TechRate

Blockchain solutions and consulting

Smart Contract Security Audit

Audit details:

Audited project:	Steak
Deployer address	0x6623774C883Af9f12e223503594Bc3B672A2DeCB
Blockchain:	Matic
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 Steak to perform an audit of smart contracts:

- *https://explorer-mainnet.maticvigil.com/address/0x52570E25abe5BfE21151CF9A3a5d6a567B79bD5A/contracts*
- *https://explorer-mainnet.maticvigil.com/address/0x2a6615238a161fABA3dC9a508f08c4320BE1a193/contracts*

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.

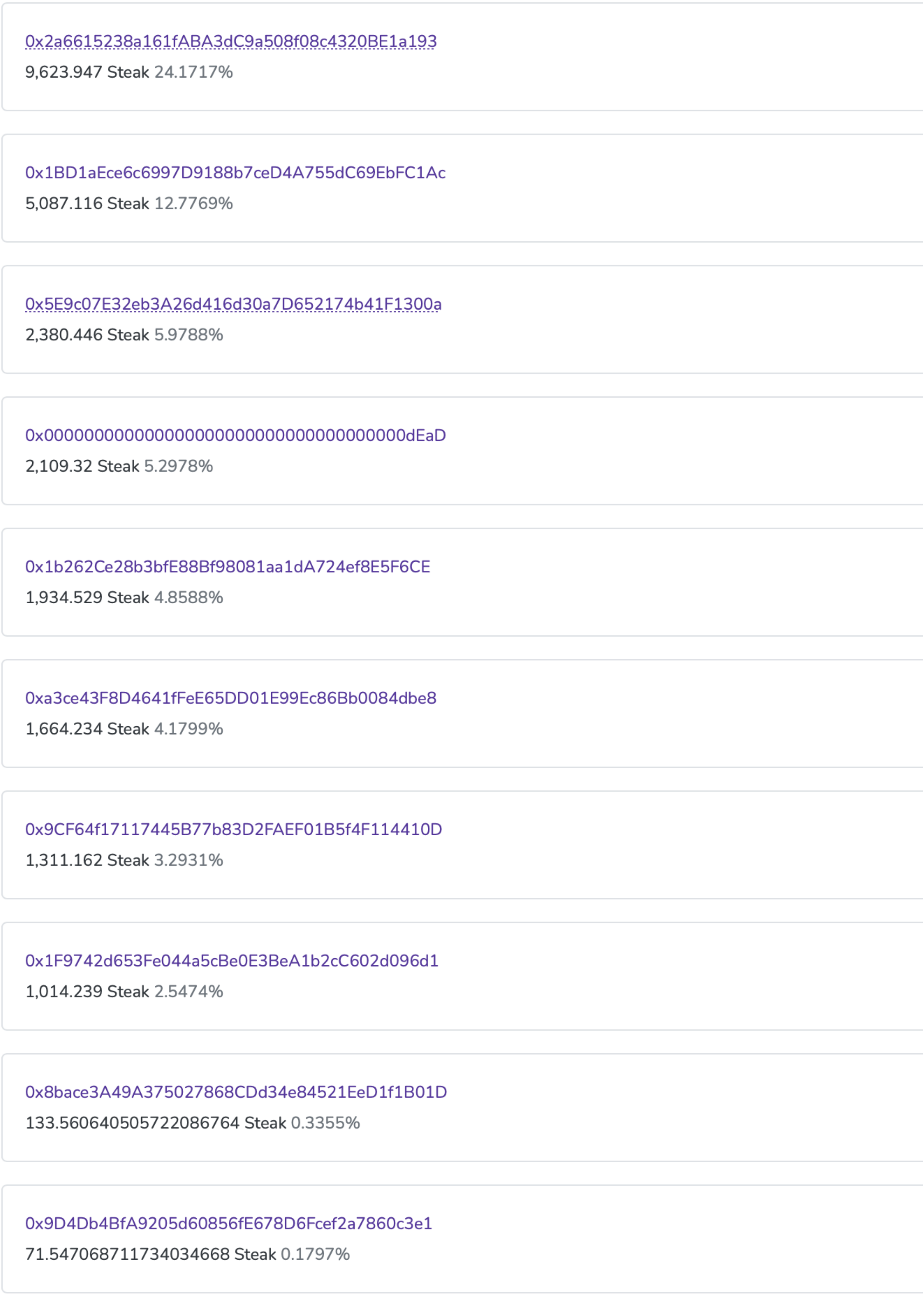
The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts details

Token contract details for 17.05.2021.

Contract name:	Steak
Compiler version:	v0.6.12+commit.27d51765
Contract address:	0x52570E25abe5BfE21151CF9A3a5d6a567B79bD5A
Total supply:	39811021684142222735934
Token ticker:	Steak
Decimals:	18
Token holders:	287
Transactions count:	910
Contract deployer address:	0x6623774C883Af9f12e223503594Bc3B672A2DeCB
Contract's current owner address:	0x2a6615238a161fABA3dC9a508f08c4320BE1a193

Steak top 10 token holders



0x2a6615238a161fABA3dC9a508f08c4320BE1a193
9,623.947 Steak 24.1717%

0x1BD1aEce6c6997D9188b7ceD4A755dC69EbFC1Ac
5,087.116 Steak 12.7769%

0x5E9c07E32eb3A26d416d30a7D652174b41F1300a
2,380.446 Steak 5.9788%

0x000000000000000000000000000000000000dEaD
2,109.32 Steak 5.2978%

0x1b262Ce28b3bfE88Bf98081aa1dA724ef8E5F6CE
1,934.529 Steak 4.8588%

0xa3ce43F8D4641fFeE65DD01E99Ec86Bb0084dbe8
1,664.234 Steak 4.1799%

0x9CF64f17117445B77b83D2FAEF01B5f4F114410D
1,311.162 Steak 3.2931%

0x1F9742d653Fe044a5cBe0E3BeA1b2cC602d096d1
1,014.239 Steak 2.5474%

0x8bace3A49A375027868CDd34e84521EeD1f1B01D
133.560640505722086764 Steak 0.3355%

0x9D4Db4BfA9205d60856fE678D6Fcef2a7860c3e1
71.547068711734034668 Steak 0.1797%

MasterChef contract details for 17.05.2021.

Contract name:	MasterChef
Compiler version:	v0.6.12+commit.27d51765
Contract address:	0x2a6615238a161fABA3dC9a508f08c4320BE1a193
Deployer address:	0x6623774C883Af9f12e223503594Bc3B672A2DeCB
Fee address:	0x1F9742d653Fe044a5cBe0E3BeA1b2cC602d096d1
Dev address:	0x1F9742d653Fe044a5cBe0E3BeA1b2cC602d096d1
Steak contract address:	
Steak per block:	500000000000000000
Contract owner address:	0xA5851adF85371Ed09B67b7478FC12dE581bE96d4
Pool length:	13
Start block:	14502500
Total alloc point:	19600
Bonus multiplier:	1

MasterChef functions outline

+ [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

+ [Int] IERC20
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #

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

+ [Lib] SafeERC20
  - [Int] safeTransfer #
  - [Int] safeTransferFrom #
  - [Int] safeApprove #
  - [Int] safeIncreaseAllowance #
  - [Int] safeDecreaseAllowance #
  - [Prv] _callOptionalReturn #

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

+  ERC20 (Context, IERC20)
   - [Pub] <Constructor> #
   - [Pub] name
   - [Pub] symbol
   - [Pub] decimals
   - [Pub] totalSupply
   - [Pub] balanceOf
   - [Pub] transfer #
   - [Pub] allowance
   - [Pub] approve #
   - [Pub] transferFrom #
   - [Pub] increaseAllowance #
   - [Pub] decreaseAllowance #
   - [Int] _transfer #
   - [Int] _mint #
   - [Int] _burn #
   - [Int] _approve #
   - [Int] _setupDecimals #
   - [Int] _beforeTokenTransfer #

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

+  ReentrancyGuard
   - [Int] <Constructor> #

+  Token (ERC20, Ownable)
   - [Pub] <Constructor> #
      - modifiers: ERC20
   - [Ext] setMinterStatus #
      - modifiers: onlyOwner
   - [Pub] mint #
      - modifiers: onlyMinters
   - [Ext] delegates
   - [Ext] delegate #
   - [Ext] delegateBySig #
   - [Ext] getCurrentVotes
   - [Ext] getPriorVotes
```

```
  - [Int] _delegate #
  - [Int] _moveDelegates #
  - [Int] _writeCheckpoint #
  - [Int] safe32
  - [Int] getChainId

+  TokenMinter (ERC20, Ownable)
  - [Pub] mint #
    - modifiers: onlyOwner
  - [Pub] burn #
    - modifiers: onlyOwner
  - [Pub] <Constructor> #
    - modifiers: ERC20
  - [Ext] setTaxAddr #
    - modifiers: onlyOwner
  - [Ext] setTax #
    - modifiers: onlyOwner
  - [Ext] setWhiteList #
    - modifiers: onlyOwner
  - [Ext] safeTokenTransfer #
    - modifiers: onlyOwner
  - [Prv] getTax
  - [Ext] delegates
  - [Ext] delegate #
  - [Ext] delegateBySig #
  - [Ext] getCurrentVotes
  - [Ext] getPriorVotes
  - [Int] _delegate #
  - [Int] _moveDelegates #
  - [Int] _writeCheckpoint #
  - [Int] safe32
  - [Int] getChainId

+  Farm (Ownable, ReentrancyGuard)
  - [Pub] <Constructor> #
  - [Ext] poolLength
  - [Pub] add #
    - modifiers: onlyOwner,nonDuplicated
  - [Pub] set #
    - modifiers: onlyOwner
  - [Int] updateStakingPool #
  - [Pub] getMultiplier
  - [Ext] pendingToken
  - [Pub] massUpdatePools #
  - [Pub] updatePool #
  - [Pub] deposit #
    - modifiers: nonReentrant
```

- [Pub] withdraw #
  - modifiers: nonReentrant
- [Pub] enterStaking #
  - modifiers: nonReentrant
- [Pub] leaveStaking #
  - modifiers: nonReentrant
- [Pub] emergencyWithdraw #
  - modifiers: nonReentrant
- [Int] safeTokenTransfer #
- [Ext] updateBonus #
  - modifiers: onlyOwner
- [Ext] updateTokenPerBlock #
  - modifiers: onlyOwner
- [Ext] setDevFeeAddr #
  - modifiers: onlyOwner
- [Ext] setTaxAddr #
  - modifiers: onlyOwner
- [Ext] setTax #
  - modifiers: onlyOwner
- [Ext] setWhiteList #
  - modifiers: onlyOwner
- [Ext] setMinterStatus #
  - modifiers: onlyOwner
- [Ext] setStartBlock #
  - modifiers: onlyOwner
- [Ext] adminMint #
  - modifiers: onlyOwner
- [Ext] setLock #
  - 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.	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 the 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 the same address will not be added twice.

Owner privileges

- ❏ Owner can change the tax address and tax fee.
- ❏ Owner can add / remove any address from the whitelist.
- ❏ Owner can BURN any amount of tokens from any address in the TokenMinter contract.
- ❏ Owner can MINT tokens through the Farm contract.
- ❏ Owner can change the minter status through the Farm contract.
- ❏ Owner can change the tax through the Farm contract.
- ❏ Owner can add to the whitelist through the Farm contract.

Conclusion

Smart contracts contain low severity issues and 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.