TerraCore Audit Report

Version 1.0.0

Serial No.: 20220109000012019

Presented by Fairyproof

January 9, 2022

FAIRYPROOF

01. Introduction

This document includes the results of the audit performed by the Fairyproof team on the Terra Core project.

Audit Start Time:

October 25, 2021

Audit End Time:

November 10, 2022

CMC ID:

4172

Audited Code's Github Repository:

https://github.com/terra-money/core/tree/v0.5.8-oracle

Audited Code's Github Commit Number When Audit Started:

0939eeb77fbb8c03ba00269e3f5ae48f687c8574

Audited Code's Github Commit Number When Audit Ended:

0939eeb77fbb8c03ba00269e3f5ae48f687c8574

The goal of this audit is to review Terra Core's implementation for its Market, Oracle and Treasury modules, study potential security vulnerabilities, its general design and architecture, and uncover bugs that could compromise the software in production.

We make observations on specific areas of the code that present concrete problems, as well as general observations that traverse the entire codebase horizontally, which could improve its quality as a whole.

This audit only applies to the specified code, software or any materials supplied by the Terra team for specified versions. Whenever the code, software, materials, settings, enviroment etc is changed, the comments of this audit will no longer apply.

— Disclaimer

Note that as of the date of publishing, the contents of this report reflect the current understanding of known security patterns and state of the art regarding system security. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk.

The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. If the audited source files are smart contract files, risks or issues introduced by using data feeds from offchain sources are not extended by this review either.

Given the size of the project, the findings detailed here are not to be considered exhaustive, and further testing and audit is recommended after the issues covered are fixed.

To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services.

FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

— Methodology

The above files' code was studied in detail in order to acquire a clear impression of how the its specifications were implemented. The codebase was then subject to deep analysis and scrutiny, resulting in a series of observations. The problems and their potential solutions are discussed in this document and, whenever possible, we identify common sources for such problems and comment on them as well.

The Fairyproof auditing process follows a routine series of steps:

1. Code review that includes the following
   i. Review of the specifications, sources, and instructions provided to Fairyproof to make sure we understand the size, scope, and functionality of the project's source code.
   ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   iii. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Fairyproof describe.
2. Testing and automated analysis that includes the following:
   i. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run the test cases.
   ii. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the source code to improve maintainability, security, and control based on the established industry and academic practices, recommendations, and research.

— Structure of the document

This report contains a list of issues and comments on all the above source files. Each issue is assigned a severity level based on the potential impact of the issue and recommendations to fix it, if applicable. For ease of navigation, an index by topic and another by severity are both provided at the beginning of the report.

— Documentation

For this audit, we used the following sources of truth about how the TerraCore system should work:

Terra Whitepaper

Terra Docs

Cosmos SDK Docs

Tendermint Docs

IBC Protocol Docs

These were considered the specification.

— Comments from Auditor

Serial Number	Auditor	Audit Time	Result
2022010900012019	Fairyproof Security Team	October 25, 2021 - November 10, 2021	Low Risk

Summary:

The Fairyproof security team used its auto analysis tools and manual work to audit the project. During the audit, 1 risk of low-severity was discovered.

02. About Fairyproof

Fairyproof is a leading technology firm in the blockchain industry, providing consulting and security audits for organizations. Fairyproof has developed industry security standards for designing and deploying blockchain applications.

03. Introduction to Terra

The Terra system is mainly composed of a Terra protocol, and two tokesn including Terra and Luna.

- The Terra protocol

The Terra protocol is the leading decentralized and open-source public blockchain protocol for algorithmic stablecoins. Using a combination of open market arbitrage incentives and decentralized oracle voting, the Terra protocol creates stablecoins that consistently track the price of any fiat currency. Users can spend, save, trade, or exchange Terra stablecoins instantly, all on the Terra blockchain. Luna provides its holders with staking rewards and governance power. The Terra ecosystem is a quickly expanding network of decentralized applications, creating a stable demand for Terra and increasing the price of Luna.

- Terra and Luna

The protocol consists of two main tokens, Terra and Luna.

Terra is a stablecoin that tracks the price of fiat currencies. Users mint new Terra by burning Luna. Stablecoins are named for their fiat counterparts. For example, the base Terra stablecoin tracks the price of the IMF's SDR, named TerraSDR, or SDT. Other stablecoin denominations include TerraUSD or UST, and TerraKRW or KRT. All Terra denominations exist in the same pool.

Luna is the Terra protocol's native staking token that absorbs the price volatility of Terra. Luna is used for governance and in mining. Users stake Luna to validators who record and verify transactions on the blockchain in exchange for rewards from transaction fees. The more Terra is used, the more Luna is worth.

04. Major functions of audited code

The audited code mainly contains three core modules:

Market: The Market module enables atomic swaps between different Terra stablecoin denominations, and between Terra and Luna. This module ensures an available, liquid market, stable prices, and fair exchange rates between the protocol's assets.

Oracle: The Oracle module provides the Terra blockchain with an up-to-date and accurate price feed of exchange rates of Luna against various Terra pegs so that the Market may provide fair exchanges between Terra<>Terra currency pairs, as well as Terra<>Luna.

Treasury: The Treasury module acts as the "central bank" of the Terra economy, measuring macroeconomic activity by observing indicators and adjusting monetary policy levers to modulate miner incentives toward stable, long-term growth..

Note: the Cosmos SDK, Tendermint Core, CosmWasm and other third party libraries that the project relies on were not covered by this audit.

05. Coverage of issues

The issues that the Fairyproof team covered when conducting the audit include but are not limited to the following ones:

- General Code Issues

- Correctness of the implementation;
- Adversarial actions and other attacks on Modules;
- Potential misuse and gaming of the all Modules;
- Attacks that impacts funds, such as the draining or the manipulation of funds;
- Mismanagement of funds via transactions;
- Modules's intended use or disrupt the execution;
- Vulnerabilities in Modules code, particularly for swapping, earning, and creating pools;
- Vulnerabilities in the interaction between Modules and other existing Cosmos-SDK modules;
- Protection against malicious attacks and other ways to exploit the Modules;
- Inappropriate permissions and excess authority;
- Data privacy, data leaking, and information integrity;
- Secure interfaces to the application (API and CLI);
- Implementation Vulnerability
- Code Improvement
- Misc

- Network Interactions

- Attacks on all Modules through the network and secure communication between Modules and network components;
- Spam attacks that bottleneck the functionality and the entire network by exhausting computation power or traffic bandwidth, leading to a monopoly of submitting transactions to the network (minimal competition among participants can lead to extreme inefficiency of price discovery);

- Economics

- Review of the economic incentives, including the intended functions and potential impact;
- Weaknesses in the economic model resulting in user attacks again the Modules;
- Attack vectors that can game the trading environment and cause unnecessary costs to ordinary users;

- Compliance with Cosmos

06. Severity level reference

Every issue in this report was assigned a severity level from the following:

Critical severity issues need to be fixed as soon as possible.

High severity issues will probably bring problems and should be fixed.

Medium severity issues could potentially bring problems and should eventually be fixed.

Low severity issues are minor details and warnings that can remain unfixed but would be better fixed at some point in the future.

Neutral is not an issue or risk but a suggestion for code improvement.

07. List of issues by severity

Index	Title	Issue/Risk	Severity	Status
FP-1	Inappropriate Descriptions	Misc	Neutral	
FP-2	Incorrect Algorithm	Implementation Vulnerability	Low	
FP-3	Inappropriate Descriptions	Misc	Neutral	
FP-4	Inappropriate Descriptions	Misc	Neutral	
FP-5	Inappropriate Descriptions	Misc	Neutral	
FP-6	Inappropriate Descriptions	Misc	Neutral	
FP-7	Redundant Code	Code Improvement	Neutral	
FP-8	Redundant Code	Code Improvement	Neutral	
FP-9	Redundant Code	Code Improvement	Neutral	
FP-10	Redundant Code	Code Improvement	Neutral	

08. Issue descriptions

[FP-1] [Inappropriate Descriptions] [Neutral]

Risk Severity: Neutral

Issue/Risk: Misc

Description:

In the `core/x/market/spec/01_concepts.md` file, the `Swap Procedure` section didn't describe the code's behavior.

Comments:

Consider changing the descriptions for `Swap Procedure` and removing the section for `Seigniorage`.

[FP-2] [Incorrect Algorithm] [Low]

Risk Severity: Low

Issue/Risk: Implementation Vulnerability

Description:

In line 81 of the `x/market/keeper/msg_server.go` file when `TerraPoolDelta` was updated `spread fee` was deducted. This is incorrect.

Recommendation:

Consider keeping this fee.

[FP-3] [Inappropriate Descriptions] [Neutral]

Risk Severity: Neutral

Issue/Risk: Misc

Description:

The descriptions in the `Prevote and Vote` section in the `core/x/oracle/spec/01_concepts.md` file didn't describe correctly the voting mechanism which was implemented in the code.

Recommendation:

Consider changing the descriptions for the voting mechanism.

[FP-4] [Inappropriate Descriptions] [Neutral]

Risk Severity: Neutral

Issue/Risk: Misc

Description:

The `Abstaining from Voting` section in the `core/x/oracle/spec/01_concepts.md` file didn't describe correctly the voting mechanism implemented in the code.

Recommendation:

Consider removing the `Abstaining from Voting` section.

[FP-5] [Inappropriate Descriptions] [Neutral]

Risk Severity: Neutral

Issue/Risk: Misc

Description:

The `Observed Indicators` section in the `core/x/treasury/spec/01_concepts.md` file didn't describe the code's behavior correctly. The seigniorage was not applied in the Columbus-5's implementation but the descriptions still had it.

Recommendation:

Consider removing the description for SR.

[FP-6] [Inappropriate Descriptions] [Neutral]

Risk Severity: Neutral

Issue/Risk: Misc

Description:

The `Monetary Policy Levers` and `Updating Policies` sections in the `core/x/treasury/spec/01_concepts.md` file didn't describe the code's behavior correctly. The seigniorage was not applied in the Columbus-5's implementation and the Oracle voters were not rewarded with seigniorage.

Recommendation:

Consider changing the descriptions for the Oracle voter's reward.

[FP-7] [Redundant Code] [Neutral]

Risk Severity: Neutral

Issue/Risk: Code Improvement

Description:

The `EndBlocker` function defined in the `core/x/treasury/abci.go` file had redundant code.

Recommendation:

Consider removing the following code:

`k.BurnCoinsFromBurnAccount(ctx)`, `k.SettleSeigniorage(ctx)`, `rewardWeight := k.UpdateRewardPolicy(ctx)` and `defer k.RecordEpochInitialIssuance(ctx)`

[FP-8] [Redundant Code] [Neutral]

Risk Severity: Neutral

Issue/Risk: Code Improvement

Description:

It was unnecessary for `UpdateIndicators` to call `k.SetSR`.

Recommendation:

Consider removing the following line:

k.SetSR(ctx, epoch, SR)

[FP-9] [Redundant Code] [Neutral]

Risk Severity: Neutral

Issue/Risk: Code Improvement

Description:

The `UpdateRewardPolicy` function defined in the `core/x/treasury/keeper/policy.go` file was redundant.

Recommendation:

Consider removing this function.

[FP-10] [Redundant Code] [Neutral]

Risk Severity: Neutral

Issue/Risk: Code Improvement

Description:

The `SettleSeigniorage` function defined in the `core/x/treasury/keeper/seigniorage.go` file was redundant.

Recommendation:

Consider removing the function.

09. Recommendations to enhance the overall security

We list some recommendations in this section. They are not mandatory but will enhance the overall security of the system if they are adopted.

- N/A