Code Security Assessment

Open Guild

Mar 9th, 2022

Table of Contents

Summary

This report has been prepared for Open Guild to discover issues and vulnerabilities in the source code of the Open Guild project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

Project Name	Open Guild
Platform	BSC
Language	Solidity
Codebase	https://github.com/OpenGuild/contracts-v1
Commit	b2e7534a7f376b2ba1ae0101e011ee58171f9236

Audit Summary

Delivery Date	Mar 09, 2022 UTC
Audit Methodology	Static Analysis, Manual Review

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Mitigated	Resolved
Critical	0	0	0	0	0	0	0
Major	2	0	0	1	0	0	1
Medium	1	0	0	0	0	0	1
Minor	1	0	0	0	0	0	1
Informational	1	0	0	0	0	0	1
Discussion	0	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
APO	AggregatePool.sol	f1872997d4ae9e1dc338ad9c05d05a98641b5849b1b0a42dd295d682d41d1bae
BPO	BasePool.sol	640930f24dcf16abc7ccf666e09004945cccba2f00053f34907c1b25150b5efd
GTO	GovernanceToken.sol	3c147df34ab379f9727b83132eb8cdce6fec954091b205d831494efa2c69b4fd
IPO	IndividualPool.sol	701be40d53a6cd8e0d466753ba7820c85dcc79b7a5441f1f742a1c83bd5478d9
PCO	ProtocolConfig.sol	1b9304e6963ccf52233b432e9995e0043377c6466750ac98fdbdb813883a7ca8

Findings

Critical	**0** (0.00%)
Major	**2** (40.00%)
Medium	**1** (20.00%)
Minor	**1** (20.00%)
Informational	**1** (20.00%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
OpenGuild-01	Centralization Related Risks	**Centralization / Privilege**	**Major**	Acknowledged
OpenGuild-02	Financial Models	Logical Issue	Medium	Resolved
BPO-01	Functions With `_` as Name Prefix Are Not `private` or `internal`	Coding Style	Informational	Resolved
IPO-01	Logic issue in `IndividualPool.investFromAggregatePool()`	Logical Issue	Major	Resolved
PCO-01	Lack of Specified Rate Range Restriction	Logical Issue	Minor	Resolved

OpenGuild-01 | Centralization Related Risks

Category	Severity	Location	Status
Centralization / Privilege	Major	Global	Acknowledged

Description

In the contract `AggregatePool` and contracts inherited from it, the role `POOL_MANAGER_ROLE` has authority over the following functions:

- setPoolAllocations(): set current `individual` pools and the allocations of every pool.

Any compromise to the `POOL_MANAGER_ROLE` account may allow a hacker to take advantage of this authority.

In the contract `AggregatePool` and contracts inherited from it, the role `OWNER_ROLE` has authority over the following functions:

- setPoolInvestmentLimit(): set the upper limit of every pool that can invest.
- setInvestorInvestmentLimit(): set the upper limit of every investor that can invest.
- addInvestors/removeInvestors(): give/cancel the authority to `invest()`, `claim()`.
- setPoolManager(): set the role of `POOL_MANAGER_ROLE` who can manage the `individual` pools and allocations.

Any compromise to the `OWNER_ROLE` account may allow a hacker to take advantage of this authority and change the basic configuration of the contract.

In the contract `IndividualPool` and contracts inherited from it, the role `RECIPIENT_ROLE` has authority over the following functions:

- withdraw(): withdraw the undeployed tokens in the contract.
- contribute(): dividend payback.

Any compromise to the `RECIPIENT_ROLE` account may allow a hacker to take advantage of this authority and take out tokens in the contract.

In the contract `IndividualPool` and contracts inherited from it, the role `validAggregatePool` has authority over the following functions:

- investFromAggregatePool(): add the undeployed amounts which can be withdrawn by the recipient.

- claimFromAggregatePool(): get the claimer's unclaimed dividends.

Any compromise to the `validAggregatePool` account may allow a hacker to take advantage of this authority and add the undeployed amounts without transferring pool tokens.

In the contract `IndividualPool` and contracts inherited from it, the role `OWNER_ROLE` has authority over the following functions:

- removeUndeployedCapital(): give back the undeployed capital.
- setMaxBalance(): set the upper limit of the amount the recipient can withdraw.
- setRecipient(): set a new recipient.

Any compromise to the `OWNER_ROLE` account may allow a hacker to take advantage of this authority and set a higher upper limit of the amount the recipient can withdraw.

In the contract `ProtocolConfig` and contracts inherited from it, the role `ADMIN` has authority over the following functions:

- setGovernanceTokenAddress(): set the address of `governanceToken`.
- setWarrantTokenAddress(): set the address of `warrantToken`.
- setTreasuryAddress(): set the address of `treasury`.
- addAggregatePool/removeAggregatePool(): make the `aggregatePool` valid/Invalid.
- addIndividualPool/removeIndividualPool(): make the `individualPool` valid/Invalid.
- setTakeRate(): set the rate of fee the recipient should pay when he/she return the dividends.

Any compromise to the `ADMIN` account may allow a hacker to take advantage of this authority and make the specified pool address valid.

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multi-signature wallets.

Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement;
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles;
  OR
- Remove the risky functionality.

Noted: Recommend considering the long-term solution or the permanent solution. The project team shall make a decision based on the current state of their project, timeline, and project resources.

Alleviation

[Open Guild]: The admin roles mentioned (`OWNER_ROLE`, ADMIN, `PAUSER_ROLE`) ensure that there is some amount of recourse in the case of smart contract bugs or exploits. The admin role will be set to a 2/3 multi-sig wallet controlled by the OpenGuild team to mitigate the impact of an individual key getting compromised. We will share the address of this wallet with the community to ensure transparency.

Unfortunately, guilds cannot use a multi-sig to accept funds as the RECIPIENT_ROLE. Crypto games such as Pegaxy (which OpenGuild guilds are focusing on for our v1) neither support WalletConnect nor have

integrations with multi-sig providers like Gnosis Safe. We ensure that these recipients are held accountable via off-chain legal contracts, which we plan on open-sourcing if possible as well.

In the future, OpenGuild will move to a DAO and use a combination of Snapshot and Gnosis Safe to ensure decision making rests with the community.

OpenGuild-02 | Financial Models

Category	Severity	Location	Status
Logical Issue	Medium	Global	Resolved

Description

The main functions of `OpenGuild` can be described as follows.

1. Users can be divided into two types: investors and guilds.
2. Investors can invest their capital by calling `AggregatePool.invest()`. This capital will be allocated to different `IndividualPools`.
3. The `Recipient` of the `IndividualPool` can withdraw capital from the `IndividualPool` if the withdrawal amount does not exceed the `recipientMaxBalance`. The `recipient` is the Guild.
4. Guilds will use the capital withdrawn from the `IndividualPool` to invest in the game NFT and can return the dividends to the investors by agreement.

Guilds are whitelisted based on their history of successful management of the P2E scholarship program (creditworthiness) and their public reputation in the P2E community (trustworthiness). In V1, the diligence process is conducted by the P2E Yield Pool (aggregate pool) manager, OpenGuild Finance. OpenGuild interviews each guild and reviews information requested from them before requiring qualified guilds to sign a legal contract and verifying the identity of the guild managers.

According to the information on the official website, guilds are appointed by OpenGuild and they will sign the legal contract. This part is not included in the scope of the audit.

And there is an issue here.

Investors can receive dividends depending on the number of shares they hold. Investors can get shares by transferring `poolTokens` to `AggregatePool`. When the contract generates dividends, investors can immediately enter and receive more dividends, which is unfair to some early investors.

Recommendation

Financial models of blockchain protocols need to be resilient to attacks. They need to pass simulations and verifications to guarantee the security of the overall protocol.

The financial model of this protocol is not in the scope of this audit.

Alleviation

[Open Guild]: Theoretically this is correct - at the time that guilds return dividends investors can immediately enter and receive more dividends to receive an outsized allocation of dividends.

In practice, this will not happen. The way that we are structuring version 1 makes it very difficult for new investors to receive more warrant tokens because we will close the pools to new investments after three days or when investors invest a total of 1 million pool tokens.

BPO-01 | Functions With `_` As Name Prefix Are Not `private` Or `internal`

Category	Severity	Location	Status
Coding Style	Informational	BasePool.sol (20220307): 40~52	Resolved

Description

Functions with names starting with `_` should be declared as `private` or `internal`.

Recommendation

Consider changing function visibility to private or internal, or removing `_` from the start of the function name.

Alleviation

The Open Guild team changed the function `__BasePool__init()` visibility to `internal` in commit `d87f5672bace5d5d210ce7e38a4a9c86ab957df9`.

IPO-01 | Logic Issue In `IndividualPool.investFromAggregatePool()`

Category	Severity	Location	Status
Logical Issue	Major	IndividualPool.sol (20220307): 170~173	Resolved

Description

The investor has transferred funds to `AggregatePool` in `AggregatePool.invest()`. The investor's balance should not be checked in the `IndividualPool.investFromAggregatePool()` again. In some cases, the investor's balance may be less than the `amount` and the function may always fail.

Recommendation

Remove the check for investor balance. Also, we suggest that the operation to transfer tokens should be moved to `IndividualPool.investFromAggregatePool()` instead of `AggregatePool.invest()`. The operation may be better with `AggregatePool.invest()` approving the allowance for the `IndividualPool` contract in advance.

Alleviation

The Open Guild team followed our advice and fixed this logic in the `d6f130ce3108d98bab4fbce95d17b3af86dc3fbb` commit.

PCO-01 | Lack Of Specified Rate Range Restriction

Category	Severity	Location	Status
Logical Issue	Minor	ProtocolConfig.sol (20220307): 189~191	Resolved

Description

The `owner` of the contract has permission to modify the fees without limitation.

Therefore, in the extreme case, that fee could be a very large amount of value, which might cause unexpected loss to the project and users.

Recommendation

We advise the client to set a reasonable range restriction for the aforementioned states to ensure the fair distribution of the fees/tokens.

Alleviation

The Open Guild team added a **require** check to the function `setTakeRate()`, the modification was supplied in commit `d87f5672bace5d5d210ce7e38a4a9c86ab957df9`.

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS

AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING

MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.