Security Review

Cantina Managed

CLEARPOOL: CPOOL TO CLEAR TOKEN SWAP

Cantina Managed review by:
Red Swan, Lead Security Researcher
Om Parikh, Security Researcher

CANTINA

October 8, 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 Low Risk . . . 4
  - 3.1.1 `owner` and LayerZero's `delegate` are not updated atomically . . . 4
- 3.2 Informational . . . 4
  - 3.2.1 Code Style . . . 4
  - 3.2.2 Upgrade to newer `solc` version . . . 5

1 Introduction

1.1 About Cantina

Cantina is a security platform that pairs a world-class network of security researchers with agentic AI to help teams building onchain find and fix real risk. Through managed reviews, competitions, and bounties, Cantina secures protocols before vulnerabilities can be exploited.
Learn more at cantina.security

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

Clearpool is the leading DeFi credit marketplace pioneering Real-World Asset lending.

From Oct 6th the Cantina team conducted a review of cpool-to-clear-token-swap on commit hash df560190. 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	1	1
Total	**3**	**2**	**1**

2.1 Scope

The security review had the following components in scope for cpool-to-clear-token-swap on commit hash df560190:

```
cpool-to-clear-token-swap/src
├── CLEAR.sol
├── CPoolToClearSwapper.sol
└── interface
    └── IExtras.sol
```

3 Findings

3.1 Low Risk

3.1.1 `owner` and LayerZero's `delegate` are not updated atomically

Severity: Low Risk.

Context: CLEAR.sol#L49-L51.

Description: The `owner` and LayerZero's `delegate` are the same address `_admin`. However, `transferOwnership` and/or `renounceOwnership` do not change the delegate when updating the owner.

In several cases, the old delegate can change the config before the `acceptOwnership` transaction and may cause side effects, such as:

- `clear.transferOwnership(...)` → `endpoint.setConfig(CLEAR, ...)` → `clear.acceptOwnership(...)` → `clear.setDelegate(newOwner);`

The same is also true for `renounceOwnership`.

Recommendation:

- Override `acceptOwnership` to handle it atomically.

```
+ /// @notice Accepts ownership and atomically transfers the LayerZero
+ ///         Endpoint delegate to the new owner.
+ function acceptOwnership() public override {
+     super.acceptOwnership();
+     setDelegate(owner());
+ }
```

- Since `renounceOwnership` creates a partial ownerless system, it can be mitigated by not allowing renouncement.

```
+ error OwnershipRenouncementDisabled();
+
+ /// @notice Ownership renouncement is disabled because CLEAR requires
+ ///         an owner for minting, LayerZero configuration, and swapper control.
+ function renounceOwnership() public pure override {
+     revert OwnershipRenouncementDisabled();
+ }
```

Clearpool: Fixed in commit f5b02b5b.

Cantina Managed: Fix verified.

3.2 Informational

3.2.1 Code Style

Severity: Informational.

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

Description: Across `CLEAR.sol` and `CPoolToClearSwapper.sol`, several stylistic inconsistencies and missing NatSpec documentation items were identified:

1. **Non-Standard Immutable Naming (`UPPER_SNAKE_CASE`)**:
   - According to the Solidity Style Guide, immutable variables and constants should follow `UPPER_SNAKE_CASE`.
   - In `CLEAR.sol`, `swapper` is declared in `camelCase`.

- In `CPoolToClearSwapper.sol`, `oldCpool` and `clear` are declared in `camelCase`.

2. **Function Ordering Violations in `CLEAR.sol`:**
   - The Solidity style convention recommends grouping functions by visibility: `external`, `public`, `internal`, and `private`, with view/pure functions placed at the end of each section.
   - In `CLEAR.sol`, the internal hook `_update` precedes the public functions `nonces` and `transferOwnership`.
3. **Incomplete NatSpec Documentation:**
   - **Constructors:** Missing `@param` tags in `CLEAR.sol` (`_lzEndpoint`, `_admin`, `_oldCpool`) and `CPoolToClearSwapper.sol` (`_oldCpool`).
   - **Internal / Helper Functions:** `_checkClearOwner` in `CPoolToClearSwapper.sol` lacks NatSpec documentation entirely.
   - **Overrides:** Overridden methods in `CLEAR.sol` (`_update`, `nonces`, `transferOwnership`, `_transferOwnership`) omit `@inheritdoc` or explanatory `@dev` annotations detailing custom behavior (e.g., the recipient exclusion guard in `_update`).
4. **Shadowing and Header Formatting Nits:**
   - **Variable Shadowing:** In `CLEAR.sol`, the `owner` parameter of `nonces(address owner)` shadows `Ownable.owner()`.
   - **Fragmented `@dev` Tags:** The file header of `CLEAR.sol` contains six separate `@dev` tags rather than a consolidated documentation block or `@custom:` tags.
   - **Wordy NatSpec:** Functions repeatedly prefix `@notice` descriptions with "This function allows...", which can be streamlined.

Recommendation: If code style is important, consider making the changes described above.

Clearpool: Acknowledged.

Cantina Managed: Acknowledged.

3.2.2 Upgrade to newer `solc` version

Severity: Informational.

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

Description: Consider upgrading from the current `0.8.28` `solc` version to the current or close to current version as there are multiple known bugs in the older version.

Recommendation:

```
- pragma solidity 0.8.28;
+ pragma solidity 0.8.37;
```

Clearpool: Fixed in commit f5b02b5b.

Cantina Managed: Fix verified.