TechRate

Blockchain solutions and consulting

Smart Contract Security Audit

Audit details:

Audited project:	Guess It
Deployer address	0x1EB34D735A7A21394cd3CF03bDb1497bFc9d75D0
Blockchain:	Binance Smart Chain
Project website:	https://docs.guessit.app

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 Guess It to perform an audit of smart contracts:

- *https://bscscan.com/address/0xd20595346ca72f82be355e2a603bfa8b0fd86e33#code*
- *https://bscscan.com/address/0x4Da11AF8176b113989EB2003fB0b10A8d225B5AE#code*
- *https://bscscan.com/address/0x7a7b6c0ebb3e8e03a62674f222445ace2172cfe4#code*
- *https://bscscan.com/address/0xf602f45b3b072217f2859e05b9a71d8e21e1f8fd#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 26.05.2021.

Contract name:	GuessItToken
Compiler version:	v0.8.4+commit.c7e474f2
Contract address:	0xd20595346ca72f82be355e2a603bfa8b0fd86e33
Total supply:	1,000,000,000
Token ticker:	GSSIT
Decimals:	18
Token holders:	2
Transactions count:	2
Rewards address:	0x7a7b6c0ebb3e8e03a62674f222445ace2172cfe4
Pancake router:	0x10ed43c718714eb63d5aa57b78b54704e256024e
Transfer percentage:	500
Contract deployer address:	0x4da11af8176b113989eb2003fb0b10a8d225b5ae
Contract’s current owner address:	0x4da11af8176b113989eb2003fb0b10a8d225b5ae

Farm contract details for 26.05.2021.

Contract name:	GuessItFarm
Compiler version:	v0.8.4+commit.c7e474f2
Contract address:	0xf602f45b3b072217f2859e05b9a71d8e21e1f8fd
Deployer address:	0x4da11af8176b113989eb2003fb0b10a8d225b5ae
Rewards address:	0x7a7b6c0ebb3e8e03a62674f222445ace2172cfe4
Native contract address:	0xd20595346ca72f82be355e2a603bfa8b0fd86e33
Native per block:	300000000000000000000
Contract owner address:	0x4da11af8176b113989eb2003fb0b10a8d225b5ae
Pool length:	0
Start block:	7739432
Total alloc point:	0
Bonus multiplier:	1

Token functions outline

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

+ [Int] IERC165
  - [Ext] supportsInterface

+  ERC165 (IERC165)
  - [Pub] supportsInterface

+ [Int] IAccessControl
  - [Ext] hasRole
  - [Ext] getRoleAdmin
  - [Ext] grantRole #
  - [Ext] revokeRole #
  - [Ext] renounceRole #

+  AccessControl (Context, IAccessControl, ERC165)
  - [Pub] supportsInterface
  - [Pub] hasRole
  - [Pub] getRoleAdmin
  - [Pub] grantRole #
  - [Pub] revokeRole #
  - [Pub] renounceRole #
  - [Int] _setupRole #
  - [Int] _setRoleAdmin #
  - [Prv] _grantRole #
  - [Prv] _revokeRole #

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

+  ReentrancyGuard
  - [Pub] <Constructor> #

+ [Int] IERC20
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] transfer #
```

```
   - [Ext] allowance
   - [Ext] approve #
   - [Ext] transferFrom #

 +  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] _beforeTokenTransfer #

 +  ERC20Burnable (Context, ERC20)
   - [Pub] burn #
   - [Pub] burnFrom #

 +  ERC20Capped (ERC20)
   - [Pub] <Constructor> #
   - [Pub] cap
   - [Int] _mint #

 + [Int] IPancakeRouter01
   - [Ext] factory
   - [Ext] WETH
   - [Ext] addLiquidity #
   - [Ext] addLiquidityETH ($)
   - [Ext] removeLiquidity #
   - [Ext] removeLiquidityETH #
   - [Ext] removeLiquidityWithPermit #
   - [Ext] removeLiquidityETHWithPermit #
   - [Ext] swapExactTokensForTokens #
   - [Ext] swapTokensForExactTokens #
   - [Ext] swapExactETHForTokens ($)
   - [Ext] swapTokensForExactETH #
   - [Ext] swapExactTokensForETH #
   - [Ext] swapETHForExactTokens ($)
   - [Ext] quote
   - [Ext] getAmountOut
   - [Ext] getAmountIn
   - [Ext] getAmountsOut
   - [Ext] getAmountsIn

 + [Int] IPancakeRouter02 (IPancakeRouter01)
   - [Ext] removeLiquidityETHSupportingFeeOnTransferTokens #
   - [Ext] removeLiquidityETHWithPermitSupportingFeeOnTransferTokens #
   - [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
   - [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens ($)
   - [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

 +  GuessItRewards (Ownable, AccessControl)
   - [Pub] <Constructor> #
   - [Ext] <Fallback> ($)
   - [Pub] getTransferRole
   - [Ext] transferRewards #

 +  GuessItToken (ERC20Burnable, ERC20Capped, AccessControl, Ownable,
ReentrancyGuard)
   - [Pub] <Constructor> #
     - modifiers: ERC20Capped,ERC20
   - [Pub] getMinterRole
   - [Pub] excludeFromFee #
     - modifiers: onlyOwner
   - [Ext] includeInFee #
     - modifiers: onlyOwner
   - [Pub] isExcludedFromFee
   - [Ext] newGame #
     - modifiers: inGameState
   - [Ext] getGame
     - modifiers: notInGameState
   - [Ext] setPrice #
     - modifiers: onlyOwner,inGameState
   - [Ext] getPrice
     - modifiers: notInGameState
   - [Ext] guess #
     - modifiers: inGameState
   - [Pub] claimableRewards
     - modifiers: inGameState
   - [Ext] withdraw #
     - modifiers: inGameState,nonReentrant
   - [Ext] setTransferPercentage #
     - modifiers: onlyOwner
   - [Ext] setRewardsPercentage #
     - modifiers: onlyOwner
```

- [Ext] setGuesserPercentage #
    - modifiers: onlyOwner
- [Pub] mint #
- [Pub] totalMinted
- [Int] _transfer #
- [Prv] _swap #
    - modifiers: lockSwap
- [Int] _mint #
- [Prv] _toLower

($) = payable function
# = non-constant function

Farm functions outline

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

+ [Int] IERC165
  - [Ext] supportsInterface

+ ERC165 (IERC165)
  - [Pub] supportsInterface

+ [Int] IAccessControl
  - [Ext] hasRole
  - [Ext] getRoleAdmin
  - [Ext] grantRole #
  - [Ext] revokeRole #
  - [Ext] renounceRole #

+ AccessControl (Context, IAccessControl, ERC165)
  - [Pub] supportsInterface
  - [Pub] hasRole
  - [Pub] getRoleAdmin
  - [Pub] grantRole #
  - [Pub] revokeRole #
  - [Pub] renounceRole #
  - [Int] _setupRole #
  - [Int] _setRoleAdmin #
  - [Prv] _grantRole #
  - [Prv] _revokeRole #

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

+ ReentrancyGuard
  - [Pub] <Constructor> #

+ [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 #

+  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] _beforeTokenTransfer #

+  ERC20Burnable (Context, ERC20)
  - [Pub] burn #
  - [Pub] burnFrom #
```

```
 +  ERC20Capped (ERC20)
    - [Pub] <Constructor> #
    - [Pub] cap
    - [Int] _mint #

 + [Int] IPancakeRouter01
    - [Ext] factory
    - [Ext] WETH
    - [Ext] addLiquidity #
    - [Ext] addLiquidityETH ($)
    - [Ext] removeLiquidity #
    - [Ext] removeLiquidityETH #
    - [Ext] removeLiquidityWithPermit #
    - [Ext] removeLiquidityETHWithPermit #
    - [Ext] swapExactTokensForTokens #
    - [Ext] swapTokensForExactTokens #
    - [Ext] swapExactETHForTokens ($)
    - [Ext] swapTokensForExactETH #
    - [Ext] swapExactTokensForETH #
    - [Ext] swapETHForExactTokens ($)
    - [Ext] quote
    - [Ext] getAmountOut
    - [Ext] getAmountIn
    - [Ext] getAmountsOut
    - [Ext] getAmountsIn

 + [Int] IPancakeRouter02 (IPancakeRouter01)
    - [Ext] removeLiquidityETHSupportingFeeOnTransferTokens #
    - [Ext] removeLiquidityETHWithPermitSupportingFeeOnTransferTokens #
    - [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
    - [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens ($)
    - [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

 +  GuessItRewards (Ownable, AccessControl)
    - [Pub] <Constructor> #
    - [Ext] <Fallback> ($)
    - [Pub] getTransferRole
    - [Ext] transferRewards #

 +  GuessItToken (ERC20Burnable, ERC20Capped, AccessControl, Ownable,
ReentrancyGuard)
    - [Pub] <Constructor> #
       - modifiers: ERC20Capped,ERC20
    - [Pub] getMinterRole
    - [Pub] excludeFromFee #
       - modifiers: onlyOwner
```

  - [Ext] includeInFee #
    - modifiers: onlyOwner
  - [Pub] isExcludedFromFee
  - [Ext] newGame #
    - modifiers: inGameState
  - [Ext] getGame
    - modifiers: notInGameState
  - [Ext] setPrice #
    - modifiers: onlyOwner,inGameState
  - [Ext] getPrice
    - modifiers: notInGameState
  - [Ext] guess #
    - modifiers: inGameState
  - [Pub] claimableRewards
    - modifiers: inGameState
  - [Ext] withdraw #
    - modifiers: inGameState,nonReentrant
  - [Ext] setTransferPercentage #
    - modifiers: onlyOwner
  - [Ext] setRewardsPercentage #
    - modifiers: onlyOwner
  - [Ext] setGuesserPercentage #
    - modifiers: onlyOwner
  - [Pub] mint #
  - [Pub] totalMinted
  - [Int] _transfer #
  - [Prv] _swap #
    - modifiers: lockSwap
  - [Int] _mint #
  - [Prv] _toLower

+ [Int] IPancakePair
  - [Ext] name
  - [Ext] symbol
  - [Ext] decimals
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transfer #
  - [Ext] transferFrom #
  - [Ext] DOMAIN_SEPARATOR
  - [Ext] PERMIT_TYPEHASH
  - [Ext] nonces
  - [Ext] permit #
  - [Ext] MINIMUM_LIQUIDITY
  - [Ext] factory

```
  - [Ext] token0
  - [Ext] token1
  - [Ext] getReserves
  - [Ext] price0CumulativeLast
  - [Ext] price1CumulativeLast
  - [Ext] kLast
  - [Ext] mint #
  - [Ext] burn #
  - [Ext] swap #
  - [Ext] skim #
  - [Ext] sync #
  - [Ext] initialize #

+ [Int] IWETH
  - [Ext] transfer #

+  GuessItFarm (Ownable, AccessControl, ReentrancyGuard)
  - [Pub] <Constructor> #
  - [Ext] getPoolInfo
  - [Ext] add #
    - modifiers: onlyOwner
  - [Ext] set #
    - modifiers: onlyOwner
  - [Pub] getMultiplier
  - [Ext] getPendingNative
  - [Pub] massUpdatePools #
  - [Pub] updatePool #
  - [Ext] deposit #
    - modifiers: nonReentrant
  - [Prv] max
  - [Prv] min
  - [Ext] withdraw #
    - modifiers: nonReentrant
  - [Ext] emergencyWithdraw #
    - modifiers: nonReentrant
  - [Int] safeNativeTransfer #
  - [Prv] _distributeDepositFee #
  - [Prv] _getNativeFinished
  - [Prv] _getRewardBlock
  - [Prv] _removeLiquidityAndSwap #
  - [Prv] _swap #


($) = 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. add function issue

Issue:

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

Recommendation:

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

Owner privileges

- ❏ Owner of the GuessItToken contract can change the price.
- ❏ Owner of the GuessItToken contract can change the transfer percentage in the range 950 - 990.
- ❏ Owner of the GuessItToken contract can change the rewards percentage in the range 300 - 1000.
- ❏ Owner of the GuessItToken contract can change the guesser percentage in the range 10-100.
- ❏ Accounts with the minter role can mint tokens up to cap.
- ❏ Owner can add the pools with big lockup periods so the users will not be able to withdraw their funds, only emergency withdraw will work.

Conclusion

Smart contracts do not contain high severity issues!

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.