
# AUDIT REPORT

## Capapult Finance Oracle Contract

Prepared by SCV-Security

On 4th December 2023

# Table of Contents

# Introduction

SCV has been engaged by Capapult Finance to conduct a comprehensive security review with the goal of identifying potential security threats and vulnerabilities within the codebase. The purpose of this audit is to evaluate the security posture of the codebase and provide actionable recommendations to mitigate any identified risks. This report presents an overview of the findings from our security audit, outlining areas of concern and proposing effective measures to enhance the codebase's security.

## Scope Functionality

The oracle contract allows the CAPA Money Market to have multiple price feeds, such as Astroport LP tokens, liquid staking derivatives, and normal price feeds. This is accomplished by defining multiple price sources and handling them appropriately to compute the asset price.

## Submitted Codebase

| Oracle Contract | |
|---|---|
| **Repository** | https://github.com/capapult-finance/capa-money-market/tree/main/contracts/oracle |
| **Commit** | `64eca48ddfeb199e3df41d3f2668c37b0d42b75c` |
| **Branch** | `main` |

## Revisions Remediation

| Oracle Contract | |
|---|---|
| **Repository** | https://github.com/capapult-finance/capa-money-market/tree/main/contracts/oracle |
| **Commit** | `ba14154953faecd45536286310cf092b75a4bc0c` |
| **Branch** | `main` |

## Methodologies

SCV performs a combination of automated and manual security testing based on the scope of testing. The testing performed is based on the extensive experience and knowledge of the auditor to provide the greatest coverage and value to Capapult Finance Testing includes, but is not limited to, the following:

- Understanding the application and its functionality purpose.
- Deploying SCV in-house tooling to automate dependency analysis and static code review.
- Analyse each line of the code base and inspect application security perimeter.
- Review underlying infrastructure technologies and supply chain security posture.

## Code Criteria

This section provides an evaluation of specific criteria aspects as described below:

- **Documentation:** Evaluating the presence and comprehensiveness of publicly available or provided explanatory information, diagram flowcharts, comments, and supporting documents to enhance code understanding.
- **Coverage:** Evaluating whether the code adequately addresses all necessary cases and scenarios, ensuring that the intended functionality or requirements are sufficiently covered.
- **Readability:** Assessing how easily the code can be understood and maintained, considering factors such as code structure, naming conventions, and overall organisation.
- **Complexity:** Evaluating the complexity of the code, including factors such as, number of lines, conditional statements, and nested structures.

The status of each criteria is categorised as either **SUFFICIENT** or **NOT-SUFFICIENT** based on the audit assessment. This categorisation provides insights to identify areas that may require further attention and improvement.

| Criteria | Status | Notes |
|---|---|---|
| **Documentation** | **SUFFICIENT** | The documentation provided by the client provides sufficient coverage of the codebase |
| **Coverage** | **SUFFICIENT** | Testing coverage is considered sufficient, although there is room for improvement as the current coverage only extends to 39.67% of the code. |
| **Readability** | **SUFFICIENT** | The codebase had good readability overall and utilised many Rust and CosmWasm best practices. |
| **Complexity** | **SUFFICIENT** | N/A |

# Findings Summary

| Summary Title | Risk Impact | Status |
|---|---|---|
| Missing address validation | INFO | RESOLVED |
| Remove default response | INFO | RESOLVED |
| Remove commented code blocks | INFO | RESOLVED |
| Remove deprecated function | INFO | ACKNOWLEDGED |
| `run_update_config` will silently perform a no-op | INFO | RESOLVED |

# Findings Technical Details

## 1. Missing address validation

| RISK IMPACT: INFO | STATUS: RESOLVED |
| --- | --- |

### Description

In `contracts/oracle/src/execute.rs:57`, the `feeder` is not validated to be a valid address. It is best practice to ensure all addresses are validated before they are stored in the contract state.

### Recommendation

Consider modifying the function to use `addr_validate` to validate the address.

## 2. Remove default response

| RISK IMPACT: INFO | STATUS: RESOLVED |
| --- | --- |

### Description

The `run_update_config` function in `contracts/oracle/src/execute.rs:30` returns a default response and does not emit any attributes. It is best practice to emit detailed attributes whenever a state change occurs.

### Recommendation

Consider emitting relevant events or attributes based on configured parameters.

### 3. Remove commented code blocks

| RISK IMPACT: INFO | STATUS: RESOLVED |
| --- | --- |

## Description

In `contracts/oracle/src/functions.rs:45` there is a commented code block. It is best practice to remove commented code blocks to improve the readability and maintainability of the codebase.

## Recommendation

Consider removing the code block if it is unused.

## 4. Remove deprecated function

| RISK IMPACT: INFO | STATUS: ACKNOWLEDGED |
| --- | --- |

### Description

The audited code frequently uses the `cosmwasm_std::serde to_binary` function.

It is suggested to use `to_json_binary` instead.

### Recommendation

Consider modifying the codebase to use `to_json_binary`.

## 5. `run_update_config` will silently perform a no-op

| RISK IMPACT: INFO | STATUS: RESOLVED |
| --- | --- |

### Description

In the `run_update_config` in `contracts/oracle/src/execute.rs:15`, the owner can supply None value. This will keep the current owner and silently pass in the function. For state-changing functionality, it is best practice to error when a change is not committed.

### Recommendation

Consider returning an error when a change is not committed.

# Document Control

| Version | Date | Notes |
|---|---|---|
| - | 23rd November 2023 | Security audit commencement date. |
| **0.1** | 30th November 2023 | Initial report with identified findings delivered. |
| **0.5** | 1st December 2023 | Fixes remediations implemented and reviewed. |
| **1.0** | 4th December 2023 | Audit completed, final report delivered. |

# Appendices

## A. Appendix – Risk assessment methodology

SCV-Security employs a risk assessment methodology to evaluate vulnerabilities and identified issues. This approach involves the analysis of both the LIKELIHOOD of a security incident occurring and the potential IMPACT if such an incident were to happen. For each vulnerability, SCV-Security calculates a risk level on a scale of 5 to 1, where 5 denotes the highest likelihood or impact. Consequently, an overall risk level is derived from combining these two factors, resulting in a value from 10 to 1, with 10 signifying the most elevated level of security risk

| Risk Level | Range |
|---|---|
| CRITICAL | 10 |
| SEVERE | From 9 to 8 |
| MODERATE | From 7 to 6 |
| LOW | From 5 to 4 |
| INFORMATIONAL | From 3 to 1 |

**LIKELIHOOD** and **IMPACT** would be individually assessed based on the below:

| Rate | LIKELIHOOD | IMPACT |
|---|---|---|
| **5** | **Extremely Likely** | Could result in severe and irreparable consequences. |
| **4** | **Likely** | May lead to substantial impact or loss. |
| **3** | **Possible** | Could cause partial impact or loss on a wide scale. |
| **2** | **Unlikely** | Might cause temporary disruptions or losses. |
| **1** | **Rare** | Could have minimal or negligible impact. |

## B. Appendix – Report Disclaimer

This report should not be regarded as an "endorsement" or "disapproval" of any specific project or team. These reports do not indicate the economics or value of any "product" or "asset" created by a team or project that engages SCV-Security for a security review. The audit report does not make any statements or warranties about the code's utility, safety, suitability of the business model, regulatory compliance of the business model, or any other claims regarding the fitness of the implementation for its purpose or its bug-free status. The audit documentation is intended for discussion purposes only. The content of this audit report is provided "as is," without representations and warranties of any kind, and SCV-Security disclaims any liability for damages arising from or in connection with this audit report. Copyright of this report remains with SCV-Security.

# THANK YOU FOR CHOOSING

SCV SECURITY

scv.services

contact@scv.services