RARI Foundation

Rari Bridged Token Security Analysis

by Pessimistic

This report is public

July 12, 2024

Abstract

In this report, we consider the security of smart contracts of Rari Bridged Token project. Our task is to find and describe security issues in the smart contracts of the platform.

Disclaimer

The audit does not give any warranties on the security of the code. A single audit cannot be considered enough. We always recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. Besides, a security audit is not investment advice.

Summary

In this report, we considered the security of Rari Bridged Token smart contracts. We described the audit process in the section below.

The audit showed one critical issue: Unprotected token registration. The audit also revealed one issue of medium severity: Project roles. Moreover, one low-severity issue was found.

All the tests passed, and the code coverage was sufficient. The documentation was private.

After the initial audit, the codebase was updated again. The developers fixed the critical issue of Unprotected token registration, commented on the medium severity issue of Project roles and fixed the low severity issue.

The number of tests stayed the same.

General recommendations

We recommend implementing CI to run tests, calculate code coverage, and analyze code with linters and security tools.

Project overview

Project description

For the audit, we were provided with Rari Bridged Token project on a public GitHub repository, commit ddf9338bcdf41d9613a75ef7cf979635242679ac.

The scope of the audit included **RariBridgedToken.sol**.

The documentation for the project included the private description.

All 4 tests pass successfully. The code coverage is 74.07%.

The total LOC of audited sources is 72.

Codebase update #1

For the recheck, we were provided with Rari Bridged Token project on a private GitHub repository, commit f22063bf636d58edd2c62ca0c88a565fe5088c9f.

The developers fixed or commented on all issues from the initial audit. The diff did not include test updates.

Audit process

We started the audit on July 8 and finished on July 10, 2024.

We inspected the materials provided for the audit. Then, we contacted the developers for an introduction to the project. During the work, we stayed in touch with the developers and discussed confusing or suspicious parts of the code.

We manually analyzed all the contracts within the scope of the audit and checked their logic. Among other, we verified the following properties of the contracts:

- Whether the code follows the Arbitrum description for the integrations;
- Whether the Arbitrum bridge can also be used between Rari and Arbitrum chains;
- Standard Solidity checks;
- The power of admin's roles.

We scanned the project with the following tools:

- Static analyzer Slither;
- Our plugin Slitherin with an extended set of rules;
- Semgrep rules for smart contracts.

We ran tests and calculated the code coverage.

We combined in a private report all the verified issues we found during the manual audit or discovered by automated tools.

We made the recheck on July 10, 2024. The developers fixed or commented on all the issues. We did not run tests as the update contained no test updates.

Manual analysis

The contracts were completely manually analyzed, their logic was checked. Besides, the results of the automated analysis were manually verified. All the confirmed issues are described below.

Critical issues

Critical issues seriously endanger project security. They can lead to loss of funds or other catastrophic consequences. The contracts should not be deployed before these issues are fixed.

C01. Unprotected token registration (fixed)

The `registerTokenOnL2` function is not protected by any roles. This allows anyone to register a custom gateway with the fake token, bridge these tokens, and receive RARI tokens.

The `registerTokenOnL2` function in the example of the integration's description with the Arbitrum bridge is also protected by the `onlyOwner` modifier.

The issue has been fixed and is not present in the latest version of the code.

Medium severity issues

Medium severity issues can influence project operation in the current implementation. Bugs, loss of potential income, and other non-critical failures fall into this category, as well as potential problems related to incorrect system management. We highly recommend addressing them.

M01. Project roles (commented)

The `MINTER_ROLE` and `DEFAULT_ADMIN_ROLE` can change the router and custom gateway addresses and mint and burn Rari tokens on Arbitrum. In the current implementation, the system depends heavily on the `MINTER_ROLE` and `DEFAULT_ADMIN_ROLE`. Thus, there are scenarios that can lead to undesirable consequences for the project and its users, e.g., if admin's private keys become compromised.

We recommend designing contracts in a trustless manner or implementing proper key management, e.g., setting up a multisig.

> *Comment from the developers:*
>
> *`DEFAULT_ADMIN_ROLE` first will be set to deployer address, later on the role will be transferred to RARI DAO governance.*
>
> *`MINTER_ROLE` will be set to custom gateway address from Arbitrum bridge.*

Low severity issues

Low severity issues do not directly affect project operation. However, they might lead to various problems in future versions of the code. We recommend fixing them or explaining why the team has chosen a particular option.

L01. Unchecked returned value (fixed)

The ERC-20 standard states:

> *"Callers MUST handle `false` from `returns (bool success)`. Callers MUST NOT assume that `false` is never returned!"*

However, the returned value from the `transferFrom` call in the `wrap` function is not checked.

The issue has been fixed and is not present in the latest version of the code.

This analysis was performed by Pessimistic:

Daria Korepanova, Senior Security Engineer
Yhtyyar Sahatov, Security Engineer
Konstantin Zherebtsov, Business Development Lead
Irina Vikhareva, Project Manager
Alexander Seleznev, CEO

July 12, 2024