HACKEN

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Customer: Toyo Verse
Date: April 05th, 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 Toyo Verse.
Approved By	Evgeniy Bezuglyi | SC Department Head at Hacken OU
Type of Contracts	ERC20 token
Platform	EVM
Language	Solidity
Methods	Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review
Website	https://toyoverse.com/
Timeline	30.03.2022 - 31.03.2021
Changelog	30.03.2022 - Initial Review 05.04.2022 - Revising

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 Toyo Verse (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 contracts.

Scope

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

Repository:
https://github.com/Toyoverse/toyo-smartcontracts
Commit:
927e9a80c8ff247ab9b5e1a7a7de8979fda98f22
Technical Documentation: Yes (https://whitepaper.toyoverse.com/)
JS tests: Yes
Contracts:
ToyoGovernanceToken.sol

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
Code review	▪ Reentrancy ▪ Ownership Takeover ▪ Timestamp Dependence ▪ Gas Limit and Loops ▪ Transaction-Ordering Dependence ▪ Style guide violation ▪ EIP standards violation ▪ Unchecked external call ▪ Unchecked math ▪ Unsafe type inference ▪ Implicit visibility level ▪ Deployment Consistency ▪ Repository Consistency
Functional review	▪ Business Logics Review ▪ Functionality Checks ▪ Access Control & Authorization ▪ Escrow manipulation ▪ Token Supply manipulation ▪ Assets integrity ▪ User Balances manipulation ▪ Data Consistency ▪ Kill-Switch Mechanism

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

Executive Summary

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

Documentation quality

The Customer provided a whitepaper with tokenomics requirements and no technical requirements. Total Documentation Quality score is **5** out of **10**.

Code quality

The total CodeQuality score is **10** out of **10**. The code follows official language style guides. Unit tests were provided.

Architecture quality

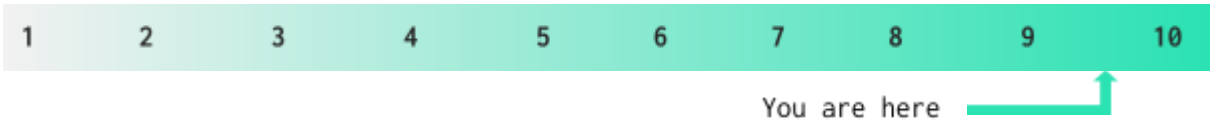
The architecture quality score is **10** out of **10**. The architecture is clear.

Security score

As a result of the audit, security engineers found no severity issues. The security score is **10** out of **10**. All found issues are displayed in the “Issues overview” section.

Summary

According to the assessment, the Customer's smart contract has the following score: **9.5**



Notices

1. There are other smart contracts in the repository that are not included in the audit scope.

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 cannot lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that cannot have a significant impact on execution

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

Findings

Critical

No critical severity issues were found.

High

1. **Token minting.**

   According to the tokenomics maximum total supply is 150,000,000, but users with the MINTER_ROLE role can mint more.

   Contracts: ToyoGovernanceToken.sol

   Function: mint

   Recommendation: remove the ability to mint more than stated in tokenomics.

   Status: Fixed (Revised Commit: 79377f1278cae55a8479ad043e984da5b14b159c)

2. **Pausing of all the token transfers.**

   Users with the PAUSER_ROLE role can pause all the token transfers anytime. Pausing functionality should be limited by clear contract rules. The documentation does not mention the functionality of stopping transfers.

   Contracts: ToyoGovernanceToken.sol

   Function: pause

   Recommendation: change pausing functionality.

   Status: Fixed (Revised Commit: 79377f1278cae55a8479ad043e984da5b14b159c)

Medium

No medium severity issues were found.

Low

1. **No caller verification in the initialization function.**

   There is no restriction that only the owner can call the initialization function.

   Contracts: ToyoGovernanceToken.sol

   Function: initialize

   Recommendation: it is better to add only owner access to the initialization function.

   Status: Mitigated. The Customer approved that function is protected.

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 by the best industry practices at the date of this report, with 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 a sufficient assessment regarding the utility and safety of the code, bug-free status, or any other contract statements. 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 cannot guarantee the explicit security of the audited smart contracts.