# Sky: chief

## Security Review

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

April 30, 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 Apr 14th to Apr 15th the Cantina team conducted a review of chief's `src` directory contents on commit hash 3c943e04. The team identified a total of **3** issues in the following risk categories:

- Critical Risk: 0
- High Risk: 0
- Medium Risk: 0
- Low Risk: 0
- Gas Optimizations: 0
- Informational: 3

The Cantina Managed team reviewed Sky's chief holistically on commit hash 3c943e04 and concluded that all issues were addressed.

# 3 Preamble

The following technical write-up with known issues was passed along to us by the Sky Team.

## 3.1 Chief Change - Technical Writeup

The current Chief change moves to operate with SKY as the governance token. Among other things, it removes the limitation for an address to lock and free in the same block, and instead disallows anyone from freeing in the same block that launch or lift were performed. There is also a cooldown period for calling lifts, so free operations are not totally starved under any circumstance.

The change still blocks launching and lifting using a flash loan, but removes the DOS vector of blocking someone else's free operation when using a shared delegate, and also allows removing the vote-delegate reserveHatch mechanism.

Some notes, disclaimers and known issues:

1. Lockstake and other related modules are assumed to also be upgraded to use SKY. This will get audited on a following iteration, along with a Chief migration library to be called from the spell.
2. The old Chief functionality for setting roles and capabilies was discarded on purpose, as it hasn't been useful.
3. Gas optimizations, although exist to some extent, were mostly discarded for the sake of leaving the old functonality and structure.
4. The fact that once there is a lift, it can cause free operations and lockstake liquidations to revert is accepted. Under regular circumstances this should happen around every 2 weeks, when a spell is scheduled. At some points in time, like on close votes, or before launch is called, lift/launch could be used for intentional DOSing. The cooldown period and the use of flashbots are assumed to mitigate that, even if additional trials are needed. Although in the previous design the reserveHatch mechanism, which was removed, could in theory guarantee a success of free, we think that the lift cooldown is enough of a mitigation.
5. As with the previous version, multi-block-mev is not addressed. The goal here was to create an increment to the previous code and not invent new mechanisms, that by themselves would come with tradeoffs. As in other governance code-bases, a 1 block delay is used. The amount of SKY in DEXs and lending markets is monitored by Sky's risk unit, with relation to risk of governance attacks.
6. Since it is now possible to lock and free in the same block, it becomes easier to cause legitimate lifts to fail by buying/lending SKY, voting for the current hat, and sandwiching the lift call (which intends to choose a new hat). This is acknowledged and is assumed to be worked-around if needed by flashbots and/or multiple trials.
7. On the initial launch period it is theoretically easier to call lift (since less SKY is needed to toggle between hats) and DOS free operations, liquidations or the intended initial lift call for address(0). It is assumed that even with the existence of the cooldown period, legitimate lifts would eventually be able to enter. We came up with a couple of good solutions to mitigate this issue, however we ended up preferring to leave the code as it is (considering it is a temporary state that once passed won't give more troubles).
8. As with the previous reserveHatch mechanism, the new implementation's cooldown period relies on an amount of blocks, and not an exact time interval. The assumption is that in case block times change drastically the Chief could be upgraded again to align to that. The selected cooldown period is expected to be chosen with thorough consideration (the current value which we think is appropriate is around 10 blocks).

# 4 Findings

## 4.1 Informational

### 4.1.1 Improvement on the `Lift` event

**Severity:** Informational

**Context:** Chief.sol#L145

**Description:** The `Lift` event that is emitted in the `lift` function could be improved to return the quantity of `approvals` as well:

```
emit Lift(whom);
```

**Recommendation:** Consider adding `approvals` topic as well.

**Sky:** Acknowledged. We will keep it as is, also as the current Chief doesn't log it.

**Cantina Managed:** Acknowledged.

### 4.1.2 Missing return parameter name

**Severity:** Informational

**Context:** Chief.sol#L78, Chief.sol#L78, Chief.sol#L78

**Description:** Throughout the contract, every return parameter is named. For example:

```
function canCall(address caller, address, bytes4) external view returns (bool ok) {
    ok = live == 1 && caller == hat;
}
```

However, the following functions do not have named return parameters: `length`, `GOV`, `MAX_YAYS`.

**Recommendation:** Consider adding names to the return parameters of these functions as well to maintain code consistency.

**Sky:** Acknowledged.

**Cantina Managed:** Acknowledged.

### 4.1.3 `yay` execution order per `slate` is non-deterministic

**Severity:** Informational

**Context:** Chief.sol#L64-L142

**Description:** When a `slate` has multiple `yays` and the slate has gathered more approvals on its `yays` than the current `hat`, anyone can call `lift(yay)` with any one of the `yays`. The execution order of a slate with multiple `yays` can be chosen by the `lift` caller and must therefore be assumed to be chosen maliciously.

**Recommendation:** If the execution order matters, consider adding only a single `yay` per slate that calls into the sub-spells in the expected order.

**Sky:** Acknowledged. This is the same as the current chief. In practice, there has always been one spell that MKR holders planned to schedule at a time, and the yays feature was used more to support moving from one hat to another safely (always protecting the hat).

If a group of MKR/SKY holders do intend to schedule 2 new spells with a specific order, and for them to be voted together, they can enforce that as part of the spells code.

**Cantina Managed:** Acknowledged.