TechRate

AUDIT COMPANY

Smart Contract Security Audit

Audit Details

Audited project

Malamute.Finance

Deployer address

0xaBa2117C7Ad34A5Cb95ea803e6Ba57BDf24F1280

Client contacts:

Malamute.Finance team

Blockchain

Ethereum

Project website:

https://malamute.finance/

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 Malamute.Finance to perform an audit of smart contracts:
https://etherscan.io/address/0xA3E7833775A9Cc022DB2c72EB900378F90cE89B4#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

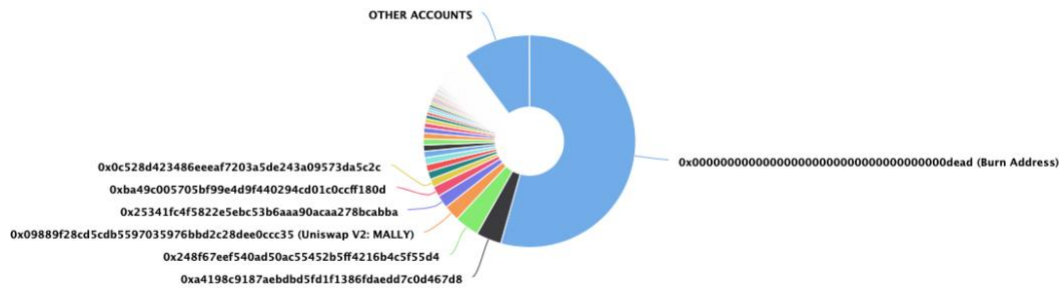## Token contract details for 03.06.2021

Contract name	Malamute.Finance
Contract address	0xA3E7833775A9Cc022DB2c72EB900378F90cE89B4
Total supply	1000000000000000000000000000
Token ticker	MALLY
Decimals	9
Token holders	2,158
Transactions count	7,008
Top 100 holders dominance	89.75%
Tax fee	2
Total fees	10400470358371349822215308
Contract deployer address	0xaBa2117C7Ad34A5Cb95ea803e6Ba57BDf24F1280
Contract's current owner address	0x0000000000000000000000000000000000000000

Malamute.Finance Token Distribution

The top 100 holders collectively own 89.75% (89,753,153,781,668,000.00 Tokens) of Malamute.Finance   Token Total Supply: 100,000,000,000,000,000.00 Token | Total Token Holders: 2,158

Malamute.Finance Top 100 Token Holders
Source: Etherscan.io

OTHER ACCOUNTS

0x0c528d423486eeeaf7203a5de243a09573da5c2c
0xba49c005705bf99e4d9f440294cd01c0ccff180d
0x25341fc4f5822e5ebc53b6aaa90acaa278bcabba
0x09889f28cd5cdb5597035976bbd2c28dee0ccc35 (Uniswap V2: MALLY)
0x248f67eef540ad50ac55452b5ff4216b4c5f55d4
0xa4198c9187aebdbd5fd1f1386fdaedd7c0d467d8

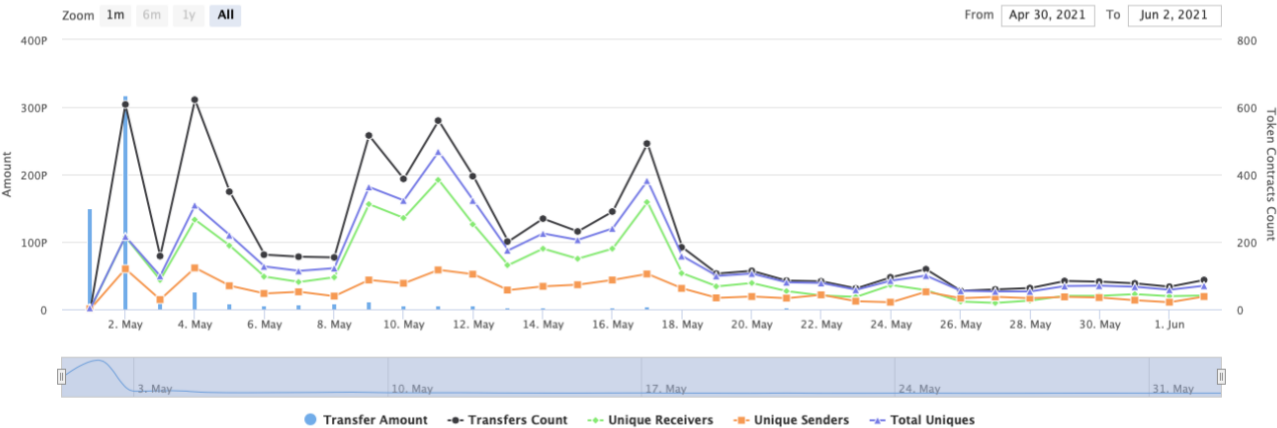0x0000000000000000000000000000000000000dead (Burn Address)

(A total of 89,753,153,781,668,000.00 tokens held by the top 100 accounts from the total supply of 100,000,000,000,000,000.00 token)

Malamute.Finance Contract Interaction Details

Time Series: Token Contract Overview   Sat 1, May 2021 - Wed 2, Jun 2021

Token Contract 0xA3E7833775A9Cc022DB2c72EB900378F90cE89B4 (Malamute.Finance)
Source: Etherscan.io

Zoom 1m 6m 1y All   From Apr 30, 2021 To Jun 2, 2021

Transfer Amount   Transfers Count   Unique Receivers   Unique Senders   Total Uniques

Malamute.Finance Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	Burn Address	54,374,985,142,692,500.393827329	54.3750%
2	0xa4198c9187aebdbd5fd1f1386fdaedd7c0d467d8	3,846,565,434,014,690.612669004	3.8466%
3	0x248f67eef540ad50ac55452b5ff4216b4c5f55d4	3,736,630,924,764,000.280313922	3.7366%
4	Uniswap V2: MALLY	2,412,698,749,005,860.511186312	2.4127%
5	0x25341fc4f5822e5ebc53b6aaa90acaa278bcabba	1,991,465,881,635,040.419249586	1.9915%
6	0xba49c005705bf99e4d9f440294cd01c0ccff180d	1,585,655,137,010,750.988911683	1.5857%
7	0x0c528d423486eeeaf7203a5de243a09573da5c2c	1,217,138,664,183,350.970544711	1.2171%
8	0x33c765f6c31fb7f9e5efef6eb11644c119ea15f2	1,178,790,529,266,990.513042773	1.1788%
9	0xc36119b212b7dfd5ec3bd46c35a4cf3f3df17f50	1,104,859,873,849,920.828008853	1.1049%
10	0xfdd9e429a159fef749e1224483295c826e23eee5	1,061,507,504,633,790.671164732	1.0615%

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)
- [Int] <Constructor> **#**
- **[Pub]** owner
- **[Pub]** renounceOwnership **#**
  - modifiers: onlyOwner
- **[Pub]** transferOwnership **#**
  - modifiers: onlyOwner

+ MalamuteFinance (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
- **[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.	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 includeAccount(address account↑) external onlyOwner() {
    require(_isExcluded[account↑], "Account is not 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.

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://etherscan.io/tx/0xbe596b406928c4c0e45b3a6b2839b1b909e9123e94b76e8e40eb11de3bf7a15e

Ownership renounce details provided by the team:
https://etherscan.io/tx/0x2b4078b0dd96aa64b61c4cd639342f4366ac9ed363ad93aa903a4c79c05bb817

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.