



# Smart Contract Security Audit

## Audit details:

Audited project:	BabyShark
Deployer address:	0x394a428265Ee9F7735F3e0904D7607C9EC28aA2F
Client contacts:	BabyShark team
Blockchain:	Binance Smart Chain
Project website:	Not provided by the BabyShark 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 BabyShark to perform an audit of smart contracts:

- <https://bscscan.com/address/0xcc9b175E4b88a22543C44F1cC65B73f63b0D4EfE#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 18.05.2021.

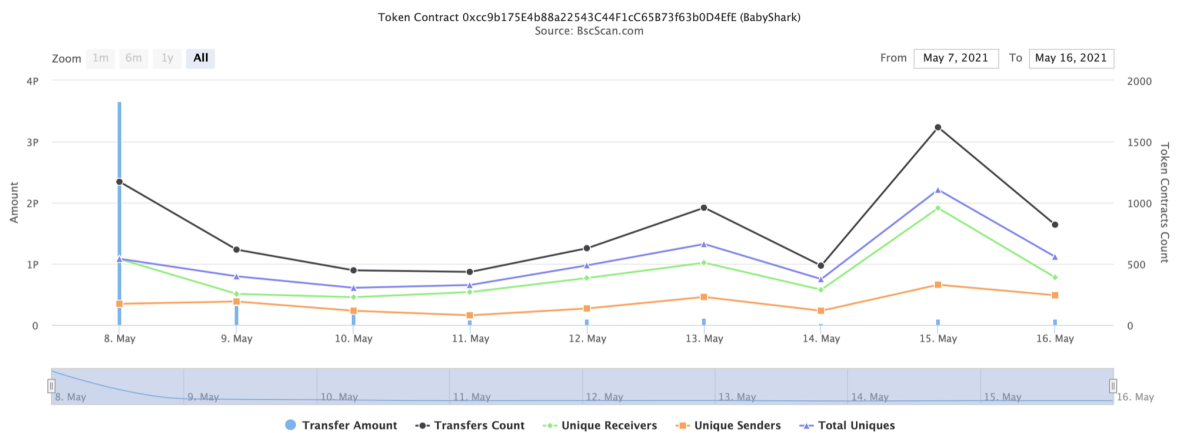
Contract name:	BabyShark
Contract address:	0xcc9b175E4b88a22543C44F1cC65B73f63b0D4EfE
Total supply:	1_000_000_000_000_000_000_000_000
Token ticker:	SHARK
Decimals:	9
Token holders:	2,593
Transactions count:	7,604
Top 100 holders dominance:	84.65%
Liquidity fee:	5
Tax fee:	3
Total fees:	51276456729510643228824
Uniswap V2 pair:	0xFD9891af26664F274af63Cbbae810Fd36422052E
Contract deployer address:	0x394a428265Ee9F7735F3e0904D7607C9EC28aA2F
Contract's current owner address:	<a href="https://bscscan.com/address/0x394a428265ee9f7735f3e0904d7607c9ec28aa2f">https://bscscan.com/address/0x394a428265ee9f7735f3e0904d7607c9ec28aa2f</a>

## BabyShark token distribution



💡 Token Total Supply: 1,000,000,000,000.00 Token | Total Token Holders: 2,593









## Sat 8, May 2021 - Sun 16, May 2021



# BabyShark top 10 token holders

Rank	Address	Quantity (Token)	Percentage
1	0x000000000000000000000000000000000000dead	200,000,000,083,787.210820625	20.0000%
2	 0xd9891af26664f274af63cbbae810fd36422052e	94,279,577,791,550.394119819	9.4280%
3	0x212bd7b9c9db1a229a13524fdb877eeee32d6a4e	22,474,479,308,341.025656384	2.2474%
4	0xc355d314bf3b0081dae779ea20ab425e620f24b6	21,445,434,468,137.685911558	2.1445%
5	0xfe965496d57587ff6bba9b37307cbd55567749c6	20,000,000,000,000	2.0000%
6	0x980d8ee1e858457af6d6ac67e0c3152cd31d29f7	19,310,943,248,995.118642179	1.9311%
7	0x5e68d93b1599340f692921028c14e3220f45e3be	18,315,953,390,606.703063391	1.8316%
8	0xd23adc76269275ef09240cdd61955827e413bdd3	14,153,124,124,247.720402171	1.4153%
9	0xdd3d6d332329753f5f163ec53dd4f7f60b9f472	13,933,835,691,260.942020797	1.3934%
10	 0x137edb03a0ad94c2feb93ba681b0b1c9af7f6c20	13,759,486,800,000.1	1.3759%

# BabyShark LP token holders

Rank	Address	Quantity	Percentage	Analytics
1	 0x142ea1e211e15e02f682f7c600156aca01531e0b	5,713.925096208397094161	80.3490%	
2	 0x00	1,323.906466253371590305	18.6167%	
3	0x07d80ae6f36a5e08dca74ce884a24d39db9934ed	73.423872256573810539	1.0325%	
4	0x254af9ecb3c74b57cc7ef26ae2847e31c08a4bcb	0.125314171222410424	0.0018%	

# 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] lockedLiquidity
- [Pub] charity
- [Pub] burn
- [Pub] renounceOwnership #
  - modifiers: onlyOwner
- [Pub] setCharityAddress #
  - modifiers: onlyOwner
- [Pub] setLockedLiquidityAddress #
  - 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 #
- + BabyShark (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] charityPercentageOfLiquidity
  - [Pub] totalCharityCollected
  - [Pub] deliver #
  - [Pub] reflectionFromToken
  - [Pub] tokenFromReflection
  - [Pub] excludeFromReward #
    - modifiers: onlyOwner
  - [Ext] includeInReward #
    - modifiers: onlyOwner
  - [Pub] devWallet
  - [Ext] setAsDevWallet #
    - modifiers: onlyOwner
  - [Priv] \_transferBothExcluded #
  - [Pub] excludeFromFee #
    - modifiers: onlyOwner

- [Pub] includeInFee #
  - 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 #
- [Prv] setDevWalletFee #
- [Pub] isExcludedFromFee
- [Prv] \_approve #
- [Prv] \_transfer #
- [Pub] collectCharity #
  - modifiers: onlyCharity
- [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.

```
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:

Use EnumerableSet instead of array or do not use long arrays.

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

❑ Owner can exclude from the fee.

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

❑ Owner can set as Dev wallet(fee increases x2)

```
function setAsDevWallet(address account) external onlyOwner() {
    _isDevWallet[account] = true;
}
```

❑ Owner can change charity and lockedLiquidity address

```
function setCharityAddress(address payable charityAddress) public virtual onlyOwner
{
    require(_charity == address(0), "Charity address cannot be changed once set");
    _charity = charityAddress;
}

function setLockedLiquidityAddress(address liquidityAddress) public virtual onlyOwner
{
    require(_lockedLiquidity == address(0), "Locked liquidity address cannot be changed once set");
    _lockedLiquidity = liquidityAddress;
}
```

## Conclusion

Smart contracts contain low severity issues. LP pair contract is not checked.

Liquidity locking details provided by the team:

<https://dxsale.app/app/pages/dxlockview?id=711&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.*