certora

Security Assessment Report

Aave Aptos Periphery

April 2025

Prepared for Aave Labs

certora

Table of content

certora

Project Summary

Project Scope

Project Name	Repository (link)	Latest Commit Hash	Platform
Aave Aptos	https://github.com/aave/aptos-v3	aaa6570	Aptos

Project Overview

This document describes the manual code review and findings of **Aave V3 on Aptos Periphery**. The work was undertaken from **Feb 2, 2025** to **Apr 7, 2025**

The following contract list is included in our scope:

`aptos-v3/aave-core/sources/aave-periphery/*`

The team performed a manual audit of all the Aptos smart contracts. During the manual audit, the Certora team discovered bugs in the Aptos smart contracts code, as listed on the following page.

Protocol Overview

Aave V3 on Aptos is a decentralized, non-custodial liquidity protocol built on the Aptos blockchain using the Move language, adapted from its original Ethereum implementation. At its core, Aave enables users to participate as liquidity providers by depositing assets into lending pools, or as borrowers who can take out loans using their deposited assets as collateral.

The periphery is a collection of contracts that extends and enhances the core functionality with components like a protocol fee collector, rewards distributor, data provider and coin migrator.

This Aptos implementation leverages Move's resource-oriented programming model and native security features while maintaining the same economic mechanics and risk parameters as the Ethereum version.

Security Considerations

Aave Aptos is an Aave V3 core and periphery code adapted from EVM (Solidity) to Aptos (Move language). The goal of this project is to deploy Aave V3 on Aptos as the first non-EVM instance of the protocol. Therefore, this code was designed to behave exactly like the EVM version, containing the same logic and flows. Since the framework and programming languages are fundamentally different, there are expected differences in the implementations as well as potential issues that may emerge.
Bearing that in mind, we conducted a thorough audit of the entire codebase, constantly comparing it with the EVM version to ensure the logic flow remained identical. We also inspected the flow and underlying assumptions from a fresh perspective of the new framework and environment on Aptos.
Note that any issues or design choices that were reported and discussed in the past on the EVM version aren't mentioned here. However, these were fully considered in the audit itself to ensure that no assumptions and limitations were further broken when moving to the new framework.

Audit Goals

1. Verify that the initialization of modules is done correctly and adheres to the equivalent initializations and configurations in EVM.

2. Verify that reward-related modules (*rewards_controller* and *emission_manager*) properly calculate, track, and distribute incentives with mathematical accuracy despite platform differences.

3. Verify that the admin-controlled ecosystem reserve is properly protected with appropriate access controls.

4. Coin conversion mechanisms between native Aptos Coin and Fungible Asset implementations should maintain precision and handle edge cases.

5. Function visibility and access control is implemented correctly using the Aptos framework.

6. Ensure UI data providers correctly report protocol state with appropriate error handling when querying missing data.

7. All numerical casting operations should maintain mathematical integrity and do not introduce vulnerabilities through truncation, overflow, or precision loss, especially in high-value scenarios.

8. Validate that the Move resource-oriented programming model correctly implements the same asset handling and transfer logic as Solidity's account-based model, with no funds being unaccounted for.

Coverage and Conclusions

1. Module initializations are done correctly and correspond to its equivalent EVM functionality.

2. The rewards controller properly tracks and calculates accrued rewards with mathematically compatible implementations that closely mirror the EVM version's functionality despite language and platform differences.

3. The admin-controlled ecosystem reserve implements appropriate access controls through ACL management, ensuring only authorized funds admins can transfer assets out of the reserve.

4. The coin migration functionality correctly handles conversion between native Aptos Coin and Fungible Asset implementations, with proper event emission and balance tracking.

5. Function visibility and access control has been implemented correctly using the Aptos framework.

6. UI data providers correctly implement view functions that query protocol state, properly handling missing data cases:
    a) Reward information is accurately presented with appropriate fallbacks for unconfigured assets,
    b) User accrued rewards are correctly calculated including both stored and pending amounts,
    c) Asset prices and other economic parameters are correctly integrated into the data presentation.

7. All numerical casting operations maintain mathematical integrity and do not introduce vulnerabilities.

8. The Move resource-oriented model successfully translates Solidity's account-based asset handling:
    a. All asset transfers maintain precise accounting with no untracked funds.
    b. The implementation properly handles Move's capability-based security model while achieving equivalent functionality to the EVM version.
    c. Resource ownership and transfer semantics correctly match the authorization flows from the original Solidity implementation.

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		

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.