yAudit

June 2026

Prepared for
Twyne

Audited by
ret2basic
adriro

Twyne 1.0.6 Update

Smart Contract Security Assessment

Contents

1 Review Summary

1.1 Protocol Overview

Twyne is a credit delegation protocol that lets borrowers rent unused borrowing power from other lenders to boost their Liquidation LTV. Lenders earn additional yield while borrowers get to ramp up their leverage or insulate their debt.

1.2 Audit Scope

This audit covers the changes between version 1.0.5 and 1.0.6 for the Twyne repository, and PR #9 for the aToken Wrapper repository, across half a day of review. The changeset in scope includes the addition of an admin for the Twyne contracts, and an admin and pause guardian for the aToken Wrapper contract.

```
├── aave-v3-aToken-wrapper
│   └── src
│       └── AaveV3ATokenWrapper.sol
└── twyne-contracts
    └── src
        ├── TwyneFactory
        │   └── CollateralVaultFactory.sol
        ├── interfaces
        │   ├── IErrors.sol
        │   └── IEvents.sol
        └── twyne
            └── VaultManager.sol
```

1.3 Risk Assessment Framework

1.3.1 Severity Classification

Severity	Description	Potential Impact
Critical	Immediate threat to user funds or protocol integrity	Direct loss of funds, protocol compromise
High	Significant security risk requiring urgent attention	Potential fund loss, major functionality disruption
Medium	Important issue that should be addressed	Limited fund risk, functionality concerns
Low	Minor issue with minimal impact	Best practice violations, minor inefficiencies
Undetermined	Findings whose impact could not be fully assessed within the time constraints of the engagement. These issues may range from low to critical severity, and although their exact consequences remain uncertain, they present a sufficient potential risk to warrant attention and remediation.	Varies based on actual severity
Gas	Findings that can improve the gas efficiency of the contracts.	Increased transaction costs
Informational	Code quality and best practice recommendations	Reduced maintainability and readability

Table 1: severity classification

1.4 Key Findings

Breakdown of Finding Impacts

Impact Level	Count
Critical	0
High	0
Medium	0
Low	0
Informational	2

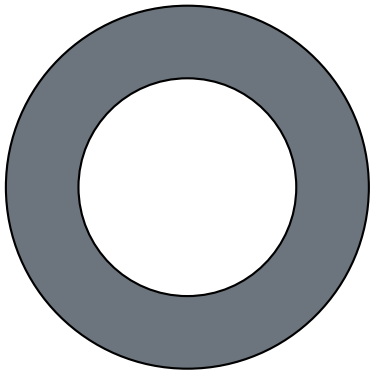

Figure 1: Distribution of security findings by impact level

1.5 Overall Assessment

The Twyne 1.0.6 update is a narrowly scoped administrative change that did not reveal any severe findings during the review. The two informational issues were promptly acknowledged by the Twyne team.

2 Audit Overview

2.1 Project Information

Protocol Name: Twyne
Repositories:

- https://github.com/0xTwyne/twyne-contracts
- https://github.com/0xTwyne/aave-v3-aToken-wrapper

Commit Hashes:

- Twyne contracts: 501185b00b83bf876a18aa875fcd099d366e0aa9
- aToken wrapper: 706aeac7fb8ad0fd4b03bca10fb3d5a93202c02a

Final Commit Hashes:

- Twyne contracts (public repository): 0c1ff9db684bea4fc6bf6fd31a3c90f7c787a1e7
- aToken wrapper: 797be33e22cce033f6d9c66650d9604ee48fc60d

2.2 Audit Timeline

The audit was conducted from June 15 to 15, 2026.

2.3 Audit Resources

- Code repositories and documentation

2.4 Critical Findings

None.

2.5 High Findings

None.

2.6 Medium Findings

None.

2.7 Low Findings

None.

2.8 Gas Savings Findings

None.

2.9 Informational Findings

2.9.1 VaultManager admin checks do not resolve EVC senders

`VaultManager.onlyCollateralVaultFactoryOrAdmin()` compares raw `msg.sender` against `admin` and `collateralVaultFactory`. Calls routed through the EVC on behalf of an authenticated account will therefore fail, particularly for the admin case, as the factory is expected to call directly.

Technical Details

`CollateralVaultFactory` inherits `EVCUtil` and overrides `_msgSender()` to resolve the current EVC account. This makes the factory EVC-aware, but the new `VaultManager.onlyCollateralVaultFactoryOrAdmin()` modifier uses `msg.sender` instead of resolving the caller with the EVC helper.
The result is inconsistent caller handling between the factory and the manager. When a privileged call reaches `VaultManager` through the EVC context, the raw sender is the EVC rather than the authenticated account.

Impact

Informational.

Recommendation

Consider using `_msgSender()` instead of `msg.sender` in `onlyCollateralVaultFactoryOrAdmin()`.

Developer Response

Acknowledged. VaultManager isn't needed or expected to be called from EVC.

2.9.2 Revert errors still refer to owner after admin split

`VaultManager`, `CollateralVaultFactory`, and `AaveV3ATokenWrapper` gate calls with the operational `admin` role, but some custom errors still refer to `owner`. Failed access checks therefore report stale role names, which can confuse integrations, monitoring, and tests that use revert data to classify failures.

Technical Details

In `VaultManager.sol`, `onlyCollateralVaultFactoryOrAdmin()` allows `admin` or `collateralVaultFactory`, but it reverts with `CallerNotOwnerOrCollateralVaultFactory()`. The `owner` role is now distinct from `admin`, since `setAdmin()` lets the owner delegate operational permissions to another account.
In `CollateralVaultFactory.sol`, `pause()` allows `admin` or `pauseGuardian`, but it reverts with `CallerNotOwnerOrPauseGuardian()`. This has the same stale owner terminology after the admin split.
In `AaveV3ATokenWrapper.sol`, `pause()` also allows `admin` or `pauseGuardian`, but it reverts with `CallerNotOwnerOrPauseGuardian()`. The wrapper initializes `admin` and `pauseGuardian` to the owner, but `setAdmin()` can later move operational permissions away from the owner.

Impact

Informational.

Recommendation

Rename the errors to match the enforced roles, such as `CallerNotAdminOrCollateralVaultFactory()` and `CallerNotAdminOrPauseGuardian()`. Changing a custom error name changes its 4-byte selector, so make the rename only if selector compatibility is acceptable across the Twyne contracts and the aToken wrapper, or keep the existing selectors and document the legacy names until a coordinated breaking release.

Developer Response

Acknowledged.

Twyne 1.0.6 Update

Completed 2026-06-15