yAudit

December 2025

Prepared for
Twyne

Audited by
Adriro
HHK

Twyne liquidation incentive

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 2 pull requests totaling approximately 250~ lines of code across 1 day of review.

```
0xTwyne/twyne-contracts/src
└── twyne
    ├── AaveV3CollateralVault.sol
    ├── CollateralVaultBase.sol
    └── EulerCollateralVault.sol
```

1.3 Risk Assessment Framework

1.3.1 Severity Classification

1.4 Key Findings

Breakdown of Finding Impacts

Impact Level	Count
Critical	0
High	0
Medium	0
Low	1
Informational	1

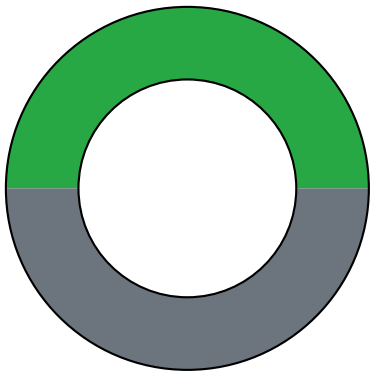

Figure 1: Distribution of security findings by impact level

1.5 Overall Assessment

2 Audit Overview

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

2.1 Project Information

Protocol Name: Twyne
Repository: https://github.com/0xTwyne/twyne-contracts
Commit Hashes:

- PR#187
- PR#220

Commit URLs:

- PR#187
- PR#220

2.2 Audit Team

Adriro, HHK

2.3 Audit Timeline

The audit was conducted from December 15 to 15, 2025.

2.4 Audit Resources

Code repositories and documentation

2.5 Critical Findings

None.

2.6 High Findings

None.

2.7 Medium Findings

None.

2.8 Low Findings

2.8.1 Missing slippage protection for liquidators

Technical Details

PR#187 introduces partial refunds for borrowers during liquidation. The refund amount depends on position health: the closer to external liquidation, the lower the refund, while liquidator incentives increase as positions become riskier for intermediate vault LPs. Liquidators must now refund borrowers a portion of the collateral, but there is no function argument that allows liquidators to set a maximum acceptable refund amount.

Impact

Low. Liquidators cannot protect against potentially unfavourable refund amounts.

Recommendation

Add a `uint256 maxBorrowerRefund` argument to `liquidate()` and revert if the computed refund exceeds it.

Developer Response

Acknowledged. We expect the liquidator to do these checks outside Twyne's core protocol.

2.9 Gas Savings Findings

None.

2.10 Informational Findings

2.10.1 Use `_getExtLiqLTV()` in `EulerCollateralVault::_invariantCollateralAmount()`

Technical Details

The implementation of `_invariantCollateralAmount()` can call `_getExtLiqLTV()` instead of repeating the expression.

Impact

Informational.

Recommendation

Use `_getExtLiqLTV()` to retrieve the liquidation LTV of Euler's target vault.

Developer Response

Acknowledged.

2.11 Final Remarks

This review focused on PR#187 of the Twyne codebase. The PR updates the collateral vaults to modify the incentive for liquidators. Instead of receiving the full collateral from the borrower, they now receive an amount based on the vault's condition. The closer to external liquidation, the higher the reward, according to a specific equation described in the whitepaper. The review did not result in any medium- or higher-severity findings, highlighting the seriousness of the Twyne team.
Towards the end of the review, PR#220 was submitted and included in the audit scope. It adds a short-circuit to `splitCollateralAfterExtLiq()` when `_maxRepay` is 0 as well as extra comments describing the computation. No new issues were found by the auditors.
The changes in PR#187 and PR#220 have been reviewed up to commit `9b2fcb3184795739431c749f4b24e5f18ebe07e8`.