Moonpool

PoseidonSwap Project

This is a free audit report

Published on 24th March, 2021

HyruleSwap Security Analysis

By MoonPool

Overview Abstract

In this report, we consider the security of the HyruleSwap 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).

We only make a quick report for 1 critical issue that we have found, the issue could lead to the loss of funds from users. To get a full report, please fill a request audit form.

Note: The report is not investment advice.
The team can publish this report to the community.

Summary

We have found 1 critical severity issue. We have notified the HyruleSwap team about the issue.

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 deployed contract **LostPartyWheel**:

Other source codes are out of scope for this report.

System Overview

The system overview is based on the document on their website here.

Findings

We have found **1** critical severity issue from a quick scanning the contract.

LostPartyWheel

[Critical] Attackable using a simple contract to bet

The **bet** function of the contract allow both contracts and normal addresses to play a bet. The flow of a bet:

1. Sender call bet(amount, color, salt) to place a bet.
2. LostPartWheel check conditions and update user's bet data, as well as the total pot data.
3. LostPartWheel collects tokens from the sender.
4. If the totalPot equals MAX_BEFORE_SPIN, it starts the spinning and updates rewards for winners.

Flow to bypass:
With a contract, it can check whether after calling the **bet** function, it wins any rewards or not. If it doesn't win any rewards, it can simply revert the whole transaction.

1. Calling the **bet** function from LostPartWheel.
2. Check if the contract is one of the winners: **rewards(address(this)) > XXX**;
3. If it is not one of the winners -> revert -> only lost some transaction fee.

4. If it is one of the winners -> call **claimRewards** and enjoy.

Since there are few conditions to start the spinning, we will need to wait until the current totalPot is around 250, then making a bet transaction with 50 Rupee.

We have made some transactions to play the bet and successfully earned **57 RUPEE** tokens.

Transaction that reverts when no winning
Transaction that wins the pot

Note: This issue has been **FIXED** in the latest the LostPartyWheel contract

Buy our team a coffee: **0xA0b91Ec92dBE8Af20fc9058bE61a2da09508255c**
Request audit from our team