

Gensyn Token Sale

Security Review

Cantina Managed review by:
Om Parikh, Security Researcher
Rvierdiiev, Security Researcher

December 18, 2025

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 Low Risk . . . 4
  - 3.1.1 `recoverTokens` allows to withdraw commitment tokens . . . 4
- 3.2 Informational . . . 4
  - 3.2.1 Remove redundant errors . . . 4
  - 3.2.2 Differences between comments and the code . . . 4
  - 3.2.3 EIP-2612 permit signatures can be front-run for `replaceBidWithPermit` . . . 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 Nov 19th to Nov 24th the Cantina team conducted a review of gensyn on commit hash 6fcaf99f. 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	0	0	0
Low Risk	1	0	1
Gas Optimizations	0	0	0
Informational	3	3	0
Total	**4**	**3**	**1**

2.1 Scope

The security review had the following components in scope for gensyn on commit hash 6fcaf99f:

```
contracts/src/GensynSale.sol
```

3 Findings

3.1 Low Risk

3.1.1 `recoverTokens` allows to withdraw commitment tokens

Severity: Low Risk

Context: GensynSale.sol#L1076-L1078

Description: Currently the `recoverTokens()` function does not restrict which tokens can be recovered. As a result, an account with the `TOKEN_RECOVERER_ROLE` can withdraw USDT and USDC, which are the core commitment tokens used by the protocol.

Recommendation: Add validation ensuring that commitment tokens (e.g., USDT and USDC) cannot be recovered through `recoverTokens()`. Only non-core, mistakenly sent tokens should be recoverable by the `TOKEN_RECOVERER_ROLE`.

Echo: Acknowledged. This is correct and deliberate. The intention was that the `recoverTokens` should allow us to recover from any scenario, e.g. if the `withdraw` function was broken for some reason.

Cantina Managed: Acknowledged.

3.2 Informational

3.2.1 Remove redundant errors

Severity: Informational

Context: GensynSale.sol#L159, GensynSale.sol#L170

Description:

```
error ZeroPrice();
error MaxAddressesPerEntityExceeded
```

Those errors are never used in the code base.

Recommendation: Remove redundant errors.

Echo: Fixed in PR 2.

Cantina Managed: Fix verified.

3.2.2 Differences between comments and the code

Severity: Informational

Context: *(No context files were provided by the reviewer)*

Description: According to the comments:

> Bids must satisfy monotonic constraints: amounts and prices can only increase, and lockup preferences can be enabled but cannot be disabled once set.

However, the implementation only enforces that the new bid's amount and price are not lower than the previous bid. This means a user can submit a bid where both the amount and price are exactly the same as the previous one.

Recommendation: Require both amount and price to be strictly greater than the previous bid's values.

Echo: Fixed in PR 2. We've updated the comment, but will opt to keep the logic as-is for simplicity, since sending bids without changes will not have any effect other than wasting the caller's gas.

Cantina Managed: Fix verified.

3.2.3 EIP-2612 permit signatures can be front-run for `replaceBidWithPermit`

Severity: Informational

Context: GensynSale.sol#L508-L516

Description: An adversary can extract permit signature from mempool and front-run to submit directly to ERC-20 contract, so `pToken.permit` would revert later when original transaction lands with same permit sig.

However, since there is `replaceBidWithApproval` is present, this doesn't lead to complete DOS.

Recommendation:

- Add a `try / catch` around `permit`, so that even if permit reverts, it can be ignored and proceed with transfering.
- Frontend must ensure `replaceBidWithApproval` is used as fallback if `replaceBidWithPermit` keeps reverting.

Echo: Fixed in PR 2.

Cantina Managed: Fix verified.