Terrapins Web Penetration and Backend Test Report

Prepared for Terrapins, 19 April 2022

Table of Contents

Introduction

SCV was engaged by Terrapins 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 strictly on the following items below:

- frontend for swapping between TPIN/gold/aUST/UST assets;
- backend worker NodeJS process PinBot;

As well as it's related codebase:

- https://github.com/terraterrapins/TerrapinWeb *da0d0d29d673af22bcd916705931d85c520a32fc*
- https://github.com/terraterrapins/PinBot *2295a7ce1c8f8c68300b4e3e88b4dfbfbee71228*

SCV notes that, the frontend had more functionality such as, user creation and user authentication handlers that were not par of this scope engagement. The Terrapins comprehensive implementation solution were also not part of this test engagement test neither.

Conclusion

Terrapins implementation does not pose a direct security risk, however, the security posture would be more robust and resilient if replaced by a CosmoWasm contract logic instead a side MongoDB/FCD databases. Terrapins team is looking into implementing such in the near future.

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 Terrapins. 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;
- Analysis each line of the code base and inspect application perimeter;
- Review underlying infrastructure technologies and supply chain security posture;

Vulnerabilities Summary

	Title and Summary	Risk	Status
1	Vulnerable library in use.	**High**	**Remediated**
2	PinBot script is not efficient	**Medium**	**Acknowledged**
3	Several hardcode variables found into the codebase	**Medium**	**Remediated**
4	Dependencies should be pinned to exact versions in package.json	**Low**	**Remediated**
5	Depending on Terra's public infrastructure to operate	**Informational**	**Acknowledged**

Detailed Vulnerabilities

1. Vulnerable library in use.

Likelihood	Impact	Risk
Possible	Severe	**High**

Description

The frontend depends on a version of the *minimist* library which has a disclosed critical vulnerability, CVE-2021-44906. This vulnerability could lead to a *Prototype Pollution* in certain circumstances that would affect the integrity of objects in the entire application.

Recommendations

Upgrade the dependency to latest using `npm audit fix`. The https://github.com/features/security can also assist in dependency updates and alerts whenever there is one.

2. PinBot script is not efficient

Likelihood	Impact	Risk
Possible	Moderate	**Medium**

Notes

The *PinBot* implementation relies on the Terra's FCD and MongoDB side databases to fully operate. Terrapins team is looking into re-designing this approach in the near future.

Description

PinBot code is used as part of Terrapins implementation that is designed scan valid transactions from the blockchain using the Terra's FCD.

The code is problematic because its not modular. The script would only flag transactions paid in UST having a fixed Fee amount.

Recommendations

It's recommended to redesign the bot to be more modular in case Fees or Denom changes.

3. Several hardcode variables found into the codebase

Likelihood	Impact	Risk
Possible	Moderate	**Medium**

Description

The codebase contains several variables (addresses, network configuration, API endpoints) that are hardcoded in several parts along the code. Using hardcoded variables, drastically reduces the code quality and increases complexity and attack surface.

Recommendations

For non-sensitive variables, use an environment configuration file. ie: *.env* file For sensitive variables, consider using a *Secret Manager* solution.

4. Dependencies should be pinned to exact versions in package.json

Likelihood	Impact	Risk
Rare	Low	**Low**

Description

Terrapins frontend contains over 45 dependencies that are not pinned to an exact version in the *package.json* file.

This can potentially allow dependency attacks, as seen with the flow of events package with *Copay Bitcoin Wallet* for example.

Recommendations

Ensure dependencies are pinned to a exact version and not to a range. From `"^1.1.0"` or `"~1.1.0"` simply do `1.1.0` to specify it.

5. Depending on Terra's public infrastructure to operate

Likelihood	Impact	Risk
Rare	Informational	**Informational**

Notes

Terrapins team is planning to deploy their own node and infrastructure that does not depending on the TFL public nodes.

Description

The Terrapins frontend depends on the public **LCD** and **FCD** to operate while interacting with Terra chain.

Terra's public infrastructure is a well known and receives a significant amount of demand causing performance issues on occasion.

Terrapins could experience downtime and/or significant delays using this public node, affecting their user's perception of the service.

Recommendations

It is recommended that Terrapins deploys the required infrastructure exclusively for their needs and use the public ones from Terra as a fallback.

By using a dedicate node, Terrapins will have greater control over the performance of their service. Which will also contribute to the further decrentralization of the Terra ecosystem they rely on.

Document control

Document changes

Version	Date	Name	Changes
0.1	2022-04-11	Vinicius Marino	Initial report
0.2	2022-04-12	Vinicius Marino	Team communication and Pre-Release
1.0	2022-04-19	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.

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