RugFreeCoins Audit

iTube™

iTube Token

Smart Contract Security Audit

September 6, 2022

Contents

Audit details

Audited project
iTube Token

Contract Address
0xB4192c333E6d38a5D2a21dcf8C83283F3c97ef3c

Client contact
iTube Team

Blockchain
Binance smart chain

Project website
https://itube.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.

✅ The owner can’t set a max transaction limit lower than 0.1% of the circulating supply.

✅ The owner can’t pause trading.

✅ The owner can't set fees over 25%.

✅ Owner can't blacklist wallets.

✅ The owner can’t set a max wallet limit lower than 1% of the circulating supply.

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

Background

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

https://bscscan.com/token/0xB4192c333E6d38a5D2a21dcf8C83283F3c97ef3c

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

.

ROADMAP

AUGUST 2022: Concept development

- Intro Website Launch
- Team Introduction
- KYC and contract audits by pinksale (including a SAFU badge)
- Marketing & Organic community growth
- Mentor & Celebrity Sign Ups
- Beta Platform Sneak Peak
- Doxxing Phase One
- BSC Token Launch
- CoinMarketCap pre-listed
- CoinGecko listing

Q4 2022: Beta launch and marketing

- Beta App Launched
- CEX Listing
- Bitmart Listing
- Github Listing
- Solidity Finance Audit
- Doxxing Phase Two
- Corporate Restructuring
- Mega Exchange Listings x 2
- Production Phase One
- International Partnered Talent Hunt
- Billboards & Publications

Q1 2023: Public Token Sale Begins

- Launch of ranking system for content bringers and viewers
- Global Inductions
- Contracting Structure with Producers
- Full Platform Launch with Complimentary Subscription to Token Holders
- International Publishing Partnerships
- International Corporate Establishment

Q2 2023: Beta App User Testing

- Partnership Expansion Program
- Application Gateway Integration
- Content acquisition & partnerships for populating content
- First iTube Original Web Series announced
- Partnerships with big influencers
- Partnerships with global events for Live Stream
- Updated website to go Live
- Top 10 exchange Listing

Q3 2023: Public Token Sale Begins

- Major Over the Top Collaborations
- iTube Reels (short videos Release)
- iTube – Live Stream platform to be launched
- The first concert to be streamed live on iTube live.
- iTube – Collaborations with influencers
- Partnership with more CEX
- Wallet & Swap functionality on Platform
- Website V2
- App analytics and advertising on platform.

Tokenomics

3% when buying & selling

- 2% of trade goes to the marketing wallet in BNB.
- 1% of trade goes to the reward wallet in BNB.

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 to staking and receive rewards.
- Anyone who’s interested in taking part in the content creation platform.
- Anyone who’s interested in taking part in the future plans of iTube.
- Anyone who’s interested in making financial transactions with any other party using iTube 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	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		**8.9/10**

Contract details

Token contract details for 6th of September 2022

Contract name	iTube
Contract address	0xB4192c333E6d38a5D2a21dcf8C83283F3c97ef3c
Token supply	100,000,000
Token ticker	iTube
Decimals	9
Token holders	1
Transaction count	1
Marketing wallet	0xc3843f10538071aa95b2f960cd55a9d39f43fd5d
Reward wallet	0xf7553b27207918c6d790797699f2e6efe92209cd
Contract deployer address	0xb39e5bFb5A012eb2E2261fBBD472842F24d57257
Contract’s current owner address	0xb39e5bfb5a012eb2e2261fbbd472842f24d57257

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
SafeMath	**Library**			
L	add	Internal 🔒		
L	sub	Internal 🔒		
L	sub	Internal 🔒		
L	mul	Internal 🔒		
L	div	Internal 🔒		
L	div	Internal 🔒		
BEP20	**Interface**			
L	getOwner	External ❗		NO❗
L	balanceOf	External ❗		NO❗
L	transfer	External ❗	🛑	NO❗

└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
Auth	**Implementation**			
└		Public ❗	🛑	NO❗
└	authorize	External ❗	🛑	onlyOwner
└	unauthorize	External ❗	🛑	onlyOwner
└	isOwner	Public ❗		NO❗
└	isAuthorized	Public ❗		NO❗
└	renounceOwnership	External ❗	🛑	onlyOwner
└	transferOwnership	External ❗	🛑	onlyOwner
└	acceptOwnership	External ❗	🛑	NO❗
IDEX Factory	**Interface**			
└	createPair	External ❗	🛑	NO❗
IDEX Router	**Interface**			

└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
iTube	**Implementation**	**BEP20, Auth**		
└		Public ❗	🛑	Auth
└		External ❗	💵	NO❗
└	getOwner	External ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	Public ❗	🛑	NO❗
└	approveMax	External ❗	🛑	NO❗
└	transfer	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
└	setMaxWalletPercent_base1000	External ❗	🛑	onlyOwner
└	setMaxTxPercent_base1000	External ❗	🛑	onlyOwner
└	_transferFrom	Internal 🔒	🛑	

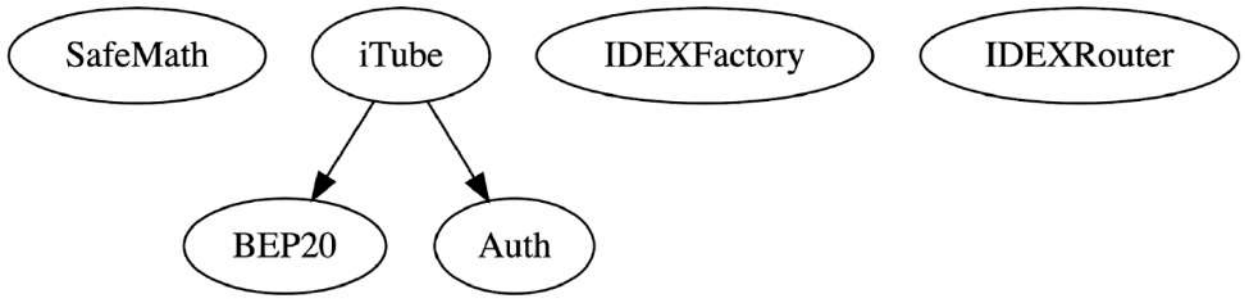
L	_basicTransfer	Internal 🔒	🛑	
L	takeFee	Internal 🔒	🛑	
L	shouldSwapBack	Internal 🔒		
L	clearStuckToken	External ❗	🛑	onlyOwner
L	tradingEnable	External ❗	🛑	onlyOwner
L	swapBack	Internal 🔒	🛑	swapping
L	manage_FeeExempt	External ❗	🛑	authorized
L	manage_TxLimitExempt	External ❗	🛑	authorized
L	manage_WalletLimitExempt	External ❗	🛑	authorized
L	update_fees	Internal 🔒	🛑	
L	setMultipliers	External ❗	🛑	authorized
L	setFees_base1000	External ❗	🛑	onlyOwner
L	setFeeReceivers	External ❗	🛑	onlyOwner
L	setSwapBackSettings	External ❗	🛑	onlyOwner
L	getCirculatingSupply	Public ❗		NO❗

L	multiTransfer	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 change the max wallet percentage minimum up to 1%

```
ftrace | funcSig
function setMaxWalletPercent_base1000(uint256 maxWallPercent_base1000⬆) external onlyOwner {
    require(maxWallPercent_base1000⬆ >= 10,"Cannot set max wallet less than 1%");
    _maxWalletToken = (totalSupply * maxWallPercent_base1000⬆ ) / 1000;
    emit config_MaxWallet(_maxWalletToken);
}
```

- The owner can change the max transaction limit minimum up to 0.2%

```
ftrace | funcSig
function setMaxTxPercent_base1000(uint256 maxTXPercentage_base1000⬆) external onlyOwner {
    require(maxTXPercentage_base1000⬆ >= 2,"Cannot set max transaction less than 0.2%");
    _maxTxAmount = (totalSupply * maxTXPercentage_base1000⬆ ) / 1000;
    emit config_MaxTransaction(_maxTxAmount);
}
```

- The owner can withdraw any bep20 tokens to the owner wallet (Cannot withdraw native tokens)

```
ftrace | funcSig
function clearStuckToken(address tokenAddress⬆, uint256 tokens⬆) external onlyOwner returns (bool success⬆) {
    require(tokenAddress⬆ != address(this), "Cannot withdraw native token");

    if(tokens⬆ == 0){
        tokens⬆ = BEP20(tokenAddress⬆).balanceOf(address(this));
    }

    emit clearToken(tokenAddress⬆, tokens⬆);

    return BEP20(tokenAddress⬆).transfer(msg.sender, tokens⬆);
}
```

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

```
// switch Trading
ftrace | funcSig
function tradingEnable() external onlyOwner {
    require(!tradingOpen,"Trading already open");
    tradingOpen = true;
    launchedAt = block.timestamp;
    emit config_TradingStatus(tradingOpen);
}
```

❖ The owner can include/exclude wallets from fees

```
ftrace | funcSig
function manage_FeeExempt(address[] calldata addresses⬆, bool status⬆) external authorized {
    require(addresses⬆.length < 501,"GAS Error: max limit is 500 addresses");
    for (uint256 i=0; i < addresses⬆.length; ++i) {
        isFeeExempt[addresses⬆[i]] = status⬆;
        emit Wallet_feeExempt(addresses⬆[i], status⬆);
    }
}
```

❖ The owner can include/exclude wallets from max transaction limit

```
ftrace | funcSig
function manage_TxLimitExempt(address[] calldata addresses⬆, bool status⬆) external authorized {
    require(addresses⬆.length < 501,"GAS Error: max limit is 500 addresses");
    for (uint256 i=0; i < addresses⬆.length; ++i) {
        isTxLimitExempt[addresses⬆[i]] = status⬆;
        emit Wallet_txExempt(addresses⬆[i], status⬆);
    }
}
```

❖ The owner can include/exclude wallets from max wallet limits

```
ftrace | funcSig
function manage_WalletLimitExempt(address[] calldata addresses⬆, bool status⬆) external authorized {
    require(addresses⬆.length < 501,"GAS Error: max limit is 500 addresses");
    for (uint256 i=0; i < addresses⬆.length; ++i) {
        isWalletLimitExempt[addresses⬆[i]] = status⬆;
        emit Wallet_holdingExempt(addresses⬆[i], status⬆);
    }
}
```

❖ The owner can change all fees total fees maximum up to 15%

```
ftrace | funcSig
function setMultipliers(uint256 _buy⬆, uint256 _sell⬆, uint256 _trans⬆) external authorized {
    sellMultiplier = _sell⬆;
    buyMultiplier = _buy⬆;
    transferMultiplier = _trans⬆;

    update_fees();
}

ftrace | funcSig
function setFees_base1000(uint256 _marketingFee⬆, uint256 _rewardFee⬆) external onlyOwner {
    marketingFee = _marketingFee⬆;
    rewardFee = _rewardFee⬆;
    totalFee = _marketingFee⬆ + _rewardFee⬆;

    update_fees();
}
```

❖ The owner can change all fee receiver wallets

```
ftrace | funcSig
function setFeeReceivers(address _marketingFeeReceiver⬆, address _rewardFeeReceiver⬆) external onlyOwner {
    require(_marketingFeeReceiver⬆ != address(0),"Marketing fee address cannot be zero address");
    require(_rewardFeeReceiver⬆ != address(0),"Team fee address cannot be zero address");

    marketingFeeReceiver = _marketingFeeReceiver⬆;
    rewardFeeReceiver = _rewardFeeReceiver⬆;

    emit Set_Wallets(marketingFeeReceiver, rewardFeeReceiver);
}
```

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

```
ftrace | funcSig
function setSwapBackSettings(bool _enabled⬆, uint256 _amount⬆) external onlyOwner {
    require(_amount⬆ < (totalSupply/10), "Amount too high");

    swapEnabled = _enabled⬆;
    swapThreshold = _amount⬆;

    emit config_SwapSettings(swapThreshold, swapEnabled);
}
```

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: **10**

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

Smart contract active ownership: **YES**