Morpho - `fixed-rate-irm`

Security Review

Cantina Managed review by:

Emanuele Ricci, Lead Security Researcher

Jonah1005, Lead Security Researcher

March 11, 2024

Contents

1 Introduction

1.1 About Cantina

Cantina is a security services marketplace that connects top security researchers and solutions with clients. Learn more at cantina.xyz

1.2 Disclaimer

Cantina Managed provides a detailed evaluation of the security posture of the code at a particular moment based on the information available at the time of the review. While Cantina Managed endeavors to identify and disclose all potential security issues, it cannot guarantee that every vulnerability will be detected or that the code will be entirely secure against all possible attacks. The assessment is conducted based on the specific commit and version of the code provided. Any subsequent modifications to the code may introduce new vulnerabilities that were absent during the initial review. Therefore, any changes made to the code require a new security review to ensure that the code remains secure. Please be advised that the Cantina Managed security review is not a replacement for continuous security measures such as penetration testing, vulnerability scanning, and regular code reviews.

1.3 Risk assessment

Severity	Description
Critical	*Must* fix as soon as possible (if already deployed).
High	Leads to a loss of a significant portion (>10%) of assets in the protocol, or significant harm to a majority of users.
Medium	Global losses <10% or losses to only a subset of users, but still unacceptable.
Low	Losses will be annoying but bearable. Applies to things like griefing attacks that can be easily repaired or even gas inefficiencies.
Gas Optimization	Suggestions around gas saving practices.
Informational	Suggestions around best practices or readability.

1.3.1 Severity Classification

The severity of security issues found during the security review is categorized based on the above table. Critical findings have a high likelihood of being exploited and must be addressed immediately. High findings are almost certain to occur, easy to perform, or not easy but highly incentivized thus must be fixed as soon as possible.

Medium findings are conditionally possible or incentivized but are still relatively likely to occur and should be addressed. Low findings a rare combination of circumstances to exploit, or offer little to no incentive to exploit but are recommended to be addressed.

Lastly, some findings might represent objective improvements that should be addressed but do not impact the project's overall security (Gas and Informational findings).

2 Security Review Summary

Morpho is a lending pool optimizer. It improves the capital efficiency of positions on existing lending pools by seamlessly matching users peer-to-peer.

Morpho's rates stay between the supply rate and the borrow rate of the pool, reducing the interests paid by the borrowers while increasing the interests earned by the suppliers. It means that you are getting boosted peer-to-peer rates or, in the worst case scenario, the APY of the pool. Morpho also preserves the same experience, liquidity and parameters (collateral factors, oracles, ...) as the underlying pool.

From Feb 19th to Feb 23rd the Cantina team conducted a review of morpho-blue-irm on commit hash 23259f88. The team identified a total of **3** issues in the following risk categories:

- Critical Risk: 0
- High Risk: 0
- Medium Risk: 1
- Low Risk: 0
- Gas Optimizations: 0
- Informational: 2

3 Findings

3.1 Medium Risk

3.1.1 High `borrowRate` values in the new fixed rate IRM could lead to user funds stuck in Morpho Blue markets

Severity: Medium Risk

Context: Morpho.sol#L483-L509, FixedRateIrm.sol#L40-L47

Description: The `FixedRateIrm` allows anyone to set a custom fixed arbitrary `borrowRate` for a specified market. The current implementation of `setBorrowRate` reverts only if the specified `newBorrowRate` value is equal to zero or if the market has been already configured with a non-zero borrow rate. This means that an arbitrary value between 1 and `type(uint256).max` is accepted as a valid `borrowRate` for a market.

While a very low borrow rate has the only consequence that borrowers will pay a tiny interest (and suppliers will earn a very tiny amount for the money lent), a high value of `borrowRate` will make the Morpho Blue operations for such market to always revert when the `_accrueInterest` function is executed. The `_accrueInterest` can revert when one of the following operations is executed:

- `wTaylorCompounded` could revert for overflow.
- `market[id].totalBorrowAssets.wMulDown` could overflow.
- `interest.toUint128()` could revert if `interest > type(uint128.max)`.
- Adding `interest` to `totalBorrowAssets` and `totalSupplyAssets` could revert if `interest` is big enough to make the sum overflow.

Such internal function is always called when one of the following functions is executed:

- `accrueInterest`
- `setFee`
- `supply`
- `withdraw`
- `borrow`
- `repay`
- `withdrawCollateral`
- `liquidate`

The only function that does not call `_accrueInterest` is `supplyCollateral`.

Let's assume that `IIrm(marketParams.irm).borrowRate(marketParams, market[id])` returns a `borrowRate` that makes `_accrueInterest` revert. User funds will be stuck forever in the Morpho Blue market in these cases:

- If a borrower provides collateral by executing `supplyCollateral`. This function never calls `_accrueInterest` and can't revert. The problem in this case is that the borrower won't be able to withdraw his collateral because `withdrawCollateral` will always revert.
- Any funds supplied to the contract at market creation or, in general, when `elapsed == 0`. After that suppliers and borrowers have provided loan/collateral tokens, those tokens will not be able to be withdrawn or borrowed because those transactions will revert when `_accrueInterest` is executed.

Note that MetaMorpho Vaults and Bundlers that interact with these reverting markets will also be influenced by these side effects.

Recommendation: Morpho should document this possible behavior and evaluate to add lower and upper bounds limit to the value that can be specified as the `newBorrowRate` input parameter of `FixedRateIrm.setBorrowRate` to prevent any Morpho Blue market that uses such IRM to revert when `_accrueInterest` is executed.

Morpho: Addressed in PR 128.

Cantina Managed: The recommendations have been implemented in PR 128. The `FixedRateIrm` now has a max borrow rate equal to the max borrow rate used for the `AdaptiveCurveIrm`.

3.2 Informational

3.2.1 Consider pre-configuring known integrators market at deployment time to avoid market-squatting

Severity: Informational

Context: FixedRateIrm.sol#L40-L47

Description: The new `FixedRateIrm` allows anyone to call `setBorrowRate(Id id, uint256 newBorrowRate)` and set the `borrowRate` for a market. Given the permissionless nature of this function, it means that anyone could frontrun a valid `setBorrowRate` call by specifying a possible malicious or useless value of `newBorrowRate`.

This is an issue that is already known to the Morpho team, and it's impossible to solve by also maintaining the permissionless nature of this IRM. Once the `borrowRate` for a market has been set, the only possibility for the market owner is to deploy a new Oracle and generate a new `marketId` to be used.

Morpho has already some Morpho Blue markets created by integrators, and it's possible that these integrators will want to experiment with the new IRM by using the same market parameters. The only solution, to avoid being frontrunned once the IRM has been deployed, is that Morpho batch the deployment of this IRM with a set of `setBorrowRate` execution with the parameters suggested by these integrators.

Recommendation: Morpho should consider bundling the deployment of the new IRM with the execution of a series of `setBorrowRate` if some of the existing integrators want to provide markets with the parameters already used (for the already deployed markets) but with the new IRM.

Morpho: This overhead of the fixed rate irm is acknowledged, for lack of a better solution than the one we have currently (batching oracle creation and rate setting).

Cantina Managed: Acknowledged.

3.2.2 Natspec documentation issues: missed parameters, typos or suggested updates

Severity: Informational

Context: FixedRateIrm.sol

Description: Various NatSpec documentation issues were found, that include missing parameters, typos or that allow general suggestions for improval:

- FixedRateIrm.sol: All the NatSpec documentation should be moved from the `FixedRateIrm` contract to the `IFixedRateIrm` interface. The current natspec can be then replaced with `/// @inheritdoc IFixedRateIrm`.

Recommendation: Morpho should consider fixing all the listed points to provide a better NatSpec documentation.

Morpho: Addressed in PR 128.

Cantina Managed: The recommendations have been implemented in PR 128.