Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

November, 2021

Audit Details

Audited project

Memeflate

Deployer address

0xfE3B9bc6E3AD8294A5196144CEf52E9167C63F02

Client contacts:

Memeflate team

Blockchain

Binance Smart Chain

Project website:

http://www.memeflate.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 Memeflate to perform an audit of smart contracts:
https://bscscan.com/address/0xafe3321309a994831884fc1725f4c3236ac79f76#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

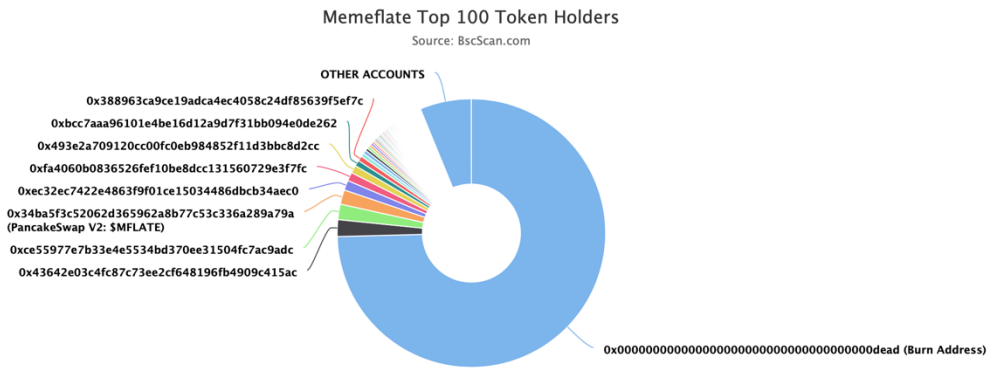
Token contract details for 07.11.2021

Contract name	Memeflate
Contract address	0xaFE3321309A994831884fc1725F4c3236AC79f76
Total supply	100,000,000,000,000,000
Token ticker	$MFLATE
Decimals	9
Token holders	6,143
Transactions count	22,832
Top 100 holders dominance	93.81%
Total fees	36,198,944,484,433,352.080087404
Contract deployer address	0xfE3B9bc6E3AD8294A5196144CEf52E9167C63F02
Contract's current owner address	0x43642e03C4fc87C73ee2CF648196Fb4909C415ac

Memeflate Token Distribution

The top 100 holders collectively own 93.81% (93,806,927,970,734,200.00 Tokens) of Memeflate

Token Total Supply: 100,000,000,000,000,000.00 Token | Total Token Holders: 6,144

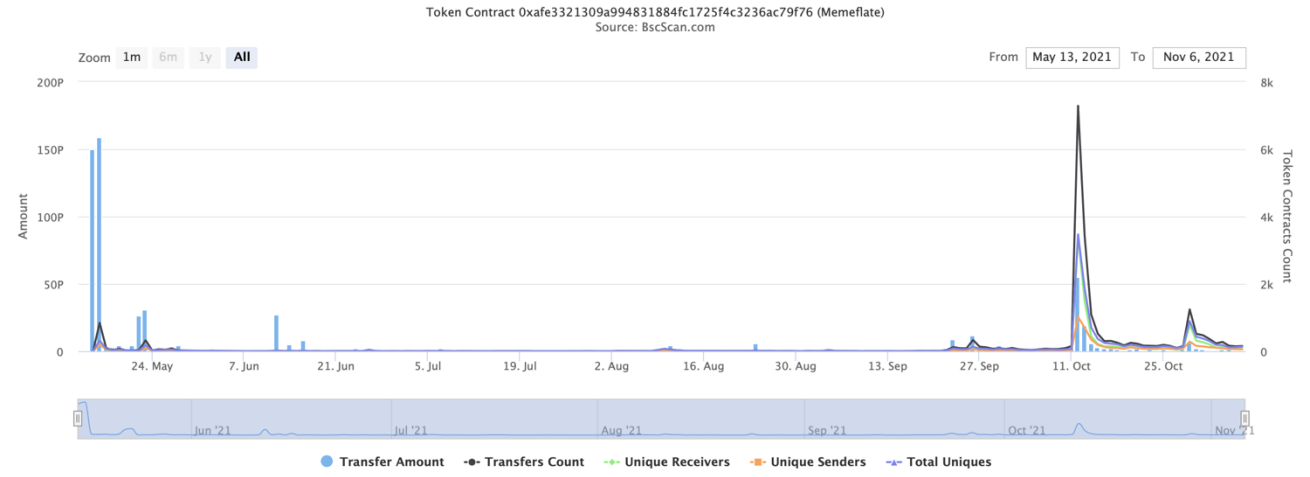


(A total of 93,806,927,970,734,200.00 tokens held by the top 100 accounts from the total supply of 100,000,000,000,000,000.00 token)

Memeflate Contract Interaction Details

Time Series: Token Contract Overview

Sat 15, May 2021 - Sat 6, Nov 2021



Memeflate Top 10 Token Holders

Rank	Address	Quantity	Percentage
1	Burn Address	74,592,508,249,027,700.962704766	74.5925%
2	0x43642e03c4fc87c73ee2cf648196fb4909c415ac	2,024,218,007,275,140.775728563	2.0242%
3	0xce55977e7b33e4e5534bd370ee31504fc7ac9adc	1,864,780,161,900,720.54823972	1.8648%
4	PancakeSwap V2: $MFLATE	1,723,783,214,563,990.9794633	1.7238%
5	0xec32ec7422e4863f9f01ce15034486dbcb34aec0	1,195,763,366,288,130.493818719	1.1958%
6	0xfa4060b0836526fef10be8dcc131560729e3f7fc	1,030,896,293,669,610.888140212	1.0309%
7	0x493e2a709120cc00fc0eb984852f11d3bbc8d2cc	939,678,608,021,171.872019676	0.9397%
8	0xbcc7aaa96101e4be16d12a9d7f31bb094e0de262	701,502,829,515,794.560534855	0.7015%
9	0x388963ca9ce19adca4ec4058c24df85639f5ef7c	620,463,122,726,469.07510196	0.6205%
10	0xe54d2a57497604a4699249f063f5879b91ecc555	450,003,067,447,255.554526286	0.4500%

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

+ [Lib] Address
  - [Int] isContract
  - [Int] sendValue #
  - [Int] functionCall #
  - [Int] functionCall #
  - [Int] functionCallWithValue #
  - [Int] functionCallWithValue #
  - [Prv] _functionCallWithValue #

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

+ MEMEFLATE (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] isExcluded
- [Pub] totalFees
- [Pub] reflect #
- [Pub] reflectionFromToken
- [Pub] tokenFromReflection
- [Pub] taxVote #
- [Pub] currentTax
- [Ext] changeTax #
  - modifiers: onlyOwner
- [Prv] burnVotes #
  - modifiers: onlyOwner
- [Pub] leadingVote
- [Pub] currentVotes
- [Ext] voteClose #
  - modifiers: onlyOwner
- [Ext] excludeAccount #
  - modifiers: onlyOwner
- [Ext] includeAccount #
  - modifiers: onlyOwner
- [Prv] _approve #
- [Prv] _transfer #
- [Prv] _transferStandard #
- [Prv] _transferToExcluded #
- [Prv] _transferFromExcluded #
- [Prv] _transferBothExcluded #
- [Prv] _reflectFee #
- [Prv] _getValues
- [Prv] _getTValues
- [Prv] _getRValues
- [Prv] _getRate
- [Prv] _getCurrentSupply

($) = 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.	Low issues
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], "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.

```
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.

2. Rounding error

Issue:

- **At each calculation with division, it is goes first. In Solidity we don't have floating points, but instead we get rounding errors.**

```
function burnVotes() private onlyOwner() {
    address dead = 0x000000000000000000000000000000000000dEaD;
    uint256 totalVotes = 0;
    for (uint256 i = 1; i < 31; i++) {
        totalVotes = totalVotes + _votes[i];
        _votes[i] = 0;
    }
    uint256 toBurn = totalVotes.div(100).mul(99);
    _transfer(_msgSender(), dead, toBurn);
}
function _getTValues(uint256 tAmount) private view returns (uint256, uint256) {
    uint256 tFee = tAmount.div(100).mul(_taxAmount);
    uint256 tTransferAmount = tAmount.sub(tFee);
    return (tTransferAmount, tFee);
}
```

Recommendation:

Do division after multiplication.

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

- Owner can change tax fee.

```
function changeTax(uint256 taxVal) external onlyOwner() {
    require(taxVal <= 30 && taxVal >= 0, "New tax value must be between 0 and 30.");
    _taxAmount = taxVal;
}
```

- Owner can close vote.

```
function voteClose() external onlyOwner() {
    uint256 newTax = leadingVote();
    burnVotes();
    _taxAmount = newTax;
}
```

- Owner can include in and exclude from reward.

```
function excludeAccount(address account) external onlyOwner() {
    require(!_isExcluded[account], "Account is already excluded");
    if(_rOwned[account] > 0) {
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}

function includeAccount(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;
        }
    }
}
```

Conclusion

Smart contracts contain low severity issues and owner privileges! 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.