

Hyperware Review Fixes

Security Review

Cantina Managed review by:
Gerard Persoon, Lead Security Researcher
Slowfi, Security Researcher

February 12, 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**
- 2.1 Scope . . . 3

3 Findings **4**
- 3.1 Medium Risk . . . 4
  - 3.1.1 `_delegatedUnlockStamps[]` isn't always updated . . . 4
- 3.2 Low Risk . . . 4
  - 3.2.1 Accounts with `address(0)` have special meaning . . . 4
  - 3.2.2 `recomputeMultipliers()` only works if all accounts are supplied . . . 4
- 3.3 Gas Optimization . . . 5
  - 3.3.1 Accounts with `balance == 0` can be ignored . . . 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

From Feb 3rd to Feb 5th the Cantina team conducted a review of protocol on commit hash 822e4852. The team identified a total of **4** issues:

Issues Found

Severity	Count	Fixed	Acknowledged
Critical Risk	0	0	0
High Risk	0	0	0
Medium Risk	1	1	0
Low Risk	2	2	0
Gas Optimizations	1	1	0
Informational	0	0	0
Total	**4**	**4**	**0**

2.1 Scope

The security review had the following components in scope for protocol on commit hash 822e4852:

```
src/registry/TokenRegistry.sol
```

3 Findings

3.1 Medium Risk

3.1.1 `_delegatedUnlockStamps[]` isn't always updated

Severity: Medium Risk

Context: TokenRegistry.sol#L110, TokenRegistry.sol#L192, TokenRegistry.sol#L465, TokenRegistry.sol#L479-L481, TokenRegistry.sol#L657, HyperGovernorToken.sol#L261-L274

Description: The value of `_delegatedUnlockStamps[]` isn't always updated:

- A decrease in locking period doesn't decrease it;
- An increase in locking period doesn't increase;
- A redelegation doesn't decrease;
- It isn't cleanedup with `withdraw()`

This can result in incorrect calculation of votes.

Recommendation: Consider updating `_delegatedUnlockStamps[]` This will require an additional datastructure

Hyperware: Delegations are stored in buckets to be able to selectively update them on delegations, locking period changes and `withdraw()` Shortening of locks is disallowed

In `updateDelegationMultipliers()` a heuristic is used to detect self delegation because that hook does not receive the delegator address.

Fixed in PR 124.

Cantina Managed: Fix verified.

3.2 Low Risk

3.2.1 Accounts with `address(0)` have special meaning

Severity: Low Risk

Context: TokenRegistry.sol#L405-L413, TokenRegistry.sol#L433, TokenRegistry.sol#L444-L454, TokenRegistry.sol#L637-L644

Description: Accounts with `address(0)` have a special meaning, because the will update `_totalMultipliers[]`.

> Note: the risk is limited because `_totalMultipliers[]` will be overwritten again at the end of the function `recomputeMultipliers()`.

Recommendation: Consider checking no acounts with `address(0)` are used.

Hyperware: Fixed in PR 124.

Cantina Managed: Fix verified.

3.2.2 `recomputeMultipliers()` only works if all accounts are supplied

Severity: Low Risk

Context: TokenRegistry.sol#L405, TokenRegistry.sol#L405-L409, TokenRegistry.sol#L444-L453

Description: The function `recomputeMultipliers()` only works correctly if all used accounts are supplied. This is because at the end all totals are calculated based on the calculated `totalMonthlyVotes[]`

With a large number of accounts this might not fit within the gas limits (especially with eip-7825 - 16M). However about 64k gas is used per account which allows for about 256 accounts. According to the project there are currently 67 active accounts, so that should be no problem.

Recommendation: Consider checking that all accounts are supplied. This can be done by checking that the sum of raw votes across accounts to equal `totalVotes`

Hyperware: Fixed in PR 124.

Cantina Managed: Fix verified.

3.3 Gas Optimization

3.3.1 Accounts with `balance == 0` can be ignored

Severity: Gas Optimization

Context: TokenRegistry.sol#L405-L418

Description: Function `recomputeMultipliers()` checks if `balance > 0` and skips some code if it doesn't. However the rest for the information for these addresses should also be 0 so they can be skipped.

Recommendation: DOublecheck the conclusion and consider changing the code to:

```
- if (balance > 0) { ... }
+ if (balance == 0) continue;
```

Hyperware: Fixed in PR 124.

Cantina Managed: Fix verified.