Security Assessment & Formal Verification Report

Lido Dual Governance v1.0.1 Hotfix

August 2025

Prepared for Lido

certora LIDO

Table of Contents

Project Summary

Project Scope

Project Name	Repository (link)	Latest Commit Hash	Platform
Lido Dual Governance v1.0.1 Hotfix	Github Link	0d31f5b	EVM

Project Overview

This document describes the specification and verification of **Lido Dual Governance v1.0.1 Hotfix** using the Certora Prover and manual code review findings. The work was undertaken on **Aug 3, 2025**.

The scope contains all changes to smart contracts within the specified commit 0d31f5b.

The Certora Prover demonstrated that the implementation of the **Solidity** contracts above is correct with respect to the formal rules written by the Certora team. In addition, the team performed a manual audit of all the Solidity contracts**.** During the verification process and the manual audit, the Certora team discovered bugs in the Solidity contracts code, as listed on the following page.

Protocol Overview

Dual Governance is a governance subsystem positioned between the Lido DAO (represented by various voting systems) and the protocol contracts it manages. Its purpose is to protect protocol users from hostile actions by the DAO, allowing them to cooperate and block any in-scope governance decision until the DAO either cancels the decision or all vetoers (w)stETH is withdrawn to ETH.

Detailed Findings

Audit Goals

1. Verify that the fix for the bug of calling the startRageQuitExtensionPeriod function is correct and prevents this attack vector.
2. Verify that the rest of the code is not affected by this fix.
3. Verify that the code does not have any other bugs.

Coverage and Conclusions

1. The fix prevents the issue completely by checking that the delay start time has not been set yet, and if so, it reverts
2. The fix is very limited (3 lines of code) and affects only the flow of calling the startRageQuitExtensionPeriod function multiple times, so it does not affect the happy flow and prevents the bad flow.
3. We did not find any other bugs in the code.

Formal Verification

Verification Notations

Formally Verified	The rule is verified for every state of the contract(s), under the assumptions of the scope/requirements in the rule.
Formally Verified After Fix	The rule was violated due to an issue in the code and was successfully verified after fixing the issue
Violated	A counter-example exists that violates one of the assertions of the rule.

Formal Verification Properties

Escrow

P-01. Finality of rage quit extension period starting time

Status: Verified

Rule Name	Status	Description	Link to rule report
finalityOfExtensionPeriodStartedAt	Verified	*This rule verifies that once the field rageQuitExtensionPeriodStartedAt is set, it can not be reset*	Report

P-02. Function startRageQuitExtensionPeriod() can not be called twice

Status: Verified

Rule Name	Status	Description	Link to rule report
canOnlyBeCalledOnce	Verified	*This rule verifies that once the function startRageQuitExtensionPeriod() is called, it reverts on future calls (including after some other function)*	*Report*

P-03. WithdrawETH can not be blocked

Status: Verified

Rule Name	Status	Description	Link to rule report
frontRunWithdrawEth	Verified	*This property ensures that if a user can withdraw he can do so if some other function is called right before*	*Report*

Disclaimer

Even though we hope this information is helpful, we provide no warranty of any kind, explicit or implied. The contents of this report should not be construed as a complete guarantee that the contract is secure in all dimensions. In no event shall Certora or any of its employees be liable for any claim, damages, or other liability, whether in an action of contract, tort, or otherwise, arising from, out of, or in connection with the results reported here.

About Certora

Certora is a Web3 security company that provides industry-leading formal verification tools and smart contract audits. Certora's flagship security product, Certora Prover, is a unique SaaS product that automatically locates even the most rare & hard-to-find bugs on your smart contracts or mathematically proves their absence. The Certora Prover plugs into your standard deployment pipeline. It is helpful for smart contract developers and security researchers during auditing and bug bounties.

Certora also provides services such as auditing, formal verification projects, and incident response.