CANTINA

MakerDAO: XChain Helpers & Spark-Gov-Relay

Security Review

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

September 9, 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.

On Aug 19th the Cantina team conducted a review of xchain-helpers and spark-gov-relay on commit hashes 07e27b6e and 38da9129 respectively.

The Cantina team reviewed MakerDAO's xchain-helpers and spark-gov-relay changes holistically on commit hashes 95edd63a and 5c166763, and determined that all issues were resolved and no new issues were identified.

The team identified a total of **7** issues in the following risk categories:

- Critical Risk: 0
- High Risk: 0
- Medium Risk: 0
- Low Risk: 0
- Gas Optimizations: 0
- Informational: 7

3 Findings

3.1 Informational

3.1.1 Obsolete `DEFAULT_ADMIN_ROLE` assignment to `Executor`'s roles

Severity: Informational

Context: Executor.sol#L53-L54

Description: In the `AccessControl` from OpenZeppelin, the `DEFAULT_ADMIN_ROLE` is already the admin of any newly defined role.

- AccessControl.sol#L39:

```
/**
 * By default, the admin role for all roles is `DEFAULT_ADMIN_ROLE`, which means
 * that only accounts with this role will be able to grant or revoke other
 * roles. More complex role relationships can be created by using
 * {_setRoleAdmin}.
```

Recommendation: Remove the `_setRoleAdmin` calls from the constructor.

Maker: Fixed in commit 5c166763.

Cantina Managed: Verified.

3.1.2 The `executionTime`can be removed from `_executeTransaction`

Severity: Informational

Context: Executor.sol#L206C14-L222

Description: The `executionTime` parameter can be removed from `_executeTransaction` as it's not used anywhere.

Recommendation: Remove the `executionTime` parameter.

Maker: Fixed in commit 5c166763.

Cantina Managed: Fixed.

3.1.3 The `gracePeriod` check can be moved inside the `_updateGracePeriod`

Severity: Informational

Context: Executor.sol#L43-L45, Executor.sol#L43-L45,

Description: The check ensuring that the new grace period is not lower than the allowed `MINIMUM_GRACE_PERIOD` can be moved to the `_updateGracePeriod` function. This will simplify the logic in both the constructor and the `updateGracePeriod` function.

Recommendation: Consider moving the check inside the `_updateGracePeriod` internal function.

Maker: Fixed in 5c166763.

Cantina Managed: Fixed.

3.1.4 Unsafe cast of the `gasLimit` in `OptimismForwarder`

Severity: Informational

Context: OptimismForwarder.sol#L23

Description: In the `OptimismForwarder` contract, the minimum gas limit parameter is received as a `uint256` and then unsafely casted to a `uint32`:

```
ICrossDomainOptimism(l1CrossDomain).sendMessage(
    target,
    message,
    uint32(gasLimit)
);
```

If the `gasLimit` is set to a value larger than what can be represented by a `uint32`, it can silently overflow, resulting in sending an incorrect minimum gas limit.

Recommendation: Consider either changing the `gasLimit` variable to `uint32` or using a safe casting approach that reverts if the value exceeds `type(uint32).max`.

Maker: Fixed in 95edd63a.

Cantina Managed: Fixed.

3.1.5 Arbitrum's Retryable Tickets cannot be cancelled.

Severity: Informational

Context: ArbitrumForwarder.sol#L43

Description: The `createRetryableTicket` function accepts several parameters, which can be reviewed in the Arbitrum Documentation. One key parameter is `callValueRefundAddress`, which is credited if the ticket times out or is canceled. More importantly, this is the address that can cancel a ticket within the 7-day window. Since this is set to `address(0)`, no one will be able to cancel the ticket. After 7 days, the ticket is dropped, and it can no longer be executed.

Recommendation: Consider whether the ability to cancel a ticket is a desired feature. If so, consider setting the `callValueRefundAddress` to a valid address. This could be a special contract that can also receive value but cannot withdraw it if burning the value is the intended behavior.

Maker: Acknowledged. We do not want transactions to be canceled.

Cantina Managed: Acknowledged.

3.1.6 Clarify ambiguous natspec for `Executor`'s `signatures` parameter

Severity: Informational

Context: IExecutor.sol#L48, IExecutor.sol#L164

Description: The `ActionsSet` and `queue`'s natspec for the `signatures` parameter states:

> Array of function signatures to encode in each call by the actions (can be empty).

The `signatures` array cannot be empty and needs to be the same size as the `targets` array of the action. However, the actual elements of the array can be the empty bytes/string in which case the `calldatas` field is used for the entire calldata.

Recommendation: Consider clarifying this in the documentation.

Maker: Fixed in commit 071ea4d0.

Cantina Managed: Fixed.

3.1.7 `Executor` does not support actions sending native tokens to EOA

Severity: Informational

Context: Executor.sol#L222

Description: All actions that can be queued and executed in the `Executor` are called using the `functionCallWithValue` helper which checks that the `target` address is a contract. The actions cannot be used to send native tokens to an EOA.

Recommendation: If sending native tokens to an EOA is a desired usecase, consider using low-level calls and checking their success state. Otherwise, consider documenting this limitation.

Maker: Acknowledged. This is the expected behavior.

Cantina Managed: Acknowledged.