HACKEN

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Customer: TrustSwap
Date: March 16th, 2022

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Document

Name	Smart Contract Code Review and Security Analysis Report for TrustSwap.
Approved by	Andrew Matiukhin | CTO Hacken OU
Type	Tokens lock
Language	Rust
Platform	Solana
Methods	Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review
Repository	https://github.com/trustswap/team-finance-solana-contracts
Commit	0067774d794f02848e50aca0be8276403ff38854
Technical Documentation	No
Tests	No
Website	https://trustswap.com/
Timeline	1 FEBRUARY 2022 - 16 MARCH 2022
Changelog	21 FEBRUARY 2022 - Initial Audit 16 MARCH 2022 - Second Review

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Table of contents

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Introduction

Hacken OÜ (Consultant) was contracted by TrustSwap(Customer) to conduct a Smart Contract Code Review and Security Analysis. This report presents the findings of the security assessment of the Customer's smart contract and its code review conducted between February 1st, 2022 - February 21st, 2022.

The second review was conducted on March 16th, 2022.

Scope

The scope of the project is smart contracts in the repository:

Repository:
https://github.com/trustswap/team-finance-solana-contracts
Commit:
0067774d794f02848e50aca0be8276403ff38854
Technical Documentation: No
Tests: No
Contracts:
./program/src/*

We have scanned this smart contract for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that are considered:

Category	Check Item
Review	▪ Business Logics Review ▪ Access Control & Authorization ▪ Assets Integrity ▪ User Balances Manipulations ▪ Data Consistency Manipulations ▪ Reentrancy ▪ Ownership Takeover ▪ Style guide Violations ▪ Unchecked math ▪ Repository Consistency

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Executive Summary

Score measurements details can be found in the corresponding section of the methodology.

Documentation quality

The customer provided superficial functional requirements and no technical requirements. Total Documentation Quality score is **2** out of **10**.

Code quality

Total CodeQuality score is **3** out of **10**. Code duplications. No unit tests were provided.

Architecture quality

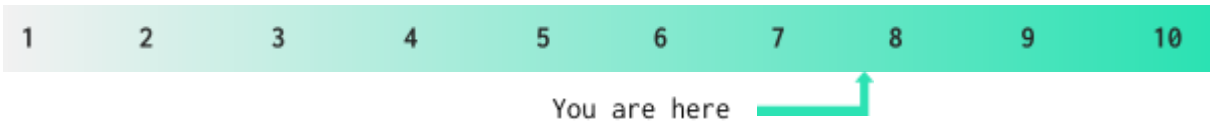
Architecture quality score is **3** out of **10**. All the logic is implemented in one file. Functions are overwhelmed with template code that could be moved to separate functions and be reused.

Security score

As a result of the audit, security engineers found **1** low severity issue. The security score is **10** out of **10**. All found issues are displayed in the “Issues overview” section of the report.

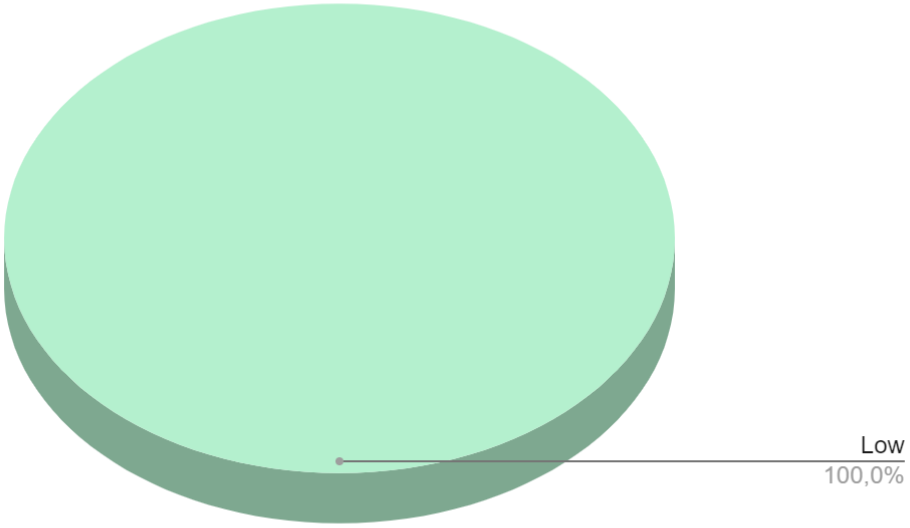
Summary

According to the assessment, the Customer's smart has the following score: **7.8**



Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Graph 1. The distribution of vulnerabilities after the audit.



Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Issues overview

Critical

No critical issues were found.

High

No high severity issues were found.

Medium

No medium severity issues were found.

Low

Code is duplicated all over the file.

File: processor.rs

Recommendation: move common code to separate functions and reuse it.

Status: acknowledged

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Disclaimers

Hacken Disclaimer

The smart contracts given for audit have been analyzed in accordance with the best industry practices at the date of this report, in relation to cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The audit makes no statements or warranties on the security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bug-free status, or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only — we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts.

Technical Disclaimer

Smart contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have vulnerabilities that can lead to hacks. Thus, the audit can't guarantee the explicit security of the audited smart contracts.