AUDITMOS

Secure your space

LUART (TOKEN)

audit / code review report

March 16, 2022

TABLE OF CONTENTS

contact@auditmos.com

AUDITMOS

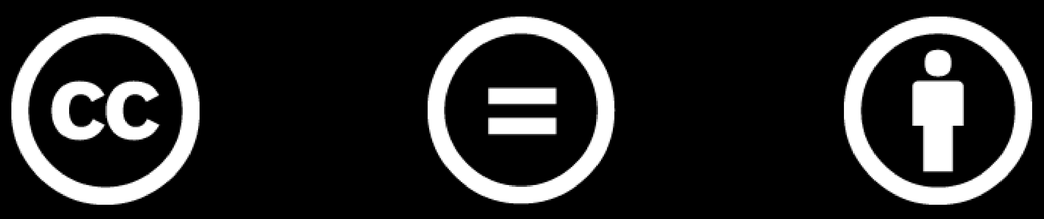
Secure your space

LUART (TOKEN)

audit / code review report

March 16, 2022

LICENSE



contact@auditmos.com

AUDITMOS

Secure your space

LUART (TOKEN)

audit / code review report

March 16, 2022

DISCLAIMER

contact@auditmos.com

AUDITMOS

Secure your space

LUART (TOKEN)

audit / code review report

March 16, 2022

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/luart-terra/luart-contracts-public
Branch	main
Commit hash	be5ee86f9f43bb3809e81274581e04bf2a450b09

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 guildlines and recommendations.
5. Preparing this report.

contact@auditmos.com

AUDITMOS

Secure your space

LUART (TOKEN)

audit / code review report

March 16, 2022

DESCRIPTION

Issues Categories:

Severity	Description
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:

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

contact@auditmos.com

AUDITMOS

Secure your space

LUART (TOKEN)

audit / code review report

March 16, 2022

AUDIT SCOPE

1. getting to know the project	✓
2. research into architecture	✓
3. manual code read	✓
4. check of permissions	✓
5. identify common Rust vulnerabilities	✓
6. test coverage	✓
7. static analysis	✓

contact@auditmos.com

AUDITMOS

Secure your space

LUART (TOKEN)

audit / code review report

March 16, 2022

FINDINGS

Finding	Severity	Status
#1 - improve tests code coverage	LOW	RESOLVED
#2 - catch common mistakes and improve your Rust code	LOW	RESOLVED

contact@auditmos.com

AUDITMOS

Secure your space

LUART (TOKEN)

audit / code review report

March 16, 2022

#1 - IMPROVE TESTS CODE COVERAGE

Severity	Status
LOW	RESOLVED

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 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.

```
Feb 27 19:19:19.123  INFO cargo_tarpaulin::report: Coverage Results:
|| Uncovered Lines:
|| contracts/token/src/contract.rs: 29, 35, 37, 40, 42-44, 48-51, 53, 58-60, 65, 67, 69-72, 74, 77, 81, 87-89, 91-92, 97
-98, 103, 108, 113-114, 120, 130, 135, 138, 146, 148-149, 151, 154, 156, 158-159, 161, 165-167, 173, 176, 185, 187-188,
190, 193, 195-196, 198-199, 201, 206-208, 214, 217, 225-227, 229-230, 233-234, 238-239, 243-244, 248-249, 253, 255, 260-
266, 268, 273-274, 277, 282-290, 293, 297-299, 301, 305-306, 308, 310, 314, 320-321, 324, 326-328, 331, 333-334, 337
|| contracts/token/src/msg.rs: 31-32, 35, 37-39, 42-44, 47-48, 50, 57-60, 62, 68-71, 73-75, 78
|| Tested/Total Lines:
|| contracts/token/src/contract.rs: 0/123 +0.00%
|| contracts/token/src/msg.rs: 0/25 +0.00%
||
0.00% coverage, 0/148 lines covered, +0% change in coverage
```

PROOF OF SOURCE

https://github.com/luart-terra/luart-contracts-public/tree/main/contracts/token

LUART (TOKEN)

audit / code review report

March 16, 2022

Severity	Status
LOW	RESOLVED

#2- CATCH COMMON MISTAKES AND IMPROVE YOUR RUST CODE

Using cargo clippy we found a collection of lints with common mistakes. Please consider them and improve your Rust code.

RECOMMENDATION

Clippy tool found 10 warnings. Complete list of these was sent to client.

The Clippy tool is a collection of lints to analyze your code so you can catch common mistakes and improve your Rust code.This tool is for helping the developer of a smart contract to write better code.

To install Clippy, enter the following:

```
rustup component add clippy
```

To run Clippy's lints on any Cargo project, enter the following:

```
cargo clippy
```

```
warning: `luart-token` (lib) generated 10 warnings
    Finished dev [unoptimized + debuginfo] target(s) in 26.31s
```

auditmos.com

AUDITMOS

Secure your space

contact@auditmos.com