Talis Protocol

Candy Machine Contract

Audit Report

Prepared for Talis, 9th June 2023

Table of Contents

Introduction

SCV was engaged by Talis Protocol 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/Talis-Art/talis_contracts_v2/tree/holger/poc_on_chain_randomness/contracts/cw721-badge
    - Code Freeze: *96a5df15b6b792acd735f0abd43ad1fc7cc3c5e5*
- https://github.com/Talis-Art/talis_contracts_v2/tree/holger/poc_on_chain_randomness/contracts/badge-minter
    - Code Freeze: *c6bfaf639e3e3bb3c2bde2a823d5c5564371c7e0*

SCV performed the revisions on suggested recommendations applied by Talis team up to the following commit hash:

- https://github.com/Talis-Art/talis_contracts_v2/commit/6a1aac43e30026ab61bbaa9303a167e25f970058

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 Talis Protocol. 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

This section below represents how *SUFFICIENT* or *NOT SUFFICIENT* each code criteria was during the assessment

Criteria	Status	Notes
Provided Documentation	**SUFFICIENT**	Detailed documentation was not provided; however, the in-line comments were deemed sufficient given the relatively low complexity of the audit scope
Code Coverage Test	**SUFFICIENT**	Testing coverage is considered sufficient, although there is room for improvement as the current coverage only extends to 48.45% of the code.
Code Readability	**SUFFICIENT**	The codebase had good readability overall and utilised many Rust and CosmWasm best practices.
Code Complexity	**SUFFICIENT**	N/A

Vulnerabilities Summary

#	Summary Title	Risk Impact	Status
1	Funds can become temporarily or permanently locked in contract	**Severe**	**Resolved**
2	Misconfiguration during instantiation will prevent future phase updates	**Medium**	**Resolved**
3	Replace Magic Numbers	**Informational**	**Resolved**
4	Inconsistent contract naming	**Informational**	**Resolved**
5	Remove commented code blocks	**Informational**	**Acknowledged**

Detailed Vulnerabilities

1 - Funds can become temporarily or permanently locked in contract

---

Risk Impact: Severe - **Status**: Resolved

Description

The `execute_withdraw` function in `contracts/badge-minter/src/contract.rs:330-408` iterates over all funds defined across all phases of the candy machine and constructs sub-messages to send these funds to the specified owner after querying their amounts. If any of the native funds specified have an amount of `0`, the message message will return an error when it is dispatched to the CosmosSDK Bank module. This is also possible for CW20 depending on the implementation of the CW20 contract and how it handles transfers with `0` amounts. If any error is encountered during this operation, then it will revert the transaction and will effectively block all withdrawals until the specific coin has a balance greater than `0`.

Most situations where this error can occur are recoverable. For example, the owner could wait until all phases are completed, or they could send the specific denom to the contract. A possible edge case that may permanently lock funds in the contract is if any of the coins specified as the payment token for a phase don’t exist or don't have send mode enabled.

Recommendations

We recommend updating the `execute_withdraw` function to first check the balances returned by the `query_contract_balance` queries, and construct messages for the balances that are greater than 0.

2 - Misconfiguration during instantiation will prevent future phase updates

---

Risk Impact: Medium - **Status**: Resolved

Description

In the `instantiate` function in `contracts/badge-minter/src/contract.rs:35`, there is no validation to ensure that phase types are properly supplied. For example, a private phase can be supplied as `msg.public_phase`. This would allow for misconfigurations to be introduced during the instantiation and during phase updates.

This can also present a situation where overlapping phases are potentially introduced. For example, if a private phase was supplied in `msg.public_phase`, its value would be retained in `line 74` so it would also be duplicated in the `CANDY_MACHINE`'s private phases. This would bypass the `check_no_overlap` validation.

Additionally, this will block future phase updates by the owner. While the initial duplication will pass during the instantiation, the validations present in `update_private_phase` and `update_public_phase` will error due to the overlapping phases.

We specify this finding as medium severity because only the owner can introduce this misconfiguration but it could potentially block important phase updates that the owner may need to make.

Recommendations

We recommend performing a validation during the instantiation that verifies that phases are the intended type before saving.

3 - Replace Magic Numbers

Risk Impact: Informational - **Status**: Resolved

Description

In the `calculate_withdraw_balance` function in `contracts/badge-minter/src/contract.rs:481` and `497` there are magic numbers used to represent the Talis protocol fee. It is best practice to create variables that represent this fee.

Recommendations

We recommend replacing the magic numbers mentioned above with a constant that is descriptive of its value and use case.

4 - Inconsistent contract naming

Risk Impact: Informational - **Status**: Resolved

Description

The `CONTRACT_NAME` of the badge-minter contract is specified as *“poc-candy-mint”*. This appears to have been copied from `contracts/poc-candy-mint/src/contract.rs:33`.

Recommendations

We recommend keeping the naming of this contract consistent and not duplicating the name of an existing contract within the project repository.

5 - Remove commented code blocks

Risk Impact: Informational - **Status**: Acknowledged

Description

There are a number of commented code blocks within the scope of this audit. It is best practice to remove these non implemented codeblocks before the contracts are released to improve the readability and maintainability of the codebase.

Recommendations

We recommend keeping the naming of this contract consistent and not duplicating the name of an existing contract within the project repository.

Document control

Version	Date	Approved by	Changes
0.1	1st June 2023	Vinicius Marino	Document Pre-Release
0.2	8th June 2023	SCV Team	Remediation Revisions
1.0	9th June 2023	Vinicius Marino	Document Release

Appendices

A. Appendix – 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:

	Rare	Unlikely	Possible	Likely
Critical	**Medium**	**Severe**	**Critical**	**Critical**
Severe	**Low**	**Medium**	**Severe**	**Severe**
Moderate	**Low**	**Medium**	**Medium**	**Severe**
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.

B. Appendix – Report Disclaimer

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts SCV-Security to perform a security review. The audit makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. The audit documentation is 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 damage arising out of, or in connection with, this audit report.