certora

Security Assessment Report

aave

Aave Stewards

August 2025

Prepared for:
Aave DAO

Code developed by TokenLogic

Table of contents

Project Summary

Project Scope

Project Name	Repository (link)	Latest Commit Hash	Platform
Rewards Steward	https://github.com/bgd-labs/aave-stewards	PR#23	Solidity

Project Overview

This document describes the manual code review findings of **Aave Stewards**. The following contract list is included in our scope:

src/finance/RewardsSteward.sol

The work was undertaken from **August 15, 2025,** to **August 18, 2025**. During this time, Certora's security researchers performed a manual audit of all the Solidity contracts**.**

Protocol Overview

RewardsSteward is a minimal Aave steward focused on incentives operations. It lets anyone trigger reward claims on behalf of the DAO's Collector, and sweep incidental assets from the steward to the Collector, while guaranteeing all value flows only to the Collector. It holds no governance power and no custodial state; it simply integrates with RewardsController once set as the authorized claimer. This design enables fast, low-friction rewards ops without requiring new votes for routine actions.

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	-	-	-
Total	-	-	-

Severity Matrix

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

Audit Goals

1. **Integration correctness**
   a. Verify function signatures and argument order match the controller API exactly: claimRewardsOnBehalf(assets, amount, COLLECTOR, COLLECTOR, reward) and claimAllRewardsOnBehalf(assets, COLLECTOR, COLLECTOR), and that observed semantics match controller expectations.
2. **Value-flow and Auth invariant**
   a. Confirm that claims succeed only after setClaimer(COLLECTOR, address(this)) is set in RewardsController, and that both claim paths send rewards exclusively to COLLECTOR with no third-party redirection surface.
3. **Rescue mechanics**
   a. Ensure ERC20 rescue is capped to the contract's actual token balance and uses safeTransfer, and ETH rescue forwards exactly address(this).balance via raw call. Verify both rescue paths send assets exclusively to COLLECTOR with no redirection surface.
4. **Permissionless claiming safety**
   a. Validate that open triggering only affects timing. Confirm Collector's receive is a no-op, streams are created only via createStream by a funds admin, incoming rewards do not create or mutate streams, and streamId increments only on explicit creation.

Coverage and Conclusions

1. **Integration correctness**
    a. Signatures and argument order match the controller definitions : claimRewardsOnBehalf(assets, amount, COLLECTOR, COLLECTOR, reward) and claimAllRewardsOnBehalf(assets, COLLECTOR, COLLECTOR). Semantics align with controller expectations and were written without parameter mismatch.
2. **Value-flow and Auth invariant**
    a. Claims succeed only when setClaimer(COLLECTOR, address(this)) is set in RewardsController. Both claim paths direct rewards to COLLECTOR. There is no call surface that redirects value to third parties.
3. **Rescue mechanics**
    a. ERC20 rescue caps to the contract's actual token balance and uses safeTransfer. ETH rescue forwards exactly address(this).balance via a raw call. Both rescue paths direct assets to COLLECTOR only.
4. **Permissionless claiming safety**

Open triggering changes only timing. Collector's receive does nothing, and stream state changes occur only through createStream by a funds admin. Incoming rewards do not create or mutate streams, and the streamId counter increments only when explicitly instructed.

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.