CANTINA

Sky:
PAU Administered Actor
Security Review

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

June 9, 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 Actors can self-call the agent to escalate roles 4

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 June 1st to June 2nd the Cantina team conducted a review of pau-administered-agent on commit hash 5c267327 (tag v1.0.0-beta.0). The team identified a total of **1** 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	0	0	0
Total	**1**	**1**	**0**

The Cantina Managed team reviewed Sky's pau-administered-agent holistically on commit hash bfaaf709 (tag v1.0.0) and concluded that all findings were addressed and no new vulnerabilities were identified.

3 Findings

3.1 Low Risk

3.1.1 Actors can self-call the agent to escalate roles

Severity: Low Risk

Context: `AdministeredAgent.sol#L110-L136`

Description: `call` and `batchCall` let actors call arbitrary targets, but do not block `target == address(this)`.

If the agent is ever added as an admin, grantor, or revoker, an actor can self-call role-management functions and inherit that role through `msg.sender == address(agent)`:

```
agent.call(
    address(agent),
    abi.encodeCall(IAdministeredAgent.addAdmin, (actor))
);
```

This breaks the intended separation between actors and role managers.

There is also a related `batchCall` concern: because each subcall is executed before the next target is processed, a self-call could mutate actor/admin/grantor/revoker state mid-batch and affect later subcalls in the same batch.

Recommendation: Reject self-calls in both `call` and `batchCall`:

```
require(target != address(this), "AdministeredAgent/invalid-target");
```

and inside the batch loop:

```
require(targets[i] != address(this), "AdministeredAgent/invalid-target");
```

This prevents actor-driven role escalation and self-reentrant state changes during batched execution.

Sky: Addressed in PR 2.

Cantina Managed: Fix verified.