
# Work X Token Security Analysis

## by Pessimistic

This report is public

December 4, 2023

# Abstract

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

The audit showed no issues. The token is ready for production deployment.

# General recommendations

We recommend implementing CI to run tests, calculate code coverage, and analyze code with linters and security tools.

# Project overview

## Project description

For the audit, we were provided with Work X Token project on a public GitHub repository, commit a1037e509c15470f0e53263da390a64d725ce176.

The scope of the audit included the **WorkToken.sol** contract.

The documentation for the project included the Work X Whitepaper.

All 28 tests pass successfully. The code coverage could not be calculated because of the issue in the `solidity-coverage` plugin.

The total LOC of audited sources is 16.

| Token details: | |
|---|---|
| Name: | Work X Token |
| Symbol: | WORK |
| Decimals: | 18 |
| Total Supply: | up to 100 000 000 |

# Audit process

We started and finished the audit on November 28, 2023.

We inspected the materials provided for the audit. Then, we contacted the developers for an introduction to the project.

During the work, we stayed in touch with the developers and discussed confusing or suspicious parts of the code.

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:

- Conformance with the ERC20 standard.

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. Overpowered roles (commented)

The system relies heavily on the roles (`DEFAULT_ADMIN_ROLE`, `MINTER_ROLE`) that can mint tokens.

There are scenarios that can lead to undesirable consequences for the project and its users, e.g. if the private key for this role becomes compromised. We recommend designing contracts in a trustless manner or implementing proper key management, e.g., setting up a multisig.

> *Comment from the developers:* *Owner role (`DEFAULT_ADMIN_ROLE`) will be transferred to the multisig after deployment. `Minter` role will only be given to other contracts we have deployed (we want to distribute and claim-vest even to ourselves for the different buckets of tokens (enterprise, marketing, etc.), so it gradually unlocks, and we are sure to have a budget in the future as well.*

This analysis was performed by Pessimistic:

Evgeny Marchenko, Senior Security Engineer
Yhtyyar Sahatov, Security Engineer
Daria Korepanova, Senior Security Engineer
Irina Vikhareva, Project Manager
Konstantin Zherebtsov, Business Development Lead

December 4, 2023