



RugFreeCoins Audit



Iqueue 10 Token

Smart Contract Security Audit

June 06, 2022

Contents

Audit details	1
Disclaimer	2
Background	3
About the project	4
Target market and the concept	7
Potential to grow with score points	9
Total Points	9
Contract details	10
Contract code function details	11
Contract description table	13
Security issue checking status	21
Audit conclusion	26

Audit details



Audited project
Iqueue 10 Token



Contract Address
0x1f7E1Aa5f1ED6928d74efF374C0E6E37964B2d77



Client contact
Iqueue 10 Team



Blockchain
Binance smart chain



Project website
<https://iq10.net/>

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 Iqueue 10 Team to perform an audit of the smart contract.

<https://bscscan.com/address/0x1f7E1Aa5f1ED6928d74efF374C0E6E37964B2d77#code>

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.

About the project

Iqueue 10 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, and **auto-staking protocol backed by Defi 3.0 yield farming** on BSC. Iqueue 10 token will bring an unparalleled, fixed APY of **1,010,101%** while imposing profound ease, simplicity, and accessibility upon all Iqueue 10 holders. Each transaction, purchase incurs a 6% fee, and sale incurs an 13% fee.

Features

- **2% of the fee when buying and 5% of the fee when selling is directed to the Iqueue 10 Buyback Assurance** which helps sustain and back the Staking Rewards provided by the Positive Rebase and these funds will be stored in a wallet for use when there's a dip.
- The **sustainability fee of 1% when buying and 2% when selling for development, and 4% for the owner** is what allows Iqueue 10 Token to hold the aforementioned promise. Tokens will be swapped into BNB and will be sent to the dev wallet. This way, Iqueue 10 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 1%**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.
- The **fee of 2% when buying and 5% when selling** is allocated for auto burn of tokens per transaction.

Roadmap

- Contact creation & development
- Contact Audit
- KYC
- Pink sale fair launch
- Fair launch refund system (FRS)
- 1000 TG members & holders
- Website redesign
- CMC listing
- CG listing
- Marketing AMA'S & Advertisements
- 5000 TG & Holders
- AIR DROP
- DAPP Release
- 10000 TG & IQ
- Certik audit
- Future plans chosen by community
- Next project announcement

Tokenomics

6% fee when buying

- 1% of trade goes to the liquidity pool.
- 1% of trade goes for the development fund in BNB
- 2% of trade goes to the burn wallet
- 2% of trade goes to manual buyback fund in BNB

13% fee when selling

- 1% of trade goes to the liquidity pool.
- 2% of trade goes for the development fund in BNB
- 5% of trade goes to the burn wallet
- 5% of trade goes to manual buyback fund 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 in receiving automatic staking and compound rewards every 10 seconds
- Anyone who's interested in receiving fixed interest in 10 seconds and 1,010,101% per year.
- Anyone who's interested in taking part in the future plans of the Iqueue 10 token.
- Anyone who's interested in making financial transactions with any other party using Iqueue 10 Token as the currency.

Core concept

Reward mechanism

2% when buying and 5% when selling of all trading fees are stored in the Iqueue 10 Token fund, which helps sustain and back the staking rewards provided by the positive rebase.

Iqueue 10 Token fund which is a separate wallet in the ecosystem. The Iqueue 10 Token fund 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 wallet.

In simple terms, the staking rewards (rebase rewards) which are distributed every 10 seconds at a rate of 1,010,101% per year are backed by the Iqueue 10 Token parameter, thus ensuring a high and stable interest rate for Iqueue 10 Token holders.

USDT Rewards

2% of each transaction gets converted to USDT and is split amongst all holders. Holders will be eligible to receive tokens in each transaction and rewards are proportional to how many tokens each individual holds.

Sustainable mechanism

The **sustainability fee of 1% when buying and 2% when selling for the development** is what allows Iqueue 10 Token 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 the dev wallet. This way, Iqueue 10 Token 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 1%, is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

The **fee of 2% when buying and 5% when selling** is allocated for auto burn of tokens per transaction.

Potential to grow with score points

1.	Project efficiency	9/10
2.	Project uniqueness	8/10
3	Information quality	8/10
4	Service quality	8/10
5	System quality	9/10
6	Impact on the community	8/10
7	Impact on the business	8/10
8	Preparing for the future	7/10
9	Smart contract security	10/10
10	Smart contract functionality assessment	10/10
Total Points		8.5/10

Contract details

Token contract details for 6th June 2022

Contract name	iqueue10
Contract address	0x1f7E1Aa5f1ED6928d74efF374C0E6E37964B2d77
Token supply	1,010,101,010
Token ticker	iq10
Decimals	5
Token holders	1
Transaction count	1
Dev Fee Receiver	0xbb83e670b41c4df06968afd81b55a71a24d97dbf
Buy back wallet	0x80d77cd06a49dbb5080cad66625dae9a9d9e69bd
Contract deployer address	0x34B04272DA7B9A597b690686c504AFFf083C1089
Contract's current owner address	0x5c6a5f9a4e0cacc397c073680bf156a36ba97443




















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















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
iqueue10	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	addLiquidity	Private 		
L	swapBack	Internal 		swapping





L	shouldTakeFee	Internal 		
L	shouldRebase	Internal 		
L	shouldSwapBack	Internal 		
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	setBotBlacklist	External !		onlyOwner

L	setLP	External !		onlyOwner
L	totalSupply	External !		NO !
L	balanceOf	Public !		NO !
L	isContract	Internal 		
L		External !		NO !
Ownable	Implementation	Context		
L		Public !		NO !
L	owner	Public !		NO !
L	renounceOwnership	Public !		onlyOwner
L	transferOwnership	Public !		onlyOwner
L	_transferOwnership	Internal 		
Context	Implementation			
L	_msgSender	Internal 		
L	_msgData	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 !
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 		
L	mod	Internal 		
L	sub	Internal 		
L	div	Internal 		
L	mod	Internal 		
SignedSafeMath	Library			
L	mul	Internal 		
L	div	Internal 		

L	sub	Internal 		
L	add	Internal 		
IUniswapV2 Router02	Interface	IUniswapV2 Router01		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External !		NO !
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External !		NO !
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External !		NO !
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External !		NO !
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 !
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 !

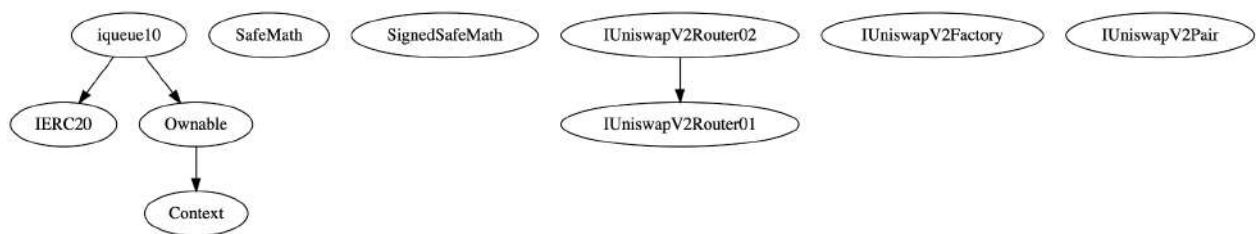
IUniswapV2Pair	Interface			
L	name	External !		NO !
L	symbol	External !		NO !
L	decimals	External !		NO !
L	totalSupply	External !		NO !
L	balanceOf	External !		NO !
L	allowance	External !		NO !
L	approve	External !		NO !
L	transfer	External !		NO !
L	transferFrom	External !		NO !
L	DOMAIN_SEPARATOR	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 !

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 issues**

No centralization issues found

Owner privileges

- ❖ The owner can enable/disable auto rebase

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

- ❖ The owner can enable/disable auto LP adding

```
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 change all buy and sell fees maximum up to 25%

```
ftrace | funcSig
function updateBuyFees(
    uint256 burn↑,
    uint256 buyBack↑,
    uint256 liquidity↑,
    uint256 dev↑
) public onlyOwner {
    buyAutoBurnFee = burn↑;
    buyBuyBackFee = buyBack↑;
    buyLiquidityFee = liquidity↑;
    buyDevFee = dev↑;

    buyTotalFee = burn↑.add(buyBack↑).add(liquidity↑).add(dev↑);

    require(buyTotalFee <= 25, "Fees can not be greater than 25%");
}

ftrace | funcSig
function updateSellFees(
    uint256 burn↑,
    uint256 buyBack↑,
    uint256 liquidity↑,
    uint256 dev↑
) public onlyOwner {
    sellAutoBurnFee = burn↑;
    sellBuyBackFee = buyBack↑;
    sellLiquidityFee = liquidity↑;
    sellDevFee = dev↑;

    sellTotalFee = burn↑.add(buyBack↑).add(liquidity↑).add(dev↑);

    require(sellTotalFee <= 25, "Fees can not be greater than 25%");
}
```

- ❖ The owner can change all swap percentages

```
ftrace | funcSig
function updateSwapPercentages(
    uint256 buyBack↑,
    uint256 liquidity↑,
    uint256 dev↑
) public onlyOwner {
    buyBackSwap = buyBack↑;
    liquiditySwap = liquidity↑;
    devSwap = dev↑;

    totalSwap = buyBack↑.add(liquidity↑).add(dev↑);
}
```

- ❖ The owner can change fee receiver wallets

```
ftrace | funcSig
function setFeeReceivers(address _buyBack↑, address _dev↑)
    external
    onlyOwner
{
    buyBackFeeWallet = _buyBack↑;
    devFeeWallet = _dev↑;
}
```

- ❖ The owner can exclude wallets from fees

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

- ❖ The owner can blacklist contracts from token contract

```
ftrace | funcSig
function setBotBlacklist(address _botAddress↑, bool _flag↑)
    external
    onlyOwner
{
    require(
        isContract(_botAddress↑),
        "only contract address, not allowed externally owned account"
    );
    blacklist[_botAddress↑] = _flag↑;
}
```

- ❖ The owner can change pair address

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

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: **9**

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

Smart contract active ownership: **YES**