



# Audit de sécurité des contrats intelligents

## Détails de l'audit :

Projet audité :	BODA
Adresse du Deployer :	0x4acd5d62f01f13a397e5bf5cdf8f4c0a69534ede
Contacts clients :	L'équipe BODA
Blockchain :	Smart Chain de Binance
Site web du projet :	<a href="https://www.bodatoken.org">https://www.bodatoken.org</a>

# Avis de non-responsabilité

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.

# Contexte

TechRate a été chargé par le BODA de réaliser un audit des contrats intelligents :

- <https://bscscan.com/address/0x81cfb5e400eb2caa319130a0dae3b32cfb19392d#code>

L'objectif de l'audit était d'atteindre les objectifs suivants :

- Assurez-vous que le contrat intelligent fonctionne comme prévu.
- Identifier les problèmes de sécurité potentiels du contrat intelligent.

Les informations contenues dans ce rapport doivent être utilisées pour comprendre l'exposition au risque du contrat intelligent, et comme guide pour améliorer la posture de sécurité du contrat intelligent en remédiant aux problèmes qui ont été identifiés.

# Détails des contrats

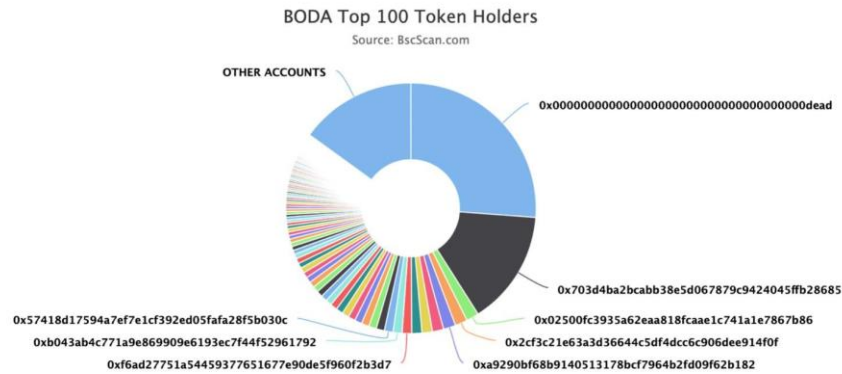
Détails du contrat des tokens pour le 06.05.2021.

Nom du contrat :	BODA
Adresse du contrat :	0x81cfb5e400eb2caa319130a0dae3b32cfb19392d
Approvisionnement total :	1_000_000_000_000_000_000_000_000
ticker du token:	BODA
Décimales :	9
Détenteurs de tokens :	1161
Nombre de transactions :	6645
Domination des détenteurs du Top 100 :	84.96 %
Commission de liquidité :	7
Frais d'impôts :	3
Total des frais :	158_388_874_733_266_276_146_654
PancakeSwap V2 paire :	0x703d4ba2bcabb38e5d067879c9424045ffb28685
Adresse du dépoyeur contractuel :	0x4acd5d62f01f13a397e5bf5cdf8f4c0a69534ede
Adresse du propriétaire actuel du contrat :	0x00

## Distribution de tokens BODA

💡 The top 100 holders collectively own 84.96% (849,644,135,423,820.00 Tokens) of BODA

💡 Token Total Supply: 1,000,000,000,000.00 Token | Total Token Holders: 1,161

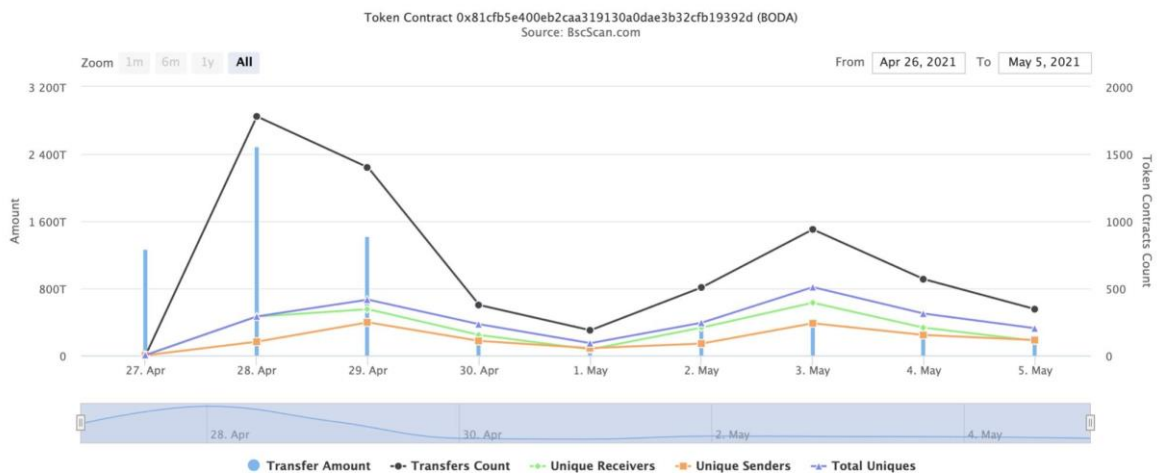


(A total of 849,644,135,423,820.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)


## Détails de l'interaction du contrat BODA

### Time Series: Token Contract Overview

Tue 27, Apr 2021 - Wed 5, May 2021



## Les 10 plus grands détenteurs de tokens BODA

Rank	Address	Quantity (Token)	Percentage
1	0x0000000000000000000000000000000000dead	261,846,632,068,150.734795191	26.1847%
2	 0x703d4ba2bcabb38e5d067879c9424045ffb28685	148,093,423,272,344.214316344	14.8093%
3	0x02500fc3935a62eaa818fcaae1c741a1e7867b86	16,182,538,750,959.627114805	1.6183%
4	0x2cf3c21e63a3d36644c5df4dcc6c906dee914f0f	16,003,463,057,035.183010255	1.6003%
5	0xa9290bf68b9140513178bcf7964b2fd09f62b182	15,949,136,925,840.478166507	1.5949%
6	0x8d76fe4578852c3bebd2034564daf3ded4d98406	14,493,183,093,023.014186082	1.4493%
7	0xe0d65e3741f1beea47cdf6a5a60cf9ef9fb8411b	13,508,986,903,972.547580741	1.3509%
8	0xf4e0e6dee785e8836f09986e1e15b6d9acc14e3d	13,019,975,334,181.42316904	1.3020%
9	0xf6ad27751a54459377651677e90de5f960f2b3d7	12,313,988,838,945.639751371	1.2314%
10	0xb043ab4c771a9e869909e6193ec7f44f52961792	11,689,496,228,270.477698551	1.1689%

## Détenteurs de tokens BODA LP

Rank	Address	Quantity	Percentage
1	 0x00	2,778.151297143223770026	83.9943%
2	0x0000000000000000000000000000000000dead	503.785866355147604036	15.2314%
3	0x07d80ae6f36a5e08dca74ce884a24d39db9934ed	24.902359302611641429	0.7529%
4	0xe9eff515b9e29c393af69d3c5905458de54fde5a	0.70863107114452575	0.0214%

# Détails des fonctions du contrat

## + [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 #
- [Priv] \_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 #

#### + BODA (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

# Problèmes de vérification de l'état

Nº	Description du problème.	Vérification du statut
1	Erreurs de compilation.	Validé
2	Conditions de course et réentrance. Conditions de course inter-fonctions.	Validé
3	Retards possibles dans la livraison des données.	Validé
4	Appelle d'Oracle.	Validé
5	En première ligne.	Validé
6	Dépendance vis-à-vis de l'horodatage.	Validé
7	Débordement et sous-exécution des nombres entiers.	Validé
8	DoS avec Revert.	Validé
9	DoS avec limite de gaz de blocage.	Faibles enjeux
10	Autorisations d'exécution des méthodes.	Validé
11	Modèle économique du contrat.	Validé
12	L'impact du taux de change sur la logique.	Validé
13	Fuites de données d'utilisateurs privés.	Validé
14	Journal des événements malveillants.	Validé
15	Champ d'application et déclarations.	Validé
16	Pointeurs de stockage non initialisés.	Validé
17	Précision arithmétique.	Validé
18	Logique de conception.	Validé
19	Conditions de course entre fonctions.	Validé
20	Mise en œuvre et utilisation sécurisées des contrats Open Zeppelin.	Validé
21	Sécurité de la fonction de repli.	Validé

# Questions de sécurité

## Problèmes de haute gravité

Aucun problème de haute gravité n'a été trouvé.

## Problèmes de gravité moyenne

Aucun problème de gravité moyenne n'a été trouvé.

## Problèmes de faible gravité

### 1. Panne de gaz

Question :

- ❑ La fonction `includeInReward()` utilise la boucle pour trouver et supprimer les adresses de la liste `_exclue`. La fonction sera interrompue avec l'exception `OUT_OF_GAS` s'il y a une longue liste d'adresses exclues.

```
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;
        }
    }
}
```

- ❑ La fonction `_getCurrentSupply` utilise également la boucle pour évaluer l'approvisionnement total. Elle peut également être interrompue par une exception `OUT_OF_GAS` si la liste des adresses exclues est longue.

```
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);
}
```

Recommandation : Utilisez `EnumerableSet` au lieu de tableau ou n'utilisez pas de tableaux longs.

# Conclusion

Les contrats intelligents ne contiennent pas de problèmes de haute gravité ! La sécurité du contrat de paire de liquidité n'est pas vérifiée car elle est hors de portée.

Note technique :

*Veillez vérifier la clause de non-responsabilité ci-dessus et noter que l'audit ne fait aucune déclaration ou garantie sur le modèle d'entreprise, l'attrait des investissements ou la durabilité du code. Le rapport est fourni pour le seul contrat mentionné dans le rapport et n'inclut pas d'autres contrats potentiels déployés par le propriétaire.*