ABegic

RMPL
Smart Contract Audit Deliverable

Date: Aug 26, 2020

Version: 1

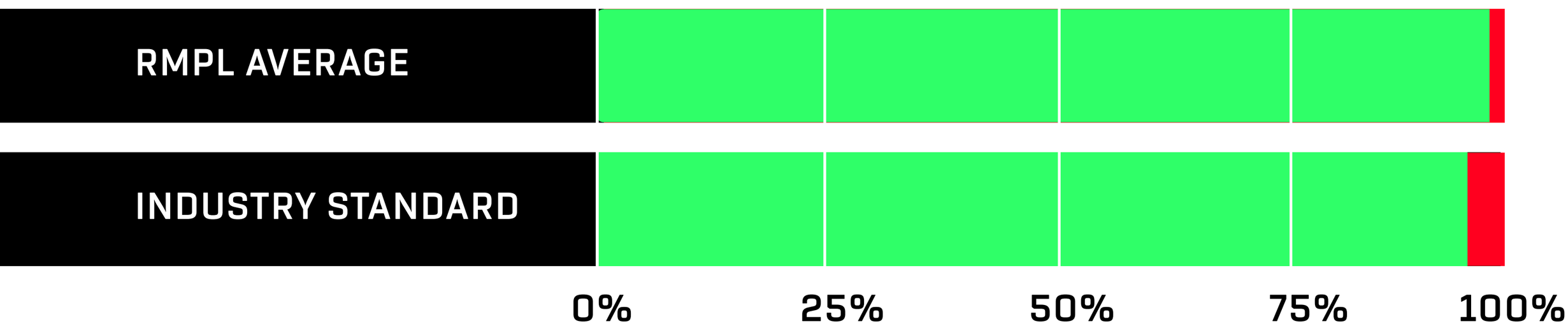
Executive Summary

This document outlines the security posture of the RMPL smart contracts as evaluated by Adis Begic, Blockchain Security Researcher. The scope of the audit was to analyze and document the codebase provided by the RMPL team for security vulnerabilities.

Contract(s) Status

No security issues were discovered during the audit of the smart contract in scope.

Code Coverage



Testable code is 97.80% while the industry standard is **95%**.

It should be noted that this document is not an endorsement of the effectiveness of the smart contracts, rather limited to an assessment of the logic and implementation. This audit should be seen as an informative practice with the intent of raising awareness on the due diligence involved in secure development and make no material statements or guarantees in regards to the operational state of the smart contract(s) post-deployment. I undertake no responsibility for any potential consequences of the deployment or use of the smart contract(s) related to the audit.

ABegic

Table of Contents

Audit Strategy & Applied Techniques

I've performed an extensive audit of the smart contract in scope, the latest version provided by RMPL on Aug 11th, 2020. The smart contract in scope was audited using the following strategy and techniques.

I've followed general best practices and industry-standard techniques to verify the implementation of the RMPL smart contracts during my assessment.

The codebase has been reviewed line-by-line and concerning security issues have been documented as discovered.

Throughout the audit, caution was taken to assess that the smart contract:

- Follows good practices with respect to efficient gas consumption, while avoiding unnecessary costs.
- Implements robust functions which are safe from well-known attack vectors i.e. reentrance.
- The logic and behavior adheres to the associated documentation and code comments.
- Distributions of tokens are designed in a sustainable way safeguarded from i.e. infinite loops.
- implements design patterns which provides the highest possible code quality.

To summarize, the auditing strategy consists largely of manual expertise.

1	Due diligence in evaluating the overall quality of the codebase.
2	Cross-comparison with other, similar smart contracts in respect to wrapped up functionality.
3	Testing the functionality against common and uncommon attack vectors.
4	Comprehensive, manual review of the codebase, line-by-line.
5	Deploying the smart contracts on testnet to run practical tests.

2.1 Summary

The RMPL codebase consists of two harmonizing smart contracts. The scope of the audit has been limited to the RMPL.sol smart contract, however, the team intent to audit the Rebaser.sol smart contract once its development has been finalized as well.

RMPL.sol is essentially a detailed ERC20 token with additional rebase functionality while Rebaser.sol provides functionality to randomize the rebasing of the token supply.

The majority of the codebase is influenced by popular and properly tested smart contracts that have been publicly disclosed by OpenZeppelin and the Ampleforth project. The smart contract implements arithmetic libraries, in the form of SafeMath.sol and SafeMathInt.sol to prevent issues such as overflows during sensitive calculations.

Altogether, it is evident that security has been a priority and the codebase of concern is a reflection of great use of defensive programming with extensive use of the CEI pattern and other best practices.

2.2 Behavioral Consistency

The RMPL.sol smart contract behaves as intended by the developer and I was not able to find an occasion in which the behavioral consistency could be disrupted nor interrupt the functional state of the smart contracts.

2.3 Coverage Results

The RMPL.sol smart contract in scope is supported by an auxiliary unit test suite that outputs a near-perfect score of 97.80% in terms of test coverage and as a result, it complies with industry standards.

For clarity of understanding, I've arranged my observations from critical to informational. The severity of each issue is evaluated based on the risk of exploitation or other unexpected behavior.

Critical

An issue flagged as critical means that it can affect the smart contract in a way that can cause serious financial implications, catastrophic impact on reputation, or disruption of core functionality.

High

An Issue flagged as high means that it can affect the ability of the smart contract to function in a significant way i.e. lead to broken execution flows or cause financial implicants.

Medium

An Issue flagged as medium means that the risk is relatively small and that the issue can not be exploited to disrupt execution flows or lead to unexpected financial implications.

Undetermined

An issue flagged as undetermined means that the impact of the discovered issue is uncertain and needs to be studied further.

Low

An Issue flagged as informational does not pose an immediate threat to disruption of functionality, however, it should be considered for security best practices or code integrity.

I am honored and grateful to have been given the opportunity to work with the RMPL project.

The smart contracts in scope form an interesting example of a token with an elastic supply model which is dynamically changing based on randomized rebasing.

No security issues were found during my assessment and the smart contract in scope pass my auditing process.

The statements made in this report do not constitute legal or investment advice and I should not be held accountable for decisions made based on them.

Furthermore, it is always a good idea to put in place a bug bounty program to encourage further analysis of the smart contract by other third parties or to engage your community.

ABegic

Appendix A

Source Code Fingerprints

FILE	FINGERPRINT
RMPL.sol	cdc30a9637efcfa3fad3031d31646cd5a4fac51898ad1847b200d9a668778220

Code Coverage

FILE	% BRANCHES	% FUNCTIONS	% LINES
RMPL.sol	96.56%	100%	97.80%
ALL FILES	96.56%*	100%*	97.80%*