DeHacker

# Code Security Assessment

# Decentraland

Aug 4th, 2023

# Contents

# Summary

DeHacker's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.
Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow/underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service/logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

# Issue Categories

Every issue in this report was assigned a severity level from the following:

### Critical severity issues

A vulnerability that can disrupt the contract functioning in a number of scenarios or creates a risk that the contract may be broken.

### Major severity issues

A vulnerability that affects the desired outcome when using a contract or provides the opportunity to use a contract in an unintended way.

### Medium severity issues

A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.

### Minor severity issues

A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.

### Informational

A vulnerability that has informational character but is not affecting any of the code.

# Overview

## Project Summary

| | |
|---|---|
| **Project Name** | Decentraland |
| **Platform** | Ethereum |
| **Website** | https://decentraland.org/ |
| **Type** | ERC-721,NFT |
| **Language** | Solidity |

## Vulnerability Summary

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

## Audit scope

| ID | File | SHA256 Checksum |
|---|---|---|
| TIE | managers/Tiers.sol | 51a28da7f517c413610e31041b66dfcc4b739e6d49187d8dfe6ca62bddf001fc |
| TPR | registries/ThirdPartyRegistry.sol | b304b05c4e5b68c8bfab2642e7c512436041c97a647639452a5013775a622327 |

# Findings

| ID | Category | Severity | Status |
|---|---|---|---|
| TIE-01 | Centralization / Privilege | Major | Acknowledged |
| TPR-01 | Centralization / Privilege | Major | Acknowledged |
| TPR-02 | Volatile Code | Medium | Acknowledged |
| TPR-03 | Logical Issue | Minor | Acknowledged |
| TPR-04 | Logical Issue | Minor | Resolved |
| TPR-05 | Control Flow | Informational | Acknowledged |

# MAJOR

## TIE-01|CENTRALIZATION RISK

| Category | Severity | Location | Status |
|---|---|---|---|
| Centralizatio n / Privilege | Major | managers/Tiers. sol | Acknowledged |

## Description

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

updatePrices(),to update the prices of tiers.
addTiers(),to add tiers

Any compromise to the owner account may allow the hacker to take advantage of this.

## Recommendation

We advise the client to carefully manage the owner account's private key to avoid any potential risks of being hacked.In general,we strongly recommend centralized privileges or roles in the protocol to be improved via adecentralized 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 termof 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 privatekey;
Introduction of a DAO/governance/voting module to increase transparency and user involvement.

# MAJOR

## TPR-01|CENTRALIZATION RISK

| Category | Severity | Location | Status |
|---|---|---|---|
| Centralization / Privilege | Major | registries/ ThirdPartyRegistry.sol | Acknowledged |

## Description

In the contract ThirdPartyRegistry,the role owner has the authority over the following functions:
setThirdPartyAgregator(),to set thirdPartyAgregator.
setFeesCollector(),to set feesCollector.
setCommittee(),to set committee.
setAcceptedToken(),to set acceptedToken.
setItemTiers(),to set itemTiers.
setInitialThirdPartyValue(),to set initialThirdPartyValue.
setInitialItemValue(),to set initialItemValue.

The role thirPartyAgregator has the authority over the following functions:
addThirdParties(),to register a new third-party in the contract.
updateThirdParties(),to update the info of third parties

The role committee has the authority over the following function:
reviewThirdParties(),to review third parties.

Any compromise to the owner,committee,thirPartyAgregator account may allow the hacker to take advantage of this.

## Recommendation

We advise the client to carefully manage the owner,committee,thirPartyAgregator accounts' private keys toavoid any potential risks of being hacked In general we strongly recommend centralized privileges or roles in theprotocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced securitypractices,e.g.,Multisignature wallets.

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

Time-lock with reasonable latency eg hours for awareness on privileged operations;

Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the privatekey;

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

# MEDIUM

## TPR-02|MANAGER CAN REMOVE/ADD OTHER MANAGERS

| Category | Severity | Location | Status |
|---|---|---|---|
| Logical Issue | Medium | registries/ ThirdPartyRegistry. sol:254 | Acknowledged |

### Description

The function updateThirdParties allows the managers of the thirdParty to add others as managers.or removeother managers If untrusted users become the manager of the thirdParty,the thirdParty team may lose thecontrol of it.

### Recommendation

We recommend the team add a third-party owner role to manage managers roles.

# MINOR

## TPR-03|THIRD PARTY DEPENDENCIES

| Category | Severity | Location | Status |
|---|---|---|---|
| Volatile Code | Minor | registries/ ThirdPartyRegistry .sol | Acknowledged |

## Description

The contract is serving as the underlying entity to interact with third party committee , acceptedToken,itemTier protocols.The scope of the audit treats rd party entities as black boxes and assume their functional correctness.However,in the real world rd parties can be compromised and this may lead to lost or stolen assets.In addition,upgrades of rd parties can possibly create severe impacts such as increasing fees of rd parties,migrating to newLP pools,etc.

## Recommendation

We understand that the business logic of ThirdPartyRegistry requires interaction with committee,acceptedToken,itemTier,etc.We encourage the team to constantly monitor the statuses of rd parties tomitigate the side effects when unexpected activities are observed.

# MINOR

## TPR-04|CHECK-EFFECTS PATTERN NOT USED

| Category | Severity | Location | Status |
|---|---|---|---|
| Logical Issue | Minor | registries/ ThirdPartyRegistry .sol:321~328 | Resolved |

## Description

State variables are changed after the transfer or external call which might leads to a re-entrancy issue.

## Recommendation

It is recommended to follow checks-effects-interactions pattern for cases like this It shields public functions fromre-entrancy attacks It's always a good practice to follow this pattern checks-effects-interactions pattern alsoapplies to ERC tokens as they can inform the recipient of a transfer in certain implementations.

Refer https//docssoliditylangorg/en/develop/security-considerationshtml?highlight=check-effects%use-the-checks-effects-interactions-pattern.

# INFORMATIONAL

## TPR-05|USERS CAN BUY ITEMSLOTS FOR ANY THIRD PARTIES

| Category | Severity | Location | Status |
|---|---|---|---|
| Control Flow | Informational | registries/ ThirdPartyRegistry .sol:301 | Acknowledged |

### Description

The function buyItemSlots() does not require the caller is the manager of the third party. We would like to confirmwith the client if the current implementation aligns with the original project design.

### Recommendation

We recommend the team only allow managers of the third parties to buy item slots.

# Disclaimer

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate.

FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

# 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.

### Coding Style

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

### Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

### 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.

## 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.

# About

DeHacker is a team of auditors and white hat hackers who perform security audits and assessments. With decades of experience in security and distributed systems, our experts focus on the ins and outs of system security. Our services follow clear and prudent industry standards. Whether it's reviewing the smallest modifications or a new platform, we'll provide an in-depth security survey at every stage of your company's project. We provide comprehensive vulnerability reports and identify structural inefficiencies in smart contract code, combining high-end security research with a real-world attacker mindset to reduce risk and harden code.

### BLOCKCHAIINS

- Ethereum
- Cosmos
- Eos
- Substrate

### TECH STACK

- Python
- Solidity
- Rust
- c++

### CONTACTS

- https://dehacker.io
- https://twitter.com/dehackerio
- https://github.com/dehacker/audits_public
- https://t.me/dehackerio
- https://blog.dehacker.io/


Aug 2023