CERTORA

Aave PR #820 (fix: zero ltv transfer) Review

Fix Overview

This change includes two updates to the current behaviour of Aave protocol v3. First, it prevents users from enabling an asset with 0 LTV as collateral. Second, when receiving an asset in isolation, it will no longer be enabled automatically as collateral, with the exception of suppliers who assign themselves with a new role called `ISOLATED_COLLATERAL_SUPPLIER` for which the old behaviour will still hold.

Scope

This PR introduces changes to the following contracts:

- Pool.sol
- BridgeLogic.sol
- LiquidationLogic.sol
- SupplyLogic.sol
- ValidationLogic.sol

Manual Review Goals

During this work, we examined the following properties via manual review:

1. For a listed asset with 0 LTV, any operation that increases a user balance of that asset must not enable the asset as collateral if it wasn't enabled as collateral before.

2. A user that holds an asset with 0 LTV can only enable that asset as collateral if it was already enabled as collateral.

3. For a listed asset in isolation with LTV greater than 0, any operation that increases a user balance of that asset must not enable the asset as collateral unless the user initiating the operation has an `ISOLATED_COLLATERAL_SUPPLIER` role.

4. For a listed asset in isolation with LTV greater than 0, any operation that increases a user balance of that asset must enable the asset as collateral if the receiving user has no other assets and the user initiating the operation has an `ISOLATED_COLLATERAL_SUPPLIER` role.

5. For a listed asset with LTV greater than 0, any operation that increases a user balance of that asset must enable the asset as collateral if either the asset is not in isolation or if there are no other assets used as collateral by the user.

Conclusions

In the previous version, when validating the use of an asset as collateral, the validation only failed if the asset was in isolation and the user held other assets as collateral. This version strengthens this validation, triggering validation failure when either the asset has 0 LTV for all users or when the asset is in isolation for users who don't have the `ISOLATED_COLLATERAL_SUPPLIER` role.

In more detail:

1. Pool.sol's public functions, which increase the user reserve balance of some assets are:

- `liquidationCall()`
- `finalizeTransfer()`
- `mintUnbacked()`
- `supply()`
- `supplyWithPermit()`
- `deposit()` (Deprecated)

2. All of the functions above call `setUsingAsCollateral()` with `usingAsCollateral = true`, which enables the asset as collateral in the user configuration.

3. All the functions above validate that `validateUseAsCollateral()` returns `true` before calling `setUsingAsCollateral()`. In the previous version, `validateUseAsCollateral()` returned `true` if either (1) the user had no assets enabled as collateral, or (2) if the user wasn't in isolation mode and the asset wasn't in isolation. This PR adds an additional constraint that the asset doesn't have 0 LTV.

4. This PR also introduces a new function called `validateAutomaticUseAsCollateral`. This function returns `false` if the asset is in isolation and if the user doesn't have the `ISOLATED_COLLATERAL_SUPPLIER` role. Otherwise, it falls back to `validateUseAsCollateral`.

5. In the new PR, all functions which increase user reserve balance check that `validateAutomaticUseAsCollateral` returns `true` before they call `setUsingAsCollateral`.

6. Following the previous bullets, we conclude that goals No. 1, 3, 4, and 5 are addressed correctly in the PR for the functions which increase user balance.

7. When calling `setUserUseReserveAsCollateral()`, user collateral is validated through `validateUseAsCollateral()`. This allows users to set an isolated asset as collateral just as before. Given this flow, we conclude that goal No. 2 is addressed correctly for `setUserUseReserveAsCollateral()`.

Issues

Severity: Low

Issue:	Front running a governance decision to set an asset LTV to 0
Description:	From the time the protocol decides to set an asset LTV to 0 until the call to `setConfiguration` is processed, a malicious user can transfer any amount of the problematic asset to honest users; by doing so, the asset will automatically be set as collateral. This will prevent those users from calling `withdraw`, `finalizeTransfer` or `setUserUseReserveAsCollateral` on their healthy assets until they withdraw the problematic asset.
Mitigation/Fix:	This is a limitation of the Aave governance system and we acknowledge it. In addition, there are situations, and probably will be more in the future, where the change of LTV to 0 would happen not via governance proposal.

Severity: Informational

Issue:	Increasing position health factor by supplying assets with 0 LTV
Description:	Setting LTV to 0 is mostly used when assets are at risk of de-pegging. However, if the asset has a relatively high liquidation threshold, users are still incentivised to supply more of it to increase their average liquidation threshold and therefore increase their health factor. This further exposes Aave to the risky asset where the initial intention was to reduce exposure.
Mitigation/Fix:	This is a broader topic, related with providing liquidity of assets with LTV 0 and their impact on the protocol. We consider it out of this scope.