cow-shed Update Audit

Gnosis Ltd - Report by Côme du Crest

2025-08-26

Table of contents

cow-shed Update Audit

This document presents the findings of a smart contract audit conducted by Côme du Crest for Gnosis Ltd.

Scope

The scope includes all contracts within `cowdao-grants/cow-shed` as of commit `0x1f2c7c4`.

The code was already audited on commit `0x0f7eaf4` and `the diff` from the last commit has been the focus of this review.

Fixes have been implemented and reviewed in commit `0x4acdd9c`.

The contracts version had been bumped to 2.0.0 in commit `0xbbfba8c` which have also been reviewed.

Context

The contracts enable pre or post hooks for CoWSwap orders to be executed via a user-controlled ERC1967 proxy.

The `GPv2Settlement` contract called by CoW solvers starts interactions with `GPv2Interaction.execute()` which calls the `HooksTrampoline` contract that forwards the hook with limited gas and does not revert on failure. This gates the batch execution contract from the hook execution.

`COWShed` then verifies that the hook data has been signed by the admin (that could be a contract) before executing the call provided by the hook data.

`COWShed` provides for an authorized function `trustedExecuteHooks()` that executes hooks without verifying signatures. This function can be called by the proxy factory or the admin only.

Status

The report has been sent to the core developer.

Fixes have been implemented and no issue remains.

Legal Information And Disclaimer

1. This report is based solely on the information provided by CoW (the “Company”), with the assumption that the information provided to Gnosis is authentic, accurate, complete, and not misleading as of the date of this report. Gnosis has not conducted any independent enquiries, investigations or due diligence in respect of the Company, its business or its operations.
2. Changes to the information contained in the documents, repositories and any other materials referenced in this report might affect or change the analysis and conclusions presented. Gnosis is not responsible for monitoring, nor will we be aware of, any future additions, modifications, or deletions to the audited code. As such, Gnosis does not assume any responsibility to update any information, content or data contained in this report following the date of its publication.
3. This report does not address, nor should it be interpreted as addressing, any regulatory, tax or legal matters, including but not limited to: tax treatment, tax consequences, levies, duties, data privacy, data protection laws, issues relating to the licensing of information technology, intellectual property, money laundering and countering the financing of terrorism, or any other legal restrictions or prohibitions. Gnosis disclaims any liability for omissions or errors in the findings or conclusions presented in this report.
4. The views expressed in this report are solely our views regarding the specific issues discussed within this report. This report is not intended to be exhaustive, nor should it be construed as an assurance, guarantee or warranty that the code is free from bugs, vulnerabilities, defects or deficiencies. Different use cases may carry different risks, and integration with third-party applications may introduce additional risks.
5. This report is provided for informational purposes only and should not be used as the basis for making investment or financial decisions. This report does not constitute investment research and should not be viewed as an invitation, recommendation, solicitation or offer to subscribe for or purchase any securities, investments, products or services. Gnosis is not a financial advisor, and this report does not constitute financial or investment advice.
6. The statements in this report should be considered as a whole, and no individual statement should be extracted or referenced independently.
7. To the fullest extent permitted by applicable laws, Gnosis disclaims any and all other liability, whether in contract, tort, or otherwise, that may arise from this report or the use thereof.

Issues

[Low] Inconsistent proxy storage layout with 1.0.0

Summary

The version 2.0.0 of `COWShed` adds a storage variable `preSignStorage` in the middle of its state struct which makes it incompatible with the existing 1.0.0 deployed proxies.

Vulnerability Detail

The `COWShedStorage` inherited by `COWShed` uses a pre-computed storage slot to store its state, however the `State` struct has been updated to add `preSignStorage` before `nonces`:

```
contract COWShedStorage {
    ...

    struct State {
        bool initialized;
        address trustedExecutor;
+       IPreSignStorage preSignStorage;
        LibBitmap.Bitmap nonces;
    }

    bytes32 internal constant STATE_STORAGE_SLOT = keccak256("COWShed.State");

    function _state() internal pure returns (State storage state) {
        bytes32 stateSlot = STATE_STORAGE_SLOT;
        assembly {
            state.slot := stateSlot
        }
    }
```

Impact

Pre-existing COWShed proxies with version 1.0.0 or 1.0.1 are incompatible with the newer version.

Code Snippets

https://github.com/cowdao-grants/cow-shed/blob/1f2c7c42de71c3259185a2e37e19527b4e996c61/src/COWShedStorage.sol#L23

Recommendation

Move the `preSignStorage` value after `nonces`.

Response

This has been fixed in commit `0x4acdd9c`