Smart Contract Security Audit

Audit details:

Audited project:	Caramel
Deployer address:	0x7290885e36355b7edf516c1d19b32141b6fe67ee
Client contacts:	Caramel team
Blockchain:	Binance Smart Chain
Project website:	Not provided

May, 2021
TechRate

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 Caramel to perform an audit of smart contracts:

- *https://bscscan.com/address/0x7ce1e651374ec5324e6f37c4ff312d53428f0d50#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

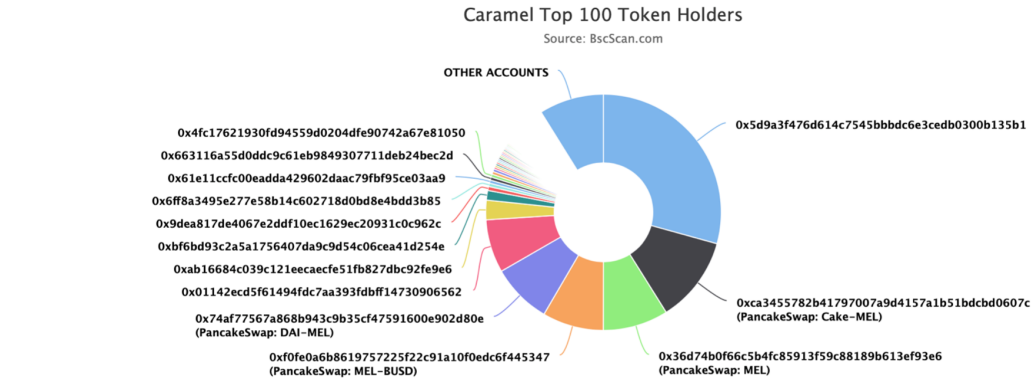
Token contract details for 05.05.2021.

Contract name:	Caramel
Contract address:	0x7ce1e651374ec5324e6f37c4ff312d53428f0d50
Total supply:	4_227_639_349_299_094_775_953_899
Token ticker:	MEL
Decimals:	18
Token holders:	2852
Transactions count:	190412
Top 100 holders dominance:	91.12 %
Compiler version:	v0.6.12+commit.27d51765
Contract deployer address:	0x7290885e36355b7edf516c1d19b32141b6fe67ee
Contract’s current owner address:	0x5d9a3f476d614c7545bbbdc6e3cedb0300b135b1

Caramel token distribution

The top 100 holders collectively own 91.12% (3,851,838.42 Tokens) of Caramel

Token Total Supply: 4,227,214.00 Token | Total Token Holders: 2,852



(A total of 3,851,838.42 tokens held by the top 100 accounts from the total supply of 4,227,214.00 token)

Caramel top 10 token holders

Rank	Address	Quantity (Token)	Percentage
1	0x5d9a3f476d614c7545bbbdc6e3cedb0300b135b1	1,238,386.482171097716595943	29.2956%
2	PancakeSwap: Cake-MEL	498,189.902502665073056215	11.7853%
3	PancakeSwap: MEL	378,411.267841378076872974	8.9518%
4	PancakeSwap: MEL-BUSD	355,657.957752897866014414	8.4135%
5	PancakeSwap: DAI-MEL	348,574.749917826369136504	8.2460%
6	0x01142ecd5f61494fdc7aa393fdbff14730906562	308,130.252773864363569306	7.2892%
7	0xab16684c039c121eecaecfe51fb827dbc92fe9e6	113,549.464574222725145208	2.6862%
8	0xbf6bd93c2a5a1756407da9c9d54c06cea41d254e	55,813.146426428171153887	1.3203%
9	0x9dea817de4067e2ddf10ec1629ec20931c0c962c	22,566.949097661131110066	0.5338%
10	0x6ff8a3495e277e58b14c602718d0bd8e4bdd3b85	19,710.959276287755614189	0.4663%

Contract functions details

+ Context
  - [Int] _msgSender
  - [Int] _msgData

+ Ownable (Context)
  - [Int] <Constructor> #
  - [Pub] owner
  - [Pub] renounceOwnership #
    - modifiers: onlyOwner
  - [Pub] transferOwnership #
    - modifiers: onlyOwner

+ [Int] IBEP20
  - [Ext] totalSupply
  - [Ext] decimals
  - [Ext] symbol
  - [Ext] name
  - [Ext] getOwner
  - [Ext] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #

+ [Lib] SafeMath
  - [Int] tryAdd
  - [Int] trySub
  - [Int] tryMul
  - [Int] tryDiv
  - [Int] tryMod
  - [Int] add
  - [Int] sub
  - [Int] mul
  - [Int] div
  - [Int] mod
  - [Int] sub
  - [Int] div
  - [Int] mod

+ BEP20 (Context, IBEP20, Ownable)
  - [Pub] <Constructor> #
  - [Ext] getOwner
  - [Pub] name
  - [Pub] symbol

  - [Pub] decimals
  - [Pub] totalSupply
  - [Pub] balanceOf
  - [Pub] transfer #
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] transferFrom #
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Pub] mint #
    - modifiers: onlyOwner
  - [Int] _transfer #
  - [Int] _mint #
  - [Int] _burn #
  - [Int] _approve #
  - [Int] _burnFrom #

+ Caramel (BEP20)
  - [Pub] <Constructor> #
  - [Pub] mint #
    - modifiers: onlyOwner
  - [Pub] burn #
  - [Pub] burnFrom #
  - [Pub] transfer #
  - [Pub] transferFrom #
  - [Ext] delegates
  - [Ext] delegate #
  - [Ext] delegateBySig #
  - [Ext] getCurrentVotes
  - [Ext] getPriorVotes
  - [Int] _delegate #
  - [Int] _moveDelegates #
  - [Int] _writeCheckpoint #
  - [Int] safe32
  - [Int] getChainId

($) = 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.	Passed
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.	High issues
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

1. Wrong transferFrom

Issue:

It is possible to call transferFrom with more tokens than allowed if the sender and recipient are not owner.

For example Alice allowed Bob to transfer 98 tokens. Bob will call the transferFrom for 100 tokens (which is more than the allowed amount!). 2 percent from 100 tokens will be burnt and then 98 tokens will be transferred and approved also 98 tokens which is allowed by Alice.

But Alice will lose 100 tokens in total instead of allowed 98.

```
function transferFrom (address sender↑, address recipient↑, uint256 amount↑) public override returns (bool) {
    if(recipient↑ != owner() && sender↑ != owner()){
        uint256 amountToBurn = amount↑.mul(2).div(100);//2% burn
        _burn(sender↑,amountToBurn);
        amount↑ = amount↑.sub(amountToBurn);
    }
    _transfer(sender↑, recipient↑, amount↑);
    _approve(sender↑,_msgSender(),_allowances[sender↑][_msgSender()].sub(amount↑, 'BEP20: transfer amount exceeds allowance'));
    return true;
}
```

Recommendation:

Call the approve before the burning and transfering.

Medium Severity Issues

1. No delegates moving in burn function.

Issue:

There are no delegates moving to the zero address in the burn function as it was moved from zero address to the recipient address in mint function.

Recommendation:

Please check the logic of the burn function.

Low Severity Issues

No low severity issues found.

Owner privileges

- Owner of the token can mint tokens without restrictions. (In this case the MasterChef contract)

Conclusion

Smart contracts contain high severity issues. Audited only the token contract.

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.