Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

November, 2021

Audit Details

Audited project

JokerFootball

Deployer address

0x0b42398fb3739912f257033Bf070C0Ef5BfB91eA

Client contacts:

JokerFootball team

Blockchain

Binance Smart Chain

Project website:

https://jokerfootballtoken.com

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 below 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 TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (TechRate) owe no duty of care towards you or any other person, nor does TechRate 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 TechRate 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, TechRate hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against TechRate, 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

TechRate was commissioned by JokerFootball to perform an audit of smart contracts:

https://www.bscscan.com/address/0x20cc0f8139663d0233def10f9b3130588ffc66a9#code

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts Details

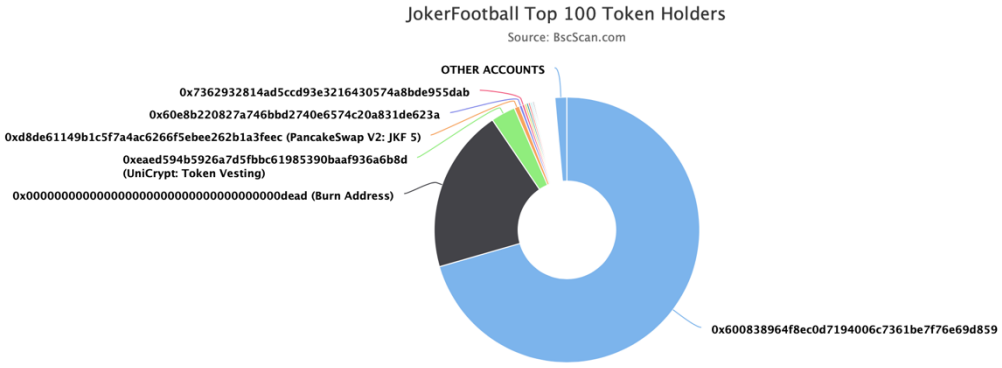
Token contract details for 16.11.2021

Contract name	JokerFootball
Contract address	0x20cc0f8139663d0233DeF10f9B3130588FFC66A9
Total supply	100,000,000,000,000,000
Token ticker	JKF
Decimals	9
Token holders	1,202
Transactions count	8,962
Top 100 holders dominance	98.54%
Buy project fee	30
Buy team fee	30
Sell project fee	90
Sell team fee	90
Uniswap V2 pair	0xd8DE61149B1c5F7a4Ac6266f5EbeE262B1A3feEC
Contract deployer address	0x0b42398fb3739912f257033Bf070C0Ef5BfB91eA
Contract's current owner address	0xF330Dd9D0CA8aBFbCb71070F5Ff614f109e2854B

JokerFootball Token Distribution

The top 100 holders collectively own 98.54% (98,542,150,789,623,500.00 Tokens) of JokerFootball

Token Total Supply: 100,000,000,000,000,000.00 Token | Total Token Holders: 1,202

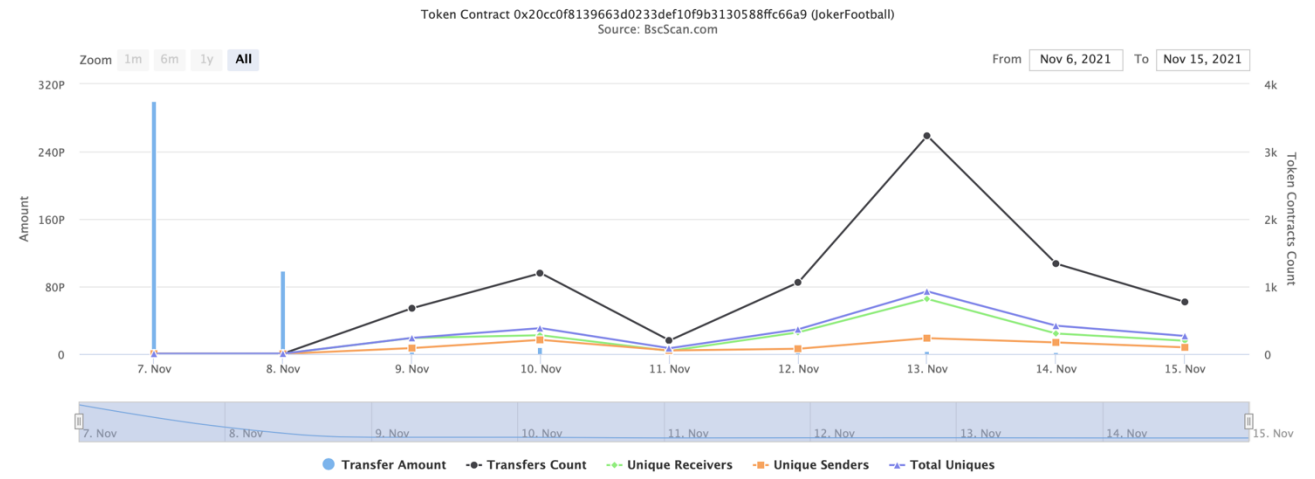


(A total of 98,542,150,789,623,500.00 tokens held by the top 100 accounts from the total supply of 100,000,000,000,000,000.00 token)

JokerFootball Contract Interaction Details

Time Series: Token Contract Overview

Sun 7, Nov 2021 - Mon 15, Nov 2021



JokerFootball Top 10 Token Holders

Rank	Address	Quantity	Percentage
1	0x600838964f8ec0d7194006c7361be7f76e69d859	70,530,525,594,200,000.2	70.5305%
2	Burn Address	20,000,000,000,000,000	20.0000%
3	UniCrypt: Token Vesting	2,989,588,469,270,000	2.9896%
4	PancakeSwap V2: JKF 5	614,777,595,628,402.801710181	0.6148%
5	0x60e8b220827a746bbd2740e6574c20a831de623a	345,387,445,385,599.43251061	0.3454%
6	0x7362932814ad5ccd93e3216430574a8bde955dab	310,086,401,290,000	0.3101%
7	0xa60434ff6c94ffb4da122ccdc39a759fb07675ab	250,000,000,000,000	0.2500%
8	0x3cf4ff7e92764bd5f560ca587d1b63e00c813d77	250,000,000,000,000	0.2500%
9	0xa079f84dfd1d390107a191ff2f4b406a0a02f05c	218,957,920,255,957.526783082	0.2190%
10	0x021a2abd74a4e155824ba3f6bebb8f6b39de9cfe	188,999,101,669,529.469988121	0.1890%

Contract functions details

+ [Int] IUniswapV2Router01
  - [Ext] factory
  - [Ext] WETH
  - [Ext] addLiquidity #
  - [Ext] addLiquidityETH ($)
  - [Ext] removeLiquidity #
  - [Ext] removeLiquidityETH #
  - [Ext] removeLiquidityWithPermit #
  - [Ext] removeLiquidityETHWithPermit #
  - [Ext] swapExactTokensForTokens #
  - [Ext] swapTokensForExactTokens #
  - [Ext] swapExactETHForTokens ($)
  - [Ext] swapTokensForExactETH #
  - [Ext] swapExactTokensForETH #
  - [Ext] swapETHForExactTokens ($)
  - [Ext] quote
  - [Ext] getAmountOut
  - [Ext] getAmountIn
  - [Ext] getAmountsOut
  - [Ext] getAmountsIn

+ [Int] IUniswapV2Router02 (IUniswapV2Router01)
  - [Ext] removeLiquidityETHSupportingFeeOnTransferTokens #
  - [Ext] removeLiquidityETHWithPermitSupportingFeeOnTransferTokens #
  - [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
  - [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens ($)
  - [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

+ [Int] IUniswapV2Factory
  - [Ext] feeTo
  - [Ext] feeToSetter
  - [Ext] getPair
  - [Ext] allPairs
  - [Ext] allPairsLength
  - [Ext] createPair #
  - [Ext] setFeeTo #
  - [Ext] setFeeToSetter #

+ [Lib] SafeMath
  - [Int] tryAdd
  - [Int] trySub
  - [Int] tryMul
  - [Int] tryDiv
  - [Int] tryMod
  - [Int] add
  - [Int] sub
  - [Int] mul
  - [Int] div
  - [Int] mod
  - [Int] sub
  - [Int] div
  - [Int] mod

+ [Int] IERC20
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #

+ [Int] IERC20Metadata (IERC20)
  - [Ext] name
  - [Ext] symbol
  - [Ext] decimals

+ Context
  - [Int] _msgSender
  - [Int] _msgData

+ Ownable (Context)
  - [Pub] <Constructor> #
  - [Pub] owner
  - [Pub] renounceOwnership #
    - modifiers: onlyOwner
  - [Pub] transferOwnership #
    - modifiers: onlyOwner
  - [Prv] _setOwner #

+ ERC20 (Context, IERC20, IERC20Metadata)
  - [Pub] <Constructor> #
  - [Pub] name
  - [Pub] symbol
  - [Pub] decimals
  - [Pub] totalSupply
  - [Pub] balanceOf
  - [Pub] transfer #
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] transferFrom #
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Int] _transfer #
  - [Int] _mint #
  - [Int] _burn #
  - [Int] _approve #
  - [Int] _beforeTokenTransfer #
  - [Int] _afterTokenTransfer #

+ JokerFootball (ERC20, Ownable)
  - [Pub] <Constructor> #
    - modifiers: ERC20
  - [Ext] <Fallback> ($)
  - [Ext] updateUniswapV2Router #
    - modifiers: onlyOwner
  - [Ext] setSwapTokensAmt #
    - modifiers: onlyOwner

- [Pub] excludeFromFees #
  - modifiers: onlyOwner
- [Pub] excludeMultipleAccountsFromFees #
  - modifiers: onlyOwner
- [Ext] blacklistAddress #
  - modifiers: onlyOwner
- [Ext] setTeamWallet #
  - modifiers: onlyOwner
- [Ext] setProjectWallet #
  - modifiers: onlyOwner
- [Ext] setBuyFee #
  - modifiers: onlyOwner
- [Ext] setSellFee #
  - modifiers: onlyOwner
- [Pub] setAutomatedMarketMakerPair #
  - modifiers: onlyOwner
- [Ext] setSwapEnabled #
  - modifiers: onlyOwner
- [Prv] _setAutomatedMarketMakerPair #
- [Pub] isExcludedFromFees
- [Int] _transfer #
- [Prv] swapAndSend #
- [Prv] swapTokensForEth #

($) = payable function
= non-constant function

Issues Checking Status

	Issue description	Checking status
1.	Compiler errors.	Passed
2.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3.	Possible delays in data delivery.	Passed
4.	Oracle calls.	Passed
5.	Front running.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow.	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Low issue
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	The impact of the exchange rate on the logic.	Passed
13.	Private user data leaks.	Passed
14.	Malicious Event log.	Passed
15.	Scoping and Declarations.	Passed
16.	Uninitialized storage pointers.	Passed
17.	Arithmetic accuracy.	Passed
18.	Design Logic.	Passed
19.	Cross-function race conditions.	Passed
20.	Safe Open Zeppelin contracts implementation and usage.	Passed
21.	Fallback function security.	Passed

Security Issues

High Severity Issues

No high severity issues found.

Medium Severity Issues

No medium severity issues found.

Low Severity Issues

1. Out of gas

Issue:

- The function `excludeMultipleAccountsFromFees()` uses the loop to exclude multiple accounts from fees. Function will be aborted with `OUT_OF_GAS` exception if there will be a long addresses list.

```
function excludeMultipleAccountsFromFees(address[] calldata accounts, bool excluded) public onlyOwner {
    for(uint256 i = 0; i < accounts.length; i++) {
        _isExcludedFromFees[accounts[i]] = excluded;
    }

    emit ExcludeMultipleAccountsFromFees(accounts, excluded);
}
```

Recommendation:
Be careful about accounts array length.

Owner privileges (In the period when the owner is not renounced)

- Owner can change Uniswap router address.

```
function updateUniswapV2Router(address newAddress) external onlyOwner {
    require(newAddress != address(uniswapV2Router), "JKF: The router already has that address");
    emit UpdateUniswapV2Router(newAddress, address(uniswapV2Router));
    uniswapV2Router = IUniswapV2Router02(newAddress);
    address _uniswapV2Pair = IUniswapV2Factory(uniswapV2Router.factory())
        .createPair(address(this), uniswapV2Router.WETH());
    uniswapV2Pair = _uniswapV2Pair;
}
```

- Owner can change swap tokens amount value.

```
function setSwapTokensAmt(uint256 amt) external onlyOwner{
    swapTokensAtAmount = amt;
}
```

- Owner can include in and exclude from fees.

```
function excludeFromFees(address account, bool excluded) public onlyOwner {
    _isExcludedFromFees[account] = excluded;

    emit ExcludeFromFees(account, excluded);
}

function excludeMultipleAccountsFromFees(address[] calldata accounts, bool excluded) public onlyOwner {
    for(uint256 i = 0; i < accounts.length; i++) {
        _isExcludedFromFees[accounts[i]] = excluded;
    }

    emit ExcludeMultipleAccountsFromFees(accounts, excluded);
}
```

- Owner can add / remove addresses from blacklist.

```
function blacklistAddress(address account, bool value) external onlyOwner{
    _isBlacklisted[account] = value;
}
```

- Owner can change team and project wallet addresses.

```
function setTeamWallet(address payable wallet) external onlyOwner{
    _teamWalletAddress = wallet;
}

function setProjectWallet(address payable wallet) external onlyOwner{
    _projectWalletAddress = wallet;
}
```

- Owner can exclude and include addresses in automatedMarketMakerPairs array.

```
function setAutomatedMarketMakerPair(address pair, bool value) public onlyOwner {
    require(pair != uniswapV2Pair, "JKF: The PancakeSwap pair cannot be removed from automatedMarketMakerPairs

    _setAutomatedMarketMakerPair(pair, value);
}
```

- Owner can change buy project, buy team, sell project and sell team fees.

```
function setBuyFee(uint16 liqFee, uint16 team) external onlyOwner {
    buyFee.projectFee = liqFee;
    buyFee.teamFee = team;
    totalBuyFee = buyFee.projectFee + buyFee.teamFee;
}

function setSellFee(uint16 liqFee, uint16 team) external onlyOwner {
    sellFee.projectFee = liqFee;
    sellFee.teamFee = team;
    totalSellFee = sellFee.projectFee + sellFee.teamFee;
}
```

- Owner can enable / disable swap.

```
function setSwapEnabled(bool value) external onlyOwner{
    swapEnabled = value;
}
```

Conclusion

Smart contracts contain low severity issues and owner privileges! Liquidity pair contract's security is not checked due to out of scope.

Liquidity locking details NOT provided by the team.

TechRate note:

Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.