Morpho Adapter Registries Security Review

Auditors

MiloTruck, Lead Security Researcher

Saw-mon and Natalie, Lead Security Researcher

Report prepared by: Lucas Goiriz

October 2, 2025

Contents

1 About Spearbit

Spearbit is a decentralized network of expert security engineers offering reviews and other security related services to Web3 projects with the goal of creating a stronger ecosystem. Our network has experience on every part of the blockchain technology stack, including but not limited to protocol design, smart contracts and the Solidity compiler. Spearbit brings in untapped security talent by enabling expert freelance auditors seeking flexibility to work on interesting projects together.

Learn more about us at spearbit.com

2 Introduction

Morpho is a trustless and efficient lending primitive with permissionless market creation.

Disclaimer: This security review does not guarantee against a hack. It is a snapshot in time of Morpho Adapter Registries according to the specific commit. Any modifications to the code will require a new security review.

3 Risk classification

Severity level	Impact: High	Impact: Medium	Impact: Low
Likelihood: high	Critical	High	Medium
Likelihood: medium	High	Medium	Low
Likelihood: low	Medium	Low	Low

3.1 Impact

- 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.

3.2 Likelihood

- High - almost certain to happen, easy to perform, or not easy but highly incentivized
- Medium - only conditionally possible or incentivized, but still relatively likely
- Low - requires stars to align, or little-to-no incentive

3.3 Action required for severity levels

- Critical - Must fix as soon as possible (if already deployed)
- High - Must fix (before deployment if not already deployed)
- Medium - Should fix
- Low - Could fix

4 Executive Summary

Over the course of 1 days in total, Morpho engaged with Spearbit to review the vault-v2-adapter-registries protocol. In this period of time a total of **3** issues were found.

Summary

Project Name	Morpho
Repository	vault-v2-adapter-registries
Commit	c1d84a55
Type of Project	DeFi, Vaults
Audit Timeline	Sep 7th to Sep 8th

Issues Found

Severity	Count	Fixed	Acknowledged
Critical Risk	0	0	0
High Risk	0	0	0
Medium Risk	0	0	0
Low Risk	1	0	1
Gas Optimizations	0	0	0
Informational	2	1	1
Total	**3**	**1**	**2**

The Spearbit team reviewed Morpho's `vault-v2-adapter-registries` holistically on commit hash dc7d306f and concluded that all findings were addressed and no new vulnerabilities were identified.

5 Findings

5.1 Low Risk

5.1.1 `isInRegistry` does not check whether parent vault was deployed by a designated `VaultV2` factory

Severity: Low Risk

Context: MorphoMarketV1Registry.sol#L18-L21, MorphoVaultV1Registry.sol#L19-L22

Description: Both `isInRegistry` implementations in `MorphoMarketV1Registry` and `MorphoVaultV1Registry` do not check whether the parent vault of the adapter was deployed by a fixed designated `VaultV2` factory.

Recommendation: Introduce an immutable `VaultV2` factory in these contracts and in `isInRegistry` check that the parent vault of the adapter was deployed by that fixed `VaultV2` factory.

Morpho: Acknowledged.

Cantina Managed: Acknowledged.

5.2 Informational

5.2.1 Duplicate

Severity: Informational

Context: RegistryList.sol#L32-L38

Description: In `RegistryList`, `addSubRegistry()` does not check if `subRegistry` already exists in the `subRegistries` list:

```
/// @dev Adding a subRegistry that reverts or makes looping too gas consuming will make new registries
↪  uneffective
/// (vaults will not be able to validate adapters that would be validated by registries that have been
↪  added after).
function addSubRegistry(address subRegistry) external {
    require(msg.sender == owner, "Not owner");
    subRegistries.push(subRegistry);
    emit AddSubRegistry(subRegistry);
}
```

As such, duplicate sub-registries can be added to the `subRegistries` list. However, note that there is no impact apart from `isInRegistry()` possibly consuming more gas.

Recommendation: This behavior could be documented.

Morpho: Acknowledged.

Cantina Managed: Acknowledged.

5.2.2 Sub-registries could break the add-only assumption for adapters

Severity: Informational

Context: RegistryList.sol#L10-L11

Description: The add only assumption needs to be transitively assumed/stated/forced for sub-registries of a registry.

Recommendation: Perhaps it can be added to the documentation.

Morpho: Fixed in commit ce533bfa.

Cantina Managed: Fix verified.