



HACKEN

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Customer: Slavi

Date: March 10, 2022

This document may contain confidential information about IT systems and the intellectual property of the Customer as well as information about potential vulnerabilities and methods of their exploitation.

The report containing confidential information can be used internally by the Customer, or it can be disclosed publicly after all vulnerabilities are fixed – upon a decision of the Customer.

Document

Name	Smart Contract Code Review and Security Analysis Report for Slavi.
Approved by	Andrew Matiukhin CTO Hacken OU Evgeniy Bezuglyi SC Department Head at Hacken OU
Type	Upgradable BEP-20 token
Platform	EVM
Language	Solidity
Methods	Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review
Repository	https://github.com/SlvLabs/SLVToken
Commit	60079F4714E4EF6E7914D79AFC56EE4224B7DA8C
Technical Documentation	NO
JS tests	NO
Deployed contract	https://bscscan.com/address/0x61656a40fc25c7e4c737723b5d3dda4e93bd08e1
Website	https://slavi.io/
Timeline	10 MARCH 2022
Changelog	10 MARCH 2022 - INITIAL AUDIT

Table of contents

Introduction	4
Scope	4
Executive Summary	5
Severity Definitions	6
Audit overview	7
Recommendations	8
Disclaimers	9

Introduction

Hacken OÜ (Consultant) was contracted by Slavi (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/SlvLabs/SLVToken>

Commit:

60079f4714e4ef6e7914d79afc56ee4224b7da8c

Technical Documentation: No

JS tests: No

Contracts:

[./SlavicoProxy.Full.sol](#)

[./Slavico.Full.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	<ul style="list-style-type: none">▪ 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	<ul style="list-style-type: none">▪ Business Logics Review▪ Functionality Checks▪ Access Control & Authorization▪ Escrow manipulation▪ Token Supply manipulation▪ Assets integrity▪ User Balances manipulation▪ Data Consistency▪ Kill-Switch Mechanism

Executive Summary

Score measurements details can be found in the corresponding section of the [methodology](#).

Documentation quality

The project has no functional or technical requirements. Total Documentation Quality score is **0** out of **10**. Weight in total score is **1**.

Code quality

The code follows official language style guides. The code is not covered with unit tests. The score is **5** out of **10**. Weight in total score is **1**.

Architecture quality

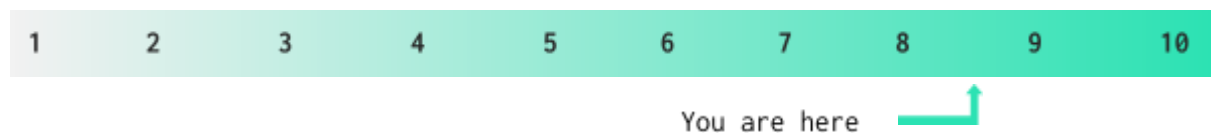
Smart contract of the project has clear architecture. Architecture quality score is **10** out of **10**. Weight in total score is **1**.

Security score

As a result of the audit, security engineers found **no** issue. Security score is **10** out of **10**. Weight in total score is **7**.

Summary

According to the assessment, the Customer's smart has the following score: **8.5**



Notices

1. The contract owner may update the contract implementation.

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

Audit overview

■■■■ Critical

No critical severity issues were found.

■■■ High

No high severity issues were found.

■■ Medium

No medium severity issues were found.

■ Low

No low severity issues were found.



Recommendations

No recommendation is provided.

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.