Terraformer - Staking Contract - Audit Report

Prepared for Terraformer, 5 May 2022

Table of Contents

Introduction

SCV was engaged by Terraformer 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/Terra-Former/audit_config_staking

Code freeze hash: *a6877e48bf69488b6343d2c45915589fa0d1a916*

Remediations were applied in the following hash: *b28709deab2a0242a0415e6700346bbabe430f79*.

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 Terraformer. 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	**Not Sufficient**	3-4	N\A
Code Coverage Test	**Not Sufficient**	3-4	N\A
Code Readability	**Sufficient**	7-8	N\A
Code Complexity	**Sufficient**	6-7	N\A

Vulnerabilities Summary

	Title and Summary	Risk	Status
1	During migration provided migration values can be lost	**Low**	**Remediated**
2	Lack of validations on migrate/instantiate functionality	**Low**	**Remediated**
3	Contract might run out-of-gas due STAKER_INFO storage size growth	**Informational**	**Acknowledged**
4	Epoch time is calculated using past blocks heights	**Informational**	**Remediated**
5	Reward Token must be the same as Staking token	**Informational**	**Remediated**
6	Specify error handlers with the appropriate handlers	**Informational**	**Remediated**

Detailed Vulnerabilities

1. During migration provided migration values can be lost

Likelihood	Impact	Risk
Possible	Low	**Low**

Description

In the event an OldConfig can't be loaded most of the migration msg values are lost. In the event an OldState can't be loaded there is no logic to set a new `total_bond_amount` or `global_reward_time` .

Recommendations

Update the migration function to ensure the desired values are used.

2. Lack of validations on migrate/instantiate functionality

Likelihood	Impact	Risk
Rare	Low	**Low**

Description

There is no validation on the `migrate`/`instantiate` function of the contract other than the existence of a previous config. During migration a human error or a wrong copy paste by the contract admin could lead to major consequences.

As an example, `penalty_payout_address: msg.penalty_payout_address` lacks address validation.

Recommendations

Enforce validation on all fields upon migration/instantiate.

3. Contract might run out-of-gas due STAKER_INFO storage size growth

Likelihood	Impact	Risk
Rare	Informational	**Informational**

Description

In the `contracts/lp_staking/src/executions.rs#L248` is noted that on every invocation of the `autostake` an unbounded range call is made to the `STAKER_INFO` struct which is then iterated on, one-by-one performing a number of computation functions and the re-saving of staker and state values.

If the list of stakers growth arbitrarily in size this may present gas errors on invocations of the autostake.

Recommendations

We recommend trim down STAKER_INFO and control its growth in size. This might be an edge case that requires further coverage testing.

4. Epoch time is calculated using past blocks heights

Likelihood	Impact	Risk
Rare	Informational	**Informational**

Description

By design Terraformer computes reward from past blocks heights which does not imply a direct security risk since it's securely implemented. However, using `time.seconds()` approach would be a more reliable source of measuring elapsed time.

Recommendations

SCV suggests the use of `env.block.time.seconds()` to calculated elapsed time.

5. Reward Token must be the same as Staking token

Likelihood	Impact	Risk
Rare	Informational	**Informational**

Description

In the `/lp_staking/src/executions.rs#241` autostake function, it appears that one of the two main checks is checking two config values against each other which is inefficient.

Recommendations

Enforce such checks on the instantiation of the contract or in a update_config method where the values are set.

6. Specify error handlers with the appropriate handlers

Likelihood	Impact	Risk
Possible	Informational	**Informational**

Description

The contract uses a generic error handler `StdError::generic_err` for all catchers which is not advised as a best practice. By using a generic error handler for all errors catchers impacts code readability and quality of the codebase.

Recommendations

We recommend the use of `ContractError` whenever a contract error is raise.

Document control

Document changes

Version	Date	Name	Changes
0.1	2022-04-19	Vinicius Marino	Initial report
0.2	2022-04-22	Vinicius Marino	Team communication and Pre-Release
1.0	2022-05-05	Vinicius Marino	Report 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.

Copyright of this report remains with the author.

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:

Likelihood Impact	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