HackSafe

Smart Contract Security Audit Report

Got Guaranteed

June 2022

Security Status

★ ★ ★ ★ ★

www.hacksafe.io

Audit Details

Audited project
Got Guaranteed

Deployer address
0x3d0aFE74c84b43b9719577D494D18f0DFe1B9e76

Client contacts
Got Guaranteed team

Blockchain
Ethereum

Website
https://gotg.world/

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) (HackSafe) owe no duty of care towards you or any other person, nor does HackSafe 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 HackSafe 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, HackSafe hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against HackSafe, 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.

Procedure

Step 1 - In-Depth Manual Review

Manual line-by-line code reviews to ensure the logic behind each function is sound and safe from various attack vectors. This is the most important and lengthy portion of the audit process (as automated tools often cannot find the nuances that lead to exploits such as flash loan attacks).

Step 2 - Automated Testing

Simulation of a variety of interactions with your Smart Contract on a test blockchain leveraging a combination of automated test tools and manual testing to determine if any security vulnerabilities exist.

Step 3 – Leadership Review

The engineers assigned to the audit will schedule meetings with our leadership team to review the contracts, any comments or findings, and ask questions to further apply adversarial thinking to discuss less common attack vectors.

Step 4 - Resolution of Issues

Consulting with the team to provide our recommendations to ensure the code's security and optimize its gas efficiency, if possible. We assist project team's in resolving any outstanding issues or implementing our recommendations.

Step 5 - Published Audit Report

Boiling down results and findings into an easy-to-read report tailored to the project. Our audit reports highlight resolved issues and any risks that exist to the project or its users, along with any remaining suggested remediation measures. Diagrams are included at the end of each report to help users understand the interactions which occur within the project.

Background

HackSafe was commissioned by Got Guaranteed token to perform an audit of smart contract:

- https://etherscan.io/address/0xceeb07dd26b36287b6d109f0b06d7e8202ce8c1d

The purpose of the audit was to achieve the

- Ensutre that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be 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.

Contract Details

Token contract details for 23.06.2022

Token Type	: ERC20
Contract name	: GOTGToken
Contract address	: 0xceEB07Dd26b36287B6d109f0b06d7e8202Ce8c1D
Compiler version	: v0.5.17+commit.d19bba13
Max Total supply	: 1,800,000,000
Token Ticker	: GOTG
Decimals	: 18
Token Holders	: 1,065
Top 100 token holder's dominance	: 100.00%
Transactions count	: 1,388
Contract deployer address	: 0x3d0aFE74c84b43b9719577D494D18f0DFe1B9e76
Owner address	: 0x2f5588CC0933b04dC76Fbb25e5D18Fd26aDCF844

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are “Secure”. This token contract does contain owner control, which do not make it fully decentralized as owner does have control over smart contract.

Insecure	Poor secured	Secure	Well-secured

You are here 

We used various tools like Slither, Mythril and Remix IDE. At the same time this finding is based on critical analysis of the manual audit. All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the issues checking status.

We found 0 critical, 0 high, 0 medium and 1 low and some very low-level issues. These issues are not critical ones.

Got Guaranteed Distribution

The top 100 holders collectively own 100.00% (1,799,998,072.92 Tokens) of Got Guaranteed

Token Total Supply: 1,800,000,000.00 Token | Total Token Holders: 1,065

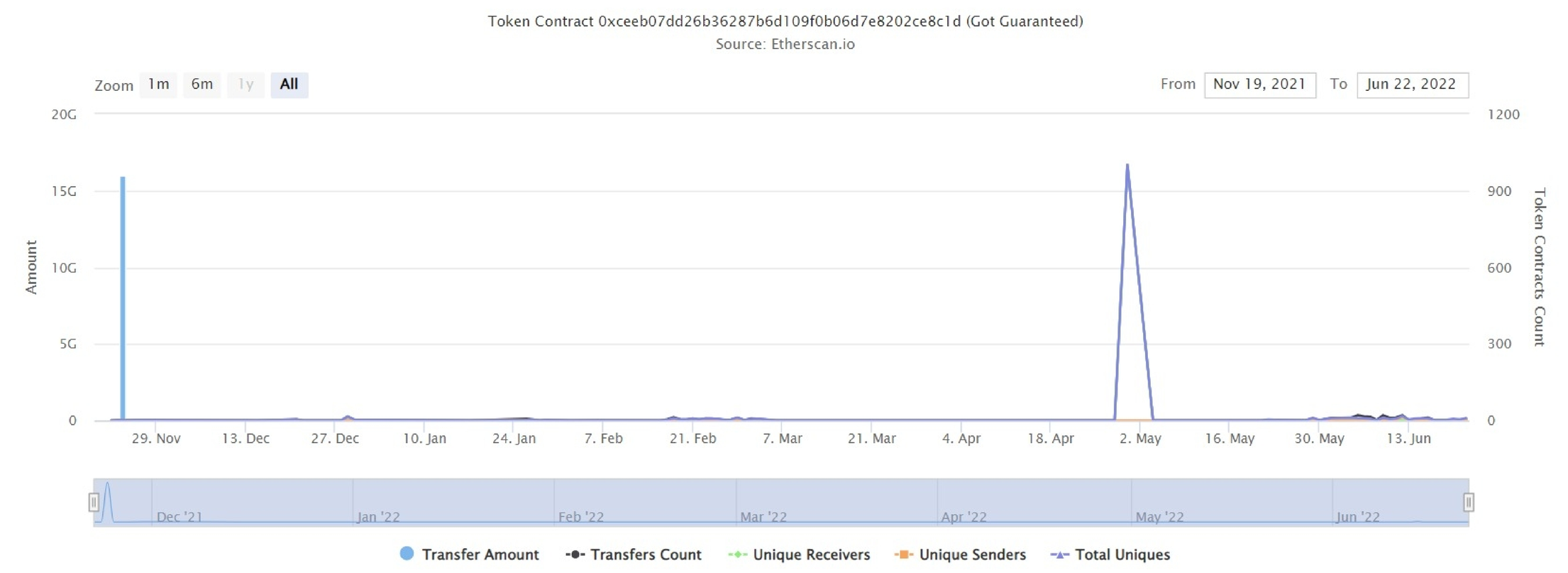
Got Guaranteed Top 20 Token Holders

(A total of 1,799,998,072.92 tokens held by the top 100 accounts from the total supply of 1,800,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x2f5588cc0933b04dc76fbb25e5d18fd26adcf844	1,500,000,000	83.3333%
2	0x54343f0c76db827a26c4160e23f6676ae567572d	218,898,137	12.1610%
3	0xbdd75b461106896f626385b63ae65e6bed0003d9	50,485,042.64678	2.8047%
4	0xc333be28ade348fa48fb42547beac01909ec7bd9	14,000,000	0.7778%
5	0x079fdfe6a9cb653207d1b1c47d92f4543026f617	10,000,000	0.5556%
6	0x355ab425f0677d7fdbbbaf6e08f37c0af702b5a0	1,000,000	0.0556%
7	0x4cce63914761d06794c819bdc50bac8e80fdd61e	1,000,000	0.0556%
8	0x67afc0f48c449a4cf491000efa45ed62489d066a	956,737.24	0.0532%
9	0x6e2bf4cca9acb7f5af493c1bf4d3a06c0d01173d	500,010	0.0278%
10	0x50a6ab44315bfe92dbc1ed9da22f4d67ee597a89	500,010	0.0278%
11	0xced434e7fbf89fdb6f35c55adcd99aff9fdfde66	500,000	0.0278%
12	0x49ca95f86ec9a7fa374b35c17eabc32842df3795	500,000	0.0278%
13	0xbe942398718d09483ea66fac8d54679e48c41548	444,510	0.0247%
14	0x9f0b7598c8739925bd07125e8b09e358ebcd87a3	200,000	0.0111%
15	0xa98c25f42fd87607ce2ef3c71f0e52320da5e3ff	200,000	0.0111%
16	0x90d0602755fbc528261c99b3e1bd48389114adce	100,000	0.0056%
17	0xf5e357cf16de3ffdd88bb63a7529467cd3fbf7e5	100,000	0.0056%
18	0x9932286853f276ed61a83752019d95bd81d0e535	100,000	0.0056%
19	0xd501cdf8b3a70e2fa22e6dc1de31a7ed041ae509	100,000	0.0056%
20	0xb0175c67a0b596505f39611c8b33b5d3cd9cc892	70,000	0.0039%

Got Guaranteed Distribution

Got Guaranteed Contract overview



Contract functions details

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

+[Int] tokenRecipient
    -[Ext] receiveApproval

+ Ownable
    -[Int] <constructor>
    -[Pub] owner
    -[Pub] isOwner
    -[Pub] renounceOwnership #
      -modifiers: onlyOwner
    -[Pub] transferOwnership #
      -modifiers: onlyOwner
    -[Int] _transferOwnership #

+Pausable (Ownable)
    -[Int] <constructor>
    -[Pub] paused
    -[Pub] pause #
      -modifiers: onlyOwner whenNotPaused
    -[Pub] unpause #
      -modifiers: onlyOwner whenPaused

+ERC20Token
    -[Pub] constructor
    -[Int] _transfer #
    -[Pub] transfer #
    -[Pub] transferFrom #
    -[Pub] approve #
    -[Pub] approveAndCall #
    -[Pub] burn #
    -[Pub] burnFrom #

Contract functions details

+GOTGToken (ERC20Token, Ownable,Pausable)
- -[Pub] <constructor>
- -[Pub] freezeAccount #
  - -modifiers: onlyOwner
- -[Pub] unFreezeAccount#
  - -modifiers: onlyOwner
- -[Pub] lockAccount#
  - -modifiers: onlyOwner
- -[Pub] unlockAccount#
  - -modifiers: onlyOwner
- -[Pub] changeName#
  - -modifiers: onlyOwner
- -[Pub] changeSymbol#
  - -modifiers: onlyOwner
- -[Int] _transfer#
  - -modifiers: whenNotPaused
- -[Pub] isAccountLocked
- -[Pub] isAccountFrozen

($) = payable function

= non-constant function

Issues Checking Status

No.	Title	Status
1.	Unlocked Compiler Version	Low issue
2.	Missing Input Validation	Passed
3.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
4.	Possible delays in data delivery	Passed
5.	Oracle calls.	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.	Private use data leaks.	Passed
13.	Malicious Event log.	Passed
14.	Scoping and Declarations.	Passed
15.	Uninitialized storage pointers.	Passed
16.	Arithmetic accuracy.	Passed
17.	Design Logic.	Passed
18.	Safe Open Zeppelin contracts implementation and usage.	Passed
19.	Incorrect Naming State Variable	Passed
20.	Compiler version too old	Passed

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution.

Security Issues

Critical Severity Issues

No critical severity issue found.

High Severity Issues

No high severity issue found.

Medium Severity Issues

No medium severity issues found.

Low Severity Issues

One low severity issue found.

1. Unlocked Compiler Version.

- **Description**

  The contract utilizes an unlocked compiler version. An unlocked compiler version in the contract's source code permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to ambiguity when debugging as compiler-specific bugs may occur in the codebase that would be difficult to identify over a span of multiple compiler versions rather than a specific one.

- **Recommendation**

  It is advisable that the compiler version is alternatively locked at the lowest version possible so that the contract can be compiled. For example, for version >=0.4.22 <0.6.0 the contract should contain the following line:

  pragma solidity 0.5.17;

Centralization

Owner Privileges

- Got Guaranteed Contract:
    - Owner can transfer ownership.
    - Owner can renounce ownership.
    - Owner can pause and unpause transfer tokens.
    - Owner can freeze and unfreeze account for some period of time.
    - Owner can lock account and unlock account.
    - Owner can change name and symbol of token.

This smart contract has some functions which can be executed by the Admin (Owner) only. If the admin wallet private key would be compromised, then it would create trouble moreover if admin have pause the transfers and renounce ownership then admin will not have any ownership to unpause the transfers which may effect users. Following are Admin functions:

- Transferownership
- Renounceownership
- Pause
- Unpause
- Freezeaccount
- Unfreezeaccount
- Lockaccount
- Unlockaccount
- Changename
- Changesymbol

Conclusion

Smart contract contains low severity issues! The further transfer and operations with the fund raised are not related to this particular contract.

HackSafe 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.