Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

July, 2021

Audit Details

Audited project

BabyDogeInu

Deployer address

0x57125dfa501388E4eb51e686367639f4F38e9284

Client contacts:

BabyDogeInu team

Blockchain

Binance Smart Chain

Project website:

www.babydogeinu.io

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 BabyDogeInu to perform an audit of smart contracts:

https://bscscan.com/address/0x5e5c9001aa81332d405d993ffd1468751d659d1e#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

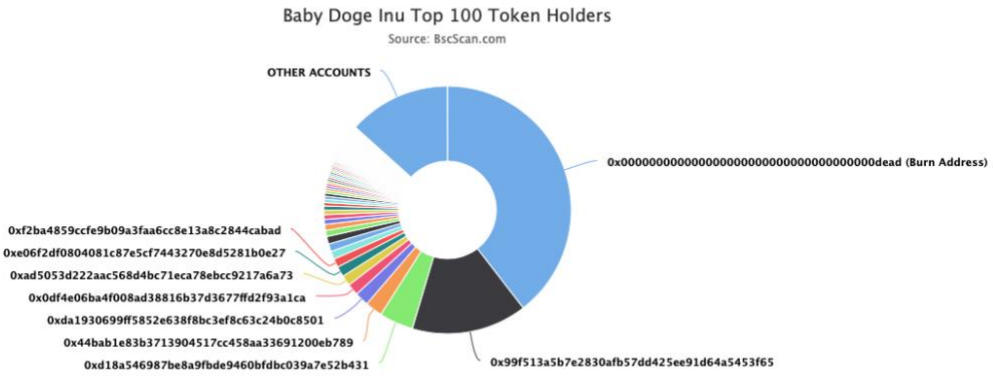
Token contract details for 15.07.2021

Contract name	BabyDogeInu
Contract address	0x5E5C9001Aa81332D405D993FFd1468751D659d1e
Total supply	1,000,000,000,000,000
Token ticker	$BABYDOGEINU
Decimals	9
Token holders	2,535
Transactions count	12,732
Top 100 holders dominance	86.64%
Liquidity fee	7
Tax fee	8
Total fees	155730986993327177358770
Uniswap V2 pair	0x99f513a5b7e2830afb57dd425ee91d64a5453f65
Contract deployer address	0x57125dfa501388E4eb51e686367639f4F38e9284
Contract's current owner address	0x0000000000000000000000000000000000000000

BabyDogeInu Token Distribution

The top 100 holders collectively own 86.64% (866,373,892,599,348.00 Tokens) of Baby Doge Inu

Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 2,535

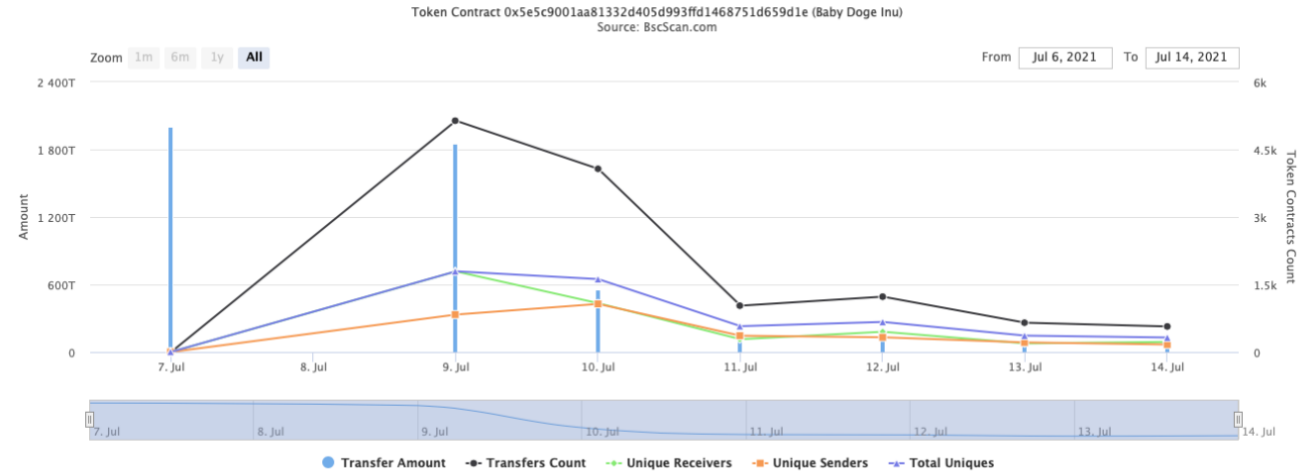


(A total of 866,373,892,599,348.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

BabyDogeInu Contract Interaction Details

Time Series: Token Contract Overview

Wed 7, Jul 2021 - Wed 14, Jul 2021



BabyDogeInu Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	Burn Address	395,280,000,000,000	39.5280%
2	0x99f513a5b7e2830afb57dd425ee91d64a5453f65	150,810,337,817,042.034964444	15.0810%
3	0xd18a546987be8a9fbde9460bfdbc039a7e52b431	44,547,119,999,999.999917875	4.4547%
4	0x44bab1e83b3713904517cc458aa33691200eb789	22,492,800,000,000.000003375	2.2493%
5	0xda1930699ff5852e638f8bc3ef8c63c24b0c8501	18,283,814,034,495.06244566	1.8284%
6	0x0df4e06ba4f008ad38816b37d3677ffd2f93a1ca	15,151,952,619,650.249071086	1.5152%
7	0xad5053d222aac568d4bc71eca78ebcc9217a6a73	13,683,780,646,531.902803223	1.3684%
8	0xe06f2df0804081c87e5cf7443270e8d5281b0e27	13,577,939,033,052.62889151	1.3578%
9	0xf2ba4859ccfe9b09a3faa6cc8e13a8c2844cabad	11,327,592,277,921.448744304	1.1328%
10	0xe70be69b76cd7e1f2c96587a077f62fd38da57fa	10,450,005,514,262.115493393	1.0450%

BabyDogeInu Top 10 Token Holders

Rank	Address	Quantity	Percentage
1	0x44bab1e83b3713904517cc458aa33691200eb789	5,749.246506011026833018	79.9001%
2	0x0000000000000000000000000000000000000000	736.530088366805847149	10.2359%
3	0x57125dfa501388e4eb51e686367639f4f38e9284	693.507909582716307454	9.6380%
4	0x07d80ae6f36a5e08dca74ce884a24d39db9934ed	15.789192732906133034	0.2194%
5	0x346600f6e6a8dc857c33c5c9a569f01f2546d694	0.346629930569073801	0.0048%
6	0x2b4839eae0afca26e0831a396fdb4d62603718bd	0.122281761193391722	0.0017%

Contract functions details

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

+ [Lib] SafeMath
  - [Int] add
  - [Int] sub
  - [Int] sub
  - [Int] mul
  - [Int] div
  - [Int] div
  - [Int] mod
  - [Int] mod

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

+ [Lib] Address
  - [Int] isContract
  - [Int] sendValue #
  - [Int] functionCall #
  - [Int] functionCall #
  - [Int] functionCallWithValue #
  - [Int] functionCallWithValue #
  - [Prv] _functionCallWithValue #

+ Ownable (Context)
  - [Pub] <Constructor> #
  - [Pub] owner
  - [Pub] renounceOwnership #
    - modifiers: onlyOwner
  - [Pub] transferOwnership #
    - modifiers: onlyOwner
  - [Pub] getUnlockTime
  - [Pub] getTime
  - [Pub] lock #
    - modifiers: onlyOwner
  - [Pub] unlock #

+ [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 #

+ BabyDogeInu (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] setTaxes #
  - modifiers: onlyOwner
- [Ext] setMaxTxAmount #
  - modifiers: onlyOwner
- [Ext] setNumTokensSellToAddToLiquidity #
  - modifiers: onlyOwner
- [Pub] setSwapAndLiquifyEnabled #
  - modifiers: onlyOwner
- [Pub] setSwapAndLiquifyByLimitOnly #
  - modifiers: onlyOwner
- [Ext] <Fallback> ($)
- [Prv] _reflectFee #
- [Prv] _getValues
- [Prv] _getTValues
- [Prv] _getRValues
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Prv] _takeLiquidity #
- [Prv] calculateTaxFee
- [Prv] calculateLiquidityFee
- [Prv] removeAllFee #
- [Prv] restoreAllFee #
- [Pub] isExcludedFromFee

- [Ext] setMarketingWalletAddress #
  - modifiers: onlyOwner
- [Ext] setTechWalletAddress #
  - modifiers: onlyOwner
- [Prv] _approve #
- [Prv] _transfer #
- [Prv] swapAndLiquify #
  - modifiers: lockTheSwap
- [Prv] swapTokensForEth #
- [Prv] addLiquidity #
- [Prv] transferToAddressETH #
- [Prv] _tokenTransfer #
- [Prv] _transferStandard #
- [Prv] _transferToExcluded #
- [Prv] _transferFromExcluded #
- [Ext] prepareForPreSale #
  - modifiers: onlyOwner
- [Ext] prepareForLaunch #
  - 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.	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.

```
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.

```
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 taxes.

```
ftrace | funcSig
function setTaxes(uint256 newRewardFee⬆, uint256 newLiquidityFee⬆, uint256 newMarketingfee⬆, uint256 newTechFee⬆) external onlyOwner() {

    _taxFee = newRewardFee⬆;
    _liquidityFee = newLiquidityFee⬆;
    _marketingFee = newMarketingfee⬆;
    _techFee = newTechFee⬆;

    _totalLiqFee = _liquidityFee.add(_marketingFee).add(_techFee);
    _prevTotalLiqFee = _totalLiqFee;
}
```

- Owner can change the maximum transaction amount.

```
ftrace | funcSig
function setMaxTxAmount(uint256 maxTxAmount⬆) external onlyOwner() {
    _maxTxAmount = maxTxAmount⬆;
}
```

- Owner can exclude from the fee.

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

- Owner can change marketing and tech wallets.

```
ftrace | funcSig
function setMarketingWalletAddress(address newAddress⬆) external onlyOwner() {
    marketingWalletAddress = payable(newAddress⬆);
}

ftrace | funcSig
function setTechWalletAddress(address newAddress⬆) external onlyOwner() {
    techWalletAddress = payable(newAddress⬆);
}
```

- Owner can set minimum number to add to liquidity.

```
function setNumTokensSellToAddToLiquidity(uint256 newLimit⬆) external onlyOwner() {
    _numTokensSellToAddToLiquidity = newLimit⬆;
}
```

- Owner can change swap and liquify settings.

```
ftrace | funcSig
function setSwapAndLiquifyEnabled(bool _enabled⬆) public onlyOwner {
    swapAndLiquifyEnabled = _enabled⬆;
    emit SwapAndLiquifyEnabledUpdated(_enabled⬆);
}

ftrace | funcSig
function setSwapAndLiquifyByLimitOnly(bool newValue⬆) public onlyOwner {
    swapAndLiquifyByLimitOnly = newValue⬆;
}
```

- Owner can enable presale and launch presets.

```
ftrace | funcSig
function prepareForPreSale() external onlyOwner {
    setSwapAndLiquifyEnabled(false);
    _taxFee = 0;
    _totalLiqFee = 0;
    _prevTotalLiqFee = 0;
    _maxTxAmount = 5000000 * 10**6 * 10**9;
}

ftrace | funcSig
function prepareForLaunch() external onlyOwner {
    setSwapAndLiquifyEnabled(true);
    _taxFee = 8;
    _totalLiqFee = _liquidityFee.add(_marketingFee).add(_techFee);
    _prevTotalLiqFee = _totalLiqFee;
    _maxTxAmount = 1000000000 * 10**6 * 10**9;
}
```

- Owner can lock and unlock. By the way, using these functions the owner could retake privileges even after the ownership was renounced.

```
ftrace | funcSig
function lock(uint256 time⬆) public virtual onlyOwner {
    _previousOwner = _owner;
    _owner = address(0);
    _lockTime = block.timestamp + time⬆;
    emit OwnershipTransferred(_owner, address(0));
}

ftrace | funcSig
function unlock() public virtual {
    require(_previousOwner == msg.sender, "You don't have permission to unlock");
    require(block.timestamp > _lockTime , "Contract is locked until 7 days");
    emit OwnershipTransferred(_owner, _previousOwner);
    _owner = _previousOwner;
}
```

Conclusion

Smart contracts contain low severity issues! Liquidity pair contract's security is not checked due to out of scope. 4/7 of the total liquidity fee goes to marketing and tech fee, 3/7 goes to liquidity.

Liquidity locking details provided by the team:
https://dxsale.app/app/v2_9/dxlockview?id=582&add=0&type=lpdefi&chain=BSC

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.