Eco PR 347

Security Review

Cantina Managed review by:

Rikard Hjort, Lead Security Researcher

Red Swan, Security Researcher

November 25, 2025

Contents

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

Eco enables apps to unlock stablecoin liquidity from any connected chain and give users the simplest onchain experience.

From Nov 17th to Nov 18th the Cantina team conducted a review of eco-routes on commit hash df431603. The team identified a total of **5** 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	5	4	1
Total	**5**	**4**	**1**

The Cantina Managed team reviewed Eco's PR 347 holistically on commit hash bb573802 (merge commit to `main` branch) and concluded that all findings were addressed and no new vulnerabilities were identified.

2.1 Scope

The security review had the following components in scope for eco-routes on commit hash df431603:

```
├── contracts
│   └── prover
│       ├── BaseProver.sol
│       └── CCIPProver.sol
└── scripts
    └── DeployCCIPProver.s.sol
```

3 Findings

3.1 Informational

3.1.1 No 0-checks on incoming message

Severity: Informational

Context: CCIPProver.sol#L78

Description: The existing provers perform 0-checks on incoming data from the bridges to catch blatant errors. For example, in `handle()` in `HyperProver` and `MetaProver`.

```
// Verify origin and sender are valid
if (origin == 0) revert ZeroDomainID();

// Validate sender is not zero
if (sender == bytes32(0)) revert SenderCannotBeZeroAddress();
```

... and `lzReceive()` in LayerZeroProver:

```
if (origin.sender == bytes32(0)) {
    revert SenderCannotBeZeroAddress();
}
```

However, `CCIPProver[] performs no such checks in`ccipReceive‘.

Recommendation: For consistency, check for `senderAddress == address(0)` and `message.sourceChainSelector == 0`, and possibly also `messageId == bytes32(0)` in `ccipReceive()`.

Eco Foundation: Fixed in PR 348.

Cantina Managed: Fix verified.

3.1.2 Prover accepts too-large proof batch sizes

Severity: Informational

Context: CCIPProver.sol#L183

Description: The `encodedProofs` bytes array can be arbitrarily large. CCIP has a documented max data size (30 KB) and per□lane execution gas ceilings Currently, the Eco protocol relies on the prover to revert on too-large proof batches. With current router implementations, it would revert on the fee request:

0x12492154714fBD28F28219f6fc4315d19de1025B

```
if (dataLength > maxDataBytes) revert MessageTooLarge(maxDataBytes, dataLength);
```

Recommendation: Best practice would be to pre□check data sizes and reject oversized batches with a clear error.

Eco Foundation: Acknowledged.

Cantina Managed: Acknowledged. The limit has been documented and it's on callers to make sure they respect the limit, or face reverts.

3.1.3 The sender of proofs must ensure that the `allowOutOfOrderExecution` flag respects chain restrictions

Severity: Informational

Context: CCIPProver.sol#L189

Description: According to the CCIP documentation:

- Default value for `allowOutOfOrderExecution` varies by chain.
- Some chains enforce specific values and will revert if not set correctly.

It is on the caller of `prove()` to check that this is set correctly. Not all chains may guarantee that it can serve transactions in order. If the caller elects to try to enforce in-order execution, the EVM2EVMOnRamp contract, called by the router, may throw `ExtraArgOutOfOrderExecutionMustBeTrue()`.

Recommendation: Document prominently. Alternatively, set `allowOutOfOrderExecution` to a static `true` value. This is more permissive, and does not revert on any chain according to official documentation This could cause problems if fulfilled intents e.g. have non-standard behavior at the address that is being withdrawn to (e.g. the withdrawal is to a contract, and that contract expects a certain order of intent fulfilments).

Eco Foundation: Fixed in PR 351.

Cantina Managed: Fix verified.

3.1.4 The provided `gasLimit` is not checked against `MIN_GAS_LIMIT`

Severity: Informational

Context: CCIPProver.sol#L188

Description/Recommendation: The caller of `prove()` on the Portal can set any gas limit they want for the CCIP call. However, like in LayerZeroProver, it should be checked that it is at least as large as `MIN_GAS_LIMIT`.

Eco Foundation: Fixed in PR 352.

Cantina Managed: Fix verified.

3.1.5 Missing documentation for the new CCIP prover in various NatSpecs

Severity: Informational

Context: Inbox.sol#L100, Inbox.sol#L149, IProver.sol#L84

Summary: There are NatSpec comments explaining the various schemes of `chainId` used for different provers, listing all (currently) available provers in Inbox.sol and IProver.sol. CCIPProver should be added to these lists.

```
/**
 * @dev WARNING: sourceChainDomainID is NOT necessarily the same as chain ID.
 *      Each bridge provider uses their own domain ID mapping system:
 *      - Hyperlane: Uses custom domain IDs that may differ from chain IDs
 *      - LayerZero: Uses endpoint IDs that map to chains differently
 *      - Metalayer: Uses domain IDs specific to their routing system
 *      - Polymer: Uses chain IDs
 *      You MUST consult the specific bridge provider's documentation to determine
 *      the correct domain ID for the source chain.
 */
```

Eco Foundation: Fixed in PR 353.

Cantina Managed: Fix verified.