# Security Assessment

# Ros

Jun 30th, 2021

# Table of Contents

# Summary

This report has been prepared for Ros to discover issues and vulnerabilities in the source code of the Ros project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases given they are currently missing in the repository;
- Provide more comments per each function for readability, especially contracts are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

# Overview

## Project Summary

| | |
|---|---|
| Project Name | Ros |
| Platform | Ethereum |
| Language | Solidity |
| Codebase | https://etherscan.io/address/0xbf759D75967caB23aE67DD72D69f161f004afb0D#code |
| Commit | |

## Audit Summary

| | |
|---|---|
| Delivery Date | Jun 30, 2021 |
| Audit Methodology | Static Analysis, Manual Review |
| Key Components | |

## Vulnerability Summary

| Vulnerability Level | Total Count | Pending | Partially Resolved | Resolved | Acknowledged | Declined |
|---|---|---|---|---|---|---|
| Critical | 0 | 0 | 0 | 0 | 0 | 0 |
| Major | 0 | 0 | 0 | 0 | 0 | 0 |
| Medium | 0 | 0 | 0 | 0 | 0 | 0 |
| Minor | 0 | 0 | 0 | 0 | 0 | 0 |
| Informational | 2 | 0 | 0 | 0 | 2 | 0 |
| Discussion | 0 | 0 | 0 | 0 | 0 | 0 |

# Audit Scope

| ID | file | SHA256 Checksum |
| --- | --- | --- |
| RRR | Rose.sol | 8c1afcc18f915d39462a62494e7649cbd734ba1b2b6e08f3fee3a95a3d812647 |

# Findings


| | |
|---|---|
| Critical | 0 (0.00%) |
| Major | 0 (0.00%) |
| Medium | 0 (0.00%) |
| Minor | 0 (0.00%) |
| Informational | 2 (100.00%) |
| Discussion | 0 (0.00%) |

| ID | Title | Category | Severity | Status |
|---|---|---|---|---|
| RRR-01 | Redundant Code | Gas Optimization | Informational | Acknowledged |
| RRR-02 | Conformance To Solidity Naming Conventions | Coding Style | Informational | Acknowledged |

# RRR-01 | Redundant Code

| Category | Severity | Location | Status |
| --- | --- | --- | --- |
| Gas Optimization | Informational | Rose.sol: 7~15 | Acknowledged |

## Description

The interface `IERC20` is actually unused.

## Recommendation

Consider removing it.

## Alleviation

No alleviation.

# RRR-02 | Conformance To Solidity Naming Conventions

| Category | Severity | Location | Status |
| --- | --- | --- | --- |
| Coding Style | Informational | Rose.sol: 29 | Acknowledged |

## Description

Rose.totalSupply (Rose.sol#29) is not in UPPER_CASE_WITH_UNDERSCORES.

## Recommendation

It is recommended to rename it to `TOTAL_SUPPLY`.

## Alleviation

No alleviation.

# Appendix

## Finding Categories

### Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

### Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

## Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

# Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without CertiK's prior written consent.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

# About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.