



Electisec

Origami - CowSwapper

Security Review Report

Prepared for: Origami

Audit Period: 2025-07-14 to 2025-07-15,
2025

Audit Performed By

Panda, Watermelon

Review Resources

<https://github.com/TempleDAO/origami/>

Commit Hash

5bf4ed59f9716416e7a793e31d2b6f1454c751d5

DISCLAIMER

This review is a code review to identify potential vulnerabilities in the code. The reviewers did not investigate security practices or operational security and assumed that privileged accounts could be trusted. The reviewers did not evaluate the security of the code relative to a standard or specification. The review may not have identified all potential attack vectors or areas of vulnerability.

Electisec and the auditors make no warranties regarding the security of the code and do not warrant that the code is free from defects. Electisec and the auditors do not represent nor imply to third parties that the code has been audited nor that the code is free from defects. By deploying or using the code, Origami and users of the contracts agree to use the code at their own risk.

Contents

Origami Review	3
1 Review Summary	3
2 Audited Files	3
3 Findings Explanation	3
4 Critical Findings	3
5 High Findings	3
6 Medium Findings	3
7 Low Findings	4
8 Gas Savings Findings	4
9 Informational Findings	4
9.1 Insufficient minimum validation for CoW Swap order expiry period	4
9.2 Reduce post hooks gas limit	4
9.3 <code>config.limitPriceAdjustmentBps</code> isn't validated when an oracle is provided	5

Origami Review

Origami provides Origami CowSwapper is used to swap farmed tokens.

1 Review Summary

The contracts of the Origami were reviewed over 2 days. 2 auditors performed the code review between 2025-07-14 and 2025-07-15, 2025. The repository was under active development during the review, but the review was limited to the latest commit at the start of the review. This was commit [5bf4ed59f9716416e7a793e31d2b6f1454c751d5](#) for the Origami repo.

2 Audited Files

```
origami
├── apps
│   └── protocol
│       ├── contracts
│           ├── common
│               └── swappers
│                   └── OrigamiCowSwapper.sol
```

3 Findings Explanation

Findings are broken down into sections by their respective impact:

- Critical, High, Medium, Low impact
 - These are findings that range from attacks that may cause loss of funds, impact control/ownership of the contracts, or cause any unintended consequences/actions that are outside the scope of the requirements.
 - Gas savings
 - Findings that can improve the gas efficiency of the contracts.
 - Informational
 - Findings including recommendations and best practices.
-

4 Critical Findings

None.

5 High Findings

None.

6 Medium Findings

None.

7 Low Findings

None.

8 Gas Savings Findings

None.

9 Informational Findings

9.1 Insufficient minimum validation for CoW Swap order expiry period

Technical Details

The `setOrderConfig()` function only checks that `expiryPeriodSecs` is non-zero:

```
1 if (config.expiryPeriodSecs == 0) revert CommonEventsAndErrors.ExpectedNonZero();
```

A value of 1 second is technically valid but impractical. The CoW Swap Watchtower polls every block, so orders with very short expiry periods will expire before they can be detected and executed.

Impact

Informational

Recommendation

A minimal value of 5 minutes seems more inline with a reasonable operational window for CoW Swap's infrastructure.

Developer Response

Acknowledged - The check is to ensure the caller has at least set it to something non-zero.

9.2 Reduce post hooks gas limit

Technical Details

The current post hook gas limit is set to 250,000 gas. Analysis of historical transactions from addresses `0x25dd72cc1de50e963948e464c09aeba0d9312349` and `0xeb32f71b700b75ad6bf84c9b0374f388740c266f` using Tenderly shows variable gas consumption based on the stalking contract used in the post hook:

- **SPK → USDS swaps:** ~60,000 gas
- **SKY → SKY swaps:** ~165,000 gas

In CoW Swap's fee structure, the full reserved gas amount (250,000) is deducted as a fee regardless of actual consumption, with no refund for unused gas. Overall yield is decreased due to inflated execution costs.

Impact

Informational.

Recommendation

Adjust the gas limits for the hooks.

Developer Response

Acknowledged.

9.3 `config.limitPriceAdjustmentBps` isn't validated when an oracle is provided

`OrigamiCowSwapper.setOrderConfig` is used by an authorized party to set crucial data to be used when creating an order for a given token.

`OrigamiSowSwapper.updateAmountsAndAdjustmentBps` may be used to update selected, existing order configurations.

Technical Details

When `config.limitPriceOracle != address(0)`, the `setOrderConfig()` method doesn't verify `config.limitPriceAdjustmentBps` to be within an expected range, whether within `[-100%, 100%]` or a stricter one.

Within `updateAmountsAndAdjustmentBps()`, the same holds: the

`limitPriceAdjustmentBps` is only validated to be non zero if `config.limitPriceOracle != address(0)`.

Impact

Informational.

Recommendation

Add validation within both mentioned methods to enforce the highlighted config value is within a selected range.

Developer Response

Updated in [PR#1937](#).