LYOPAY

SMART CONTRACT AUDIT

LYO Credit (LYO)

ZOKYO.

March 11th, 2022 | v. 1.0

PASS

Zokyo’s Security Team has concluded that this smart contract passes security qualifications to be listed on digital asset exchanges.

SCORE

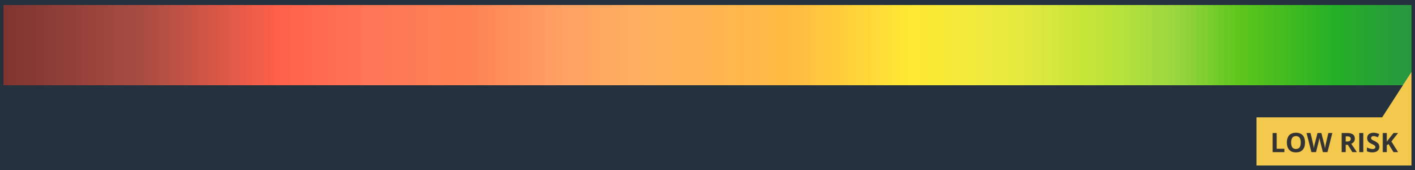
98

TECHNICAL SUMMARY

This document outlines the overall security of the LYO smart contracts, evaluated by Zokyo's Blockchain Security team.

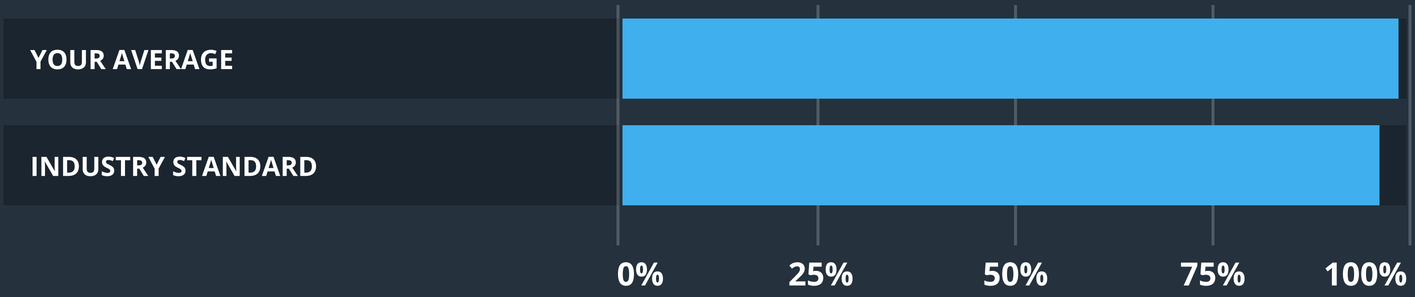
The scope of this audit was to analyze and document LYO smart contract codebase for quality, security, and correctness.

Contract Status



There were some no critical and medium issues found during the audit.

Testable Code



The testable code is 100% which is above the industry standard of 95%.

It should be noted that this audit is not an endorsement of the reliability or effectiveness of the contract, rather limited to an assessment of the logic and implementation. In order to ensure a security of the contract we at Zokyo recommend that the LYO team put in place a bug bounty program to encourage further and active analysis of the smart contract.

TABLE OF CONTENTS

AUDITING STRATEGY AND TECHNIQUES APPLIED

The LYO smart contract's source code was taken from the deployed contract provided by the LYO team
Original contract
https://bscscan.com/address/0xc0b7d0da96c310dc1ec81163506144c6ad28a381 (proxy)
https://bscscan.com/address/0xb46e10416664b9849a707e29974fd32d1c4b2c65 (impl)

Final contract:
https://bscscan.com/address/0x9bad6C75b5a4E72dF8147cc89d068cc848648e59 (proxy)
https://bscscan.com/address/0x740d3ed0f865dc8bec02915f5d498aabbd2aafd5 (impl)

Within the scope of this audit Zokyo auditors have reviewed the following contract(s):

- LYO.sol

Throughout the review process, care was taken to ensure that the contract:

- Implements and adheres to existing standards appropriately and effectively;
- Documentation and code comments match logic and behavior;
- Distributes tokens in a manner that matches calculations;
- Follows best practices in efficient use of resources, without unnecessary waste;
- Uses methods safe from reentrance attacks;
- Is not affected by the latest vulnerabilities;
- Whether the code meets best practices in code readability, etc.

Zokyo's Security Team has followed best practices and industry-standard techniques to verify the implementation of LYO smart contracts. To do so, the code is reviewed line-by-line by our smart contract developers, documenting any issues as they are discovered. Part of this work includes writing a unit test suite. In summary, our strategies consist largely of manual collaboration between multiple team members at each stage of the review:

1 Due diligence in assessing the overall code quality of the codebase.

2 Cross-comparison with other, similar smart contracts by industry leaders.

3 Testing contract logic against common and uncommon attack vectors.

4 Thorough, manual review of the codebase, line-by-line.

EXECUTIVE SUMMARY

There were no critical issues found during the audit. Nevertheless, there were found several issues conencted to the ownership and access. Also, there were few issues with code style and performance. Nevertheless - all issues were successfully resolved by LYO team.

Also, the contract provided by the LYO team had no native unit tests coverage, nevertheless Zokyo security team provided the full coverage in order to ensure the whole logic.

It needs to be mentioned, that the token contract is upgradeable, thus its implementation can be updated. After the contact with the LYO team we ensured that the upgradeability will not be used out of critical situations.

STRUCTURE AND ORGANIZATION OF DOCUMENT

For ease of navigation, sections are arranged from most critical to least critical. Issues are tagged “Resolved” or “Unresolved” depending on whether they have been fixed or addressed. Issues tagged “Verified” contain unclear or suspicious functionality that either needs explanation from the Customer’s side or it is an issue that the Customer disregards as an issue. Furthermore, the severity of each issue is written as assessed by the risk of exploitation or other unexpected or otherwise unsafe behavior:

Critical

The issue affects the contract in such a way that funds may be lost, allocated incorrectly, or otherwise result in a significant loss.

High

The issue affects the ability of the contract to compile or operate in a significant way.

Medium

The issue affects the ability of the contract to operate in a way that doesn’t significantly hinder its behavior.

Low

The issue has minimal impact on the contract’s ability to operate.

Informational

The issue has no impact on the contract’s ability to operate.

COMPLETE ANALYSIS

HIGH | RESOLVED

Owner is not initialized

LYO.sol, initializer()
Owner is not initialized. The issue is marked as high and not critical, because owner is used for funds recovering only.

Recommendation:
Add owner initialization.

MEDIUM | RESOLVED

SafeERC20 usage preferable

TokenRecover, recoverERC20()
Consider usage of SafeTransfer library, because unsafe token transfer will revert with the non-standard implementation of the ERC20 standard (like for example USDT has).

Recommendation:
Use SafeERC20 library.

LOW | RESOLVED

Ownable and AccessControl mixed

LYO.sol utilizes both ownership and access control with roles, which is over-complification of the contract, as owner can be set up as a separate role or the admin itself.

Recommendation:
Omit Ownable functionality and use AccessControl only.

LOW | RESOLVED

Extra check

LYO.sol, renounceRole() contains extra check, since 2 of 3 existing “require” statements are enough to cover all cases.

Recommendation:
Remove extra check.

INFORMATIONAL | UNRESOLVED

Upgradable token

LYO.sol
The token is implemented as upgradable, thus the owner of the token (deployer) has full control over the contract. Thus, this fact should be reflected in the audit report.

Recommendation:
Provide a non-upgradable token or provide the guarantees for the users against the contract modification.

	LYO.sol
Re-entrancy	Pass
Access Management Hierarchy	Pass
Arithmetic Over/Under Flows	Pass
Unexpected Ether	Pass
Delegatecall	Pass
Default Public Visibility	Pass
Hidden Malicious Code	Pass
Entropy Illusion (Lack of Randomness)	Pass
External Contract Referencing	Pass
Short Address/ Parameter Attack	Pass
Unchecked CALL Return Values	Pass
Race Conditions / Front Running	Pass
General Denial Of Service (DOS)	Pass
Uninitialized Storage Pointers	Pass
Floating Points and Precision	Pass
Tx.Origin Authentication	Pass
Signatures Replay	Pass
Pool Asset Security (backdoors in the underlying ERC-20)	Pass

CODE COVERAGE AND TEST RESULTS FOR ALL FILES

Tests written by Tarantino team

As part of our work assisting LYO team in verifying the correctness of their contract code, our team was responsible for writing integration tests using Truffle testing framework.

Tests were based on the functionality of the code, as well as review of the LYO contract requirements for details about issuance amounts and how the system handles these.

```
Contract: LYO.initialize() function
      ✓ Correct roles (301ms)
      ✓ Token in Alice`s account (234ms)
Contract: LYO.decimals() function
      ✓ Correct decimals (245ms)
Contract: LYO.pause() function
      ✓ Alice can pause contract (328ms)
      ✓ Bob (haven`t pauser role) can`t pause contract (615ms)
Contract: LYO.unpause() function
      ✓ Alice can pause and unpause contract (396ms)
      ✓ Bob (haven`t pauser role) can`t unpause paused contract (250ms)
Contract: LYO.renounceRole() function
      ✓ Alice can grant PAUSER_ROLE to Bob and Bob can renounce it (214ms)
      ✓ Negative result if sender tries to renounce not himself (240ms)
      ✓ Alice can grant FROZEN_ROLE to Bob and Bob can`t renounce it (243ms)
Contract: LYO.recoverERC20() function
      ✓ Positive status (204ms)
Contract: LYO._beforeTokenTransfer() function
      ✓ Correct work if sender and _from has not FROZEN_ROLE (170ms)
      ✓ FROZEN_ROLE sender can`t transfer (234ms)
      ✓ FROZEN_ROLE _from can`t transfer (364ms)
  14 passing (5s)
```

FILE	% STMTS	% BRANCH	% FUNCS	% LINES	% UNCOVERED LINES
contracts/	100.00	100	100.00	100.00	
LYO.sol	100.00	100	100.00	100.00	
All files	100.00	100	100.00	100.00	

We are grateful to have been given the opportunity to work with the LYO team.

The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them.

Zokyo's Security Team recommends that the LYO team put in place a bug bounty program to encourage further analysis of the smart contract by third parties.

ZOKYO.