Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

June, 2021

Audit Details

Audited project

PolyGold

Deployer address

0xb642B476f4C5516a887bf073b32880C3257C1334

Client contacts:

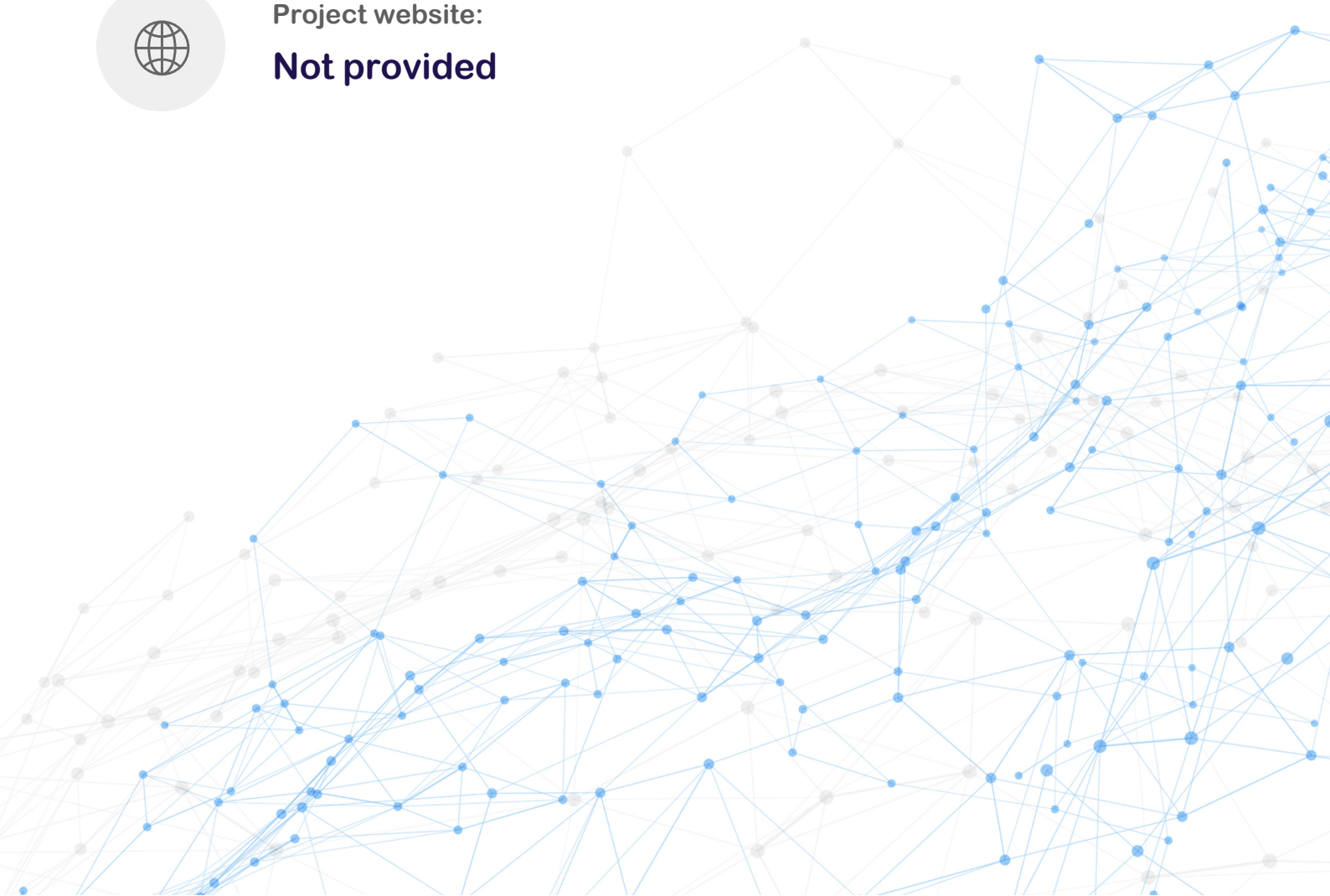
PolyGold team

Blockchain

Matic

Project website:

Not provided



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 PolyGold to perform an audit of smart contracts:

- https://explorer-mainnet.maticvigil.com/address/0x0184316f58B9A44aCDD3e683257259dC0CF2202a/transactions
- https://explorer-mainnet.maticvigil.com/address/0xe0e400617A20ADee7B2034324C3fa4C37bce97E8/transactions
- https://explorer-mainnet.maticvigil.com/address/0x6496841DCA098Bc829eb012423aC40f539394bf6/transactions

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 04.06.2021

Contract name	PolyGold
Contract address	0x0184316f58B9A44aCDD3e683257259dC0CF2202a
Total supply	12224064499999999934210
Token ticker	POLYGOLD
Decimals	18
Token holders	900
Transactions count	252996
Contract deployer address	0xb642B476f4C5516a887bf073b32880C3257C1334
Contract's current owner address	0xe0e400617a20adee7b2034324c3fa4c37bce97e8

PolyGold Top 10 Token Holders

0xe0e400617A20ADee7B2034324C3fa4C37bce97E8
4,322.749 POLYGOLD 35.3598%

0x9Caf7BbeB28E4e69DbB255017D231d98d0e1CA9C
3,912.035 POLYGOLD 32.0002%

0xBfbb75Bc384F70A2cF3567b9e2d89cc475a213Cf
3,402.897 POLYGOLD 27.8355%

0x0000000000000000000000000000000000000dEaD
206.096889957958664869 POLYGOLD 1.6859%

0xAAA7ea392E80da7182b55582D9470c1B3e9bfDB5
169.66263078897112151 POLYGOLD 1.3878%

0x64DD8df35518887ae74e0E73CF289A3c50646181
31.616371033342698694 POLYGOLD 0.2586%

0x383855Aac8c3B6d85c97881990da115899D069E3
29.275164468023987171 POLYGOLD 0.2395%

0xeCC612Da4452347943BDB02a20eAc558E9A79C1b
21.792072678728986802 POLYGOLD 0.1783%

0xC40ba5621434659D042f8c9ef1f0F8D5e273e701
18.842032840027401954 POLYGOLD 0.1541%

0xD7f665Fa15a5Df1e3846B1377911Cfe052857B2F
17.1839016189635972 POLYGOLD 0.1406%

MasterChef functions details

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

+ [Int] IBEP20
  - [Ext] totalSupply
  - [Ext] decimals
  - [Ext] symbol
  - [Ext] name
  - [Ext] getOwner
  - [Ext] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #

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

+ BEP20 (Context, IBEP20, Ownable)
  - [Pub] <Constructor> #
  - [Ext] getOwner
  - [Pub] name
  - [Pub] symbol
  - [Pub] decimals
  - [Pub] totalSupply
  - [Pub] balanceOf
  - [Pub] transfer #
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] transferFrom #
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Pub] mint #
    - modifiers: onlyOwner
  - [Int] _transfer #
  - [Int] _mint #
  - [Int] _burn #
  - [Int] _approve #
  - [Int] _burnFrom #

+ GoldToken (BEP20)
  - [Pub] mint #
    - modifiers: onlyOwner
  - [Ext] delegates
  - [Ext] delegate #
  - [Ext] delegateBySig #

- [Ext] getCurrentVotes
- [Ext] getPriorVotes
- [Int] _delegate #
- [Int] _moveDelegates #
- [Int] _writeCheckpoint #
- [Int] safe32
- [Int] getChainId

+ [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] SafeBEP20
- [Int] safeTransfer #
- [Int] safeTransferFrom #
- [Int] safeApprove #
- [Int] safeIncreaseAllowance #
- [Int] safeDecreaseAllowance #
- [Prv] _callOptionalReturn #

+ [Lib] SafeMath
- [Int] add
- [Int] sub
- [Int] sub
- [Int] mul
- [Int] div
- [Int] div
- [Int] mod
- [Int] mod

+ MasterChef (Ownable)
- [Pub] <Constructor> #
- [Ext] poolLength
- [Pub] add #
  - modifiers: onlyOwner
- [Pub] set #
  - modifiers: onlyOwner
- [Pub] getMultiplier
- [Ext] pendingEgg
- [Pub] massUpdatePools #
- [Pub] updatePool #
- [Pub] deposit #
- [Pub] withdraw #
- [Pub] emergencyWithdraw #
- [Int] safeEggTransfer #
- [Pub] dev #

- [Pub] setFeeAddress #
- [Pub] updateEmissionRate #
  - modifiers: onlyOwner
- [Pub] updateStartBlock #
  - 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:
`add(uint256 _allocPoint, ...)`, `set(uint256 _pid, ...)` and `updateEmissionRate()` could invoke `massUpdatePools()` function, that can fail due to 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 same address will not be added twice.

Owner privileges

- Owner can change start block value.

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.