Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

August, 2021

Audit Details

Audited project

Calcifer

Deployer address

0x23a29f6700282e127de4f42e8624484870d7817f

Client contacts:

Calcifer team

Blockchain

Binance Smart Chain

Project website:

Not provided

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 Calcifer to perform an audit of smart contracts:

- https://bscscan.com/address/0x9d13cc6fabde882e059413c524a32ba5befebd8b#code
- https://bscscan.com/address/0x0efEc11A28c8cA0Dc05941da21989904181bff59#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 22.08.2021

Contract name	Calcifer
Contract address	0x9D13Cc6FABDe882E059413c524a32BA5befebD8b
Total supply	80,000
Token ticker	CALCIFER
Decimals	18
Token holders	1
Transactions count	2
Top 100 holders dominance	100.00%
Contract deployer address	0x23a29f6700282e127de4f42e8624484870d7817f
Contract's current owner address	0x0efec11a28c8ca0dc05941da21989904181bff59
Contract's current operator address	0x23a29f6700282e127de4f42e8624484870d7817f

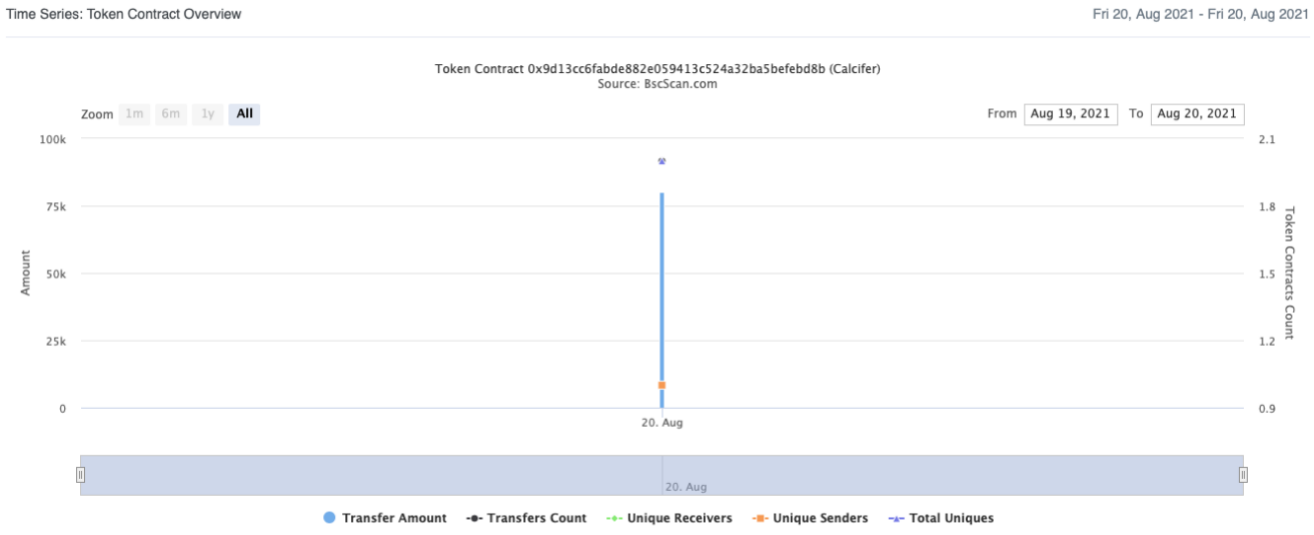
Calcifer Token Distribution

The top 100 holders collectively own 100.00% (80,000.00 Tokens) of Calcifer

Token Total Supply: 80,000.00 Token | Total Token Holders: 1

(A total of 80,000.00 tokens held by the top 100 accounts from the total supply of 80,000.00 token)

Calcifer Contract Interaction Details



Calcifer Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	0x23a29f6700282e127de4f42e8624484870d7817f	80,000	100.0000%

MasterChef functions details

+ [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

+ [Lib] SafeBEP20
  - [Int] safeTransfer #
  - [Int] safeTransferFrom #
  - [Int] safeApprove #
  - [Int] safeIncreaseAllowance #
  - [Int] safeDecreaseAllowance #
  - [Prv] _callOptionalReturn #

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

+ [Int] ICalciferReferral
  - [Ext] recordReferral #
  - [Ext] recordReferralCommission #
  - [Ext] getReferrer
+ [Int] IBEP20
  - [Ext] totalSupply
  - [Ext] decimals
  - [Ext] symbol
  - [Ext] name
  - [Ext] getOwner
  - [Ext] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #
+ Context
  - [Int] _msgSender
  - [Int] _msgData

+ 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 #

+ [Lib] Address
- [Int] isContract
- [Int] sendValue #
- [Int] functionCall #
- [Int] functionCall #
- [Int] functionCallWithValue #
- [Int] functionCallWithValue #
- [Int] functionStaticCall
- [Int] functionStaticCall
- [Int] functionDelegateCall #
- [Int] functionDelegateCall #
- [Prv] _verifyCallResult

+ Calcifer (BEP20)
- [Pub] <Constructor> #
  - modifiers: BEP20
- [Pub] mint #
  - modifiers: onlyOwner
- [Int] _transfer #
  - modifiers: antiWhale
- [Pub] maxTransferAmount
- [Pub] isExcludedFromAntiWhale
- [Pub] isExcludedFromTransferTax
- [Pub] updateTransferTaxRate #
  - modifiers: onlyOperator
- [Pub] updateBurnRate #
  - modifiers: onlyOperator
- [Pub] updateMaxTransferAmountRate #
  - modifiers: onlyOperator
- [Pub] setExcludedFromAntiWhale #
  - modifiers: onlyOperator
- [Pub] setExcludedFromTransferTax #
  - modifiers: onlyOperator
- [Pub] operator
- [Pub] transferOperator #
  - modifiers: onlyOperator

  - [Ext] delegates
  - [Ext] delegate #
  - [Ext] delegateBySig #
  - [Ext] getCurrentVotes
  - [Ext] getPriorVotes
  - [Int] _delegate #
  - [Int] _moveDelegates #
  - [Int] _writeCheckpoint #
  - [Int] safe32
  - [Int] getChainId

+ HowlsCastle (Ownable)
  - [Pub] <Constructor> #
  - [Ext] poolLength
  - [Ext] remainRewards
  - [Pub] add #
    - modifiers: onlyOwner
  - [Pub] set #
    - modifiers: onlyOwner
  - [Pub] getMultiplier
  - [Ext] pendingCALCIFER
  - [Pub] canHarvest
  - [Pub] harvestTax
  - [Pub] massUpdatePools #
  - [Pub] updatePool #
  - [Pub] deposit #
  - [Pub] withdraw #
  - [Pub] emergencyWithdraw #
  - [Int] payOrLockupPendingCALCIFER #
  - [Int] safeCALCIFERTransfer #
  - [Pub] setBoostAmounts #
    - modifiers: onlyOwner
  - [Pub] setPoolBoost #
    - modifiers: onlyOwner
  - [Pub] addUserBoostByOperator #
    - modifiers: onlyOperator
  - [Pub] setDevAddress #
  - [Pub] setFeeAddress #
  - [Ext] updateOperator #
    - modifiers: onlyOwner
  - [Pub] updateEmissionRate #
    - modifiers: onlyOwner
  - [Pub] updateEmissionHalving #
    - modifiers: onlyOwner
  - [Int] autoReduceEmissionRate #
  - [Pub] setReferralContract #
    - modifiers: onlyOwner
  - [Pub] setReferralCommissionRate #
    - modifiers: onlyOwner
  - [Int] payReferralCommission #
  - [Pub] setStartRewardBlock #
    - modifiers: onlyOwner

($) = 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.	Low 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

No high severity issues found.

Medium Severity Issues

No medium severity issues found.

Low Severity Issues

1. Block gas limit

Issue:
`add(uint256 _allocPoint, ...)`, `set(uint256 _pid, ...)` and `updateEmissionRate()` could invoke `massUpdatePools()` function, that can fail due to block gas limit if the pool size is too big.

2. `add` function issue

Issue:
If some LP token is added to the contract twice using function add, then the total amount of reward in function `updatePool` will be incorrect.

Recommendation:
Add the mapping from address to bool and check that same address will not be added twice.

3. Reentrancy issue

Issue:
Withdraw and deposit functions do not have mechanism to help prevent reentrant calls to the functions.

Recommendation:
Add reentrancy guard.

Notes:

- There is sending tokens to the dead address in overridden _transfer functions, instead of burning them in token contract.
- 1/20 of user rewards mints to dev address and the same amount goes to treasury address.
- The same amount as mints to user as reward also mints to referral.

Owner privileges

- Owner can change boost amounts.
- Owner can set pool boost.
- Owner can change contract operator.
- Dev address can change dev address.
- Fee address can change fee address.
- Owner can change calciferPerBlock.
- Owner can change emission halving settings.
- Owner can change referral address.
- Owner can change referral commission rate.
- Owner can change start block.
- Operator can change user boost amount.
- Operator can change the transfer tax rate.
- Operator can change the burn rate.
- Operator can change the max transfer amount rate.
- Operator can exclude from antiWhale and transfer tax.

Conclusion

Smart contracts contain low severity issues. Liquidity pair contract's security is not checked due to out of scope. The further transfers and operations with the funds raise are not related to this particular contract.

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.