Protocol Audit Report

Version 1.0

Audit Mandalorian

April 22, 2025

Protocol Audit Report

Audit Mandalorian

April 19, 2025

Prepared by: Cyfrin
Lead Auditors: - Din Audit (Senior Security Researcher) - Cara Dune (Solidity Specialist) - IG-88 (Automated Scanner Engineer)

Table of Contents

- Table of Contents
- Protocol Summary
- Disclaimer
- Risk Classification
- Audit Details
  - Scope
  - Roles
- Executive Summary
  - Issues found
- Findings
- High
  - Reentrancy Risk in `withdrawStake()`
- Medium
  - Incorrect Chainlink Price Feed Address
  - Missing Access Control for `mintCowNFT()`

- Low
  - Event Emission Missing on State Changes
- Informational
- Gas

Protocol Summary

CowSwap is a decentralized cattle exchange platform that tokenizes livestock assets as NFTs on Ethereum. Users can stake, trade, and redeem cattle-backed tokens representing real-world cows. The protocol uses Chainlink oracles to verify location, weight, and health metadata for each animal.

Disclaimer

The Audit Mandalorian team makes all effort to find as many vulnerabilities in the code in the given time period, but holds no responsibilities for the findings provided in this document. A security audit by the team is not an endorsement of the underlying business or product. The audit was time-boxed and the review of the code was solely on the security aspects of the Solidity implementation of the contracts.

Risk Classification

		Impact		
		High	Medium	Low
Likelihood	High	H	H/M	M
	Medium	H/M	M	M/L
	Low	M	M/L	L

We use the CodeHawks severity matrix to determine severity. See the documentation for more details.

Audit Details

Scope

The audit covered the following contracts and files: - `CowToken.sol` - `StakingManager.sol` - `CowRegistry.sol` - `ChainlinkOracleConsumer.sol` - `CowNFT.sol`

Timeframe: April 12–18, 2025
Tools used: Slither, Foundry, MythX, Manual review

Roles

Address	Role
0xA1b2…D3f4	Protocol Owner
0x1234…5678	Cow NFT Minter
0x9aBc…Ef90	Oracle Operator

Executive Summary

The audit team performed a comprehensive review of the CowSwap protocol's smart contracts. The contracts follow best practices in most cases, but several issues were identified that could impact security or performance.

A critical bug in `StakingManager.sol` could allow early withdrawal without penalty, and an incorrect price feed configuration may cause undercollateralization of NFT tokens. Overall, the protocol is well structured but requires fixes for medium and high-severity issues before launch.

Issues found

- 1 High
- 2 Medium
- 1 Low
- 2 Informational
- 1 Gas optimization

Findings

High

Reentrancy Risk in `withdrawStake()`

File: `StakingManager.sol`
Impact: High
Description: The function `withdrawStake()` updates user balances after an external call to the cow NFT contract. This creates a reentrancy risk that could be exploited to withdraw multiple times.
Recommendation: Apply the checks-effects-interactions pattern to move state updates before external calls.

Medium

Incorrect Chainlink Price Feed Address

File: `ChainlinkOracleConsumer.sol`
Impact: Medium
Description: The configured address for the cow weight price feed points to the ETH/USD aggregator.
Recommendation: Verify and update to the appropriate cow weight data feed address.

Missing Access Control for `mintCowNFT()`

File: `CowNFT.sol`
Impact: Medium
Description: Anyone can call `mintCowNFT()` and mint unlimited NFTs.
Recommendation: Restrict function to only allow calls from the protocol owner or authorized role.

Low

Event Emission Missing on State Changes

File: `CowRegistry.sol`
Impact: Low
Description: Critical functions modify mappings without emitting events, making off-chain tracking

difficult.

Recommendation: Emit events when registering or updating cow data.

Informational

- Code lacks NatSpec comments
- Unused variable `pendingWithdrawals` in `StakingManager.sol`

Gas

- Replace `require(x == true)` with `require(x)` in multiple locations
- Consider using `unchecked` math where overflows are impossible
