Yieldoor Audit Report

Prepared by 0xSimao

Version 2.0

November 30, 2025

Contents

1 About 0xSimao

0xSimao is an indepedent security researcher, #2 on Sherlock, top-ranked on Cantina and Code4rena, and a member of Blackthorn, a leading auditing firm. Previously, he served as Head of Security at Three Sigma.

0xSimao has placed in the Top 3 in 28 public audits and has led over 60 private engagements. For private audits or collaboration opportunities, feel free to reach out to him on X (@0xSimao), Telegram (@OxSimao), or Discord (@0xSimao).

2 Disclaimer

0xSimao makes every effort to find as many vulnerabilities in the code as possible in the given time but holds no responsibility for the findings in this document. A security audit by the team does not endorse the underlying business or product. The audit was time-boxed and the review of the code was solely on the security aspects of the solidity implementation of the contracts.

3 Risk Classification

	Impact: High	**Impact: Medium**	**Impact: Low**
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

4 Protocol Summary

Loop Vaults are designed for users who want to earn high fixed yields by leveraging interest rate arbitrage in a fully automated and capital-efficient manner. Built for stablecoin holders, these vaults harness advanced DeFi primitives like Pendle's PT tokens and Morpho's lending markets to generate leveraged returns—without requiring users to manage positions manually.

5 Audit Scope

The scope was the migration of the code to libraries and the rebalancing functionality.

6 Executive Summary

Over the course of 2 days, 0xSimao conducted an audit on the Yieldoor smart contracts provided by Yieldoor. In this period, a total of 5 issues were found.

Summary

Project Name	Yieldoor
Repository	loopedVault
Commit	753ba227a910. . .
Fix Commit	940648c0c27a. . .
Audit Timeline	June 20th - June 23rd
Methods	Manual Review, Stateful Fuzzing

Issues Found

Critical Risk	0
High Risk	0
Medium Risk	2
Low Risk	2
Informational	1
Gas Optimizations	0
Total Issues	5

Summary of Findings

[M-1] Vesting interest is not reset to 0 in case there is no interest in LoopedVault11::updateTotalAssets()	Resolved
[M-2] Code has typos in RebalanceLogic::repayAavePosition() leading to undefined behaviour	Resolved
[L-1] RebalanceLogic::rebalanceCallback() doesn't update lastTotalAssets when rebalancing from morpho to morpho	Resolved
[L-2] Inconsistent updateTotalAssets modifier usage	Resolved
[I-1] Duplicated approval in RebalanceLogic::rebalanceCallback()	Resolved

7 Findings

7.1 Medium Risk

7.1.1 Vesting interest is not reset to 0 in case there is no interest in LoopedVault11::updateTotalAssets()

Description: `LoopedVault1_1::_updateTotalAssets()` doesn't set to 0 vesting interest when `vestingInterestPreFee` is null, so the new period will be vesting with the interest from the previous period. This is highly unlikely given that some interest should always accrue.

Recommended Mitigation: Set the vesting interest to 0 when the new interest is 0.

0xSimao:

Fixed in #ad00401.

7.1.2 Code has typos in RebalanceLogic::repayAavePosition() leading to undefined behaviour

Description: `RebalanceLogic::repayAavePosition()` code is switched and the collateral or debt amounts intended to be dealt with will not be the actual amounts used. Due to several checks within the code it is unlikely it would still go through and cause any issues, other than the collateral and borrowed amounts not being correct. Technically it could be possible to borrow too much and be liquidated or similar, but very unlikely.

Recommended Mitigation: Fix the correct collateral/debt variables and the limiting of the amount to withdraw.

0xSimao:

Fixed in #ad00401.

7.2 Low Risk

7.2.1 RebalanceLogic::rebalanceCallback() doesn't update lastTotalAssets when rebalancing from morpho to morpho

Description: All actions that instantly change the value of the vault, namely slippage from swaps, are taken into account by also reducing `lastTotalAssets` resulting in the next vesting amount being unaffected. However, this is not performed when rebalancing morpho to morpho, which does a swap and changes the value of the vault.

Recommended Mitigation: Adjust `lastTotalAssets` when rebalancing morpho to morpho. Aave to morpho or morpho to aave shouldn't change the value of the vault.

0xSimao:

Fixed in #ad00401.

7.2.2 Inconsistent updateTotalAssets modifier usage

Description: `LoopedVaultV1_1::rebalance()` doesn't have the `updateTotalAssets` modifier, contrarily to the rest of the functions. Due to the fact that the real new total assets and the stored lastTotalAssets change in the same amount, triggering the vesting period update or not has no impact in the instant total assets, as `newVestingInterest = newTotalAssets - lastTotalAssets` is the same (minus some negligible rounding), and the vesting amount is exactly the same.

For deposits/withdrawals there is also no difference, except that users want to vest as soon as they deposit.

Recommended Mitigation: Keep it consistent between functions.

0xSimao:

Fixed in #940648c.

7.3 Informational

7.3.1 Duplicated approval in RebalanceLogic::rebalanceCallback()

Description: `RebalanceLogic::rebalanceCallback()` approves morpho twice, 1st and 2nd.

Recommended Mitigation: Remove the 2nd approval (or 1st).

0xSimao:

Fixed in #ad00401.