Powerledger × PESSIMISTIC

SECURITY ANALYSIS

by Pessimistic

This report is public

September 26, 2024

Abstract 2
Disclaimer 2
Summary 2
General recommendations 2
Project overview 3
  Project description 3
Audit process 4
Manual analysis 5
  Critical issues 5
  Medium severity issues 5
  Low severity issues 5
  Notes 5
    N01. Project roles 5

ABSTRACT

In this report, we consider the security of smart contracts of Powerledger escrow project. Our task is to find and describe security issues in the smart contracts of the platform.

DISCLAIMER

The audit does not give any warranties on the security of the code. A single audit cannot be considered enough. We always recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. Besides, a security audit is not investment advice.

SUMMARY

In this report, we considered the security of Powerledger escrow smart contracts. We described the audit process in the section below.

The audit did not reveal any issues.

GENERAL RECOMMENDATIONS

We do not have any further recommendations.

PROJECT OVERVIEW

Project description

For the audit, we were provided with Powerledger escrow project on a private GitHub repository, commit b51a79ba15f23e17f1347553e59b00c610d454dc.

The scope of the audit included everything.

The documentation for the project included `README.md`.

All 14 tests pass successfully. The code coverage is 57.14%.

The total LOC of audited sources is 63.

AUDIT PROCESS

We started the audit on September 16, 2024, and finished on September 17, 2024.

We inspected the materials provided for the audit. Then, we contacted the developers for an introduction to the project. After a discussion, we performed preliminary research.

We manually analyzed all the contracts within the scope of the audit and checked their logic. Among other, we verified the following properties of the contracts:

- Whether the integration of the new contract with the deployed token works properly.

We scanned the project with the following tools:

- Static analyzer Slither;
- Our plugin Slitherin with an extended set of rules;
- Semgrep rules for smart contracts.

We ran tests and calculated the code coverage.

We combined in the report all the verified issues we found during the manual audit or discovered by automated tools.

MANUAL ANALYSIS

The contracts were completely manually analyzed, their logic was checked. Besides, the results of the automated analysis were manually verified. All the confirmed issues are described below.

Critical issues

Critical issues seriously endanger project security. They can lead to loss of funds or other catastrophic consequences. The contracts should not be deployed before these issues are fixed.

The audit showed no critical issues.

Medium severity issues

Medium severity issues can influence project operation in the current implementation. Bugs, loss of potential income, and other non-critical failures fall into this category, as well as potential problems related to incorrect system management. We highly recommend addressing them.

The audit showed no issues of medium severity.

Low severity issues

Low severity issues do not directly affect project operation. However, they might lead to various problems in future versions of the code. We recommend fixing them or explaining why the team has chosen a particular option.

The audit showed no issues of low severity.

Notes

N01. Project roles

The project includes an admin role, which is initially granted to two addresses. Internally, the contract utilizes `DEFAULT_ADMIN_ROLE`, which has extended permissions. As a result, each admin can grant and revoke the role, i.e., modify the list of admins.

Moreover, leakage/compromise of the admin's private key may lead to fund loss.

This analysis was performed by Pessimistic:

Evgeny Marchenko, Senior Security Engineer
Yhtyyar Sahatov, Security Engineer
Konstantin Zherebtsov, Business Development Lead
Irina Vikhareva, Project Manager
Alexander Seleznev, CEO

September 26, 2024