Apollo DAO - Steak Contracts - Audit Report

Prepared for Apollo DAO, 18 August 2022

Table of Contents

Introduction

SCV was engaged by Apollo DAO to assist in identifying security threats and vulnerabilities that have the potential to affect their security posture. Additionally, SCV will assist the team in understanding the risks and identifying potential mitigations.

Scope

SCV performed the security assessment on the following codebase:

- https://github.com/apollodao/steak-contracts/tree/dev/vault-token-abstraction
- Code Freeze: *6f385c78c884b0b4ab480eac2a118549fc8b4e59*

Remediations were applied into the following commits:

- https://github.com/apollodao/steak-contracts/commit/3b59e47747ae446cdf5b9f0101dcabfc1e3d00b6
- https://github.com/apollodao/steak-contracts/commit/798b30b3c5b78948acd3a382ef0ff2adb2f08f4e

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 Apollo DAO. Testing includes, but is not limited to, the following:

- Understanding the application and its code base 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 and Test Coverage

SCV used a scale from **0** to **10** that represents how **SUFFICIENT(6-10)** or **NOT SUFFICIENT(0-5)** each code criteria was during the assessment:

Criteria	Status	Scale Range	Notes
Provided Documentation	**Sufficient**	7-8	N/A
Code Coverage Test	**Sufficient**	7-8	N/A
Code Readability	**Sufficient**	8-9	N/A
Code Complexity	**Sufficient**	6-7	N/A

Vulnerabilities Summary

	Title and Summary	Risk	Status
1	Add additional validation during instantiation to prevent misconfigurations	**Informational**	**Remediated**
2	Overflow checks not set for release profile	**Informational**	**Remediated**
3	Remove print statements	**Informational**	**Remediated**
4	compute_redelegations_for_removal call is unnecessary if validator has 0 delegation	**Informational**	**Remediated**

Detailed Vulnerabilities

1. Add additional validation during instantiation to prevent misconfigurations

Likelihood	Impact	Risk
Unlikely	Informational	**Informational**

Description

The initial parameters in *instantiate* in *steak-contracts/packages/steak/src/execute.rs:27* are lacking some validation that could be implemented to reduce the risk of a misconfigured instantiation state. *msg.validators* is a vector of validator addresses, but it is not deduplicated. In addition *performance_fee* in *contracts/packages/steak/src/execute.rs:61* should be checked to ensure that its value does not exceed a specified maximum value or at least that it does not exceed *1.0*.

We classify this finding as informational because these misconfigurations are unlikely as only the owner can introduce this misconfiguration.

Recommendations

We recommend ensuring that the vector of validators is deduplicated before saving. In addition, we recommend enforcing a maximum value for *performance_fee*.

2. Overflow checks not set for release profile

Likelihood	Impact	Risk
Unlikely	Informational	**Informational**

Description

Even though this check is implicitly applied to all packages from the workspace's Cargo.toml, we recommend also explicitly enabling overflow checks in every individual package. This helps prevent unintended consequences when the codebase is refactored in the future.

Recommendations

We recommend explicitly enabling overflow checks in both contracts/osmosis_hub/Cargo.toml and contracts/cw20_hub/Cargo.toml.

3. Remove print statements

Likelihood	Impact	Risk
Unlikely	Informational	**Informational**

Description

In *steak-contracts/packages/steak/src/execute.rs:214* and *215* there are two print statements that should be removed before the code is moved to production. While print statements will have no negative impact on the validator nor will they present any non-determinism issues, it is best practice to remove all print statements.

Recommendations

We recommend removing the print statements in *steak-contracts/packages/steak/src/execute.rs:214* and *215*. We recommend adding the specified values to the *InvalidCoinSent* error rather than printing them to standard output.

4. compute_redelegations_for_removal call is unnecessary if validator has 0 delegation

Likelihood	Impact	Risk
Unlikely	Informational	**Informational**

Description

The *remove_validator* in *steak-contracts/packages/steak/src/execute.rs:757* removes a validator specified by the contract's owner and redelegates that validator's delegations in *compute_redelegations_for_removal*. This is first done by querying the validator's current delegation with *query_delegation*. If the validator has a *delegation_to_remove* of *0* the *remove_validator* function does not need to call the *compute_redelegations_for_removal* function. We classify this as unlikely due to the fact that only the owner can call this function and that the edge case is unlikely to occur.

Recommendations

We recommend checking that *delegation_to_remove* is not 0 before calling the *compute_redelegations_for_removal* function.

Document control

Document changes

Version	Date	Name	Changes
0.1	2022-08-12	Vinicius Marino	Initial report
0.2	2022-08-13	Vinicius Marino	Team communication and Pre-Release
1.0	2022-08-18	Vinicius Marino	Remediations Review and Document Release

Document contributors

Name	Role	Email address
Vinicius Marino	Security Specialist	vini@scv.services

Appendices

Appendix A: Report Disclaimer

The content of this audit report is provided “As is”, without representations and warranties of any kind.

The author and their employer disclaim any liability for damage arising out of, or in connection with, this audit report.

Appendix B: Risk assessment methodology

A qualitative risk assessment is performed on each vulnerability to determine the impact and likelihood of each.

Risk rate will be calculated on a scale. As per criteria Likelihood vs Impact table below:

Impact \ Likelihood	Rare	Unlikely	Possible	Likely
Critical	Medium	High	Critical	Critical
Severe	Low	Medium	High	High
Moderate	Low	Medium	Medium	High
Low	Low	Low	Low	Medium
Informational	Informational	Informational	Informational	Informational

LIKELIHOOD:

- **Likely**: likely a security incident will occur;
- **Possible**: It is possible a security incident can occur;
- **Unlikely**: Low probability a security incident will occur;
- **Rare**: In rare situations, a security incident can occur;

IMPACT:

- **Critical**: May cause a significant and critical impact;
- **Severe**: May cause a severe impact;
- **Moderate**: May cause a moderated impact;
- **Low**: May cause low or none impact;
- **Informational**: May cause very low impact or none.


SCV