RUGDOG

SMART CONTRACT SECURITY AUDIT

Final report

Plan: Simple

Decentraland

July 2022

rugdog.net

the@rugdog.net

CONTENTS

INTRODUCTION

RugDog has performed an audit of Decentraland smart contracts.

Decentraland is EVM-compatible software for Polygon Network, aimed at creating a shared virtual world.

Name	Decentraland
Audit date	2022-07-10 - 2022-07-12
Language	Solidity
Network	Polygon Network

CONTRACTS CHECKED

Name	Address
ERC721Bid.sol	0d58d3858ed194f4a3a33092a159c016c683e9def4782138bb9f4fdae2b87e39
MarketplaceV2.sol	e1253eda9e0b5a8b69b99fe42af09c4b9c8dafc4a9681c70f1e34c40bb00b943
BidStorage.sol	1180603d7158931ea5fc310cda743172f1e507690a46aecb6b4282b5e4ce6ae1
RoyaltiesManager.sol	562fbbfbfc2e66d2c6d7064aaee740bfae8610a9b5ddb9b5b3e16cefaee18e4f

AUDIT PROCESS

The code was audited by the team according to the following order:

Automated analysis

- Scanning the project's smart contracts with several publicly available automated Solidity analysis tools
- Manual confirmation of all the issues found by the tools

Manual audit

- Thorough manual analysis of smart contracts for security vulnerabilities
- Smart contracts' logic check

ATTACKS CHECKED

Title	Check result
Unencrypted Private Data On-Chain	✓ passed
Code With No Effects	✓ passed
Message call with hardcoded gas amount	✓ passed
Typographical Error	✓ passed
DoS With Block Gas Limit	✓ passed
Presence of unused variables	✓ passed
Incorrect Inheritance Order	✓ passed
Requirement Violation	✓ passed
Weak Sources of Randomness from Chain Attributes	✓ passed
Shadowing State Variables	✓ passed

Incorrect Constructor Name	✓ passed
Block values as a proxy for time	✓ passed
Authorization through tx.origin	✓ passed
DoS with Failed Call	✓ passed
Delegatecall to Untrusted Callee	✓ passed
Use of Deprecated Solidity Functions	✓ passed
Assert Violation	✓ passed
State Variable Default Visibility	✓ passed
Reentrancy	✓ passed
Unprotected SELFDESTRUCT Instruction	✓ passed
Unprotected Ether Withdrawal	✓ passed
Unchecked Call Return Value	✓ passed
Floating Pragma	✓ passed
Outdated Compiler Version	✓ passed
Integer Overflow and Underflow	✓ passed
Function Default Visibility	✓ passed

CLASSIFICATION OF ISSUES

High severity	Issues leading to assets theft, locking or any other loss of assets or leading to contract malfunctioning.
Medium severity	Issues that can trigger a contract failure of malfunctioning.
Low severity	Issues that do now affect contract functionality. For example, unoptimised gas usage, outdated or unused code, code styleviolations, etc.

ISSUES

High severity issues

1. Excessive owner power (ERC721Bid.sol)

The role _owner in the ERC721Bid contract has excessive power over a set of functions, affecting the variables within them:

setFeesCollectorCutPerMillion setRoyaltiesCutPerMillion setFeesCollector setRoyaltiesManager pause

It's possible for eternal attackers as well as _owner to abuse these abilities.

Recommendation: We recommend taking security measures to avoid hacker attacks or involving a decentralized mechanism.

2. Excessive owner power (MarketplaceV2.sol)

The role _owner in the MarketplaceV2 contract has excessive power over a set of functions, affecting variables within them:

setPublicationFee setFeesCollectorCutPerMillion setRoyaltiesCutPerMillion setFeesCollector setRoyaltiesManager

Recommendation: We recommend taking security measures to avoid hacker attacks or involving a decentralized mechanism.

Medium severity issues

No issues were found

Low severity issues

1. Unfixed Pragma (ERC721Bid.sol)

Pragma in the contract should be fixed to the version with which the contract will be deployed. Otherwise, issues may occur after the compilation at a specific version or above it.

2. Emit events missing (ERC721Bid.sol)

When it's called, the pause function has to emit appropriate events.

CONCLUSION

Decentraland ERC721Bid.sol, MarketplaceV2.sol, BidStorage.sol, RoyaltiesManager.sol contracts were audited. 2 high, 2 low severity issues were found.

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 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 without RugDog prior written consent.

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 RugDog 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.

RUGDOG

WOOF!

rugdog.net

the@rugdog.net