Security Review

Cantina Managed

COINBASE: SOLANA-STABLE-SWAPPER

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

CANTINA

August 19, 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 Default address fee recipient . . . 4
  - 3.1.2 Incorrect value used in test . . . 4
- 3.2 Informational . . . 4
  - 3.2.1 User Funded ATA Creation for Fee Recipient . . . 4
  - 3.2.2 Missing non-zero `min_amount_out` check in Solana swap compared to EVM . . . 5
  - 3.2.3 Hot key comment not accurate . . . 5
  - 3.2.4 `MIGRATION_GROWTH` is only used in comments . . . 5
  - 3.2.5 `.key()` versus `Pubkey` . . . 6
  - 3.2.6 Initial state of a newly added token . . . 6
  - 3.2.7 ”Liquidity paused” not clear . . . 6
  - 3.2.8 Comments in `stable-swapper.ts` are obsolete . . . 6

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

Base is a secure, low-cost, builder-friendly Ethereum L2 built to bring the next billion users onchain.

From Aug 13th to Aug 14th the Cantina team conducted a review of stable-swapper on commit hash 397bfa36. The team identified a total of **10** issues:

Issues Found

Severity	Count	Fixed	Acknowledged
Critical Risk	0	0	0
High Risk	0	0	0
Medium Risk	0	0	0
Low Risk	2	2	0
Gas Optimizations	0	0	0
Informational	8	6	2
Total	**10**	**8**	**2**

2.1 Scope

The security review had the following components in scope for stable-swapper on commit hash 397bfa36:

```
solana/programs/stable-swapper/src
├── constants.rs
├── errors.rs
├── lib.rs
├── state.rs
└── utils.rs
```

3 Findings

3.1 Low Risk

3.1.1 Default address fee recipient

Severity: Low Risk.

Context: lib.rs#L25-L30, lib.rs#L517.

Description: The original `initialize()` allowed the default address for the fee recipient.

The new `initialize()` checks for this via `require_recipient_set()`.

Recommendation: Consider checking this in `do_migrate_authorities()`.

Coinbase: Fixed in PR 20.

Cantina Managed: Fix verified.

3.1.2 Incorrect value used in test

Severity: Low Risk.

Context: stable-swapper.ts#L1371, stable-swapper.ts#L1448-L1457.

Description: A test for `Unpause` uses `pauseAuthority`, which looks to be incorrect.

Recommendation: Consider changing the test to:

```
    .accounts({
    pool,
-   pauseAuthority: pauseAuthority.publicKey,
+   unpauseAuthority: unpauseAuthority.publicKey,
  })
-  .signers([pauseAuthority.payer])
+  .signers([unpauseAuthority.payer])
```

Coinbase: Fixed in PR 20.

Cantina Managed: Fix verified.

3.2 Informational

3.2.1 User Funded ATA Creation for Fee Recipient

Severity: Informational.

Context: lib.rs#L353, lib.rs#L772-L779.

Description: When `update_fee_recipient` is called, only the pubkey is updated in the pool state. No associated token accounts are created for the new fee recipient. During subsequent swaps, the `fee_recipient_token_account` uses `init_if_needed` with the swapping user as payer. This means the first user to swap each supported token after a fee recipient change will pay approximately 0.002 SOL to create the new recipients ATA.

In contrast, when adding a new token via `add_supported_token`, the `configure_authority` pays for the fee recipients ATA creation.

Recommendation: Consider requiring the `configure_authority` to pre create all necessary ATAs for the new fee recipient within the `update_fee_recipient` instruction. This would align the cost model with `add_supported_token` and shift the burden to the party making the configuration change.

Coinbase: Good point, but this is out of scope for this review.

Cantina Managed: Acknowledged.

3.2.2 Missing non-zero `min_amount_out` check in Solana swap compared to EVM

Severity: Informational.

Context: StableSwapper.sol#L333-L344, lib.rs#L170-L173, lib.rs#L240-L242.

Description: The Solana swap instruction validates that `amount_in` is greater than zero and that the computed `amount_out` is greater than zero, but it never requires `min_amount_out` to be greater than zero. Because the only slippage guard is `amount_out` ≥ `min_amount_out`, a caller who passes `min_amount_out` equal to zero reduces that guard to a comparison that always succeeds, so slippage protection is effectively turned off for that swap.

The EVM implementation does not permit this. Its swap function explicitly rejects a zero minimum, forcing every caller to commit to a real minimum output before the swap executes. The relevant EVM guard is on L344 in function `swap()`.

This is a behavioral divergence between the two chains for the same product. The two implementations are meant to mirror each other, yet the Solana side accepts an input that the EVM side treats as invalid. This is a small parity and hardening note to further tighten the codebase.

Recommendation: Add an explicit check that `min_amount_out` is greater than zero at the start of the swap instruction, mirroring the EVM `CannotBeZeroAmount` guard so both chains enforce the same invariant:

```
  require!(amount_in > 0, LiquidityError::InvalidAmount);
+   require!(min_amount_out > 0, LiquidityError::InvalidAmount);
```

This forces every caller to specify a real minimum output and removes the ability to submit a swap with slippage protection disabled, bringing the Solana behavior back in line with the EVM contract.

Coinbase: Fixed in PR 20.

Cantina Managed: Fix verified.

3.2.3 Hot key comment not accurate

Severity: Informational.

Context: state.rs#L15.

Description: The hot key comment is not accurate because there is a list of allowed `withdraw_recipients`.

Recommendation: Consider changing the comment in the following way:

```
- /// Hot key allowed to withdraw liquidity (only to `withdraw_recipient`).
+ /// Hot key allowed to withdraw liquidity (only to one of the
↪  `withdraw_recipients`).
```

Coinbase: Fixed in PR 20.

Cantina Managed: Fix verified.

3.2.4 `MIGRATION_GROWTH` is only used in comments

Severity: Informational.

Context: lib.rs#L510, lib.rs#L601, state.rs#L53.

Description: The variable `MIGRATION_GROWTH` is only used in comments.

Recommendation: Consider removing the variable or turning the line into a comment.

Coinbase: Fixed in PR 20.

Cantina Managed: Fix verified.

3.2.5 `.key()` versus `Pubkey`

Severity: Informational.

Context: lib.rs#L25-L37, lib.rs#L66-L72.

Description: Function `initialize()` uses `.key()` while `migrate_authorities()` and most other functions uses `Pubkey`. The result is the same but it's inconsistent.

Recommendation: Consider using the same convention everywhere. However, because it changes the interface, it might not be worth the trouble.

Coinbase: Acknowledged.

Cantina Managed: Acknowledged.

3.2.6 Initial state of a newly added token

Severity: Informational.

Context: StableSwapper.sol#L333-L353, StableSwapper.sol#L406-L420, lib.rs#L86-L111, lib.rs#L170-L183.

Description: The initial state of a newly added token is different between EVM and Solana. On the EVM the initial state is that `swappable = false` and on Solana the initial state is `vault.disabled = false;`, which means it is swappable.

Both versions are meant to have the same functionality.

Recommendation: Consider using the same approach on both versions.

Coinbase: Fixed in PR 20.

Cantina Managed: Fix verified.

3.2.7 "Liquidity paused" not clear

Severity: Informational.

Context: errors.rs#L7-L8, lib.rs#L304-L306.

Description: "Liquidity paused" means withdrawals paused, however, swaps and direct deposits are still allowed. So the naming `LiquidityPaused` and the error message are confusing.

Recommendation: Consider renaming `LiquidityPaused` to `WithdrawalPaused`, both the error string and the variable name.

Coinbase: Fixed in PR 20.

Cantina Managed: Fix verified.

3.2.8 Comments in `stable-swapper.ts` are obsolete

Severity: Informational.

Context: stable-swapper.ts#L341, stable-swapper.ts#L1401, stable-swapper.ts#L1450.

Description: Some comments in `stable-swapper.ts` are obsolete and confusing.

Recommendation: Consider changing the comments in the following way, or remove the comments.

```
- .pauseWithdraws() // swapsPaused=null, liquidityPaused=true
+ .pauseWithdraws() // Sets liquidityPaused=true; swapsPaused is unchanged
- .pauseSwaps() // swapsPaused, liquidityPaused
+ .pauseSwaps() // Sets swapsPaused=true; liquidityPaused is unchanged
- .unpauseSwaps() // swapsPaused, liquidityPaused
+ .unpauseSwaps() // Sets swapsPaused=false; liquidityPaused is unchanged
```

Coinbase: Fixed in PR 20.

Cantina Managed: Fix verified.