Maker DAO - D3M implementation for Morpho Security Review

Auditors

Christoph Michel, Lead Security Researcher

M4rio.eth, Security Researcher

Shung, Associate Security Researcher

Report prepared by: Lucas Goiriz

March 26, 2024

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

The Maker Protocol, also known as the Multi-Collateral Dai (MCD) system, allows users to generate Dai (a decentralized, unbiased, collateral-backed cryptocurrency soft-pegged to the US Dollar) by leveraging collateral assets approved by the Maker Governance, which is the community organized and operated process of managing the various aspects of the Maker Protocol.

Disclaimer: This security review does not guarantee against a hack. It is a snapshot in time of D3M implementation for Morpho 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 3 days in total, MakerDAO engaged with Spearbit to review the D3M implementation for Morpho protocol. In this period of time a total of **7** issues were found.

Summary

Project Name	MakerDAO
Repository	D3M implementation for Morpho
Commit	698ec5...e9c2b8
Type of Project	DeFi
Audit Timeline	Mar 18 to Mar 25

Issues Found

Severity	Count	Fixed	Acknowledged
Critical Risk	0	0	0
High Risk	0	0	0
Medium Risk	0	0	0
Low Risk	2	0	2
Gas Optimizations	0	0	0
Informational	5	0	5
Total	**7**	**0**	**7**

5 Findings

5.1 Low Risk

5.1.1 `maxDeposit()` deposit DoS on duplicate markets

Severity: Low Risk

Context: MetaMorpho.sol#L799, MetaMorpho.sol#L507

Description: The `MetaMorpho.maxDeposit` function that is used by the pool has the following limitation:

> @dev Warning: May be higher than the actual max deposit due to duplicate markets in the supplyQueue.

Depositing with a `maxDeposit` might revert if the Metamorpho contract is misconfigured.

Recommendation: Ensure there are no duplicate markets in `Metamorpho`'s `supplyQueue`.

MakerDAO: Acknowledged. We are aware of this risk. Care will be taken by the allocator not to add duplicate markets.

Spearbit: Acknowledged by the client.

5.1.2 DAI can be lost if the underlying Metamorpho curator force-removes a market

Severity: Low Risk

Context: MetaMorpho.sol#L353, D3MHub.sol#L291

Description: The Metamorpho curator can force-remove markets. This will lead to a drop in `totalAssets()` as the removed market's deposited DAI balance is lost for all Metamorpho vault share token holders. The lost DAI is still in circulation in the underlying Morpho Blue's market.

Recommendation: Forced removals come with a `timelock` delay. Metamorpho's `submitMarketRemoval` and all markets' `config[id].removableAt` timestamps should be monitored and a max withdrawal should be attempted before the forced market removal (in case the vault allocated tokens to this market).

MakerDAO: Acknowledged. MakerDAO is the owner (+ curator) of the Metamorpho vault. It will be noted that this should not be called unless the position in that market is first unwound via setting the supply cap to 0 and reallocating funds away.

Spearbit: Acknowledged by the client.

5.2 Informational

5.2.1 The `deposit`/`withdraw` function points to Morpho documentation

Severity: Informational

Context: D3M4626TypePool.sol#L104, D3M4626TypePool.sol#L104

Description: We can deduct from the name of the pool that this pool can be used for a general `ERC4626` vault. The pool also implements general `ERC4626` calls, nothing specific to `Morpho` implementation.

The documentation on the `deposit`/`withdraw` function points to Morpho repository while instead it should point to the `ERC4626` EIP definition.

Recommendation: Consider pointing to the EIP-4626 documentation as the `D3M4626TypePool` can be used with any `ERC4626` vault implementation.

MakerDAO: Acknowledged. This will be specific to Morpho for now despite the naming. It's not clear we will even need another D3M.

Spearbit: Acknowledged by the client.

5.2.2 Centralization risk on `D3MOperatorPlan`

Severity: Informational

Context: D3MOperatorPlan.sol#L79

Description: Within the `D3MOperatorPlan` the function `getTargetAssets` returns the targeted DAI supply that should be targeted by the pool.

We can see that in the other `D3M` operators these values are calculated based on onchain values (D3MAaveV2TypeRateTargetPlan.sol#L173) while in the plan for the `D3M4626TypePool` the target assets are set by a guarded function called `setTargetAssets`.

This function resides under `operator` role. If this role is malicious or calls a misconfigured value, then the Hub will be able to wrongfully wind/unwind assets from the pool.

Recommendation: Consider having this under a trusted multisig wallet.

MakerDAO: Acknowledged. The operator role will be under a multisig.

Spearbit: Acknowledged by the client.

5.2.3 The `vat` is not verified if the `hub` changes

Severity: Informational

Context: D3M4626TypePool.sol#L96-L99

Description: A new hub address can be set without any kind of verification to ensure that the associated `VAT` on the new hub is equal to the saved `VAT`. This could allow an arbitrary address to be set as the hub, which posses a security risk because the `Pool` will use a different `VAT` than the `Hub`.

Recommendation: There are two recommendations that can be implemented:

- Prior to the assignment of data to the hub variable, a check should confirm that the `VAT` associated with the new hub is equal to the current `VAT`.
- Consider while setting the new `hub`, a configuration check is done within the script to make sure the `VAT`s are equal.

MakerDAO: Acknowledged. We are careful with the configuration, and this matches the other D3Ms.

Spearbit: Acknowledged by the client.

5.2.4 Missing rewards collection method

Severity: Informational

Context: D3M4626TypePool.sol#L37, UniversalRewardsDistributor.sol#L133

Description: The `D3M4626TypePool` contract is built with Metamorpho contracts in mind. Morpho has a rewards distributor contract. The pool contract might receive reward tokens at some point.

Recommendation: Consider implementing a generic `skim(address token, address recipient) external auth` function to claim reward tokens that can skim any `token != vault` as the pool contract should only hold vault shares.

MakerDAO: Acknowledged. We purposefully excluded rewards as we don't expect them, and if that changes we can update the pool later.

Spearbit: Acknowledged by the client.

5.2.5 Two ways to disable `D3MOperatorPlan`

Severity: Informational

Context: D3MOperatorPlan.sol#L70-L71

Description: The canonical way to disable the `D3MOperatorPlan` is by calling its `disable` function defined as part of the `ID3MPlan` interface and used by the `D3Mom`. The second way to disable it is by calling `file("enabled", 0)`, however, this will not emit the `Disable` event that `disable()` emits.

Recommendation: Consider only allowing `file` to *enable* the `enabled` flag, or emitting the `Disabled` event also in `file`.

MakerDAO: Acknowledged. We are matching the other versions of this.

Spearbit: Acknowledged by the client.