Maker Dao - DSS Conduit

Security Review

Cantina Managed review by:
Christoph Michel, Lead Security Researcher
M4rio.eth, Security Researcher

November 24, 2023

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

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.

From Oct 17th to Oct 18th the Cantina team conducted a review of dss-conduits on commit hash eaf761f5.

***Note**: The original review commit hash was b8b3dac3, but the final commits that were signed off correspond to commit hash eaf761f5 as stated above.*

The team identified a total of **2** issues in the following risk categories:

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

3 Findings

3.1 Informational

3.1.1 Add proper documentation / spec

Severity: Informational

Context: README.md

Description: The documentation in the `README` for the `ArrangerConduit` contract does not reflect the contract's functionality well. It seems to be currently written for a generic `Conduit` but there are some important differences:

1. It is missing a description and the important role of the `arranger`, along with their ability to draw and return funds.
2. It only describes the `deposit` and `withdraw` functionality, but misses the required functionality for subDAOs to request funds via `requestFunds` so they can withdraw.
3. The `withdraw` function is described as:

   > This can pull funds atomically from a yield bearing strategy in the case of DeFi protocols, or can pull the funds directly from the Conduit in the case of a Real World Asset strategy where the permissioned actor has returned the funds manually.

   The `withdraw` function cannot pull funds from other protocols itself. The contract always requires the funds to have already been returned by the permissioned actor prior to the `withdraw` call. Consider removing the first part of the description.

Recommendation: Consider properly documenting the functionality of the contract.

Maker: Addressed in PR 15.

Cantina: Fixed, the arranger conduit is documented in the wiki and a link to the wiki has been added to the README.

3.1.2 Arranger has full control over deposited funds

Severity: Informational

Context: ArrangerConduit.sol#L165-L205

Description: The `ArrangerConduit` contains a special role called `arranger` which acts like a fund manager. This role can draw funds by calling `drawFunds` and then mark them as returned by calling the `returnFunds` function. We must mention that:

- During the draw, the `arranger` can not draw more than the `availableFunds`. The `availableFunds` is the total balance of an asset that the conduit has, without the funds that were marked as `withdrawable`
- During the return of the funds, there is no call that transfers the funds in the contract, we expect for the funds to first be transferred to the conduit, then the `returnFunds` function to be called.
- When calling the return funds function, a `fundRequestId` must be sent, which means that a `fundRequest` call must be performed because the `returnFunds` function requires a `fundRequestId` which needs to be filled with the `returnAmount`.

Various scenarios can happen:

- The arranger does not respond to a fund request or denies a fund request by using a `returnAmount` of `0` (or returns less than requested, `returnAmount < fundRequest.amountRequested`).
- The arranger can also return a `returnAmount` greater than the requested funds, resulting in the subDAO being able to withdraw more funds than they expected.

Recommendation: Our recommendation is to make sure that the `arranger` role is safely guarded as it is in full control of the entire deposited funds. Furthermore, consider documenting the arranger and its trust assumptions in a "Roles/Permissions" or "Security Model" section.

Maker: Acknowledged. We are aware of this. Arranger is a very privileged role and should be treated as such.

Cantina: Acknowledged.

4 Additional Comments

The Cantina team reviewed MakerDao's dss-conduits changes holistically on commit hash 367d0e477fc8da7a61ee53e9ea8d3837667c8ef1 and determined that all issues were resolved and no new issues were identified.