AUDIT REPORT

DojoSwap

Injera Contract

Prepared by SCV-Security

On 12th July 2024

Table of Contents

Introduction

SCV has been engaged by Dojo Trading to conduct a comprehensive security review with the goal of identifying potential security threats and vulnerabilities within the codebase. The purpose of this audit is to evaluate the security posture of the codebase and provide actionable recommendations to mitigate any identified risks. This report presents an overview of the findings from our security audit, outlining areas of concern and proposing effective measures to enhance the codebase's security.

Scope Functionality

The Red Bank contract is a decentralized lending protocol that allows users to perform deposit and borrow operations. If an account is unhealthy, liquidators can liquidate the borrower to ensure the protocol's solvency.

Submitted Codebase

red-bank	
Repository	https://github.com/dojo-trading/injera
Commit	`edc74bf1ba4d9de53728bc155355f6f21720f4c1`
Contract	`red-bank`
Branch	`master`

Revisions Codebase

red-bank	
Repository	https://github.com/dojo-trading/injera
Commit	`20cae65e4beb13c766fd40b84b7c6198a679a14d`
Contract	`red-bank`
Branch	`master`

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 Dojo Trading. Testing includes, but is not limited to, the following:

- Understanding the application and its functionality 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

This section provides an evaluation of specific criteria aspects as described below:

- **Documentation:** Evaluating the presence and comprehensiveness of publicly available or provided explanatory information, diagram flowcharts, comments, and supporting documents to enhance code understanding.
- **Coverage:** Evaluating whether the code adequately addresses all necessary cases and scenarios, ensuring that the intended functionality or requirements are sufficiently covered.
- **Readability:** Assessing how easily the code can be understood and maintained, considering factors such as code structure, naming conventions, and overall organisation.
- **Complexity:** Evaluating the complexity of the code, including factors such as, number of lines, conditional statements, and nested structures.

The status of each criteria is categorised as either **SUFFICIENT** or **NOT-SUFFICIENT** based on the audit assessment. This categorisation provides insights to identify areas that may require further attention and improvement.

Criteria	Status	Notes
Documentation	**SUFFICIENT**	N/A
Coverage	**SUFFICIENT**	N/A
Readability	**SUFFICIENT**	N/A
Complexity	**SUFFICIENT**	N/A

Findings Summary

Summary Title	Risk Impact	Status
The first depositor can inflate the utilization rate to steal funds	CRITICAL	RESOLVED
Deprecated `to_binary` and `from_binary` functions are used	INFO	ACKNOWLEDGED
Unneeded migration features	INFO	ACKNOWLEDGED

Findings Technical Details

1. The first depositor can inflate the utilization rate to steal funds

RISK IMPACT: CRITICAL	STATUS: RESOLVED

Description

The `update_interest_rates` function in `contracts/red-bank/src/interest_rates.rs:102-116` does not limit the utilization rate below 100%. This is problematic because the first depositor can transfer funds directly to the contract and borrow them, causing the borrowing ratio to exceed the lending ratio, thereby inflating the utilization rate and profiting from the abnormally high lending interest.

An example attack flow:

1. The attacker becomes the first depositor when a new market (e.g., INJ) is created.
2. The attacker deposits 1 INJ to the contract.
3. The attacker donates 1000 INJ to the contract.
4. The attacker deposits sufficient collateral with another account to borrow all the donated funds.
5. With another account, the attacker receives a high lending interest accrued to their account.

This could happen when a new market is being created. To reproduce this issue, the SCV-Security team provides the following PoC, which is available here.

Recommendation

Consider performing a `min()` check to ensure the utilization rate does not exceed 100%.

2. Deprecated `to_binary` and `from_binary` functions are used

RISK IMPACT: INFORMATIONAL	STATUS: ACKNOWLEDGED

Description

The `to_binary` and `from_binary` functions are implemented in several instances of the codebase. However, these functions are unsupported and deprecated.

Recommendation

Consider removing the deprecated functions and using the `to_json_binary` and `from_json` functions instead.

3. Unneeded migration features

RISK IMPACT: INFORMATIONAL	**STATUS:** ACKNOWLEDGED

Description

The `migrate` entry point in `contracts/red-bank/src/contract.rs:262-265` implements a migration feature for the previous version of the contract to be migrated into a newer version. Since this is a new contract to be deployed on Injective, there are no existing deployments, which means the migration feature is unneeded.

This also includes the `UpdateAssetCollateralStatus` message and the `MIGRATION_GUARD` state in `contracts/red-bank/src/contract.rs:133-136` and `contracts/red-bank/src/state.rs:18`.

Recommendation

Consider removing the unneeded migration features.

Document Control

Version	Date	Notes
-	12th June 2024	Security audit commencement date.
0.1	28th June 2024	Initial report with identified findings delivered.
0.5	12th July 2024	Fixes remediations implemented and reviewed.
1.0	12th July 2024	Audit completed, final report delivered.

Appendices

A. Appendix - Risk assessment methodology

SCV-Security employs a risk assessment methodology to evaluate vulnerabilities and identified issues. This approach involves the analysis of both the LIKELIHOOD of a security incident occurring and the potential IMPACT if such an incident were to happen. For each vulnerability, SCV-Security calculates a risk level on a scale of 5 to 1, where 5 denotes the highest likelihood or impact. Consequently, an overall risk level is derived from combining these two factors, resulting in a value from 10 to 1, with 10 signifying the most elevated level of security risk

Risk Level	Range
CRITICAL	10
SEVERE	From 9 to 8
MODERATE	From 7 to 6
LOW	From 5 to 4
INFORMATIONAL	From 3 to 1

LIKELIHOOD and **IMPACT** would be individually assessed based on the below:

Rate	LIKELIHOOD	IMPACT
5	**Extremely Likely**	Could result in severe and irreparable consequences.
4	**Likely**	May lead to substantial impact or loss.
3	**Possible**	Could cause partial impact or loss on a wide scale.
2	**Unlikely**	Might cause temporary disruptions or losses.
1	**Rare**	Could have minimal or negligible impact.

B. Appendix - Report Disclaimer

This report should not be regarded as an "endorsement" or "disapproval" of any specific project or team. These reports do not indicate the economics or value of any "product" or "asset" created by a team or project that engages SCV-Security for a security review. The audit report does not make any statements or warranties about the code's utility, safety, suitability of the business model, regulatory compliance of the business model, or any other claims regarding the fitness of the implementation for its purpose or its bug-free status. The audit documentation is intended 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 damages arising from or in connection with this audit report. Copyright of this report remains with SCV-Security.

THANK YOU FOR CHOOSING

SCV SECURITY

🌐 scv.services

✉ contact@scv.services