HACKEN

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Customer: Block Square
Date: March 28th 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 Block Square.
Approved by	Andrew Matiukhin | CTO Hacken OU Evgeniy Bezuglyi | SC Department Head at Hacken OU
Type	ERC20 token; Staking
Platform	EVM
Language	Solidity
Methods	Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review
Repository	https://github.com/blocksquare/oceanpoint-contracts/blob/v1.1.0/contracts/OceanStaking.sol - Initial Audit https://github.com/blocksquare/oceanpoint-contracts/commit/2c9c0885b9970ffc9470f60b7dd047fad2c0aabd - Remediation Check
Commit	bff5a4ceede98bcfc9e1dfac7a20ce0e324c8bca - Initial Audit 2c9c0885b9970ffc9470f60b7dd047fad2c0aabd - Remediation Check
Deployed contract	No
Technical Documentation	Yes
JS tests	Yes
Website	https://blocksquare.io/
Timeline	09 MARCH 2022 - 28 MARCH 2022
Changelog	09 MARCH 2022 - Initial Audit 21 MARCH 2022 - Revising 28 MARCH 2022 - Remediation check

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 Block Square (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/blocksquare/oceanpoint-contracts/blob/v1.1.0/contracts/OceanStaking.sol - Initial Check
https://github.com/blocksquare/oceanpoint-contracts/commit/2c9c0885b9970ffc9470f60b7dd047fad2c0aabd - Remediation Check
Commit: bff5a4ceede98bcfc9e1dfac7a20ce0e324c8bca - Initial Check
2c9c0885b9970ffc9470f60b7dd047fad2c0aabd - Remediation Check
Technical Documentation: Yes
JS tests: Yes
Contracts:
OceanStaking.sol
OceanStakinProxy.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 ▪ DoS with (Unexpected) Throw ▪ DoS with Block Gas Limit ▪ Transaction-Ordering Dependence ▪ Style guide violation ▪ Costly Loop ▪ ERC20 API violation ▪ Unchecked external call ▪ Unchecked math ▪ Unsafe type inference ▪ Implicit visibility level ▪ Deployment Consistency ▪ Repository Consistency ▪ Data Consistency

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

Functional review	▪ Business Logics Review ▪ Functionality Checks ▪ Access Control & Authorization ▪ Escrow manipulation ▪ Token Supply manipulation ▪ Assets integrity ▪ User Balances manipulation ▪ Data Consistency manipulation ▪ Kill-Switch Mechanism ▪ Operation Trails & Event Generation

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 both a high level and detailed documentation. These documentations explained the workflow of the protocol. Total Documentation Quality score is **10** out of **10**.

Code quality

The total CodeQuality score is **8** out of **10**. The code follows most of the official guides and best practices. Unit tests were provided.

Architecture quality

The architecture quality score is **10** out of **10**. The code generally follows the best practices.

Security score

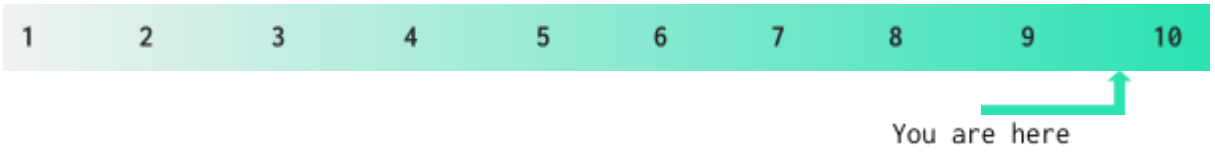
As a result of the audit, security engineers found **2** medium, **3** low severity issues. The security score is **5** out of **10**. All found issues are displayed in the “Issues overview” section.

The revised code contains **1** medium and **2** low severity issues. As a result, the security score became **10** out of **10**.

As a result of the remediations review, Customers’ smart contracts contain **no** issues. Thus their security score is **10** out of **10**.

Summary

According to the assessment, the Customer's smart has the following score: **9,8**



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 asset 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

Audit overview

Critical

No critical issues.

High

No high issues.

Medium

1. **Function Call Order Assumption**

   The withdraw function aims to withdraw a given amount of tokens from the contract. However, assume that the traders make no deposits to this contract, this will make the bst token balance 0 (because the bstAmount will be zero) which will make your contract try sending 0 funds to the trader. This will cause unnecessary Gas consumption.

   Contracts: OceanStaking.sol

   Function: -

   Recommendation: Implement control mechanisms for the bst token balance.

   Status: Fixed

Low

1. **Missing Zero Address Validation**

   In the initialize(), there needs to be a zero address check for *bst* and *owner* parameters.

   Contracts: OceanStaking.sol

   Function: initialize() between lines #25 - #32,

   Recommendation: Implement a missing zero address check.

   Status: Fixed

2. **Use of Hardcoded Values**

   In the initialize functions hardcoded values are used in calculations

   Contracts: OceanStaking.sol

   Function: initialize() between lines #25 - #32,

   Recommendation: Move hardcoded values to constants.

   Status: Fixed

3. **Functions that can be Declared as *external***

   To save Gas, public functions that are never called in the contract should be declared as *external*.

   Contracts: OceanStaking.sol

   Function:initialize()

   Recommendation: Aforementioned should be declared as *external*.

   Status: Fixed

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 it 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.