Smart Contract Security Audit

Audit Details

Audited project

CARDANA

Deployer address

0x26dF87b59Da66a6482d1a45dd456c733745f8248

Client contacts:

CARDANA team

Blockchain

Binance Smart Chain

Project website:

https://cardana.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 CARDANA to perform an audit of smart contracts:

https://bscscan.com/address/0x88C28D94C260340BEBE32e3574D10380d0bf236a#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 04.08.2021

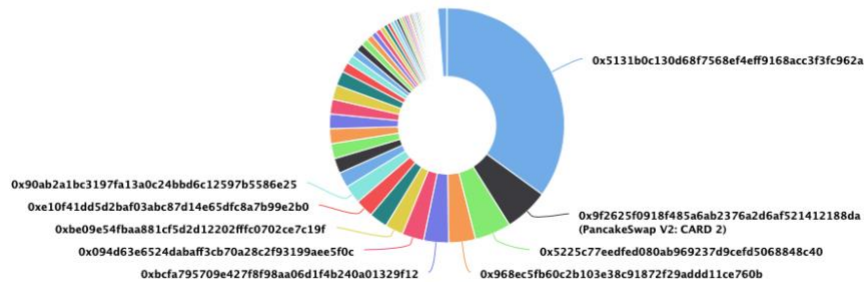|	
Contract name	CARDANA
Contract address	0x88C28D94C260340BEBE32e3574D10380d0bf236a
Total supply	1,000,000,000
Token ticker	CARD
Decimals	9
Token holders	217
Transactions count	807
Top 100 holders dominance	98.74%
Liquidity fee	5
Tax fee	5
Total fees	36583059007798279
Uniswap V2 pair	0x9f2625f0918f485a6ab2376a2d6af521412188da
Contract deployer address	0x26dF87b59Da66a6482d1a45dd456c733745f8248
Contract's current owner address	0x26df87b59da66a6482d1a45dd456c733745f8248

CARDANA Token Distribution

Cardana Top 100 Token Holders
Source: BscScan.com

0x5131b0c130d68f7568ef4eff9168acc3f3fc962a

0x90ab2a1bc3197fa13a0c24bbd6c12597b5586e25
0xe10f41dd5d2baf03abc87d14e65dfc8a7b99e2b0
0xbe09e54fbaa881cf5d2d12202fffc0702ce7c19f
0x094d63e6524dabaff3cb70a28c2f93199aee5f0c
0xbcfa795709e427f8f98aa06d1f4b240a01329f12

0x9f2625f0918f485a6ab2376a2d6af521412188da
(PancakeSwap V2: CARD 2)
0x5225c77eedfed080ab969237d9cefd5068848c40
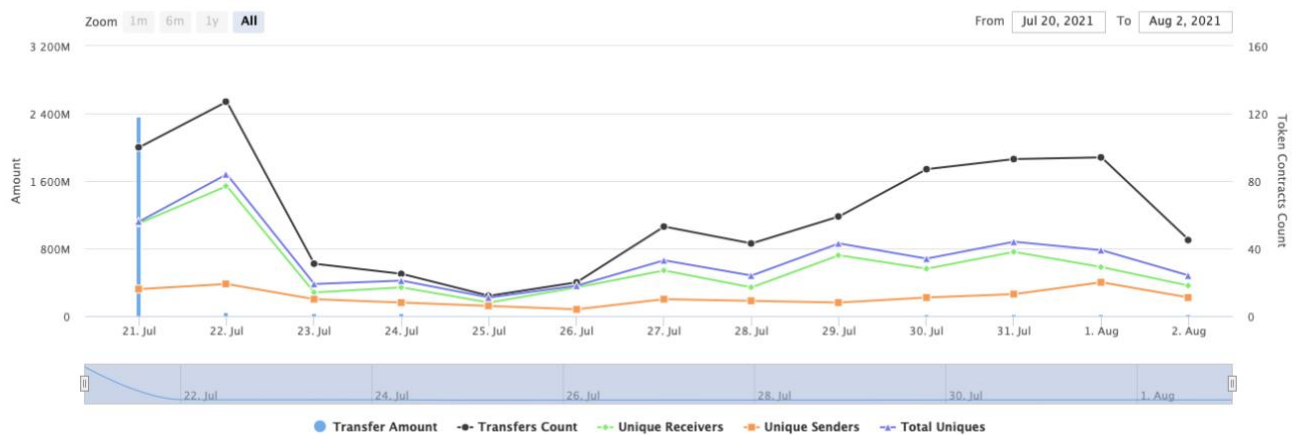0x968ec5fb60c2b103e38c91872f29addd11ce760b

(A total of 987,386,012.94 tokens held by the top 100 accounts from the total supply of 1,000,000,000.00 token)

CARDANA Contract Interaction Details

Time Series: Token Contract Overview

Wed 21, Jul 2021 - Mon 2, Aug 2021

Token Contract 0x88C28D94C260340BEBE32e3574D10380d0bf236a (Cardana)
Source: BscScan.com

CARDANA Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	0x5131b0c130d68f7568ef4eff9168acc3f3fc962a	350,000,000	35.0000%
2	📄 PancakeSwap V2: CARD 2	60,709,939.762161681	6.0710%
3	0x5225c77eedfed080ab969237d9cefd5068848c40	50,401,772.964236515	5.0402%
4	0x968ec5fb60c2b103e38c91872f29addd11ce760b	36,701,655.069385335	3.6702%
5	0xbcfa795709e427f8f98aa06d1f4b240a01329f12	34,414,753.258679628	3.4415%
6	0x094d63e6524dabaff3cb70a28c2f93199aee5f0c	30,034,625.578553491	3.0035%
7	0xbe09e54fbaa881cf5d2d12202fffc0702ce7c19f	25,035,014.827196556	2.5035%
8	0x887c66d5f03f58c94abd3871616db2d19a382788	25,000,000	2.5000%
9	0xe10f41dd5d2baf03abc87d14e65dfc8a7b99e2b0	25,000,000	2.5000%
10	0x90ab2a1bc3197fa13a0c24bbd6c12597b5586e25	25,000,000	2.5000%

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
+ [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 **#**

+ Cardana (Context, IERC20, Ownable)
- **[Pub]** <Constructor> **#**
- **[Pub]** name
- **[Pub]** symbol
- **[Pub]** decimals

- **[Pub]** totalSupply
- **[Pub]** balanceOf
- **[Pub]** transfer **#**
- **[Pub]** multiTransfer **#**
- **[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
- **[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.

- **The function multiTransfer() uses the loop for transferring amounts from array to recipients from array. It also could be aborted with OUT_OF_GAS exception if there will be a long recipients/amounts list.**

```solidity
function multiTransfer(address[] memory recipients↑, uint256[] memory amounts↑) public returns (bool) {
    require(recipients↑.length == amounts↑.length, "Different length of data");

    for (uint i = 0; i < recipients↑.length; i++) {
        _transfer(_msgSender(), recipients↑[i], amounts↑[i]);
    }

    return true;
}
```

Recommendation:
Check that the arrays length is not too big.

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

- **Owner can change the tax and liquidity fee.**

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

function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner() {
    _liquidityFee = liquidityFee;
}
```

- **Owner can exclude from the fee.**

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

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://dxsale.app/app/v2_9/dxlockview?id=0&add=0x26dF87b59Da66a6482d1a45dd456c733745f8248&type=lplock&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.