Electisec Origami hOHM Migrator Review

Review Resources:

- contract

Auditors:

- Adriro
- Panda

Table of Contents

Review Summary

Origami hOHM Migrator

The Origami hOHM Migrator is a specialized contract designed to consolidate various loans into a single hOHM position.

The contracts of the Origami hOHM Migrator Repo were reviewed over two days. Two auditors performed the code review between March 18th and 19th, 2025. The repository was under active development during the review, but the review was limited to the latest commit 6c8c13a02377686b537a236e0335d2234a2fab63 of the Origami hOHM Migrator repo.

Scope

The scope of the review consisted of the following contracts at the specific commit:

```
apps/protocol/contracts/investments/olympus/OrigamiCoolerMigrator.sol
```

After the findings were presented to the Origami hOHM Migrator team, fixes were made and included in several PRs.

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 hOHM Migrator and users of the contracts agree to use the code at their own risk.

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.

Critical Findings

None.

High Findings

None.

Medium Findings

None.

Low Findings

None.

Gas Saving Findings

1. Gas - Unnecessary safe casts

Several occurrences where a value of type `uint256` is safe-cast to `uint128` and stored back into a `uint256`.

Technical Details

- OrigamiCoolerMigrator.sol#L279
- OrigamiCoolerMigrator.sol#L280
- OrigamiCoolerMigrator.sol#L317
- OrigamiCoolerMigrator.sol#L385
- OrigamiCoolerMigrator.sol#L386

Impact

Gas savings.

Recommendation

Remove the type cast.

Developer Response

Fixed in 2b076d1c.

2. Gas - Replace _ `clearinghouses` with three constants

Technical Details

The contract `OrigamiCoolerMigrator` currently uses a struct AllClearinghouses to store three constant addresses representing different clearinghouse versions:

```
AllClearinghouses private _clearinghouses;

constructor(
    // ... other parameters
    address[3] memory clearinghouses_ // v1.1, v1.2, v1.3 in order
) OrigamiElevatedAccess(initialOwner_) {
    // ... other initializations
    _clearinghouses = AllClearinghouses(clearinghouses_[0], clearinghouses_[1],
```

```
clearinghouses_[2]);
}
```

Impact

Gas savings

Recommendation

```
// ... existing code ...

/// @notice Clearing houses for different cooler versions
address private immutable clearinghouse_v1_1;
address private immutable clearinghouse_v1_2;
address private immutable clearinghouse_v1_3;

constructor(
    // ... other parameters
    address[3] memory clearinghouses_ // v1.1, v1.2, v1.3 in order
) OrigamiElevatedAccess(initialOwner_) {
    // ... other initializations
    clearinghouse_v1_1 = clearinghouses_[0];
    clearinghouse_v1_2 = clearinghouses_[1];
    clearinghouse_v1_3 = clearinghouses_[2];
}

// ... existing code ...

/// @inheritdoc IOrigamiCoolerMigrator
function getClearinghouses() external override view returns (AllClearinghouses
memory) {
    return AllClearinghouses(clearinghouse_v1_1, clearinghouse_v1_2,
clearinghouse_v1_3);
}

// ... update all other references from _clearinghouses.v1_1 to clearinghouse_v1_1,
etc.
```

Developer Response

Fixed in ed04e6.

Informational Findings

None.

Final Remarks

Electisec reviewed the Origami Cooler migrator mechanism and found no major security issues or vulnerabilities in its implementation.