

BITLOCUS

audit / code review report

November 25, 2021

TABLE OF CONTENTS

1. License
2. Disclaimer
3. Approach and methodology
4. Description
5. Findings

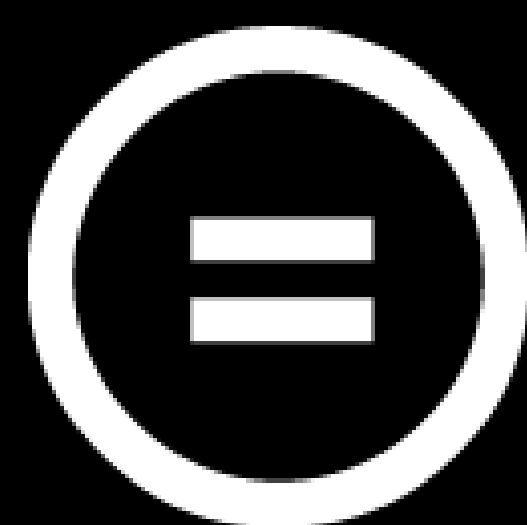
BITLOCUS

audit / code review report

November 25, 2021

LICENSE

Attribution-NonCommercial 4.0 International (CC BY-NC 4.0)



BITLOCUS

audit / code review report

November 25, 2021

DISCLAIMER

THE CONTENT OF THIS AUDIT REPORT IS PROVIDED “AS IS”, WITHOUT REPRESENTATIONS AND WARRANTIES OF ANY KIND.

THE AUTHOR AND HIS EMPLOYER DISCLAIM ANY LIABILITY FOR DAMAGE ARISING OUT OF, OR IN CONNECTION WITH, THIS AUDIT REPORT.

COPYRIGHT OF THIS REPORT REMAINS WITH THE AUTHOR.

BITLOCUS

audit / code review report

November 25, 2021

APPROACH AND METHODOLOGY

PURPOSE

1. Determine the correct operation of the protocol, according to the design specification.
2. Identify possible vulnerabilities that could be exploited by an attacker.
3. Detect errors in the smart contract that could lead to unexpected behavior.
4. Analyze whether best practices were followed during development.
5. Make recommendations to improve security and code readability.

CODEBASE

Repository	https://github.com/Bitlocus/terraswap/tree/main/contracts/terraswap_token
Branch	main
Commit hash	191c1fb11e84771a022d793b70b9fe70988e50d3

METHODOLOGY

1. Reading the available documentation and understanding the code.
2. Doing automated code analysis and reviewing dependencies.
3. Checking manually source code line by line for security vulnerabilities.
4. Following guidelines and recommendations.
5. Preparing this report.

BITLOCUS

audit / code review report

November 25, 2021

DESCRIPTION

Issues Categories:

<u>Severity</u>	<u>Description</u>
CRITICAL	vulnerability that can lead to loss of funds, failure to recover blocked funds, or catastrophic denial of service.
HIGH	vulnerability that can lead to incorrect contract state or unpredictable operation of the contract.
MEDIUM	failure to adhere to best practices, incorrect usage of primitives, without major impact on security.
LOW	recommendations or potential optimizations which can lead to better user experience or readability.

Each issue can be in the following state:

<u>State</u>	<u>Description</u>
PENDING	still waiting for resolving
ACKNOWLEDGED	know but not planned to resolve for some reasons
RESOLVED	fixed and deployed

BITLOCUS

audit / code review report

November 25, 2021

FINDINGS

<u>Finding</u>	<u>Severity</u>	<u>Status</u>
#1 - improve tests code coverage	LOW	ACKNOWLEDGED

BITLOCUS

audit / code review report

November 25, 2021

#1 - IMPROVE TESTS CODE COVERAGE

Test Coverage is an important indicator of software quality and an essential part of software maintenance. It helps in evaluating the effectiveness of testing by providing data on different coverage items. It is a useful tool for finding untested parts of a code base. Test coverage is also called code coverage in certain cases.

Test coverage can help in monitoring the quality of testing and assist in directing the test generators to create test cases that cover areas that have not been tested. It helps in determining a quantitative measure of Test coverage, which is an indirect measure of quality and identifies redundant test cases that do not increase coverage.

RECOMMENDATION

It is highly recommended to test all of the functions (especially those with private access) and have high ratio of test coverage. It is recommended to use code coverage reporting tool for the Cargo build system for example [cargo-tarpaulin](#).

<u>Severity.</u>	<u>Status</u>
LOW	ACKNOWLEDGED

PROOF OF SOURCE

https://github.com/Bitlocus/terraswap/tree/main/contracts/terraswap_token/src

auditmos.com

AUDITMOS

Secure your space

contact@auditmos.com
