Security Assessment Report

Guardian Access Control

05/24

Prepared for
Aave

Table of content

Project Summary

Project Scope

Repo Name	Repository	Commits	Compiler version	Platform
aave–delivery –infrastructure	[Github Repository](#)	[d104463](#)	0.8.19	Multiple

Project Overview

This document describes the specification and verification of the Granular-Guardian-Access-Control using manual code review. The work was undertaken from **02 May 2024** to **07 May 2024**.

The following contract list is included in our scope:

- `GranularGuardianAccessControl.sol`
- `IGranularGuardianAccessControl.sol`

The team performed a manual audit of all the Solidity contracts. During the manual audit, the Certora team discovered bugs in the Solidity contracts code, as listed below.

Protocol Overview

The contract GranularGuardianAccessControl functions as a Guardian for aDI. This new contract adds granularity to the aDI guardian executable methods, so that the DAO can better manage aDI with different roles.

Audit Goals

Verify that the above mentioned contract is implemented correctly and have not introduced a security bug.

Coverage

We determined that the code matches the description of the contract, as it allows for a more granular access control to the CROSS_CHAIN_CONTROLLER contract.
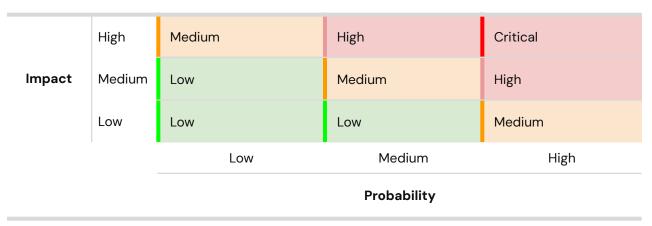
Findings Summary

The table below summarizes the findings of the review, including details on type and severity.

Severity	Discovered	Acknowledged	Code Fixed
Critical			
High			
Medium			
Low	1	yes	6b0370e
Informational			
Total	1		

Severity Matrix

Impact	High	Medium	High	Critical
	Medium	Low	Medium	High
	Low	Low	Low	Medium
		Low	Medium	High

Probability

Detailed Findings

ID	Title	Severity
Low-1	Faulty Role Assigning	Low

Low Severity Issues

Low-1: Faulty Role Assigning

Issue: The retry guardian might not be assigned with the RETRY role.

Description:
The following if statement (GranularGuardianAccessControl.sol, line 44 in the constructor):

```
if (initialGuardians.solveEmergencyGuardian != address(0)) {
  _grantRole(RETRY_ROLE, initialGuardians.retryGuardian);
}
```

checks whether initialGuardians.solveEmergencyGuardian != address(0) while the intention was to check whether initialGuardians.retryGuardian != address(0).
So in case that the field solveEmergencyGuardian (of initialGuardians) is 0, and the field retryGuardian isn't 0, the retryGuardian wouldn't be granted RETRY_ROLE.

Aave's response:
Issue was acknowledged and the conditional was updated to reflect that initialGuardians.retryGuardian != address(0) was checked when assigning the RETRY_ROLE.

Disclaimer

The Certora Prover takes a contract and a specification as input and formally proves that the contract satisfies the specification in all scenarios. Notably, the guarantees of the Certora Prover are scoped to the provided specification and the Certora Prover does not check any cases not covered by the specification.

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.