



Offchain Labs Upgrade Executor

Security Assessment (Summary Report)

July 30, 2025

Prepared for:

Harry Kalodner, Steven Goldfeder, and Ed Felten

Offchain Labs

Prepared by: **Jaime Iglesias**

Table of Contents

Table of Contents	1
Project Summary	2
Project Targets	3
Executive Summary	4
A. Code Quality Findings	5
About Trail of Bits	6
Notices and Remarks	7

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:

Benjamin Samuels, Engineering Director, Blockchain
benjamin.samuels@trailofbits.com

The following consultants were associated with this project:

Jaime Iglesias, Consultant
jaime.iglesias@trailofbits.com

Project Timeline

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

Date	Event
July 21, 2025	Pre-project kickoff call
July 24, 2025	Delivery of report draft
July 30, 2025	Delivery of final summary report

Project Targets

The engagement involved reviewing and testing the following target.

Upgrade Executor

Repository	https://github.com/OffchainLabs/upgrade-executor
Version	eea9d607464abb8683a3fe526acbf6d877f924e0
Type	Solidity
Platform	Arbitrum

Executive Summary

Engagement Overview

Offchain Labs engaged Trail of Bits to review the security of the Upgrade Executor, specifically commit [eaa9d60](#). The Upgrade Executors are smart contracts that are part of Arbitrum's governance and are responsible for the execution of critical protocol upgrades (on Arbitrum One, Nova, and Ethereum), system parameter changes, and other changes as instructed by the DAO (via proposal votes and through the L1 Timelock contract) or by the Arbitrum security council.

A team of one consultant conducted the review from July 21 to July 22, 2025, for a total of one engineer-day of effort. With full access to source code and documentation, we performed manual review of the aforementioned commit.

Observations and Impact

Commit [eaa9d60](#) introduces a new `executeCall` function for performing protocol upgrades (as explained by the related [DAO proposal](#)). This new function allows the Upgrade Executor to make an arbitrary `call` to a contract (as opposed to the current implementation, which can perform only arbitrary `delegatecalls` via the `execute` function). The goal is to simplify the upgrade process—for example, by removing the need to create [action contracts](#) for certain upgrades that could be performed via a direct call from the Upgrade Executor.

The review focused exclusively on this new functionality. We assessed whether the function is correctly implemented, whether its implementation aligns with the DAO proposal, and whether it introduces any unexpected side effects.

The review did not reveal any security-relevant issues.

Recommendations

We recommend implementing the recommendation provided in the [Code Quality Findings appendix](#).

A. Code Quality Findings

The following finding is not associated with any specific vulnerabilities. However, fixing it will enhance code readability and may prevent the introduction of vulnerabilities in the future.

- **The Upgrade Executor repository does not have any build instructions whatsoever.** Adding these instructions would help reviewers and developers to quickly build the project and its dependencies along with running its tests and/or other relevant scripts.

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 assessments, supporting client organizations in the technology, defense, blockchain, and finance industries, as well as government entities. Notable clients include HashiCorp, Google, Microsoft, Western Digital, Uniswap, Solana, Ethereum Foundation, Linux Foundation, and Zoom.

To keep up to date with our latest news and announcements, please follow [@trailofbits on X](#) or [LinkedIn](#), 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

© 2025 by Trail of Bits, Inc.

All rights reserved. Trail of Bits hereby asserts its right to be identified as the creator of this report in the United Kingdom.

This report is considered by Trail of Bits to be 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 the express written permission of Trail of Bits.

The sole canonical source for Trail of Bits publications is the [Trail of Bits Publications page](#). Reports accessed through any source 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.