Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

July, 2021

Audit Details

Audited project

DINIZIA

Deployer address

0x71c87346fd1C86BE81FB461443f102cF67b08bFC

Client contacts:

DINIZIA team

Blockchain

Binance Smart Chain

Project website:

https://dinizia.net/

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 DINIZIA to perform an audit of smart contracts:

https://bscscan.com/address/0x8b0926a1133be813a88a9c29303ebe65655087ab#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 12.10.2021

Contract name	DINIZIA
Contract address	0x8b0926a1133Be813a88A9C29303EBE65655087Ab
Total supply	10,000,000,000
Token ticker	DINI
Decimals	9
Token holders	7
Transactions count	7
Top 100 holders dominance	100.00%
Liquidity fee	4
Tax fee	2
Total tax fees	0
Uniswap V2 pair	0x8be4f0706cbd5c45da74a326141d53599d7f7568
Contract deployer address	0x71c87346fd1C86BE81FB461443f102cF67b08bFC
Contract's current owner address	0x4a262f6b263d63bc065c2b1cebe1670d2ce4c3db

DINIZIA Token Distribution

The top 100 holders collectively own 100.00% (10,000,000,000.00 Tokens) of DINIZIA

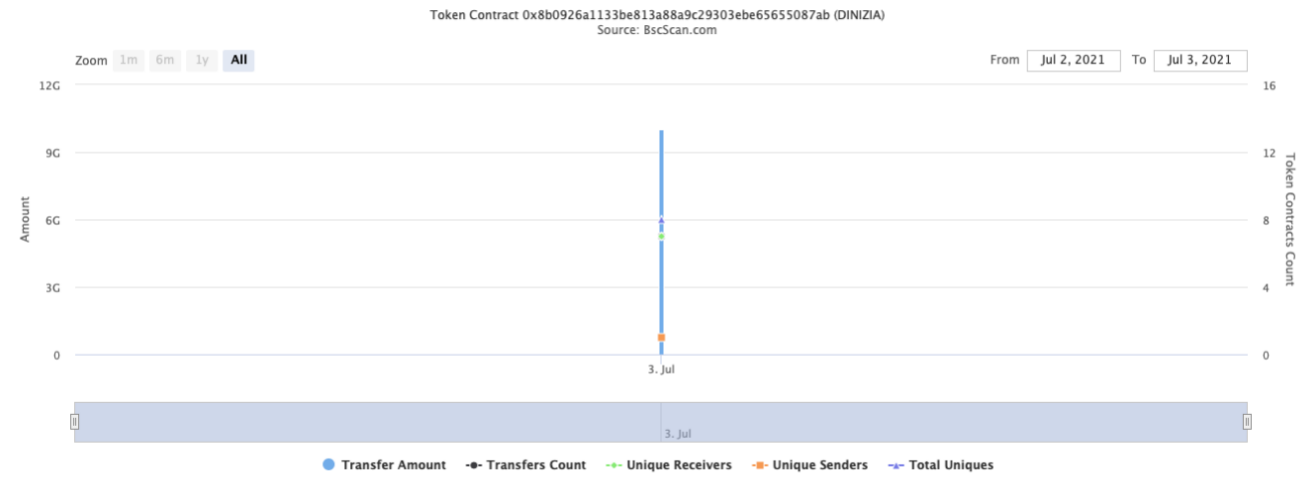
Token Total Supply: 10,000,000,000.00 Token | Total Token Holders: 7

(A total of 10,000,000,000.00 tokens held by the top 100 accounts from the total supply of 10,000,000,000.00 token)

DINIZIA Contract Interaction Details

Time Series: Token Contract Overview

Sat 3, Jul 2021 - Sat 3, Jul 2021



DINIZIA Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	0x4ed6bb7216076888b9103448c48ae509eca81eb3	4,000,000,000	40.0000%
2	0x566af44bcb2b1f13938eb34c83ded13fa1353fb1	2,500,000,000	25.0000%
3	0x25013ad82c85d876c99965ac7097c28351ecd43b	1,200,000,000	12.0000%
4	0x86add57df784a1b3f1e5b1155c7688d856ded4af	1,000,000,000	10.0000%
5	0xd7b538fb925423337d440888695fc3ae01499793	800,000,000	8.0000%
6	0x0f0dbdc1ec233d8edb1aa96e3b79beef262afdb0	300,000,000	3.0000%
7	0x17ae46b27127836fa321f1adf4987d6e0cb8eb43	200,000,000	2.0000%

Contract functions details

+ Context
  - [Int] _msgSender
  - [Int] _msgData
+ [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
  - [Int] ceil
+ [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
+ [Int] IUniswapV2Factory
  - [Ext] createPair #
+ [Int] IUniswapV2Pair
  - [Ext] sync #
+ [Int] IUniswapV2Router01
  - [Ext] factory
  - [Ext] WETH
  - [Ext] addLiquidity #
  - [Ext] addLiquidityETH ($)
+ [Int] IUniswapV2Router02 (IUniswapV2Router01)
  - [Ext] removeLiquidityETHSupportingFeeOnTransferTokens #
  - [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #
  - [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
  - [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens ($)
+ RewardWallet
  - [Pub] <Constructor> #
+ Balancer

```
  - [Pub] <Constructor> #
+ DINIZIA (Context, IERC20, Ownable)
  - [Pub] <Constructor> #
  - [Pub] name
  - [Pub] symbol
  - [Pub] decimals
  - [Pub] totalSupply
  - [Int] find2Percent
  - [Pub] balanceOf
  - [Pub] transfer #
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] transferFrom #
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Pub] isExcluded
  - [Pub] reflectionFromToken
  - [Pub] tokenFromReflection
  - [Ext] excludeAccount #
    - modifiers: onlyOwner
  - [Ext] includeAccount #
    - modifiers: onlyOwner
  - [Prv] _approve #
  - [Prv] _transfer #
  - [Prv] collectFee #
  - [Prv] _getReflectionRate
  - [Prv] swapAndLiquify #
    - modifiers: lockTheSwap
  - [Prv] swapTokensForEth #
  - [Prv] addLiquidity #
  - [Ext] setPair #
    - modifiers: onlyOwner
  - [Ext] setTaxless #
    - modifiers: onlyOwner
  - [Ext] setSwapAndLiquifyEnabled #
    - modifiers: onlyOwner
  - [Ext] setFeeActive #
    - modifiers: onlyOwner
  - [Ext] setTaxFee #
    - modifiers: onlyOwner
  - [Ext] setBurnFee #
    - modifiers: onlyOwner
  - [Ext] setLiquidityFee #
    - modifiers: onlyOwner
  - [Ext] setMaxTxAmount #
    - modifiers: onlyOwner
  - [Ext] setCharityFee #
    - modifiers: onlyOwner
  - [Ext] setMinTokensBeforeSwap #
    - modifiers: onlyOwner
  - [Ext] <Fallback> ($)
($) = 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 includeAccount () 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 includeAccount(address account) external onlyOwner() {
    require(_isExcluded[account], "ERC20: Account is already included");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tokenBalance[account] = 0;
            _isExcluded[account] = false;
            _excluded.pop();
            break;
        }
    }
}
```

- The function _getReflectionRate() 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 _getReflectionRate() private view returns (uint256) {
    uint256 reflectionSupply = _reflectionTotal;
    uint256 tokenSupply = _tokenTotal;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (
            _reflectionBalance[_excluded[i]] > reflectionSupply ||
            _tokenBalance[_excluded[i]] > tokenSupply
        ) return _reflectionTotal.div(_tokenTotal);
        reflectionSupply = reflectionSupply.sub(
            _reflectionBalance[_excluded[i]]
        );
        tokenSupply = tokenSupply.sub(_tokenBalance[_excluded[i]]);
    }
    if (reflectionSupply < _reflectionTotal.div(_tokenTotal))
        return _reflectionTotal.div(_tokenTotal);
    return reflectionSupply.div(tokenSupply);
}
```

Recommendation:

Check that the excluded array length is not too big.

2. No checking if charity address is excluded

Issue:

- There is no checking if charity address is excluded from reward in _transfer function, so if it would be, token balance of charity address won't increase.

Recommendation:

Check charity address to be excluded from reward and increase _tokenBalance with proper value if needed.

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

- Owner can change the tax, burn and liquidity fee.

```
ftrace | funcSig
function setTaxFee(uint256 fee↑) external onlyOwner {
    _taxFee = fee↑;
}

ftrace | funcSig
function setBurnFee(uint256 fee↑) external onlyOwner {
    _burnFee = fee↑;
}

ftrace | funcSig
function setLiquidityFee(uint256 fee↑) external onlyOwner {
    _liquidityFee = fee↑;
}
```

- Owner can change the maximum transaction amount.

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

- Owner can change uniswapV2Pair.

```
ftrace | funcSig
function setPair(address pair↑) external onlyOwner {
    uniswapV2Pair = pair↑;
}
```

- Owner can exclude from the taxes.

```
ftrace | funcSig
function setTaxless(address account↑, bool value↑) external onlyOwner {
    isTaxless[account↑] = value↑;
}
```

- Owner can disable and enable fees.

```
ftrace | funcSig
function setFeeActive(bool value↑) external onlyOwner {
    isFeeActive = value↑;
}
```

- Owner can change dev fee.

```
function setCharityFee(uint256 amount⬆) external onlyOwner {
    charityFee= amount⬆;
}
```

- Owner can change minimum amount of tokens needed to swap.

```
ftrace | funcSig
function setMinTokensBeforeSwap(uint256 amount⬆) external onlyOwner {
    minTokensBeforeSwap = amount⬆;
}
```

Conclusion

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

Liquidity locking details 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.