Sky: stUSDS

Security Review

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

August 18, 2025

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

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

From Jul 28th to Aug 8th the Cantina team conducted a review of stusds on commit hash 09b7eee5. The team identified a total of **1** issues in the following risk categories:

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	1	0	1
Total	**1**	**0**	**1**

The Cantina Managed team reviewed Sky's `stusds` holistically on commit hash 09b7eee5 and concluded that all the issues were addressed and no new vulnerabilities were identified.

3 Findings

3.1 Informational

3.1.1 Mismatch between main functions and the `preview*()` counterpart behaviour when `chi == 0`

Severity: Informational

Context: StUsds.sol#L429-L436

Description: Once a bad-debt slash drives chi to 0, calling _drip() returns 0, which then causes a revert either via division by zero or a custom revert in the following functions:

- `mint()`.
- `deposit()`.
- `withdraw()`.
- `redeem()`.

We will use the deposit flow as an example, but the same behavior is present in all the aforementioned functions.

Under the ERC-4626 guidelines, `previewDeposit()` may fail under the same vault-level conditions that would make `deposit()` revert:

> MUST NOT revert due to vault-specific user/global limits. MAY revert due to other conditions that would also cause `deposit()` to revert.

This creates ambiguous behavior: `deposit()` reverts when `chi == 0`, but `previewDeposit()` does not.

Recommendation: Document this behavior. The EIP uses "MAY revert," so aligning the two paths is not strictly required, but an explicit specification will help third parties understand this difference.

Sky: The team has acknowledged the difference.

Cantina Managed: Acknowledged by the team.