ Owner can change lp pair address

```
ftrace | funcSig
function setLPPair(address _lpPair↑) external onlyOwner {
    lpPair = _lpPair↑;
}

ftrace | funcSig
```

- ❖ Owner can withdraw any bep20 token and bnb from the contract

```
ftrace | funcSig
function withdrawToken(
    address _token↑,
    address _to↑
) external onlyOwner returns (bool _sent↑) {
    require(_token↑ != address(0), "_token address cannot be 0");
    uint256 _contractBalance = IERC20(_token↑).balanceOf(address(this));
    _sent↑ = IERC20(_token↑).transfer(_to↑, _contractBalance);
}

// withdraw ETH if stuck or someone sends to the address
ftrace | funcSig
function withdrawStuckETH() external onlyOwner {
    bool success;
    (success, ) = address(msg.sender).call{value: address(this).balance}(
        ""
    );
}
```

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: **3**

Solidity code functional issue level: **PASS**

Number of owner privileges: **10**

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

Smart contract active ownership: **ACTIVE**