TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

June, 2021

Audit Details

Audited project

FootballHooligans

Deployer address

0xB188111E4AadF5881Ef0071b991c0359c2f52058

Client contacts:

FootballHooligans team

Blockchain

Binance Smart Chain

Project website:

Not provided by FootballHooligans team

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 FootballHooligans to perform an audit of smart contracts:

https://bscscan.com/address/0x03ce7649838a811c26c9eabe85a1884e3e0120d4#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.
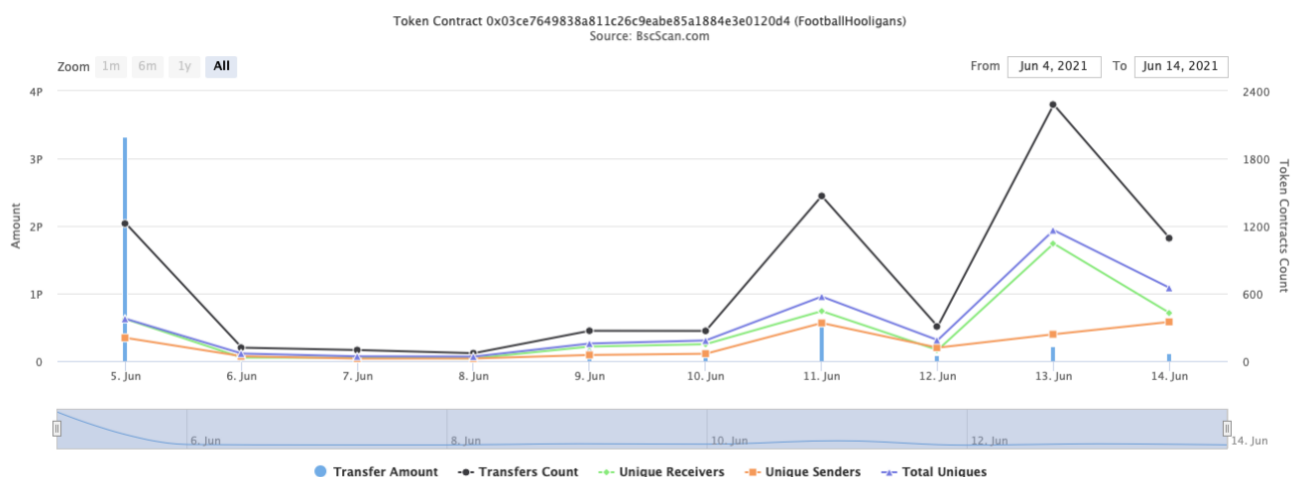
Contracts Details

Token contract details for 15.06.2021

Contract name	FootballHooligans
Contract address	0x03cE7649838A811c26C9eabe85A1884E3E0120D4
Total supply	1,000,000,000,000,000
Token ticker	FBH
Decimals	9
Token holders	2,078
Transactions count	8,493
Top 100 holders dominance	83.91%
Liquidity fee	3
Tax fee	4
Total fees	134910833422650661790895
Uniswap V2 pair	0x98c542bf4e61ac502150df89902a77cd53a19dcb
Contract deployer address	0xB188111E4AadF5881Ef0071b991c0359c2f52058
Contract's current owner address	0x0000000000000000000000000000000000000000

FootballHooligans Token Distribution

FootballHooligans Top 100 Token Holders
Source: BscScan.com

OTHER ACCOUNTS

0x0000000000000000000000000000000000000000dead (Burn Address)

0xcf52072cddd1b6bdadc9bd5ad957350e28344079
0x98c542bf4e61ac502150df89902a77cd53a19dcb (PancakeSwap V2: FBH 5)
0x2d045410f002a95efcee67759a92518fa3fce677
0xe0a78da7ba8e4709bd037450e0e119ef2633df5d

(A total of 839,140,564,192,637.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

FootballHooligans Contract Interaction Details

Time Series: Token Contract Overview   Sat 5, Jun 2021 - Mon 14, Jun 2021

Token Contract 0x03ce7649838a811c26c9eabe85a1884e3e0120d4 (FootballHooligans)
Source: BscScan.com

Zoom 1m 6m 1y All   From Jun 4, 2021 To Jun 14, 2021

● Transfer Amount -●- Transfers Count -+- Unique Receivers -■- Unique Senders -▲- Total Uniques

FootballHooligans Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	Burn Address	500,000,000,000,000	50.0000%
2	0xe0a78da7ba8e4709bd037450e0e119ef2633df5d	27,043,057,224,457.817828658	2.7043%
3	📄 0x2d045410f002a95efcee67759a92518fa3fce677	25,167,529,203,163.811945005	2.5168%
4	📄 PancakeSwap V2: FBH 5	19,643,899,920,525.754534694	1.9644%
5	0xcf52072cddd1b6bdadc9bd5ad957350e28344079	13,632,088,317,910.526206587	1.3632%
6	0x4c94e7cabe3468bf0c3d7ee42384085cacb6cb26	10,355,821,865,836.543444196	1.0356%
7	0x84e3ab7aa6eac4991fdef6763486be74282a348b	10,125,980,017,087.499945901	1.0126%
8	0x37a24bbc545b6445999ea63b80cd02e88fbf6f7e	10,123,990,399,622.066569199	1.0124%
9	0x1c014ccfc84c6e1b3c27287a257ccc352c0339e3	7,872,723,548,929.166727295	0.7873%
10	0x3c8e571302f3d11d8d292e041b6a5ccebbd18086	7,720,858,487,932.292889115	0.7721%

Contract functions details

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

+ [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

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

+ [Lib] Address
- **[Int]** isContract
- **[Int]** sendValue **#**
- **[Int]** functionCall **#**
- **[Int]** functionCall **#**
- **[Int]** functionCallWithValue **#**
- **[Int]** functionCallWithValue **#**
- **[Int]** functionStaticCall
- **[Int]** functionStaticCall
- **[Int]** functionDelegateCall **#**
- **[Int]** functionDelegateCall **#**
- **[Prv]** _verifyCallResult

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

+ [Int] IUniswapV2Factory
- **[Ext]** feeTo
- **[Ext]** feeToSetter

- **[Ext]** getPair
- **[Ext]** allPairs
- **[Ext]** allPairsLength
- **[Ext]** createPair **#**
- **[Ext]** setFeeTo **#**
- **[Ext]** setFeeToSetter **#**

+ **[Int]** IUniswapV2Pair
- **[Ext]** name
- **[Ext]** symbol
- **[Ext]** decimals
- **[Ext]** totalSupply
- **[Ext]** balanceOf
- **[Ext]** allowance
- **[Ext]** approve **#**
- **[Ext]** transfer **#**
- **[Ext]** transferFrom **#**
- **[Ext]** DOMAIN_SEPARATOR
- **[Ext]** PERMIT_TYPEHASH
- **[Ext]** nonces
- **[Ext]** permit **#**
- **[Ext]** MINIMUM_LIQUIDITY
- **[Ext]** factory
- **[Ext]** token0
- **[Ext]** token1
- **[Ext]** getReserves
- **[Ext]** price0CumulativeLast
- **[Ext]** price1CumulativeLast
- **[Ext]** kLast
- **[Ext]** mint **#**
- **[Ext]** burn **#**
- **[Ext]** swap **#**
- **[Ext]** skim **#**
- **[Ext]** sync **#**
- **[Ext]** initialize **#**

+ **[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 **#**

+ **FootballHooligans (Context, IERC20, Ownable)**
  - **[Pub]** <Constructor> **#**
  - **[Pub]** name
  - **[Pub]** symbol
  - **[Pub]** decimals
  - **[Pub]** totalSupply
  - **[Pub]** balanceOf
  - **[Pub]** transfer **#**
  - **[Pub]** allowance
  - **[Pub]** approve **#**
  - **[Pub]** transferFrom **#**
  - **[Pub]** increaseAllowance **#**
  - **[Pub]** decreaseAllowance **#**
  - **[Pub]** isExcludedFromReward
  - **[Pub]** totalFees
  - **[Pub]** deliver **#**
  - **[Pub]** reflectionFromToken
  - **[Pub]** tokenFromReflection
  - **[Pub]** excludeFromReward **#**
    - modifiers: onlyOwner
  - **[Ext]** includeInReward **#**
    - modifiers: onlyOwner
  - **[Prv]** _transferBothExcluded **#**
  - **[Pub]** excludeFromFee **#**
    - modifiers: onlyOwner
  - **[Pub]** includeInFee **#**
    - modifiers: onlyOwner
  - **[Ext]** setTaxFeePercent **#**
    - modifiers: onlyOwner
  - **[Ext]** setmarketingFeePercent **#**
    - modifiers: onlyOwner
  - **[Ext]** setLiquidityFeePercent **#**
    - modifiers: onlyOwner
  - **[Ext]** setMaxTxPercent **#**
    - modifiers: onlyOwner
  - **[Pub]** setSwapAndLiquifyEnabled **#**
    - modifiers: onlyOwner
  - **[Ext]** <Fallback> **($)**
  - **[Prv]** _reflectFee **#**
  - **[Prv]** _getValues
  - **[Prv]** _getTValues
  - **[Prv]** _getRValues
  - **[Prv]** _getRate
  - **[Prv]** _getCurrentSupply
  - **[Prv]** _takeLiquidity **#**
  - **[Prv]** _takemarketing **#**

- **[Prv]** calculateTaxFee
- **[Prv]** calculatemarketingFee
- **[Prv]** calculateLiquidityFee
- **[Prv]** removeAllFee **#**
- **[Prv]** restoreAllFee **#**
- **[Pub]** isExcludedFromFee
- **[Prv]** _approve **#**
- **[Prv]** _transfer **#**
- **[Prv]** swapAndLiquify **#**
  - modifiers: lockTheSwap
- **[Prv]** swapTokensForEth **#**
- **[Prv]** addLiquidity **#**
- **[Prv]** _tokenTransfer **#**
- **[Prv]** _transferStandard **#**
- **[Prv]** _transferToExcluded **#**
- **[Prv]** _transferFromExcluded **#**

($) = 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 issues
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 **includeInReward()** uses the loop to find and remove addresses from the **_excluded** list. Function will be aborted with **OUT_OF_GAS** exception if there will be a long excluded addresses list.

```solidity
function includeInReward(address account↑) external onlyOwner() {
    require(_isExcluded[account↑], "Account is already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account↑) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account↑] = 0;
            _isExcluded[account↑] = false;
            _excluded.pop();
            break;
        }
    }
}
```

- The function **_getCurrentSupply** also uses the loop for evaluating total supply. It also could be aborted with **OUT_OF_GAS** exception if there will be a long excluded addresses list.

```solidity
function _getCurrentSupply() private view returns (uint256, uint256) {
    uint256 rSupply = _rTotal;
    uint256 tSupply = _tTotal;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (
            _rOwned[_excluded[i]] > rSupply ||
            _tOwned[_excluded[i]] > tSupply
        ) return (_rTotal, _tTotal);
        rSupply = rSupply.sub(_rOwned[_excluded[i]]);
        tSupply = tSupply.sub(_tOwned[_excluded[i]]);
    }
    if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
    return (rSupply, tSupply);
}
```

Recommendation:
Check that the excluded array length is not too big.

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

- **Owner can change the tax, marketing and liquidity fee.**

```
ftrace | funcSig
function setTaxFeePercent(uint256 taxFee↑) external onlyOwner() {
    _taxFee = taxFee↑;
}


ftrace | funcSig
function setmarketingFeePercent(uint256 marketingFee↑) external onlyOwner() {
    _marketingFee = marketingFee↑;
}


ftrace | funcSig
function setLiquidityFeePercent(uint256 liquidityFee↑) external onlyOwner() {
    _liquidityFee = liquidityFee↑;
}
```

- **Owner can change the maximum transaction amount.**

```
function setMaxTxPercent(uint256 maxTxPercent↑) external onlyOwner() {
    _maxTxAmount = _tTotal.mul(maxTxPercent↑).div(
        10**2
    );
}
```

- **Owner can exclude from the fee.**

```
function excludeFromFee(address account↑) public onlyOwner {
    _isExcludedFromFee[account↑] = true;
}
```

Conclusion

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

Liquidity locking details provided by the team:
https://dxsale.app/app/pages/dxlockview?id=0&add=0xB188111E4AadF5881Ef0071b991c0359c2f52058&type=lplock&chain=BSC

Ownership renounce details provided by the team:
https://bscscan.com/tx/0x2e86d0db05f5b0bf586ccc348cb91d753ec363447d082a6e860a078494c90689

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.