CYBRO × PESSIMISTIC

SECURITY ANALYSIS

by Pessimistic

This report is public

December 9, 2024

ABSTRACT

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

The audit showed one issue of medium severity: Project's role. Also, several low-severity issues were found. All tests passed. The code coverage is not measurable, but it is not critical for this code because of its simplicity.

After the initial audit, the Project's role issue of medium severity was partially addressed as the developers immediately provided the address of the Multisig wallet.

GENERAL RECOMMENDATIONS

We recommend fixing the mentioned issues and creating more comprehensive public documentation with more details about the process: what the allocation means, how the rewards are calculated, how the `force` should work, etc.

PROJECT OVERVIEW

Project description

For the audit, we were provided with CYBRO Staking project on a public GitHub repository, commit c95aef189e3be077d89a846db787b1ecb056f313.

The scope of the audit included:

- **CYBROStaking.sol**;
- **LockedCYBRO.sol**;
- **LockedCYBROStaking.sol**.

The project's description included the following link.

All 10 tests pass successfully. The code coverage is not measured due to the "stack too deep" problem.

The total LOC of audited sources is 215.

AUDIT PROCESS

We started the audit on December 8 and finished it on December 9, 2024.

During the work, we stayed in touch with the developers. We also discussed general questions about project motivation, number of tests, and code coverage.

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 all users will be able to retrieve their tokens in emergencies;
- The impact of owner rights;
- Whether tokens are properly accounted for.

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 also sent the results to the developers in the text file.

We ran tests and tried to calculate the code coverage, but it could not be measured because of the "stack too deep" problem.

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

The developers provided the comment for M01 on December 9, 2024.

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.

M01. Project's roles

In the current implementation, the system depends heavily on the owner role. Thus, there are scenarios that can lead to undesirable consequences for the project and its users, e.g., if owner's private keys become compromised.

- In the **CYBROStaking** contract, the owner can:
    - Front-run the `stake` function and change the APR or lock time of staking;
    - Censor users by changing the `minBalance` value;
    - Withdraw user rewards through the `withdrawFunds` function.
- **(addressed)** In the **LockedCYBRO** contract, the owner can:
    - Censor users by removing them from minter or transfer whitelists;
    - Change the minting mode in the `setMintableByUsers` function.

We recommend designing contracts in a trustless manner or implementing proper key management, e.g., setting up a multisig.

> `Comment from the developers:`
> The **LockedCYBRO** contract has already been deployed. The owner of the **LockedCYBRO** contract is the Multisig wallet with 0x66E424337c0f888DCCbCf2e0730A00A526D716f6 address.

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.

L01. Code quality

Consider reading the `users[msg.sender]` storage variable to the local one at line 97 in the `CYBROStaking.withdraw` function to avoid creating the additional local variable `balance` at line 106 and to improve code readability.

L02. Unnecessary inheritance

The **LockedCYBROStaking** contract is inherited from the **CYBROStaking** and **Ownable** contracts. However, the **CYBROStaking** contract has already been inherited from the **Ownable**.

Consider removing unused import at line 6 and the inheritance from the **Ownable** contract in the **LockedCYBROStaking** contract.

L03. Unused import

The **LockedCYBROStaking** and **CYBROStaking** contracts import the same contracts: **IERC20Metadata** and **SafeERC20**. However, the **LockedCYBROStaking** contract is inherited from the **CYBROStaking** contract.

Consider removing unnecessary imports at lines 7 and 8 from the **LockedCYBROStaking** contract.

This analysis was performed by Pessimistic:

Daria Korepanova, Senior Security Engineer
Evgeny Bokarev, Junior Security Engineer
Irina Vikhareva, Project Manager
Alexander Seleznev, CEO

December 9, 2024