TFM - Router Contract - Audit Report

Prepared for TFM, 7 June 2022

Table of Contents

Introduction

SCV was engaged by TFM 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/tfm-com/audit_splitting_router

Code freeze hash: *20eb3e2af1fc57c0acdd13c86bd63d7563b3d5ad*

Remediations were applied into the following hash: *ff13920e9d8102ed56716345e93a87bc10adc56d*.

TFM also upgrade contracts to suit Terra 2.0 on the following codehash:

- *b4f2b412ec75d98afe43865dfef497b6cc4f4715*.

The implementation was found to be secured.

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 TFM. 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**	6-7	N\A
Code Coverage Test	**Sufficient**	6-7	operations.rs lacks coverage
Code Readability	**Sufficient**	7-8	N\A
Code Complexity	**Sufficient**	6-7	N\A

Vulnerabilities Summary

	Title and Summary	Risk	Status
1	Duplicate handler in ExecuteMsg for pair in tfm package	**Informational**	**Acknowledged**
2	Lack of test Coverage on router operations	**Informational**	**Acknowledged**
3	Specify error handlers with the appropriate handlers	**Informational**	**Remediated**
4	Unused variables and functions in code	**Informational**	**Remediated**

Detailed Vulnerabilities

1. Duplicate handler in ExecuteMsg for pair in tfm package

Likelihood	Impact	Risk
Unlikely	Informational	**Informational**

Description

The `tfm` package was included with the router contract. In the `/packages/tfm/src/pair.rs#L57` file, there is a duplicated `swap` handler.

Recommendations

Remove the swap method or rename it to a non clashing CamelCase name if necessary.

2. Lack of test Coverage on router operations

Likelihood	Impact	Risk
Rare	Informational	**Informational**

Notes

TFM team advices that, tests is extensively via on-chain.

Description

Besides the total test coverage *(76.48%)* across the entire contract, the `operations.rs` file has almost no test coverage. This is a critical component of the router and should be tested extensively.

Recommendations

Enforce test coverage to the operations of the router contract to ensure they perform a swap, fabricate assets into the appropriate msgs and also cover possible edge cases.

3. 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.

4. Unused variables and functions in code

Likelihood	Impact	Risk
Rare	Informational	**Informational**

Description

In `contract.rs`, there is a number of values which appears not to be used anywhere in the code contract.

That include, `first_operation#617` and `last_operation#618` for example.

Additionally, there are a number of places where a variant of `SwapOperation` is used without the `_var` convention for unused values.

Recommendations

It's recommended to append a `_var` name convention for unused values. Also, considering making a case where the `searching` in the `contract.rs#606` can be set to **`true`**.

Document control

Document changes

Version	Date	Name	Changes
0.1	2022-05-02	Vinicius Marino	Initial report
0.2	2022-05-03	Vinicius Marino	Team communication and Pre-Release
1.0	2022-05-03	Vinicius Marino	Document Release
1.1	2022-06-07	Vinicius Marino	Terra v2 Updates
1.2	2022-06-07	Vinicius Marino	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.

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