Offchain Labs Custom Fee ERC-20 Bridge Upgrade and EIP-7702 Fixes

Security Assessment (Summary Report)

March 31, 2025

Prepared for:

Harry Kalodner, Lee Bousfield, Steven Goldfeder, and Ed Felten

Offchain Labs

Prepared by: **Gustavo Grieco**

Table of Contents

Project Summary

Contact Information

The following project manager was associated with this project:

Mary O'Brien, Project Manager
mary.obrien@trailofbits.com

The following engineering director was associated with this project:

Josselin Feist, Engineering Director, Blockchain
josselin.feist@trailofbits.com

The following consultant was associated with this project:

Gustavo Grieco, Consultant
gustavo.grieco@trailofbits.com

Project Timeline

The significant events and milestones of the project are listed below.

Date	Event
January 31, 2025	Delivery of report draft
February 2, 2025	Report readout meeting
March 31, 2025	Delivery of final summary report

Executive Summary

Engagement Overview

Offchain Labs engaged Trail of Bits to review the security of the ERC-20 bridge upgrade process for custom fee token chains. Specifically, we reviewed the patched version of the `ERC20Bridge` contract with a function to fix decimals, the Orbit chains action to upgrade `ERC20Bridge` to 2.1.2, and pre-BoLD EIP-7702 fixes in commit `961a49` and PR #39 (at commit `f4f878e`).

The first two files in scope allow users to upgrade from version 1.1.x of the ERC-20 bridge to version 2.2.1 following some specific instructions when the chain is using a custom fee token. Additionally, this engagement included the review of EIP-7702 fixes applied to a non-BoLD Nitro branch to allow chain owners to receive such fixes without forcing them to upgrade to BoLD. In both cases, invalid upgrade paths should not be possible to perform.

One consultant conducted the review from January 27 to January 31, 2025, for a total of one engineer-week of effort. With full access to source code and documentation, we performed manual review and dynamic testing of the code in scope.

Observations and Impact

The code review uncovered a single informational-severity issue that could allow users to upgrade a contract using an incorrect version during the upgrade of non-BoLD rollups. Additionally, users should be aware that EIP-7702 can cause some unexpected behavior changes, as explained in our previous EIP-7702 audit. We recommend that users carefully review the Arbitrum documentation if they are going to actively use such a feature to interact with the Arbitrum rollups.

Recommendations

Based on the findings identified during the security review, Trail of Bits recommends that Offchain Labs take the following steps:

- Remediate the finding disclosed in this report as part of a direct remediation or any refactoring that may occur when addressing other recommendations.

- Complement the command-line tool to check for valid upgrades with a centralized document explaining valid upgrades and important information for performing the upgrades.

Detailed Findings

1. An invalid upgrade for non-BoLD rollup is possible	
Severity: **Informational**	Difficulty: **High**
Type: Undefined Behavior	Finding ID: TOB-ARBFIX-1
Target: `NitroContracts2Point1Point3UpgradeAction.sol`	

Description

The Orbit action in scope allows users to execute an invalid upgrade path, from 3.x.x to 2.2.3.

EIP-7702 allows EOAs to set their code. Some changes in the Arbitrum contract are required in order to correctly support this EIP. The following code triggers the actual upgrade of the sequencer inbox and the inbox for some specific versions if the rollups are not going to support BoLD:

```
  function perform(address inbox, ProxyAdmin proxyAdmin) external {
        address bridge = IInbox(inbox).bridge();
        address sequencerInbox = IInbox(inbox).sequencerInbox();

        bool isERC20 = false;

        // if the bridge is an ERC20Bridge below v2.x.x, revert
        try IERC20Bridge(bridge).nativeToken() returns (address) {}
        catch {
            isERC20 = true;
            // it is an ERC20Bridge, check if it is on v2.x.x
            try IERC20Bridge_v2(address(bridge)).nativeTokenDecimals() returns
(uint8) {}
            catch {
                // it is not on v2.x.x, revert
                revert("NitroContracts2Point1Point3UpgradeAction: bridge is an
ERC20Bridge below v2.x.x");
            }
        }

        // upgrade the sequencer inbox
        proxyAdmin.upgrade({
            proxy: TransparentUpgradeableProxy(payable((sequencerInbox))),
            implementation: newSequencerInboxImpl
        });
        // upgrade the inbox
        proxyAdmin.upgrade({
            proxy: TransparentUpgradeableProxy(payable((inbox))),
            implementation: isERC20 ? newERC20InboxImpl : newEthInboxImpl
        });
    }
}
```

Figure 1.1: The `depositEth` function (`src/bridge/Inbox.sol#L202–L214`)

The code correctly checks that the upgrade cannot be performed from contract versions below 2.x.x. However, it does not check if the contract is in 3.x.x, as the check for `nativeTokenDecimals` also passes if the contract is already upgraded to the latest version.

Exploit Scenario

Alice performs the upgrade using a contract that is already in 3.x.x, causing the contract to be downgraded.

Recommendations

Short term, add a check to verify that the contract version is not 3.x.x.

Long term, review each upgrade action for both lower and upper bounds of valid versions.

About Trail of Bits

Founded in 2012 and headquartered in New York, Trail of Bits provides technical security assessment and advisory services to some of the world's most targeted organizations. We combine high-end security research with a real-world attacker mentality to reduce risk and fortify code. With 100+ employees around the globe, we've helped secure critical software elements that support billions of end users, including Kubernetes and the Linux kernel.

We maintain an exhaustive list of publications at https://github.com/trailofbits/publications, with links to papers, presentations, public audit reports, and podcast appearances.

In recent years, Trail of Bits consultants have showcased cutting-edge research through presentations at CanSecWest, HCSS, Devcon, Empire Hacking, GrrCon, LangSec, NorthSec, the O'Reilly Security Conference, PyCon, REcon, Security BSides, and SummerCon.

We specialize in software testing and code review projects, supporting client organizations in the technology, defense, and finance industries, as well as government entities. Notable clients include HashiCorp, Google, Microsoft, Western Digital, and Zoom.

Trail of Bits also operates a center of excellence with regard to blockchain security. Notable projects include audits of Algorand, Bitcoin SV, Chainlink, Compound, Ethereum 2.0, MakerDAO, Matic, Uniswap, Web3, and Zcash.

To keep up to date with our latest news and announcements, please follow @trailofbits on Twitter and explore our public repositories at https://github.com/trailofbits. To engage us directly, visit our "Contact" page at https://www.trailofbits.com/contact, or email us at info@trailofbits.com.

Trail of Bits, Inc.
228 Park Ave S #80688
New York, NY 10003
https://www.trailofbits.com
info@trailofbits.com

Notices and Remarks

Copyright and Distribution

Trail of Bits considers this report public information; it is licensed to Offchain Labs under the terms of the project statement of work and has been made public at Offchain Labs' request. Material within this report may not be reproduced or distributed in part or in whole without Trail of Bits' express written permission.

The sole canonical source for Trail of Bits publications is the Trail of Bits Publications page. Reports accessed through sources other than that page may have been modified and should not be considered authentic.

Test Coverage Disclaimer

All activities undertaken by Trail of Bits in association with this project were performed in accordance with a statement of work and agreed upon project plan.

Security assessment projects are time-boxed and often reliant on information that may be provided by a client, its affiliates, or its partners. As a result, the findings documented in this report should not be considered a comprehensive list of security issues, flaws, or defects in the target system or codebase.

Trail of Bits uses automated testing techniques to rapidly test the controls and security properties of software. These techniques augment our manual security review work, but each has its limitations: for example, a tool may not generate a random edge case that violates a property or may not fully complete its analysis during the allotted time. Their use is also limited by the time and resource constraints of a project.