Sky: dss-exec-lib

Security Review

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

January 29, 2026

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

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 are 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

Sky Protocol is a decentralised protocol developed around the USDS stablecoin.

From Jan 5th to Jan 16th the Cantina team conducted a review of dss-exec-lib on commit hash 3996d708. The team identified a total of **2** issues:

Issues Found

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

The Cantina Managed team reviewed Sky's dss-exec-lib holistically on commit hash 74f7115d and concluded that all findings were addressed and no new vulnerabilities were identified.

3 Findings

3.1 Informational

3.1.1 Missing checks in the `DssExecLib.sol`

Context: (*See each case below*)

Severity: Informational

Description:

1. DssExecLib.sol#L832-L835: We can observe an asymmetry between: `setIlkMaxLiquidationAmount` and `setIlkMinVaultAmount`, the `setIlkMaxLiquidationAmount` is missing the `hole` against `dust`
2. DssExecLib.sol#L883-L886: `tip` is a `uint192` when we file it `else if (what == "tip") tip = uint192(data);` the `_amount * RAD` could exceed the `uint192`.

Recommendation: Fix the checks mentioned above.

Sky:

1. This introduces a mandatory order when performing operations:

   If a new ilk is being initialized or if we are increasing `dust` beyond the current value of `hole`, we need to call `setIlkMaxLiquidationAmount` first and the `setIlkMinVaultAmount`. Luckily the order is already correct in `addNewCollateral`.

   If we are decreasing `hole` below `dust`, we need the reverse order.

   Fixed in commit fbb6cb7.
2. Fixed in commit 2c88b26.

Cantina Managed: Fix verified.

3.1.2 Minor issues, Typos & Documentation

Context: (*See each case below*)

Severity: Informational

Description:

1. DssExecLib.sol#L354: Consider removing the ”legacy” `flip` function in favor of `clip`, implementing the same code.
2. DssExecLib.sol#L880: The `tip` is always a fixed, flat fee, not a ”max amount”. Consider updating the comment.
3. DssExecLib.sol#L992: Consider also setting the `calc` parameters (`tau/cut/step` depending on what calc it is) in `addCollateralBase`.
4. DssExecLib.sol#L298-L300: The `GOV_GUARD` key is not present in the chainlog.
5. DssExecLib.sol#L1045: `addNewCollateral` does not add the `ilk` to the `LINE_MOM`. (`LineMomLike(chainlog.getAddress("LINE_MOM")).addIlk(cfg.ilk)`)

Recommendation: Consider fixing the issues mentioned above.

Sky:

1. It's a candidate for removal in the next iteration.
2. Fixed in commit 3bedd216.
3. Acknowledged. It's kind of hard to encapsulate that into this call because of the limited polymorphic capabilities of Solidity. The function to set the `calc` parameters was usually called right after `addNewCollateral`, like in DssSpell.sol#L93-L95.
4. This key was removed after this set of changes were made. Will remove it in the next iteration.

5. This is another action that is performed separately after the call to `addNewCollateral` in the same spell.

Cantina Managed: Verified.