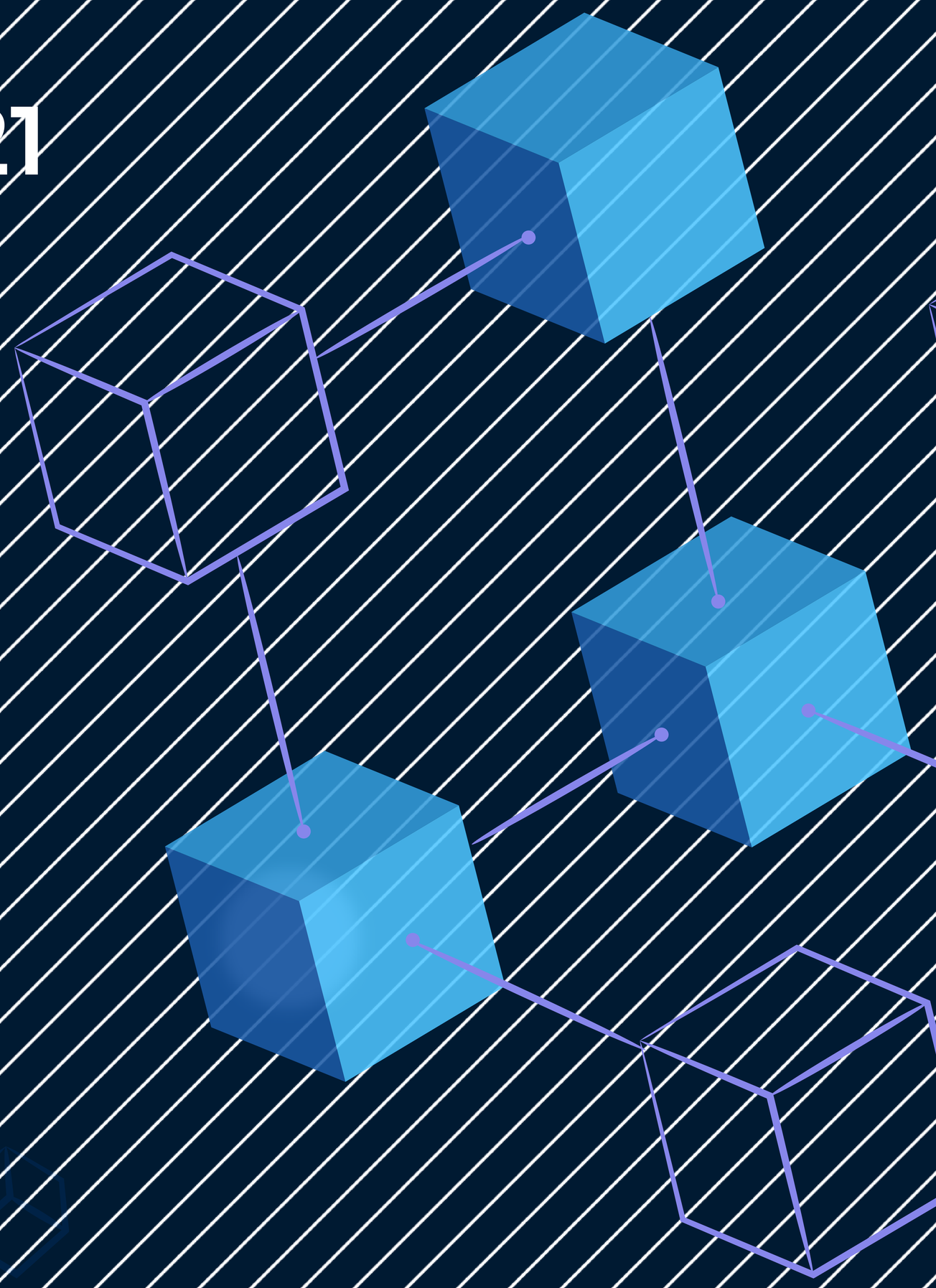
QuillAudits

Audit Report
December, 2021



For

RARX

Contents

Scope of the Audit

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

Checked Vulnerabilities

We have scanned the smart contract for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that we considered:

- Re-entrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Use of tx.origin
- Exception disorder
- Gasless send
- Balance equality
- Byte array
- Transfer forwards all gas
- BEP20 API violation
- Malicious libraries
- Compiler version not fixed
- Redundant fallback function
- Send instead of transfer
- Style guide violation
- Unchecked external call
- Unchecked math
- Unsafe type inference
- Implicit visibility level

Techniques and Methods

Throughout the audit of smart contract, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments match logic and expected behaviour.
- Token distribution and calculations are as per the intended behaviour mentioned in the whitepaper.
- Implementation of **BEP20** token standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods and tools were used to review all the smart contracts.

Structural Analysis
In this step, we have analysed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.

Static Analysis
Static analysis of smart contracts was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of smart contracts.

Code Review / Manual Analysis
Manual analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analysed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of the automated analysis were manually verified.

Gas Consumption
In this step, we have checked the behaviour of smart contracts in production. Checks were done to know how much gas gets consumed and the possibilities of optimization of code to reduce gas consumption.

Tools and Platforms used for Audit
Remix IDE, Truffle, Truffle Team, Solhint, Mythril, Slither, Solidity statistic analysis, Theo.

Issue Categories

Every issue in this report has been assigned to a severity level. There are four levels of severity, and each of them has been explained below.

Risk-level	Description
High	A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality, and we recommend these issues be fixed before moving to a live environment.
Medium	The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems, and they should still be fixed.
Low	Low-level severity issues can cause minor impact and or are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.
Informational	These are severity issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Number of issues per severity

Type	High	Medium	Low	Informational
Open	0	0	0	0
Acknowledged	0	0	2	0
Closed	0	0	0	0

Introduction

During the period of **16th Dec 2021 to 18th Dec 2021** - QuillAudits Team performed a security audit for Rarx smart contracts.

The code for the audit was taken from the following official link: https://bscscan.com/address/0x152149e684b0566f1dbdce6ed03b0ef113917103#code

Issues Found – Code Review / Manual Testing

High severity issues

No issues were found.

Medium severity issues

No issues were found.

Low severity issues

1. **No way to burn tokens externally**

   Description
   _burn() and _burnFrom() are internal functions. So they cannot be called externally from the smart contract if the owner of the token wants to burn some supply. But as mentioned in the specification docs, the team plans to burn the remaining tokens after sale.

   Remediation
   It is advised to add an onlyOwner burn function and call the above internal functions in them if the owner wants to burn any Rarx tokens in future.

   Status: Acknowledged

2. **Renounce Ownership**

   Description
   Usually, the contract's owner is the account that deploys the contract. As a result,the owner is able to perform certain privileged activities on his behalf. The renounceOwnership function is used in smart contracts to renounce ownership. Otherwise, if the contract's ownership has not been transferred previously, it will never have an Owner, which is risky.

Remediation
It is advised that the Owner cannot call renounceOwnership without first transferring ownership to a different address. Additionally, if a multi-signature wallet is utilized, executing the renounceOwnership method for two or more users should be confirmed. Alternatively, the Renounce Ownership functionality can be disabled by overriding it.
Refer this post for additional info -
https://www.linkedin.com/posts/razzor_github-razzorsecrazzorsec-contracts-activity-6873251560864968705-HOS8

Status: Acknowledged

Informational issues

No issues were found.

Functional test

Function Names	Testing results	Logical results	Overall results
transfer	Passed	Passed	Passed
allowance	Passed	Passed	Passed
approve	Passed	Passed	Passed
transferFrom	Passed	Passed	Passed
increaseAllowance	Passed	Passed	Passed
decreaseAllowance	Passed	Passed	Passed
mint	Passed	Passed	Passed
renounceOwnership	Passed	Passed	Passed
transferOwnership	Passed	Passed	Passed

Closing Summary

Overall, smart contracts are very well written and adhere to guidelines. Two low Severity issues found during the Audit, which has been Acknowledged by the Rarx Team.

Disclaimer

Quillhash audit is not a security warranty, investment advice, or an endorsement of Rarx. This audit does not provide a security or correctness guarantee of the audited smart contracts. 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. Securing smart contracts is a multistep process. One audit cannot be considered enough; we recommend that the Rarx Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.

Audit Report
December, 2021

For

RARX

QuillAudits

Canada, India, Singapore, United Kingdom

audits.quillhash.com

audits@quillhash.com