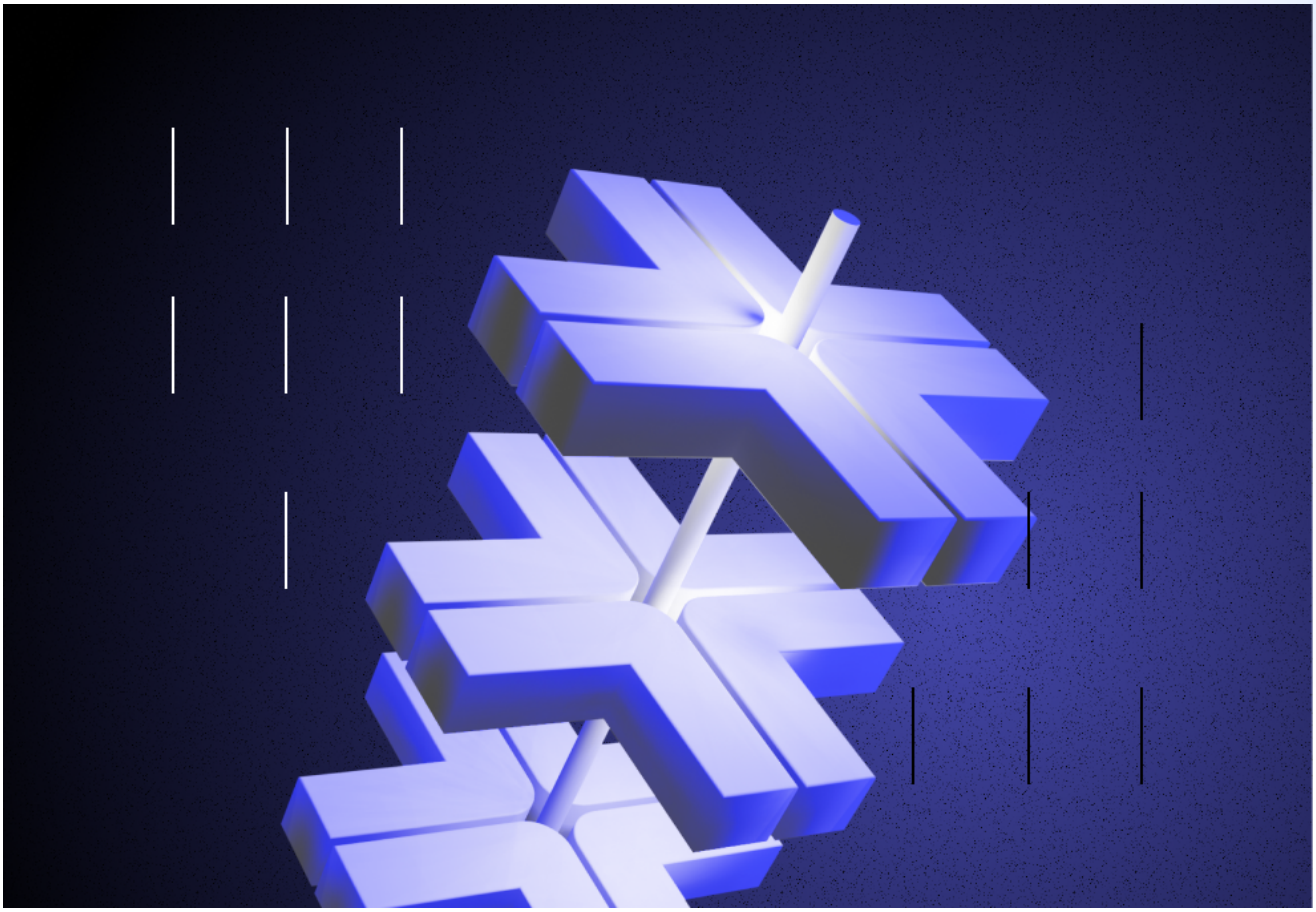
< Back to articles

Axelar: Ethereum Contracts Audit Summary

Andrea Nováková // AUDITS, ETHEREUM JULY 25, 2022



Between **July 4 and July 12, 2022**, Axelar engaged Ackee Blockchain to review and audit changes on several Ethereum contracts. The entire audit process was conducted with a total time commitment of **5 engineering days**. We now publish a summary of our results.

METHODOLOGY

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

We start by reviewing the specifications, sources, and instructions provided to us, which is essential to ensure we understand the project's size, scope, and functionality. This is followed by due diligence using the automated Solidity analysis tools and Slither.

In addition to tool-based analysis, we continue with a detailed manual code review, which is the process of reading the source code line by line to identify potential vulnerabilities or code duplications. When the code review is complete, we run unit tests to ensure the system works as expected and potentially write missing unit or fuzzy tests. We also deploy the contracts locally and try to attack and break the system.

SCOPE

The audit was performed on two repositories: we audited commit *9f9ca0d* of the axelarnetwork/axelar-cgp-solidity repository (file: contracts/AxelarAuthMultisig.sol) and commit *327c543* of the axelarnetwork/axelar-xc20-wrapper repository (file: contracts/*).

During the security review, **we paid particular attention to**:

- validating the upgradeability pattern;
- detecting possible reentrancies in the code;
- ensuring access controls are not too relaxed or too strict;
- looking for common issues such as data validation.

FINDINGS

Here we present our findings.

Critical severity

No critical severity issues were found.

High severity

H1: Ignored return values on LocalAsset interface

Medium severity

M1: Floating dependency on AxelarGateway

Low severity

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

No low severity issues were found.

Warning severity

W1: Pitfalls of upgradeability

W2: The owner can change arbitrarily operatorship and potentially cause DoS

W3: XC20Wrapper owner has escalated priviliges

W4: Missing unit tests

W5: Usage of solc optimizer

Informational severity

I1: Typo in the variable name

I2: Missing events

CONCLUSION

Our review resulted in **9 findings** ranging from *Informational* to *High* severity.

After the audit, **we recommended Axelar to**:

- create documentation including NatSpec comments;
- reconsider the current upgradeability pattern;
- write unit tests for XC20 Wrapper;
- address all other reported issues.

Update: On **July 25, 2022**, Axelar provided an updated codebase that addresses the reported issue. The updated commit for XC20 Wrapper was *4340a2f* and after reporting an incorrect fix in H1F, it was changed to *dd49548*. No changes have been made in the Solidity CGP Gateway. Some of the findings were fixed (H1, M1, W4). A detailed discussion of the exact status of each issue can be found in Appendix D of the report.

Ackee Blockchain's full *Axelar: Ethereum contracts* audit report with a more detailed description of all findings and recommendations can be found here.

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

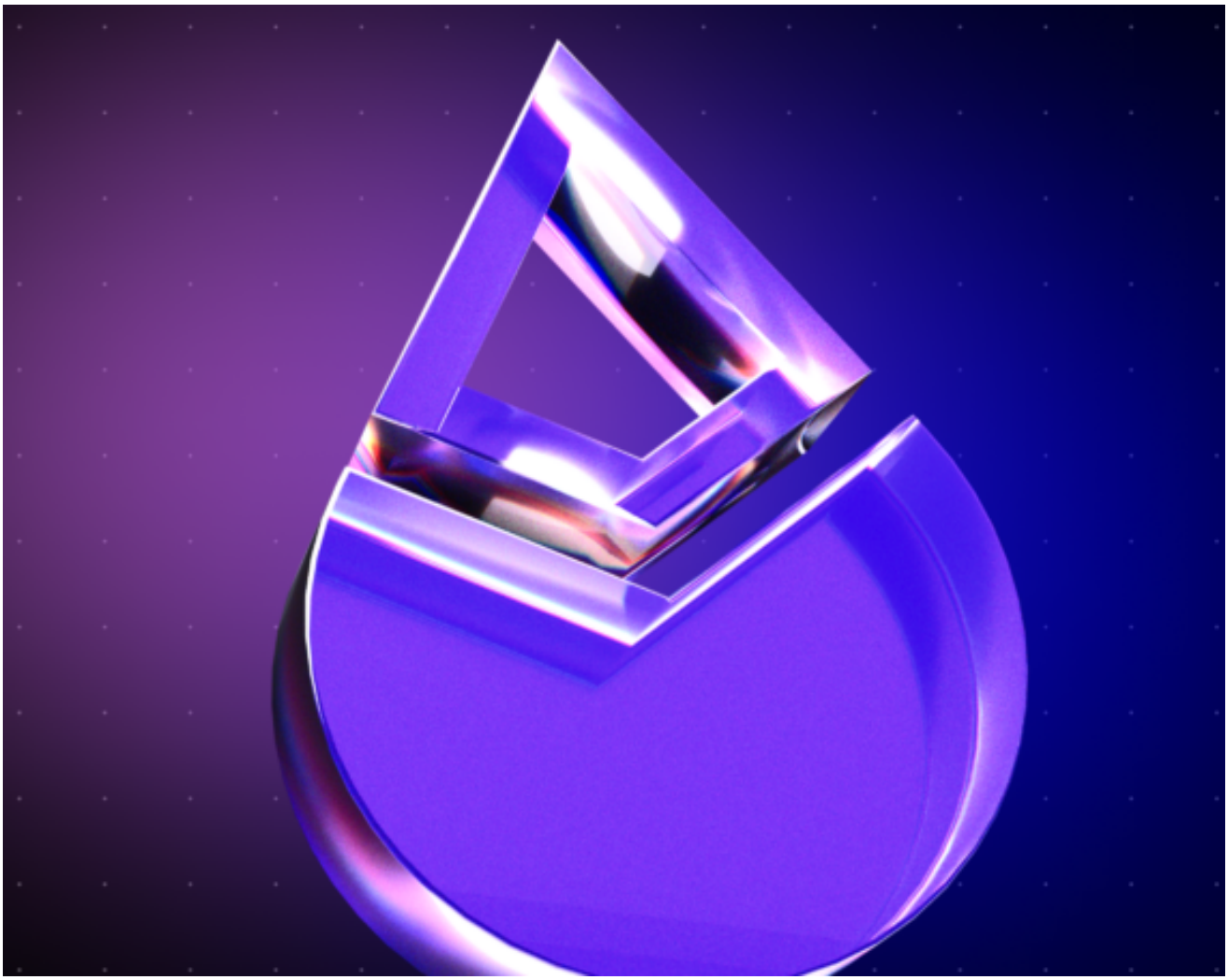
We were delighted to audit **Axelar** and look forward to working with them again.

Audit Axelar Blockchain Cryptocurrency Ethereum EVM Findings Fuzzy Testing Security Smart Contract Vulnerability

Andrea Nováková

You May Also Like

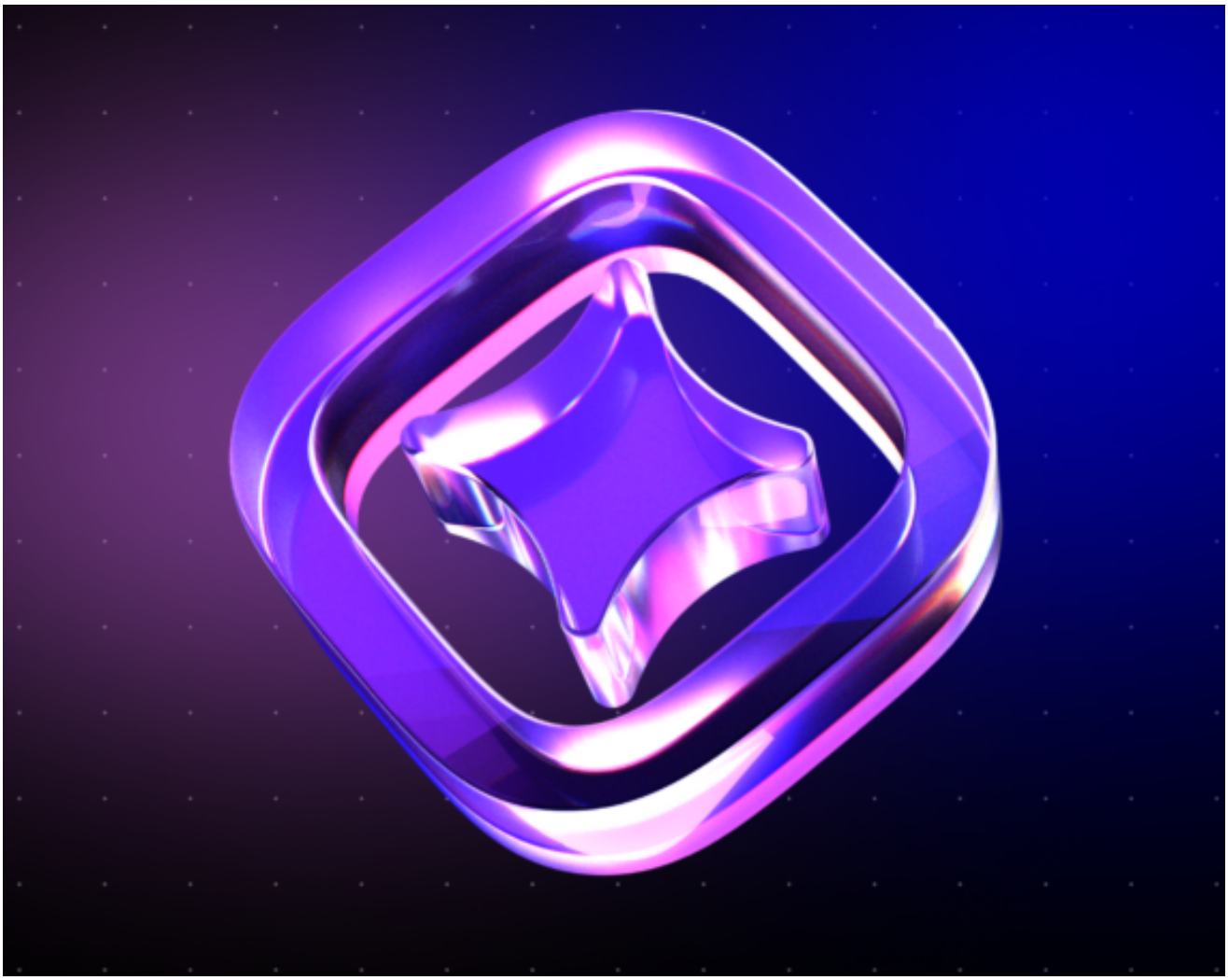
Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF



// AUDITS ETHEREUM WAKE AUGUST 9, 2024

Lido Simple Delegation Audit Summary

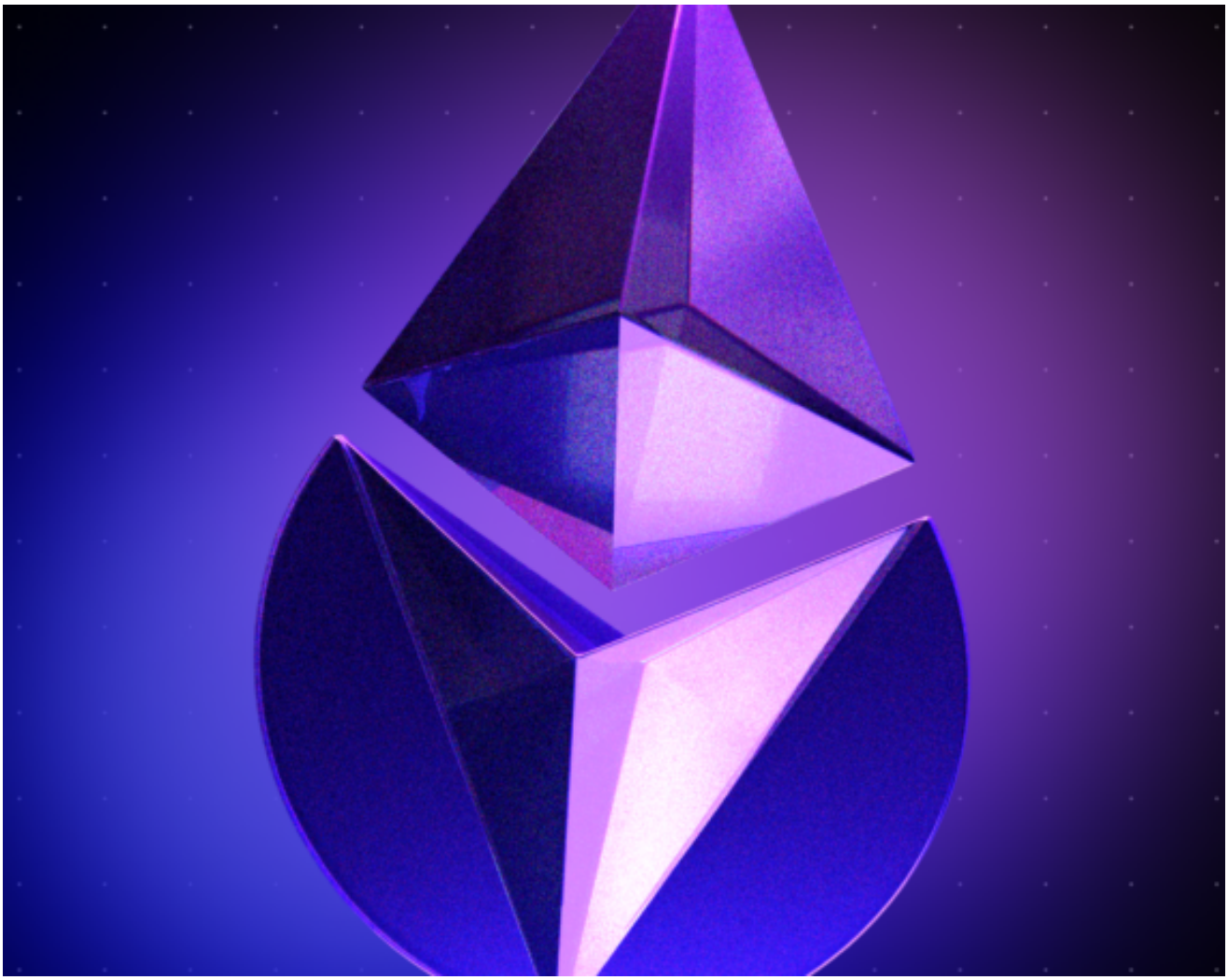
Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF



// AUDITS ETHEREUM WAKE AUGUST 2, 2024

Rhinestone Module Registry Audit Summary

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF



// AUDITS ETHEREUM WAKE JULY 29, 2024

Lido stETH on Optimism Audit Summary

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF