TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

July, 2021

Audit Details

Audited project

Baby Pig Token

Deployer address

0xECbAb2a565dbCd1398D7Ea90273391e61B42B8e0

Client contacts:

Baby Pig Token team

Blockchain

Binance Smart Chain

Project website:

https://www.babypig.org/

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 Baby Pig Token to perform an audit of smart contracts:
https://bscscan.com/token/0x7b0b4c304b9149b692c1acbc9db4f4a1c17b2b91

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 03.08.2021

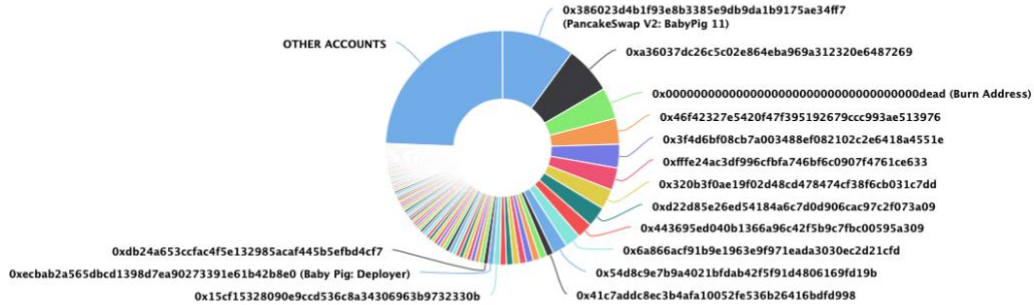|	
Contract name	Baby Pig Token
Contract address	0x7b0b4C304B9149B692c1Acbc9dB4F4a1c17B2B91
Total supply	784,044,330,000
Token ticker	BabyPig
Decimals	9
Token holders	7,629
Transactions count	28,037
Top 100 holders dominance	75.59%
Liquidity fee	22
Tax fee	8
Total fees	257643159344828841303
Uniswap V2 pair	0x386023d4b1f93e8b3385e9db9da1b9175ae34ff7
Contract deployer address	0xECbAb2a565dbCd1398D7Ea90273391e61B42B8e0
Contract's current owner address	0x54d8c9e7b9a4021bfdab42f5f91d4806169fd19b

Baby Pig Token Token Distribution

The top 100 holders collectively own 75.59% (592,643,274,101.98 Tokens) of Baby Pig Token | Token Total Supply: 784,044,330,000.00 Token | Total Token Holders: 7,629

Baby Pig Token Top 100 Token Holders
Source: BscScan.com

0x386023d4b1f93e8b3385e9db9da1b9175ae34ff7
(PancakeSwap V2: BabyPig 11)

0xa36037dc26c5c02e864eba969a312320e6487269

0x000000000000000000000000000000000000dead (Burn Address)

0x46f42327e5420f47f395192679ccc993ae513976

0x3f4d6bf08cb7a003488ef082102c2e6418a4551e

0xfffe24ac3df996cfbfa746bf6c0907f4761ce633

0x320b3f0ae19f02d48cd478474cf38f6cb031c7dd

0xd22d85e26ed54184a6c7d0d906cac97c2f073a09

0x443695ed040b1366a96c42f5b9c7fbc00595a309

0x6a866acf91b9e1963e9f971eada3030ec2d21cfd

0x54d8c9e7b9a4021bfdab42f5f91d4806169fd19b

0x41c7addc8ec3b4afa10052fe536b26416bdfd998

OTHER ACCOUNTS

0xdb24a653ccfac4f5e132985acaf445b5efbd4cf7

0xecbab2a565dbcd1398d7ea90273391e61b42b8e0 (Baby Pig: Deployer)

0x15cf15328090e9ccd536c8a34306963b9732330b

(A total of 592,643,274,101.98 tokens held by the top 100 accounts from the total supply of 784,044,330,000.00 token)

Baby Pig Token Contract Interaction Details

Time Series: Token Contract Overview   Thu 10, Jun 2021 - Sun 1, Aug 2021

Token Contract 0x7b0b4c304b9149b692c1acbc9db4f4a1c17b2b91 (Baby Pig Token)
Source: BscScan.com
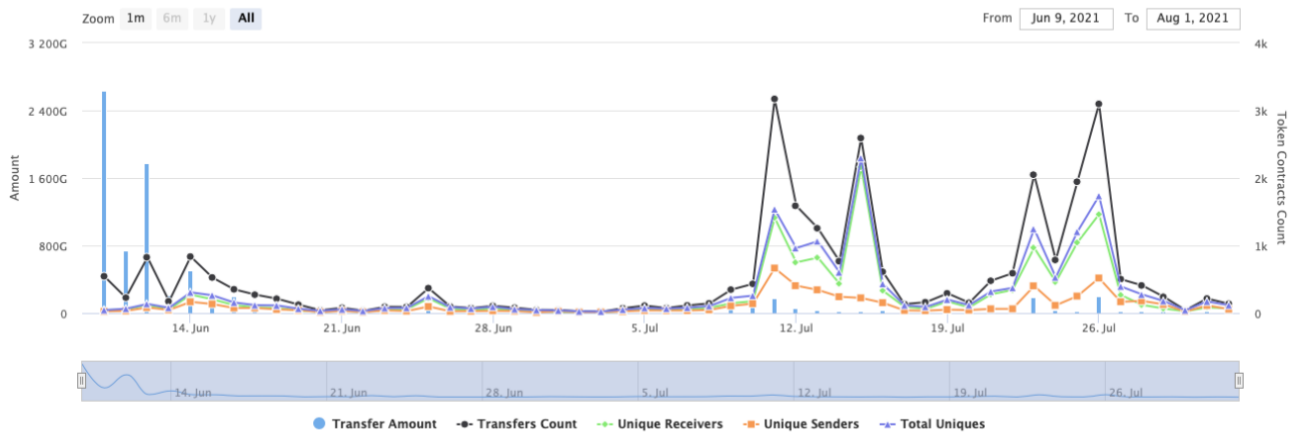
Baby Pig Token Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	📄 PancakeSwap V2: BabyPig 11	78,838,287,724.383332919	10.0553%
2	📄 0xa36037dc26c5c02e864eba969a312320e6487269	51,537,113,196.230907192	6.5732%
3	Burn Address	33,811,414,719.246556932	4.3124%
4	0x46f42327e5420f47f395192679ccc993ae513976	28,046,301,152.107494375	3.5771%
5	📄 0x3f4d6bf08cb7a003488ef082102c2e6418a4551e	25,673,935,754.828416897	3.2746%
6	0xfffe24ac3df996cfbfa746bf6c0907f4761ce633	24,155,580,547.978242484	3.0809%
7	0x320b3f0ae19f02d48cd478474cf38f6cb031c7dd	21,865,132,739.80858237	2.7888%
8	0xd22d85e26ed54184a6c7d0d906cac97c2f073a09	21,815,165,975.236044264	2.7824%
9	0x443695ed040b1366a96c42f5b9c7fbc00595a309	17,034,942,498.747608455	2.1727%
10	0x6a866acf91b9e1963e9f971eada3030ec2d21cfd	16,841,625,292.96659708	2.1480%

Baby Pig Token Top 10 LP Token Holders

Rank	Address	Quantity	Percentage
1	📄 0x3f4d6bf08cb7a003488ef082102c2e6418a4551e	154.345922	58.2165%
2	0x54d8c9e7b9a4021bfdab42f5f91d4806169fd19b	38.071476195729817537	14.3599%
3	📄 0xa36037dc26c5c02e864eba969a312320e6487269	30.242	11.4067%
4	0xdeb7bda3d5763a6cab3e013a3897e862f5e30666	9.831760665433018812	3.7084%
5	0x6851cba782e51687f517c2bfc011f09ae0832910	6.838008399089803856	2.5792%
6	0xd22d85e26ed54184a6c7d0d906cac97c2f073a09	1.773041715473953247	0.6688%
7	0xdb4348068af94b3059b61a474761e287b6c198ed	1.278449269038969892	0.4822%
8	0xb52f97570039662e7586e4eb47fe7b463dbd0b40	1.262705450236471618	0.4763%
9	0x128d986eed853e3249123731d5aeab2085ae398c	1.123017567498274582	0.4236%
10	0xc442456398fe364bcc7e250060c18519011388fc	0.965255364525497293	0.3641%

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)
- [Int] <Constructor> **#**
- **[Pub]** owner
- **[Pub]** renounceOwnership **#**
  - modifiers: onlyOwner
- **[Pub]** transferOwnership **#**
  - modifiers: onlyOwner
- **[Pub]** geUnlockTime
- **[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 **#**

+ **BabyPig** (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]** 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]** calculateTaxFee
  - **[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 and liquidity fee.**

```
function setTaxFeePercent(uint256 taxFee) external onlyOwner() {
    _taxFee = taxFee;
}

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;
}
```

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

```
//Locks the contract for owner for the amount of time provided
function lock(uint256 time) public virtual onlyOwner {
    _previousOwner = _owner;
    _owner = address(0);
    _lockTime = now + time;
    emit OwnershipTransferred(_owner, address(0));
}

//Unlocks the contract for owner when _lockTime is exceeds
function unlock() public virtual {
    require(_previousOwner == msg.sender, "You don't have permission to unlock");
    require(now > _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.

Liquidity locking details provided by the team:
https://deeplock.io/lock/0x386023d4b1f93E8b3385e9dB9DA1B9175ae34Ff7

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.