yAudit

November 24, 2025

Prepared for
Origami

Audited by
Panda
Watermelon

Origami - CowSwapper

Smart Contract Security Assessment

Contents

1 Review Summary

1.1 Protocol Overview

Origami CowSwapper is used to swap farmed tokens

1.2 Audit Scope

This audit covers 1 smart contracts totaling approximately 120 lines of code across two days of review.

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

1.3 Risk Assessment Framework

1.3.1 Severity Classification

1.4 Key Findings

Breakdown of Finding Impacts

Impact Level	Count
Critical	0
High	0
Medium	0
Low	0
Informational	3

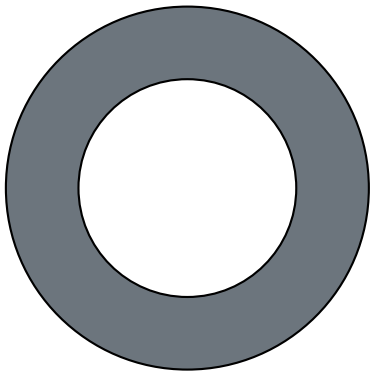

Figure 1: Distribution of security findings by impact level

1.5 Overall Assessment

No concerns

Severity	Description	Potential Impact
Critical	Immediate threat to user funds or protocol integrity	Direct loss of funds, protocol compromise
High	Significant security risk requiring urgent attention	Potential fund loss, major functionality disruption
Medium	Important issue that should be addressed	Limited fund risk, functionality concerns
Low	Minor issue with minimal impact	Best practice violations, minor inefficiencies
Undetermined	Findings whose impact could not be fully assessed within the time constraints of the engagement. These issues may range from low to critical severity, and although their exact consequences remain uncertain, they present a sufficient potential risk to warrant attention and remediation.	Varies based on actual severity
Gas	Findings that can improve the gas efficiency of the contracts.	Reduced transaction costs
Informational	Code quality and best practice recommendations	Improved maintainability and readability

Table 1: severity classification

2 Audit Overview

2.1 Project Information

Protocol Name: Origami
Repository: https://github.com/TempleDAO/origami/
Commit Hash: 5bf4ed59f9716416e7a793e31d2b6f1454c751d5
Commit URL:
https://github.com/TempleDAO/origami/blob/5bf4ed59f9716416e7a793e31d2b6f1454c751d5

2.2 Audit Team

Panda, Watermelon

2.3 Audit Resources

Category	Mark	Description
Access Control	High	good

Table 2: Code Evaluation Matrix

3 Origami Review

Origami provides Origami CowSwapper is used to swap farmed tokens.

3.1 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.

3.2 Audited Files

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

3.3 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.

3.4 Critical Findings

None.

3.5 High Findings

None.

3.6 Medium Findings

None.

3.7 Low Findings

None.

3.8 Gas Savings Findings

None.

3.9 Informational Findings

3.9.1 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.

3.9.2 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.

3.9.3 `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.