Smart Contract Security Audit

Audit details:

Audited project:	Marshmallow Defi
Deployer address	0xe07573beb21aabf59862d3cc40d3ced019b172a5
Blockchain:	Binance Smart Chain
Project website:	https://marshmallowdefi.com

April, 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 Marshmallow Defi to perform an audit of smart contracts:

- *https://bscscan.com/address/0x787732f27D18495494cea3792ed7946BbCFF8db2#code*
- *https://bscscan.com/address/0x8932a6265b01d1d4e1650feb8ac38f9d79d3957b#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.

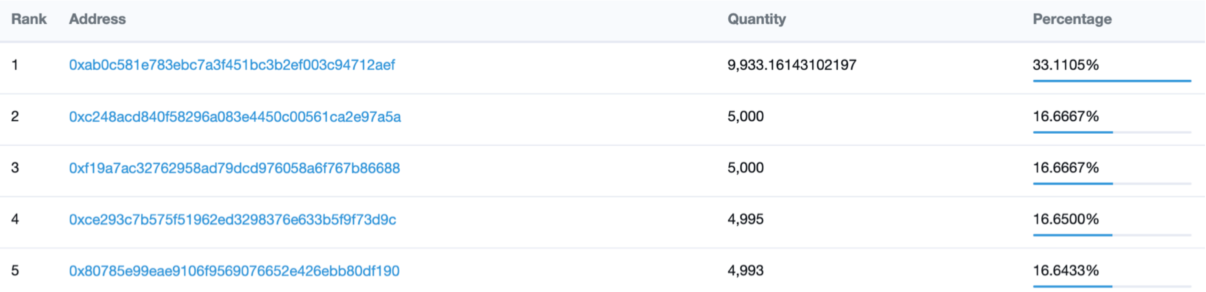
The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts details

Token contract details for 09.04.2021.

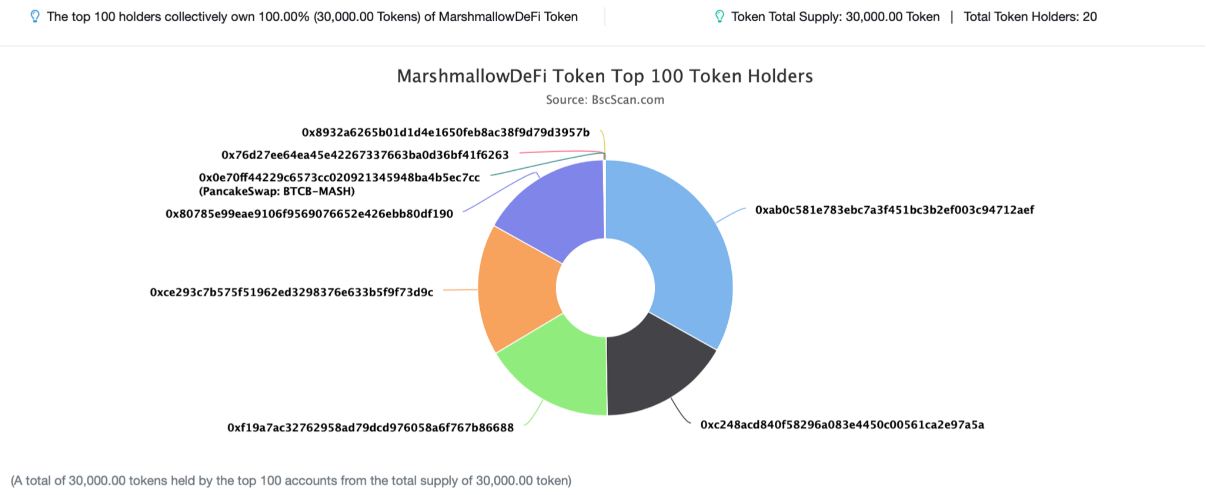
Contract name:	Marshmallow Defi
Compiler version:	v0.6.12+commit.27d51765
Contract address:	0x787732f27D18495494cea3792ed7946BbCFF8db2
Total supply:	30_000
Token ticker:	MASH
Decimals:	18
Token holders:	20
Transactions count:	88
Top 100 holders dominance:	100 %
Contract deployer address:	0xe07573beb21aabf59862d3cc40d3ced019b172a5
Contract’s current owner address:	0xe07573beb21aabf59862d3cc40d3ced019b172a5

Marshamllow Defi top 5 token holders



Rank	Address	Quantity	Percentage
1	0xab0c581e783ebc7a3f451bc3b2ef003c94712aef	9,933.16143102197	33.1105%
2	0xc248acd840f58296a083e4450c00561ca2e97a5a	5,000	16.6667%
3	0xf19a7ac32762958ad79dcd976058a6f767b86688	5,000	16.6667%
4	0xce293c7b575f51962ed3298376e633b5f9f73d9c	4,995	16.6500%
5	0x80785e99eae9106f9569076652e426ebb80df190	4,993	16.6433%

Marshamllow Defi top 100 token distribution



The top 100 holders collectively own 100.00% (30,000.00 Tokens) of MarshmallowDeFi Token

Token Total Supply: 30,000.00 Token | Total Token Holders: 20

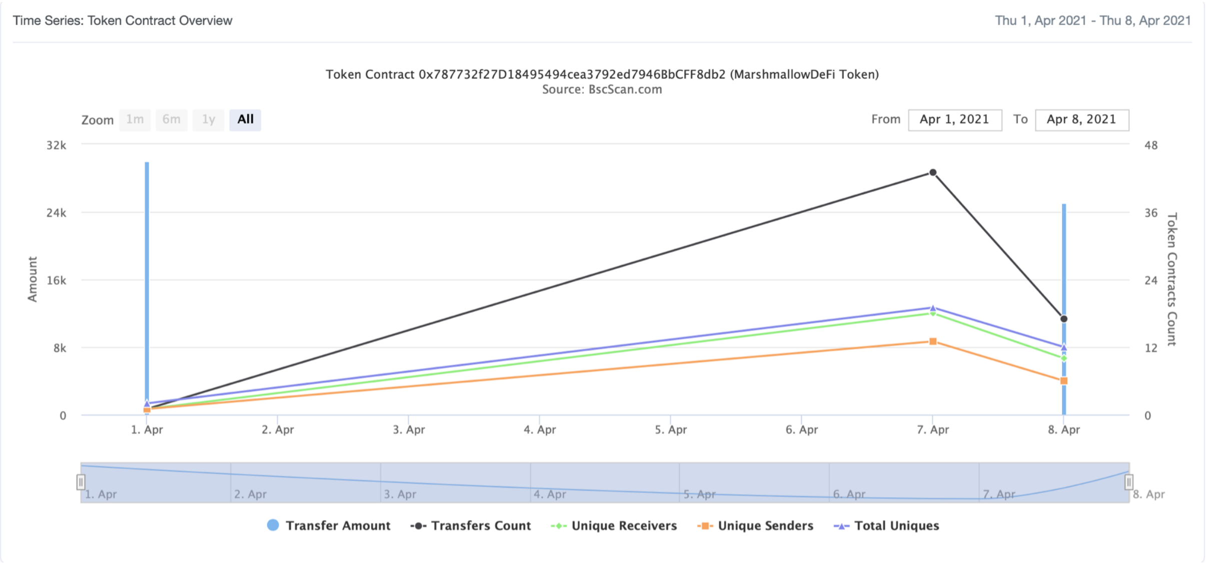
MarshmallowDeFi Token Top 100 Token Holders

Source: BscScan.com

0x8932a6265b01d1d4e1650feb8ac38f9d79d3957b
0x76d27ee64ea45e42267337663ba0d36bf41f6263
0x0e70ff44229c6573cc020921345948ba4b5ec7cc (PancakeSwap: BTCB-MASH)
0x80785e99eae9106f9569076652e426ebb80df190
0xab0c581e783ebc7a3f451bc3b2ef003c94712aef
0xce293c7b575f51962ed3298376e633b5f9f73d9c
0xf19a7ac32762958ad79dcd976058a6f767b86688
0xc248acd840f58296a083e4450c00561ca2e97a5a

(A total of 30,000.00 tokens held by the top 100 accounts from the total supply of 30,000.00 token)

Marshamllow Defi contract interaction details



Time Series: Token Contract Overview

Thu 1, Apr 2021 - Thu 8, Apr 2021

Token Contract 0x787732f27D18495494cea3792ed7946BbCFF8db2 (MarshmallowDeFi Token)

Source: BscScan.com

Zoom 1m 6m 1y **All**

From Apr 1, 2021 To Apr 8, 2021

Masterchef contract details for 09.04.2021.

Contract name:	MasterChef
Compiler version:	v0.6.12+commit.27d51765
Contract address:	0x8932a6265b01d1d4e1650feb8ac38f9d79d3957b
Dev address:	0xe07573beb21aabf59862d3cc40d3ced019b172a5
Fee address:	0xca7ae1e969f8986db45ae58150bfc9cc17332812
Mash contract address:	0x787732f27d18495494cea3792ed7946bbcff8db2
Mash per block:	1_000_000_000_000_000_000
Contract owner address:	0xe07573beb21aabf59862d3cc40d3ced019b172a5
Pool length:	16
Start block:	6445090
Total alloc point:	2650
Bonus multiplier:	1
Max deposit fee:	100 %

MasterChef contract Pools info:

Pool with id 0:

```
lpToken   address :  0x87C182EDB12f74d561519AB586205fE6CD75363a
allocPoint   uint256 :  500
lastRewardBlock   uint256 :  6445090
accEggPerShare   uint256 :  0
depositFeeBP   uint16 :  0
```

Pool with id 1:

```
lpToken   address :  0x7621886AC71e985DBea4f3f563BBB5a7865876A8
allocPoint   uint256 :  500
lastRewardBlock   uint256 :  6445090
accEggPerShare   uint256 :  0
depositFeeBP   uint16 :  0
```

Pool with id 2:

```
lpToken   address :  0x1B96B92314C44b159149f7E0303511fB2Fc4774f
allocPoint   uint256 :  100
lastRewardBlock   uint256 :  6445090
accEggPerShare   uint256 :  0
depositFeeBP   uint16 :  500 (5%)
```

Pool with id 3:

```
lpToken   address :  0x7561EEe90e24F3b348E1087A005F78B4c8453524
allocPoint   uint256 :  100
lastRewardBlock   uint256 :  6445090
accEggPerShare   uint256 :  0
depositFeeBP   uint16 :  500 (5%)
```

Pool with id 4:

```
lpToken   address :  0x787732f27D18495494cea3792ed7946BbCFF8db2
allocPoint   uint256 :  300
lastRewardBlock   uint256 :  6445090
accEggPerShare   uint256 :  0
depositFeeBP   uint16 :  0
```

Pool with id 5:

lpToken *address* : 0x0E09FaBB73Bd3Ade0a17ECC321fD13a19e81cE82
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 6445090
accEggPerShare *uint256* : 0
depositFeeBP *uint16* : 500 (5%)

Pool with id 6:

lpToken *address* : 0xC9849E6fdB743d08fAeE3E34dd2D1bc69EA11a51
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 6445090
accEggPerShare *uint256* : 0
depositFeeBP *uint16* : 500 (5%)

Pool with id 7:

lpToken *address* : 0xC9849E6fdB743d08fAeE3E34dd2D1bc69EA11a51
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 6445090
accEggPerShare *uint256* : 0
depositFeeBP *uint16* : 500 (5%)

Pool with id 8:

lpToken *address* : 0xcF6BB5389c92Bdda8a3747Ddb454cB7a64626C63
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 6445090
accEggPerShare *uint256* : 0
depositFeeBP *uint16* : 500 (5%)

Pool with id 9:

lpToken *address* : 0xd456Be0fF7007B3d8ad656136487A23e771F5762
allocPoint *uint256* : 200
lastRewardBlock *uint256* : 6445090
accEggPerShare *uint256* : 0
depositFeeBP *uint16* : 500 (5%)

Pool with id 10:

lpToken *address* : 0xbb4CdB9CBd36B01bD1cBaEBF2De08d9173bc095c
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 6445090
accEggPerShare *uint256* : 0
depositFeeBP *uint16* : 500 (5%)

Pool with id 11:

lpToken *address* : 0xe9e7CEA3DedcA5984780Bafc599bD69ADd087D56
allocPoint *uint256* : 50
lastRewardBlock *uint256* : 6445090
accEggPerShare *uint256* : 0
depositFeeBP *uint16* : 500 (5%)

Pool with id 12:

lpToken *address* : 0x7130d2A12B9BCbFAe4f2634d864A1Ee1Ce3Ead9c
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 6445090
accEggPerShare *uint256* : 0
depositFeeBP *uint16* : 500 (5%)

Pool with id 13:

lpToken *address* : 0x0E70ff44229c6573cC020921345948ba4b5Ec7CC
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 6445090
accEggPerShare *uint256* : 0
depositFeeBP *uint16* : 0

Pool with id 14:

lpToken *address* : 0x9F8223B4b616AA9becB599c93b0430c6beF0443A
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 6445090
accEggPerShare *uint256* : 0
depositFeeBP *uint16* : 0

Pool with id 15:

lpToken *address* : 0x16940Bc578c30C7c10a2cf8A150b98A8b1CEe152
allocPoint *uint256* : 100
lastRewardBlock *uint256* : 6445090
accEggPerShare *uint256* : 0
depositFeeBP *uint16* : 0

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.	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

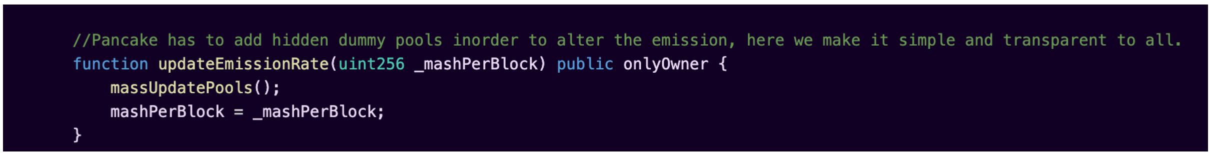
No medium severity issues found.

Low Severity Issues

1. Block gas limit

Issue:

The updateEmissionRate function can fail due to block gas limit if the pool size is too big.

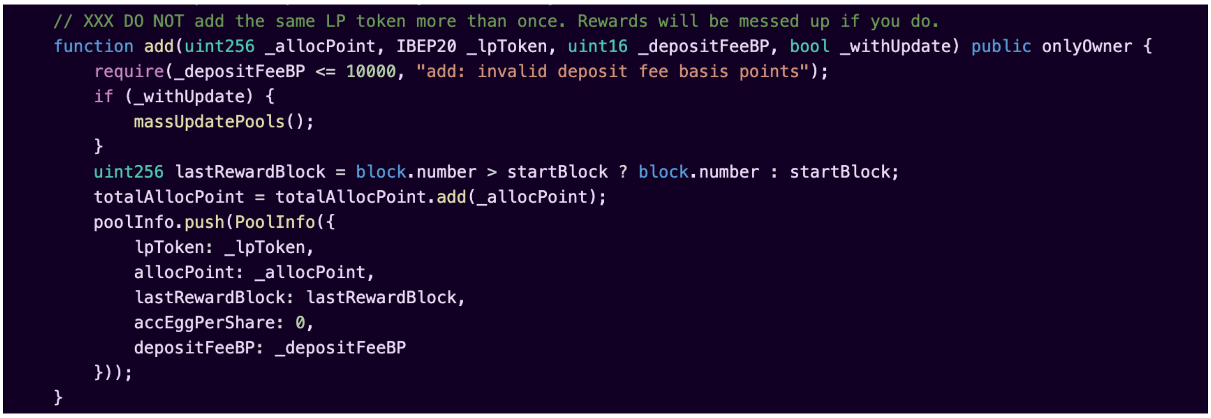


```
    //Pancake has to add hidden dummy pools inorder to alter the emission, here we make it simple and transparent to all.
    function updateEmissionRate(uint256 _mashPerBlock) public onlyOwner {
        massUpdatePools();
        mashPerBlock = _mashPerBlock;
    }
```

2. add function issue

Issue:

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



```
    // XXX DO NOT add the same LP token more than once. Rewards will be messed up if you do.
    function add(uint256 _allocPoint, IBEP20 _lpToken, uint16 _depositFeeBP, bool _withUpdate) public onlyOwner {
        require(_depositFeeBP <= 10000, "add: invalid deposit fee basis points");
        if (_withUpdate) {
            massUpdatePools();
        }
        uint256 lastRewardBlock = block.number > startBlock ? block.number : startBlock;
        totalAllocPoint = totalAllocPoint.add(_allocPoint);
        poolInfo.push(PoolInfo({
            lpToken: _lpToken,
            allocPoint: _allocPoint,
            lastRewardBlock: lastRewardBlock,
            accEggPerShare: 0,
            depositFeeBP: _depositFeeBP
        }));
    }
```

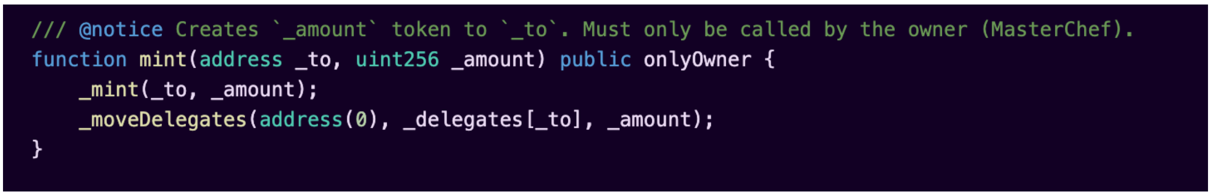
Recommendation:

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

Owner privileges

1. Owner privileges

Owner can mint any amount of tokens using function mint, until he transfers it to MasterChef contract.



```
/// @notice Creates `_amount` token to `_to`. Must only be called by the owner (MasterChef).
function mint(address _to, uint256 _amount) public onlyOwner {
    _mint(_to, _amount);
    _moveDelegates(address(0), _delegates[_to], _amount);
}
```

Conclusion

Smart contracts do not contain any high severity issues! However, there are noticed some owner privileges.

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.