Contract Audit Results

Prepared on: Feb 09, 2021

Contract: AUD430

Prepared by:

Charles Holtzkampf

Sentnl

Prepared for:

Michael Yeates

Dacoco GmbH

Table of Contents

Executive Summary

This document outlines any issues found during the audit of the contracts:

- teleporteos
- teleporteth

The contract has 0 flaws or security vulnerabilities. The risk associated with this contract is low

REMARK	MINOR	MAJOR	CRITICAL
0	0	0	0

Severity Description

REMARK

Remarks are instances in the code that are worthy of attention, but in no way represent a security flaw in the code. These issues might cause problems with the user experience, confusion with new developers working on the project, or other inconveniences.

Things that would fall under remarks would include:

- Instances where best practices are not followed
- Spelling and grammar mistakes
- Inconsistencies in the code styling and structure

MINOR

Issues of Minor severity can cause problems in the code, but would not cause the code to crash unexpectedly or for funds to be lost. It might cause results that would be unexpected by users, or minor disruptions in operations. Minor problems are prone to become major problems if not addressed appropriately.

Things that would fall under minor would include:

- Logic flaws (excluding those that cause crashes or loss of funds)
- Code duplication
- Ambiguous code

MAJOR

Issues of major security can cause the code to crash unexpectedly, or lead to deadlock situations.

Things that would fall under major would include:

- Logic flaws that cause crashes
- Timeout exceptions
- Incorrect ABI file generation
- Unrestricted resource usage (for example, users can lock all RAM on contract)

CRITICAL

Critical issues cause a loss of funds or severely impact contract usage.

Things that would fall under critical would include:

- Missing checks for authorization
- Logic flaws that cause loss of funds
- Logic flaws that impact economics of system
- All known exploits (for example, on_notification fake transfer exploit)

Methodology

Throughout the review process, we check that the token contract:

- Documentation and code comments match logic and behaviour
- Is not affected by any known vulnerabilities

Our team follows best practices and industry-standard techniques to verify the proper implementation of the smart contract. Our smart contract developers reviewed the contract line by line, documenting any issues as they were discovered.

Our strategies consist largely of manual collaboration between multiple team members at each stage of the review, including:

I. Due diligence in assessing the overall code quality of the codebase.
II. Testing contract logic against common and uncommon attack vectors.
III. Thorough, manual review of the codebase, line-by-line.

Our testing includes

- Overflow Audit
- Authority Control Audit Authority Vulnerability Audit
- Authority Excessive Audit
- Safety Design Audit Hard-coded Audit
- Show coding Audit
- Abnormal check Audit
- Type safety Audit
- Denial of Service Audit
- Performance Optimization Audit
- Design Logic Audit
- False Notice Audit
- False Error Notification Audit

- Counterfeit Token Audit
- Random Number Security Audit
- Rollback Attack Audit

Contract Files

Filename	SHA256
teleporteos.cpp	485228b05402565a05b922884dd1d79f6eb03f0bdc8963fde09cfb948e70c123
teleporteos.hpp	3a04ba8878dab408920837cf2f2a575edf79be6b261e0daaf1c43da3ba2e09b5
teleport.sol	e7b1ad0206a4d73621bdfaad3248cb59c096ce1fa934cdca1797a4ff592df3da