



Sky: stUSDS Standby Spells

Security Review

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

June 3, 2026

Contents

1	Introduction	2
1.1	About Cantina	2
1.2	Disclaimer	2
1.3	Risk assessment	2
1.3.1	Severity Classification	2
2	Security Review Summary	3
3	Findings	4
3.1	Low Risk	4
3.1.1	StUsdsWipeParamSpell.done() can report line wipe completion before the effective Vat line is zero	4
3.2	Informational	4
3.2.1	Unnecessary STUSDS_MOM-is-ward-on-STUSDS requirement from halt/diss done() checks	4
3.2.2	Minor issues, Typos & Documentation	5

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 Mar 2nd to Mar 10th the Cantina team conducted a review of [emergency-spells](#) on commit hash [4ad87da7](#). The team identified a total of **3** issues:

Issues Found

Severity	Count	Fixed	Acknowledged
Critical Risk	0	0	0
High Risk	0	0	0
Medium Risk	0	0	0
Low Risk	1	1	0
Gas Optimizations	0	0	0
Informational	2	2	0
Total	3	3	0

The Cantina Managed team reviewed Sky's [emergency-spells](#) holistically on commit hash [f32bd6f7](#) and concluded that all findings were addressed and no new vulnerabilities were identified.

3 Findings

3.1 Low Risk

3.1.1 `StUsdsWipeParamSpell.done()` can report line wipe completion before the effective Vat line is zero

Severity: Low Risk

Context: `src/stusds/StUsdsWipeParamSpell.sol#L78-L80`

Description: For LINE and BOTH, the spell executes `stUsdsMom.zeroLine(address(stUsdsRateSetter))`, while `done()` only checks whether `stUsds.line()` and `stUsdsRateSetter.maxLine()` are zero.

However, zeroing `stUsds.line()` is not sufficient to stop borrowing immediately. In `StUsds`, `file("line", data)` only updates the internal line value and explicitly notes that `drip()` must be called right after if the new value should have an immediate effect on `vat[ilk].line`. The actual Vat sync happens in `_setLine()`, which is only reached through `drip()` and other later flows. `StUsdsMom.zeroLine()` does not call `stusds.drip()`.

As a result, `done()` can return `true` while the effective `Vat.ilks(ilk).line` is still non-zero, so the spell can be reported as complete before borrowing is actually stopped at the Vat level.

Recommendation: Consider making the line wipe path synchronize the effective Vat line in the spell by calling `stusds.drip()` after `stusds.file("line", 0)`. Independently, consider updating `done()` for LINE and BOTH to verify the effective `Vat.ilks(ilk).line == 0`.

Sky: Fixed in commit [a262fdf](#). `stUsdsMom.zeroLine` calls `drip` to apply the zero line immediately.

Cantina Managed: Fix verified.

3.2 Informational

3.2.1 Unnecessary STUSDS_MOM-is-ward-on-STUSDS requirement from halt/diss `done()` checks

Severity: Informational

Context: `src/stusds/StUsdsRateSetterHaltSpell.sol`, `src/stusds/StUsdsRateSetterDissBudSpell.sol`

Description: `StUsdsRateSetterHaltSpell` and `StUsdsRateSetterDissBudSpell` set `done()` to `true` when `stUsds.wards(address(stUsdsMom)) == 0`:

- `src/stusds/StUsdsRateSetterHaltSpell.sol#L68`.
- `src/stusds/StUsdsRateSetterDissBudSpell.sol#L74`.

This ward relation is unnecessary for both actions because both spells delegate to `STUSDS_MOM` methods that do not require writing to `STUSDS`:

- `StUsdsRateSetterHaltSpell`: `stUsdsMom.haltRateSetter(rateSetter)` → `rateSetter.file("bad", 1)`.
- `StUsdsRateSetterDissBudSpell`: `stUsdsMom.dissRateSetterBud(rateSetter, bud)` → `rateSetter.diss(bud)`.

The extra requirement can be removed for these two spells without affecting execution safety for their actual call paths. This is distinct from `StUsdsWipeParamSpell`, where `STUSDS_MOM` mutates `stUSDS` and therefore needs `stUsds.wards(address(stUsdsMom))`.

Recommendation:

- Remove the `stUsds.wards(address(stUsdsMom))` guard from `done()` in:
 - `src/stusds/StUsdsRateSetterHaltSpell.sol`.
 - `src/stusds/StUsdsRateSetterDissBudSpell.sol`.
- Keep the `stUsdsMom-on-stUsdsRateSetter` and `stUsdsRateSetter-on-stUsds` checks because those are required for these execution paths.

Sky: Fixed in the following commits:

- ddcab3ec.
- a6e7d601.
- eac8b2b0.

Cantina Managed: Verified.

3.2.2 Minor issues, Typos & Documentation

Severity: Informational

Context: (See each case below)

Description:

- `StUsdsWipeParamSpell.sol#L109`: Comments says `// Ignore StUsds instances that have not relied on StUsdsMom.` but the check is against `stUsdsRateSetter`, not `stUsdsMom`. Should be `// Ignore StUsds instances that have not relied on stUsdsRateSetter.`
- `StUsdsWipeParamSpell.sol#L95`: Natspec says `In both cases` but lists three conditions. Should be `In such cases`.
- `StUsdsWipeParamSpell.sol#L71`: `description()` returns `"Emergency Spell | stUSDS | halt: "` but the spell wipes parameters, not halts. Should be `"Emergency Spell | stUSDS | wipe param: "`.
- All `src/stusds/` files including `StUsdsRateSetterDissBudSpell.sol`: Copyright header says `"© 2024"` but files were written in December 2025 / January 2026. Update to `"© 2025"`.

Recommendation: Consider fixing the comments and copyright years mentioned above.

Sky: Fixed in the following commits:

1. e6c41ec5.
2. de7ac3ba.
3. 84494084.
4. 9da109f0.

Cantina Managed: Verified.