Moonpool

Great Navigation Coin

This is a free audit report

Published on 4th April, 2021

Great Navigation Coin Security Analysis

By MoonPool

Overview Abstract

In this report, we consider the security of the Great Navigation Coin project. Our main task is to find and describe security issues in the smart contracts of the platform to the team.

Limitations and use of the report

Broadly speaking, the assessments can not uncover all vulnerabilities of the given smart contracts, thus, it is not guaranteed that the system is secured even if no vulnerabilities are found. The focus of the assessments was limited to only 1 given smart contract, other contracts were excluded (including external libraries or third party codes).

Note: The report is not investment advice.

The team can publish this report to the community.

Summary

We have found 1 medium severity issue and 4 low severity issues.

Recommendations

We recommend the team to fix all issues, as well as tests coverage to ensure the security of the contracts.

Assessment Overview

Scope of the audit

Source codes for the audit was initially taken from the GitHub repo with the following **1** contract:

1. GNCToken.sol

Other source codes are out of scope for this report.

System Overview

1. GNCToken :

- A standard ERC20 token with a fixed total supply of 300M tokens.

Findings

We have found **1** medium severity issue and **4** low severity issues.

GNCToken

[Medium] **Multiple Withdrawal Attack** possible with approve function

This is a well-known attack in a ERC20 token, can read more details here. Consider to make changes to prevent the attack. Marking the issue as **Medium** since it is easy to fix.

[Low] Redundant implementation of fallback function with revert.

If the contract doesn't implement the fallback function, it will revert all ETH or BNB that is wrongly sent to the contract, thus, there is no need to implement the fallback function with only revert statement.

[Low] **balanceOf/allowance** Remove unused return value

In the function **balanceOf**, the return value ***balance*** is unused. Consider to either remove o**r assign balance = balances[_owner].**
Similarly for **allowance** function**.**

[Low] Upgrade to a (fixed) newer solidity version

The latest version is 0.8.x, while the contract is using ^0.5.16. Consider to use a fixed and newer version for the source code.

[Low] Code Convention

1. Incorrect indentation in line 96 for emit event call.
2. No visibility for attributes: **balances** and **allowed** don't have visibility state. Consider to use *private*, *public*, etc.

Testing

The team has not provided any tests for given contracts. We recommend adding full tests coverage to ensure the logic works as expected and the security of the smart contracts.