certora

Security Assessment Report

aave

Collector operational upgrade

Jan-2025

Prepared for:

Aave DAO

Code developed by:

BORED
GHOSTS
DEVELOPING

Table of content

Project Summary

Project Scope

Project Name	Repository (link)	Latest Commit Hash	Platform
Collector operational upgrade	Aave-v3-origin collector-upgrade-rev 6	9e52bd6 cdb6438	EVM/Solidity 0.8

Project Overview

This document describes the specification of the "Collector operational upgrade: multiple funds admin support" manual code review findings. The work was done on 20-21 Jan **2025**.

The scope of our review is on the following contracts:

On Aave-v3-origin:

- src/contracts/treasury/Collector.sol

On collector-upgrade-rev6:

- src/CollectorWithCustomImpl.sol
- src/CollectorWithCustomImplZkSync.sol
- src/UpgradePayload.sol

The team performed a manual audit of all the Solidity contracts**.** During the verification process and the manual audit, no bug was discovered. (Anyhow we have one informational issue that we list below.)

Protocol Overview

The contracts under review are the upgrade and deployment of the new implementation of the Aave Collector contracts across networks, to unify its implementation and enable the finance contributors of the DAO to progress on initiatives like the Finance Stewards.

Coverage

1. With respect to manual auditing we went over the ARFC and the code and we saw that the implementation matches the described intention.
2. **Storage Layout Changes**
    a. In Collector.sol, changes have been made to the order and inclusion of storage properties. By transitioning from VersionedInitializable.sol and ReentrancyGuard.sol to AccessControlUpgradeable.sol and ReentrancyGuardUpgradeable.sol, the *_status*, *lastInitializedRevision*, and *_gap* have been removed. Additionally, on zkSync, the bool private initializing from VersionedInitializable.sol has also been removed.
    b. These modifications result in the storage layout shifting by 53 slots (54 slots on zkSync). To maintain alignment, a new *_gap* has been introduced in Collector.sol. The size of the added *_gap* has been verified, ensuring that the storage layout remains consistent and secure. Most importantly, *_nextStreamId* and *_streams* retain their positions in the same storage slots as before.
        i. On zkSync, because no streams had been added yet, it is possible to configure *_nextStreamId* during initialization to start at 10,000 as it was previously, while utilizing a new storage slot for *_streams*.
    c. We reviewed the Collector instances across all AAVE deployed chains and confirmed that zkSync is the only chain with a different storage layout.
        i. During which we found out the Linea instance is the same as zkSync which means it also needed to be included in the special deployment of CollectorWithCustomImplZkSync.sol.
3. **Shifting from require(...) to revert**
    a. We reviewed the transition from require(...) to revert and ensured that all necessary logical adjustments were implemented. No issues were identified during this review.
4. **Allow multiple Fund Admin entities**
    a. This is possible by using the new logic provided on AccessControlUpgradeable.sol which enables the option to allow multiple admin entities.

Findings Summary

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

Severity	Discovered	Confirmed	Fixed
Critical			
High			
Medium			
Low			
Informational	1		
Total			

Severity Matrix

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

Detailed Findings

ID	Title	Severity	Status
I-01	Linea have the same storage layout as zkSync	Informational	

Informational Severity Issues

I-01. Linea has the same storage layout as zkSync

Description:
The storage layout on Linea and zkSync utilizes an additional storage slot for VersionedInitializable.sol. zkSync was configured with a special implementation during deployment. Upon reviewing all AAVE instances, we identified that in the newly yet to be deployed Linea instance, the initializing variable from VersionedInitializable.sol was also used. This means that during the upgrade, the Linea collector must be included in the special implementation used for zkSync. Due to the fact that Linea is yet to be deployed, even if it were deployed using the regular implementation, the collector would not be adversely affected. The only difference would be that *_nextStreamId* would be initialized to 0 instead of 10,000.

BGD-labs response: Linea will be included in the upgrade proposal and use the same storage layout shift as zksync.

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.