CANTINA

MakerDAO: Spark ALM Controller

Security Review

Cantina Managed review by:
M4rio.eth, Security Researcher
Jonatas Martins, Associate Security Researcher

October 23, 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

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 16th to Oct 18th the Cantina team conducted a review of spark-alm-controller on commit hash e082b200.

The Cantina team reviewed MakerDAO's spark-alm-controller changes holistically on commit hash 6058f68f and determined that all issues were resolved and no new issues were identified.

The team identified a total of **5** issues in the following risk categories:

- Critical Risk: 0
- High Risk: 0
- Medium Risk: 0
- Low Risk: 0
- Gas Optimizations: 0
- Informational: 5

3 Findings

3.1 Informational

3.1.1 Missing domain checks in the initialization

Severity: Informational

Context: ControllerInit.sol#L131 | ControllerInit.sol#L298

Description: The initialization scripts add minters for various CCTP domains but there is no check that those domains are valid.

```
for (uint256 i = 0; i < mintRecipients.length; i++) {
        controller.setMintRecipient(mintRecipients[i].domain, mintRecipients[i].mintRecipient);
}
```

Recommendation: Consider adding a function that checks for a specific domain if it's part of the `CCTPForwarder` domains.

Maker: Acknowledged. For this we will just verify as part of the spell review process.

Cantina Managed: Acknowledged by the client.

3.1.2 Improved admin check in the initialization script for the Controllers

Severity: Informational

Context: See below

Description: Both the `MainnetController` and `ForeignController` initialization scripts include checks to verify if the Proxy and the Rate Limiter have the same admin as the controller.

In the `MainnetController`, these checks are split across two functions: `subDaoInitFull` handles the Proxy and Rate Limiter checks, while `subDaoInitController` takes care of the controller check.

In contrast, the `ForeignController` performs all the checks in one place, as shown below:

```
require(almProxy.hasRole(DEFAULT_ADMIN_ROLE, addresses.admin)   == true,
↪  "ForeignControllerInit/incorrect-admin-almProxy");
require(rateLimits.hasRole(DEFAULT_ADMIN_ROLE, addresses.admin) == true,
↪  "ForeignControllerInit/incorrect-admin-rateLimits");
require(controller.hasRole(DEFAULT_ADMIN_ROLE, addresses.admin) == true,
↪  "ForeignControllerInit/incorrect-admin-controller");
```

This difference in approach may introduce confusion, potentially leading to unintended restrictions or logic issues.

Recommendation: To avoid confusion and maintain consistency, we recommend the following design options:

1. Assume that all components (Proxy, Rate Limiter, and Controller) share the same admin. In this case, combine all the checks into a single function.
2. If each component can have a different admin, modify the initialization to use separate parameters, such as `addresses.proxyAdmin`, `addresses.rateLimitsAdmin`, and `addresses.controllerAdmin`. Additionally, if the intention is for these to be the same, add a check to ensure that all of them are equal.

Maker: Fixed in PR-44

Cantina Managed: Verified.

3.1.3 Missing `PSM` checks within the initialization of the Foreign Controller

Severity: Informational

Context: ControllerInit.sol#L252

Description: The `ForeignController` expects a PSM3 instance that has USDC, USDS and SUSDS as the underlying assets. The initialization script currently it's missing to add a check that the PSM that is received as parameter and configured it has the correct configuration: has 3 assets and they are USDC, USDS and SUSDS.

Recommendation: Consider adding the following checks:

- psm.usdc() is the `addresses.usdc`
- psm.usds() is the `addresses.usds` (this needs to be added in the addresses)
- psm.suds() is the `addresses.susds` (this needs to be added in the addresses)

Maker: Fixed in PR-44

Cantina Managed: Verified.

3.1.4 Revoking the `oldController` can silently fail, leading to an unsuccessful role revocation

Severity: Informational

Context: ControllerInit.sol#L111-L114 | ControllerInit.sol#L271-L273

Description: In the initialization script for the Controller, there is a condition that triggers a revoke if `oldController` parameter is set, this assumes that both the old proxy and the rate limiter are controlled by the same `oldController`. However, an issue arises if the two contracts are not actually controlled by the same controller. In this case, the conditional check leads to a silent failure due to the behavior of the `revoke` function from OpenZeppelin's AccessControl contract, which does not revert when the role does not exist. This can leave the false impression that the `oldController` was revoked, even though it may still be active.

Recommendation: Consider adding conditional revokes that verify whether the `oldController` is currently the controller before proceeding with the revoke. If the `oldController` is not the current controller, we should revert, avoiding the silent failure.

Maker: Fixed in PR-44

Cantina Managed: Verified.

3.1.5 Unused `pocket` function

Severity: Informational

Context: MainnetController.sol#L35

Description: The `pocket` function of the `IPSMLike` interface it's not used.

Recommendation: Consider removing it

Maker: Fixed in PR-44

Cantina Managed: Fixed.