JuicyBet Token Security Analysis

by Pessimistic

This report is public

May 22, 2024

Abstract

In this report, we consider the security of smart contracts of JuicyBet Token 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 JuicyBet Token smart contracts. We described the audit process in the section below.

The initial audit did not show any issues. All tests passed.

After the initial audit, we checked the updated codebase. We did not find any issues. All tests passed, and the code coverage decreased.

General recommendations

We have no recommendations.

Project overview

Project description

For the audit, we were provided with JuicyBet Token project on a public GitHub repository, commit 4a396a3ed64410caa501bb834b5167059cc22af9, and a deployed contract address 0x5C59514c9F05D3820CA246b8dAEB4E61da6F1773.

The scope of the audit included the **JuicySips.sol** file.

The project has no documentation.

16 tests out of 16 pass successfully. The code coverage is 85.71%.

The total LOC of audited sources is 33.

Token details

Name:	JuicySips
Symbol:	JSP
Decimals:	18
Total Supply:	888,888,888

Codebase update #1

After the initial audit, the codebase was updated, and we were provided with commit 70953a090c06beddb09daf52101b0e6736bb8b6d. The contract was redeployed to the 0x4B6f82a4eD0B9E3767F53309b87819a78d041A7f address.

All 16 tests passed. The code coverage decreased to 66.67%.

Audit process

We made the audit on May 8, 2024.

We inspected the materials provided for the audit.

We manually analyzed the token and checked its logic. Among other, we verified the following properties of the contract:

- Conformance with the ERC20 standard;
- The provided contract address 0x5C59514c9F05D3820CA246b8dAEB4E61da6F1773 source code is equivalent to the audited code;
- Correctness of initialization.

We ran tests, calculated the code coverage, and combined all results in a private report.

We checked the updated version of the code on May 22, 2024. The developers redeployed the contract to the new 0x4B6f82a4eD0B9E3767F53309b87819a78d041A7f address.

We re-ran the tests, calculated the code coverage, and updated the report.

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.

This analysis was performed by Pessimistic:

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

May 22, 2024