Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

November, 2021

Audit Details

Audited project

Junior Shiba

Deployer address

0x4799716e16aa0b4c2e2288e375ddddb1c5d46016

Client contacts:

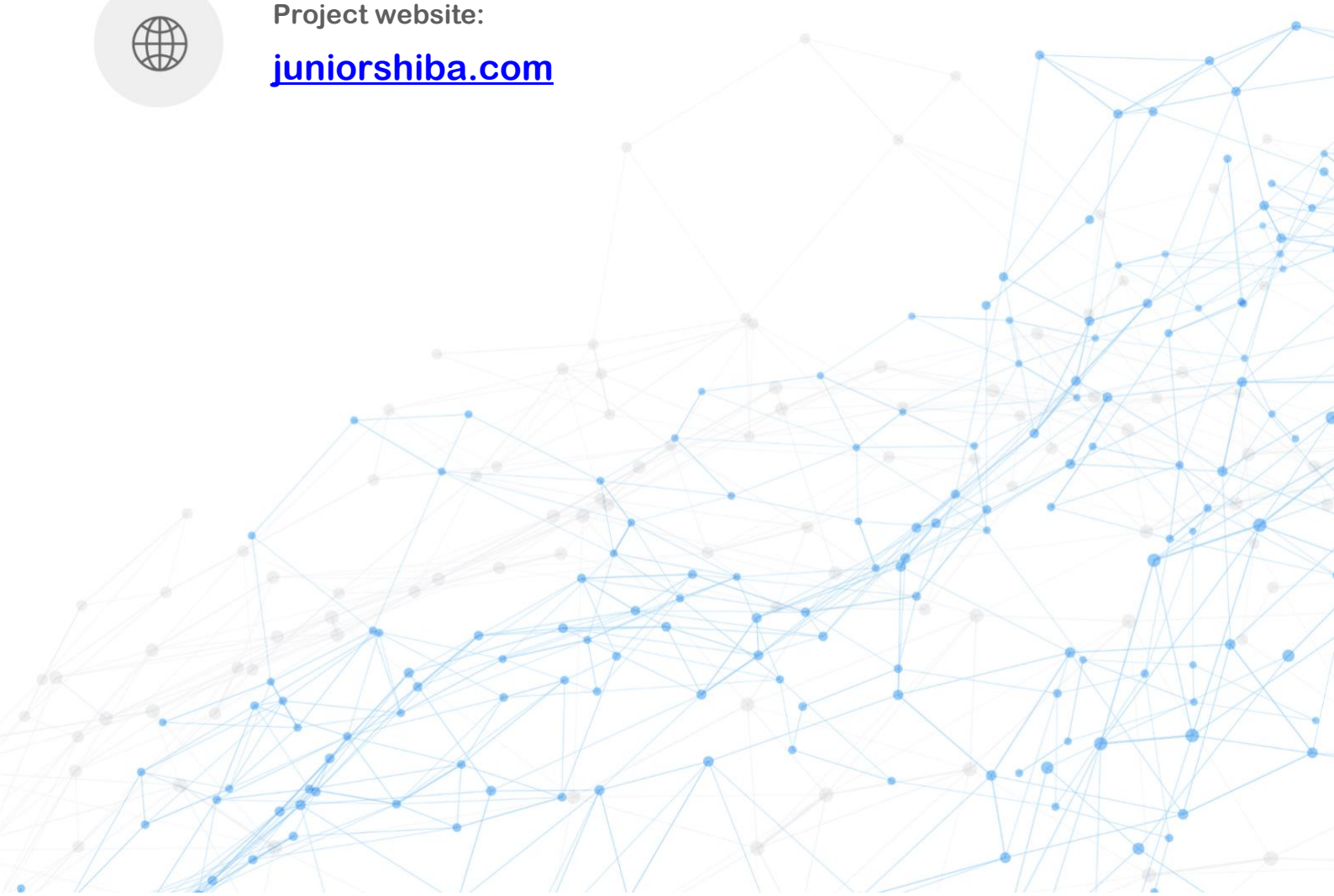
Junior Shiba team

Blockchain

Ethereum

Project website:

juniorshiba.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 Junior Shiba to perform an audit of smart contracts:

https://etherscan.io/address/0x73ee71cb9f0276f093f113c94c084a7a58ffd1e9#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 15.11.2021

Contract name	Junior Shiba
Contract address	0x73eE71Cb9F0276f093F113c94C084a7A58FFD1E9
Total supply	1,000,000,000,000,000
Token ticker	JRSHIB
Decimals	18
Token holders	5
Transactions count	5
Top 100 holders dominance	100.00%
Tax fee	3
Team fee	7
Uniswap Pair	0x97fed55bc2c982c611f051139c24bdb207fc6ccd
Contract deployer address	0x4799716e16aa0b4c2e2288e375ddddb1c5d46016
Contract’s current owner address	0x4799716e16aa0b4c2e2288e375ddddb1c5d46016

Junior Shiba Token Distribution

The top 100 holders collectively own 100.00% (1,000,000,000,000,000.00 Tokens) of Junior Shiba

Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 5

(A total of 1,000,000,000,000,000.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

Junior Shiba Contract Interaction Details

Time Series: Token Contract Overview

Tue 9, Nov 2021 - Wed 10, Nov 2021

Junior Shiba Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	0x4799716e16aa0b4c2e2288e375ddddb1c5d46016	500,000,000,000,000	50.0000%
2	0xbd2ab61c66e0564bc5f77564ce0ea74b425f4c32	200,000,000,000,000	20.0000%
3	Black Hole: 0x000...dEaD	200,000,000,000,000	20.0000%
4	0x8e790ae6fe1c6851aa91e74321afddf51e1c1011	50,000,000,000,000	5.0000%
5	0xe896ccf278196d052e717b4b0420121d39b69050	50,000,000,000,000	5.0000%

Contract functions details

+ [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

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

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

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

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

+ 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

+ JRSHIB (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] isExcluded
- [Ext] setExcludeFromFee #
  - modifiers: onlyOwner
- [Ext] enableDisableBuySellLimit #
  - modifiers: onlyOwner
- [Pub] totalFees
- [Pub] deliver #
- [Pub] reflectionFromToken
- [Pub] tokenFromReflection
- [Ext] excludeAccount #
  - modifiers: onlyOwner
- [Ext] includeAccount #
  - modifiers: onlyOwner
- [Prv] removeAllFee #
- [Prv] restoreAllFee #
- [Pub] isExcludedFromFee
- [Prv] _approve #
- [Prv] _transfer #
- [Prv] swapTokensForEth #
  - modifiers: lockTheSwap
- [Prv] sendETHToTeam #
- [Ext] manualSwap #
  - modifiers: onlyOwner
- [Ext] manualSend #
  - modifiers: onlyOwner
- [Ext] setSwapEnabled #
  - modifiers: onlyOwner
- [Ext] setCooldownEnabled #
  - modifiers: onlyOwner
- [Prv] _tokenTransfer #
- [Prv] _transferStandard #
- [Prv] _transferToExcluded #

- [Prv] _transferFromExcluded #
- [Prv] _transferBothExcluded #
- [Prv] _takeTeam #
- [Prv] _reflectFee #
- [Ext] <Fallback> ($)
- [Prv] _getValues
- [Prv] _getTValues
- [Prv] _getRValues
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Prv] _getTaxFee
- [Prv] _getMaxTxAmount
- [Pub] getMaxTxAmount
- [Pub] getNumOfTokensToExchangeForTeam
- [Pub] _getETHBalance
- [Ext] _setTaxFee #
  - modifiers: onlyOwner
- [Ext] _setTeamFee #
  - modifiers: onlyOwner
- [Ext] _setOpsTeamWallet #
  - modifiers: onlyOwner
- [Ext] _setMaxTxAmount #
  - modifiers: onlyOwner

($) = 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.	Passed
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

No low severity issues found.

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

- Owner can change the tax and team fee.

```
function _setTaxFee(uint256 taxFee⬆) external onlyOwner() {
    require(taxFee⬆ >= 1 && taxFee⬆ <= 3, 'taxFee should be between 1 and 3');
    _taxFee = taxFee⬆;
}

ftrace | funcSig
function _setTeamFee(uint256 teamFee⬆) external onlyOwner() {
    require(teamFee⬆ >= 1 && teamFee⬆ <= 7, 'teamFee should be between 1 and 7');
    _teamFee = teamFee⬆;
}
```

- Owner can change the maximum transaction amount.

```
function _setMaxTxAmount(uint256 maxTxAmount⬆) external onlyOwner()
{
    require(maxTxAmount⬆ >= MIN_BUY_SELL_TXN_AMOUNT  , 'maxTxAmount should be greater than MIN_BUY_SELL_TXN_AMOUNT');
    uint256 _tempMaxTxAmount     = maxTxAmount⬆ * (10**18);
    _maxTxAmount = _tempMaxTxAmount;
}
```

- Owner can exclude from the fee.

```
function setExcludeFromFee(address account⬆, bool excluded⬆) external onlyOwner() {
    _isExcludedFromFee[account⬆] = excluded⬆;
}
```

- Owner can enable/disable buy/sell limit.

```
function enableDisableBuySellLimit(bool _buySellLimitEnabled⬆) external onlyOwner()
{
    buySellLimitEnabled = _buySellLimitEnabled⬆;
}
```

- Owner can manually swap and send ETH to team.

```
ftrace | funcSig
function manualSwap() external onlyOwner() {
    uint256 contractBalance = balanceOf(address(this));
    swapTokensForEth(contractBalance);
}

ftrace | funcSig
function manualSend() external onlyOwner() {
    uint256 contractETHBalance = address(this).balance;
    sendETHToTeam(contractETHBalance);
}
```

- Owner can enable/disable swap and cooldown.

```
ftrace | funcSig
function setSwapEnabled(bool enabled⬆) external onlyOwner(){
    swapEnabled = enabled⬆;
}

ftrace | funcSig
function setCooldownEnabled(bool enabled⬆) external onlyOwner() {
    cooldownEnabled = enabled⬆;
}
```

- Owner can change _opsTeamWalletAddress.

```
function _setOpsTeamWallet(address payable opsTeamWalletAddress⬆) external onlyOwner() {
    _opsTeamWalletAddress = opsTeamWalletAddress⬆;
}
```

Conclusion

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

Liquidity locking details are 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.