Coinbase: commerce-payments PR 90

Security Review

Cantina Managed review by:
Sujith Somraaj, Lead Security Researcher
Cryptara, Security Researcher

July 24, 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 Informational . . . 4
  - 3.1.1 Zero-fee documentation conflicts with fixed fee receiver validation . . . 4
  - 3.1.2 Capture fragmentation bypasses the aggregate minimum fee . . . 4
  - 3.1.3 Changed event signatures silently break offchain consumers filtering by `topic0` . . . 4

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

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

From Jul 20th to Jul 21st the Cantina team conducted a review of commerce-payments on commit hash 6b0a5ef3. 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	0	0	0
Gas Optimizations	0	0	0
Informational	3	3	0
Total	**3**	**3**	**0**

2.1 Scope

The security review had the following components in scope for commerce-payments on commit hash 6b0a5ef3:

```
src/AuthCaptureEscrow.sol
```

3 Findings

3.1 Informational

3.1.1 Zero-fee documentation conflicts with fixed fee receiver validation

Severity: Informational.

Context: AuthCaptureEscrow.sol#L527-L529.

Description: The documentation in `docs/Fees.md:133` states that `feeAmount = 0 with feeReceiver = address(0)` succeeds when `PaymentInfo.feeReceiver` is a fixed nonzero address. However, `_validateFee()` unconditionally compares both receiver addresses and reverts with `InvalidFeeReceiver`. Integrations following the documentation may therefore unexpectedly fail `charge()` or `capture()` calls.

Recommendation: Align the documentation and implementation. Either require the configured receiver in the example, or skip receiver matching when `feeAmount == 0`.

Coinbase: Fixed in commit 94367139.

Cantina Managed: Verified fix. No code changes made, the fix is documentation-only.

3.1.2 Capture fragmentation bypasses the aggregate minimum fee

Severity: Informational.

Context: AuthCaptureEscrow.sol#L517-L519.

Description: Each capture independently computes `minFee = floor(captureAmount × minFeeBps / 10,000)`. An authorized operator can fragment a payment into sufficiently small captures so every minimum rounds to zero.

For a zero-decimal token, a 100-token authorization with a fixed 1% fee requires 1 token when captured once, but 100 one-token captures each accept `feeAmount = 0`. Exploitation is constrained by operator trust (as fee receiver will be the operator at present), transaction costs, and token denomination. If the operator trust model changes this may become profitable for low-decimal high-value tokens.

Recommendation: Document this as "accepted risk". Alternatively, explicitly define `minFeeBps` as a per-capture guarantee and restrict uneconomically small captures.

Coinbase: Fixed in commit 94367139.

Cantina Managed: Verified fix. No code changes made, the fix is documentation-only.

3.1.3 Changed event signatures silently break offchain consumers filtering by `topic0`

Severity: Informational.

Context: AuthCaptureEscrow.sol#L78-L85, AuthCaptureEscrow.sol#L93.

Description: The PR changes `uint16 feeBps` to `uint256 feeAmount` in the `PaymentCharged` and `PaymentCaptured` events. Because an event's `topic0` is the `keccak256` of its full signature, both events get new selectors.

Offchain indexers, reconciliation and monitoring that filter logs by the old `topic0` will not error — they will simply observe zero events from the new deployment. The failure mode is silent data loss rather than a crash, so it can go unnoticed.

Two aggravating factors:

1. The currently deployed `AuthCaptureEscrow` instances on Base mainnet/Sepolia emit the old-format events, so both formats will coexist during migration.

2. The third data field changes meaning from a rate (bps) to an absolute token amount; a consumer migrated to the new address but decoding with the old ABI would misinterpret values (e.g. reading `25e6` as bps) rather than fail.

Recommendation: Call out the selector change explicitly in release/migration notes, publish both `topic0` values for integrators, and version the deployment docs + SDK ABIs together with the redeployment.

Coinbase: Fixed in commit 94367139.

Cantina Managed: Verified fix. No code changes made, the fix is documentation-only.