CERTIK

Preliminary Comments

GAIA - DAI

Dec 13th, 2021

Table of Contents

Summary

This report has been prepared for GAIA to discover issues and vulnerabilities in the source code of the GAIA - DAI 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	GAIA - DAI
Platform	polygon
Language	Solidity
Codebase	https://polygonscan.com/address/0xaBE3AB72b608237d80bE59854bD9aD74c35F5b4F
Commit	

Audit Summary

Delivery Date	Dec 13, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	RewardPool

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
Critical	0	0	0	0	0	0
Major	1	1	0	0	0	0
Medium	0	0	0	0	0	0
Minor	0	0	0	0	0	0
Informational	5	5	0	0	0	0
Discussion	2	2	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
RPC	RewardPool.sol	f79b54a5e46f568b37afe55c88796c3256a3dcf2596a7da92d7a34a7db960ad9

Overview

Gaia is part of the new generation of gaming being built on the blockchain which gives players full ownership of their characters and rewards them for playing in a "play to earn" model of gaming.

External Dependencies

The scope of the audit treats third-party entities as black boxes and assumes their functional correctness. However, in the real world, third parties can be compromised and this may lead to lost or stolen assets.

There are a few dependent injection contracts and addresses in the current project:

Contract `RewardPool`:

- `token`: `0x723B17718289A91AF252D616DE2C77944962d122`;
- `depositToken`: `0x885eb7D605143f454B4345aea37ee8bc457EC730`;
- `_owner`;
- `rewardDistribution`.

Currently, the `_owner` and the `rewardDistribution` are attached to address 0xA1d5bd7298D35d8d5c8210aBe78cc901E25B50A9

We assume these contracts or addresses are valid and non-vulnerable actors and implement proper logic to collaborate with the current project.

Privileged Functions

In the contract `RewardPool`, the role `owner` has the authority over the following function:

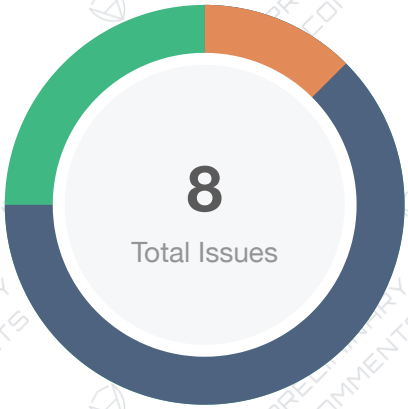
- `renounceOwnership()` to renounce ownership;
- `transferOwnership()` to transfer ownership;
- `setRewardDistribution()` to update the address of `rewardDistribution`;
- `destroyContract()` to call the `selfdestruct` method on the contract thereby transfering balance to the owner.

In the contract `RewardPool`, the address `rewardDistribution` has the authority over the following function:

- `notifyRewardAmount()` to update the `rewardRate`, `lastUpdateTime`, and `periodFinish`;
- `withdrawTOKEN()` to withdraw tokens from the contract.

To improve the trustworthiness of the project, dynamic runtime updates in the project should be notified to the community. Any plan to invoke the aforementioned functions should be also considered to move to the execution queue of the `Timelock` contract.

Findings



Critical	**0** (0.00%)
Major	**1** (12.50%)
Medium	**0** (0.00%)
Minor	**0** (0.00%)
Informational	**5** (62.50%)
Discussion	**2** (25.00%)

ID	Title	Category	Severity	Status
RPC-01	Centralization Risk	**Centralization / Privilege**	**Major**	Pending
RPC-02	Unhandled Return Value	Logical Issue	Informational	Pending
RPC-03	Improper Usage of `public` and `external` Type	Gas Optimization	Informational	Pending
RPC-04	Missing Emit Events	Coding Style	Informational	Pending
RPC-05	Variables that Could be Declared as `constant`	Gas Optimization	Informational	Pending
RPC-06	Unlocked compiler version	Language Specific	Informational	Pending
RPC-07	Discussion on Token Transfer Flow	Logical Issue	Discussion	Pending
RPC-08	Potential Reward Miscalculation	Logical Issue	Discussion	Pending

RPC-01 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	Major	projects/RewardPool.sol (e98d39f): 238	Pending

Description

In the contract `RewardPool`, the role `owner` has the authority over the following function:

- `renounceOwnership()` to renounce ownership;
- `transferOwnership()` to transfer ownership;
- `setRewardDistribution()` to update the address of `rewardDistribution`;
- `destroyContract()` to call the `selfdestruct` method on the contract thereby transfering balance to the owner.

In the contract `RewardPool`, the address `rewardDistribution` has the authority over the following function:

- `notifyRewardAmount()` to update the `rewardRate`, `lastUpdateTime`, and `periodFinish`;
- `withdrawTOKEN()` to withdraw tokens from the contract.

Any compromise to the `owner` and `rewardDistribution` accounts may allow the hacker to take advantage of these functions.

As of Dec-13th-2021, the `_owner` and the `rewardDistribution` are address 0xA1d5bd7298D35d8d5c8210aBe78cc901E25B50A9, which is an EOA.

Recommendation

We advise the client to carefully manage the `owner` and `rewardDistribution` accounts' private keys to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

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

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key;

- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

RPC-02 | Unhandled Return Value

Category	Severity	Location	Status
Logical Issue	Informational	projects/RewardPool.sol (e98d39f): 585, 501, 495, 617	Pending

Description

The return values of `transferFrom()` and `transfer()` are not properly handled. For example,

```
495  depositToken.transferFrom(msg.sender, address(this), amount);
```

```
501  depositToken.transfer(msg.sender, amount);
```

```
585  token.transfer(msg.sender, reward);
```

```
617  token.transfer(msg.sender, amount);
```

`transferFrom()` and `transfer()` are not void-return functions per `IERC20` interface. Ignoring the return values of the functions might cause some unexpected exceptions, especially if the called functions do not revert automatically on failure.

Recommendation

We recommend checking the return values of the aforementioned functions and handling both success and failure cases based on the business logic.

RPC-03 | Improper Usage of `public` and `external` Type

Category	Severity	Location	Status
Gas Optimization	Informational	projects/RewardPool.sol (e98d39f): 196~199, 170~172, 205~207, 492~496, 498~502	Pending

Description

`public` functions that are never called by the contract could be declared as `external`. `external` functions are more efficient than `public` functions.

Recommendation

Consider using the external attribute for public functions that are never called within the contract.

RPC-04 | Missing Emit Events

Category	Severity	Location	Status
Coding Style	Informational	projects/RewardPool.sol (e98d39f): 236~241	Pending

Description

There should always be events emitted in the sensitive functions that are controlled by centralization roles.

Recommendation

It is recommended emitting events for the sensitive functions that are controlled by centralization roles.

RPC-05 | Variables that Could be Declared as `constant`

Category	Severity	Location	Status
Gas Optimization	Informational	projects/RewardPool.sol (e98d39f): 479, 506	Pending

Description

The linked variables could be declared as `constant` since these state variables are never modified.

Recommendation

We recommend to declare these variables as `constant`.

RPC-06 | Unlocked compiler version

Category	Severity	Location	Status
Language Specific	Informational	projects/RewardPool.sol (e98d39f): 7	Pending

Description

The contract has unlocked compiler version. An unlocked compiler version in the source code of the contract 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 different compiler versions. This can lead to an ambiguity when debugging as compiler specific bugs may occur in the codebase that would be hard to identify over a span of multiple compiler versions rather than a specific one.

Recommendation

We advise that the compiler version is instead locked at the lowest version possible that the contract can be compiled at. For example, for version `v0.5.5` the contract should contain the following line:

```
pragma solidity 0.5.5;
```

RPC-07 | Discussion on Token Transfer Flow

Category	Severity	Location	Status
Logical Issue	Discussion	projects/RewardPool.sol (e98d39f): 590	Pending

Description

According to the current implementation, the `rewardDistribution` address will update the reward by calling `notifyRewardAmount()`.

However, the contract implementation itself does not guarantee there is sufficient token to be distributed to users. Currently, the project should manually transfer reward `token` to the contract.

The concern is, if there is not enough token transferred to the current address/contract, users might not be able to get rewards as expected.

We would like to check with the team if the tokens will be guaranteed in this centralized way.

RPC-08 | Potential Reward Miscalculation

Category	Severity	Location	Status
Logical Issue	Discussion	projects/RewardPool.sol (e98d39f): 572, 566~567	Pending

Description

When an user firstly stakes tokens by calling the `stake()` function, modifier `updateReward()` will be indirectly invoked, where `reward[account]` is calculated as follows:

```
546 balanceOf(account)
547     .mul(rewardPerToken().sub(userRewardPerTokenPaid[account]))
548     .div(1e18)
549     .add(rewards[account]);
```

Since this would be the first time the said user stakes, `userRewardPerTokenPaid[account]` and `rewards[account]` are initialized with zero values. Therefore, `rewards[account]` would equal `balanceOf(account).mul(rewardPerToken()).div(1e18)`. Assuming this user is not the first one to stake, `rewardPerToken()` would equal

```
546 rewardPerTokenStored.add(
547     lastTimeRewardApplicable()
548         .sub(lastUpdateTime)
549         .mul(rewardRate)
550         .mul(1e18)
551         .div(totalSupply())
552 );
```

Here, `totalSupply()` would not include this particular user's staked amount, since `super.stake(amount)` is only invoked after the modifier `updateReward(msg.sender)` finishes execution in the `stake()` function. Therefore, `totalSupply()` only reflects the total supply before the stake. As a result, the result of `rewardPerToken()` is a product of total supply before the said user stakes, thereby assigning a higher than normal value to `rewards[account]`.

The same applies to the `withdraw()` function.

Recommendation

We want to confirm with the client that this is the intended design.

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.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

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.

Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.

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.