Security Assessment & Formal Verification Report

Risk Steward

June-2024

Prepared for
AAVE

Table of content

Project Summary

Project Scope

Project Name	Repository (link)	Latest Commit Hash	Platform
Risk-Steward	aave-v3-risk-steward	6ca003d	EVM/Solidity 0.8

Project Overview

This document describes the specification and verification of the **Risk-Steward** using the Certora Prover and manual code review findings. The work was undertaken from **5 June 2024 to 20 June 2024**.

The following contract list is included in our scope:

- RiskSteward
- IRiskSteward

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, no bug was discovered.

Protocol Overview

The RiskSteward is a smart contract to which the Aave Governance gives POOL_ADMIN the role over all v3 instances, controlled by a 2-of-2 multisig of the risk providers, and heavily constrained on what can do and how by its own logic. More specifically the following risk params could be changed by the RiskStewards: Supply Caps, Borrow Caps, LTV, Liquidation Threshold, Liquidation Bonus, Debt Ceiling, Base variable borrow rate, Slope 1, Slope 2, Optimal point, and Cap parameters for PriceCapAdapters (CAPO).

Coverage

1. We wrote several rules in order to check the validity of the various update functions of the contract.
2. With respect to manual auditing we have checked the following:
    a. All the functions have the correct visibility and correct modifiers.
    b. Checked that the _updateWithinAllowedRange function and all the places which calls it work as intended.
    c. Checked that the Risk Steward will not be allowed to set the values of supply cap, borrow cap, debt ceiling, LTV, Liquidation Threshold, Liquidation Bonus to 0 no matter if the maxPercentChange has been configured to 100% or more.
    d. Checked that no asset parameter can be configured more than once every minDelay time.

Findings Summary

We didn't find any bugs/issues with the Risk Steward.

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

In the table below we specify all the formally verified rules that we wrote for the verification of the risk-steward, and give a detailed description for them. A link to the Certora's prover report can be found here.

P-01. updateCaps_validity

Status: Verified	Property Assumptions:

Rule Name	Status	Description	Rule Assumptions
updateCaps_validity	Verified	*The rule checks that:* *1. After a successed call to updateCaps(..), the fields supplyCapLastUpdated and borrowCapLastUpdated (of the struct Debounce) get the value of current timestamp.* *2. The function AaveV3ConfigEngine.updateCaps(..) is called.*	*The size of the array passed to the function updateCaps(..) is at most of size 2.*

P-02. updateRates_validity

Status: Verified	Property Assumptions:

Rule Name	Status	Description	Rule Assumptions
updateRates_validity	Verified	*The rule checks that:* *1. After a successed call to updateRates(..), the fields optimalUsageRatio, baseVariableBorrowRate, variableRateSlope1, and variableRateSlope2 (of the struct Debounce) get the value of current timestamp.* *2. The function AaveV3ConfigEngine.updateRateStrategies(..) is called.*	*The size of the array passed to the function updateRates(..) is at most of size 2.*

P-03. updateCollateralSide_validity

Status: Verified	Property Assumptions:

Rule Name	Status	Description	Rule Assumptions
updateCollateralSide_validity	Verified	*The rule checks that:* *1. After a successed call to updateCollateralSide(..), the fields ltvLastUpdated, liquidationThresholdLastUpdated, liquidationBonusLastUpdated, and debtCeilingLastUpdated (of the struct Debounce) get the value of current timestamp.* *2. The function AaveV3ConfigEngine.updateCollateralSide(..) is called.*	*The size of the array passed to the function updateCollateralSide(..) is at most of size 2.*

P-04. updateLstPriceCaps_validity

Status: Verified	Property Assumptions:

Rule Name	Status	Description	Rule Assumptions
updateLstPriceCaps_validity	Verified	*The rule checks that:* *1. After a successed call to updateLstPriceCaps(..), the field priceCapLastUpdated (of the struct Debounce) gets the value of current timestamp.* *2. The function IPriceCapAdapter.setCapParameters(..) is called.*	*The size of the array passed to the function updateLstPriceCaps(..) is at most of size 2.*

P-05. updateStablePriceCaps_validity

Status: Verified	Property Assumptions:

Rule Name	Status	Description	Rule Assumptions
updateStablePriceCaps_validity	Verified	*The rule checks that:* *1. After a successed call to updateStablePriceCaps(..), the field priceCapLastUpdated (of the struct Debounce) gets the value of current timestamp.* *2. The function IPriceCapAdapterStable.setPriceCap(..) is called.*	*The size of the array passed to the function updateStablePriceCaps(..) is at most of size 2.*

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.