HACKEN

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Customer: Brokoli
Date: September 17th, 2021

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 Brokoli.
Approved by	Andrew Matiukhin | CTO Hacken OU
Type	ERC20 token; Transfer controller
Platform	Ethereum / Solidity
Methods	Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review
Repository	1. https://github.com/BrokoliNetwork/contract-ido 2. https://github.com/BrokoliNetwork/contract-token/releases/tag/audit
Commit	1. B6431662EFAA8F0B3C50BA034EBD0DED34433A70 2. BD9600B4AE857948B114AC466356042B37264BC9
Technical Documentation	NO
JS tests	YES
Timeline	25 AUGUST 2021 - 17 SEPTEMBER 2021
Changelog	02 SEPTEMBER 2021 - INITIAL AUDIT 14 SEPTEMBER 2021 - SECOND REVIEW 17 SEPTEMBER 2021 - THIRD 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 Brokoli (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 August 25th, 2021 - September 2nd, 2021. The second review conducted on September 14th, 2021. The third review conducted on September 17th, 2021.

Scope

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

Repository 1:
https://github.com/BrokoliNetwork/contract-ido
Commit 1:
b6431662efaa8f0b3c50ba034ebd0ded34433a70
Repository 2:
https://github.com/BrokoliNetwork/contract-token/releases/tag/audit
Commit 2:
bd9600b4ae857948b114ac466356042b37264bc9

Technical Documentation: No
JS tests: Yes
Contracts:
claim\BaseClaim.sol
claim\VestedClaim.sol
interfaces\IIDO.sol
test\BaseClaimTest.sol
test\erc20.sol
test\IDOTest.sol
test\VestedClaimTest.sol
whitelisted\Whitelisted.sol
Claim.sol
IDO.sol

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

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
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

Executive Summary

According to the assessment, the Customer's smart contracts are secured but the rewards token could be taken by the owner.

Insecure	Poor secured	Secured	Well-secured

You are here ↑

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

Our team performed an analysis of code functionality, manual audit, and automated checks with Mythril and Slither. All issues found during automated analysis were manually reviewed, and important vulnerabilities are presented in the Audit overview section. All found issues can be found in the Audit overview section.

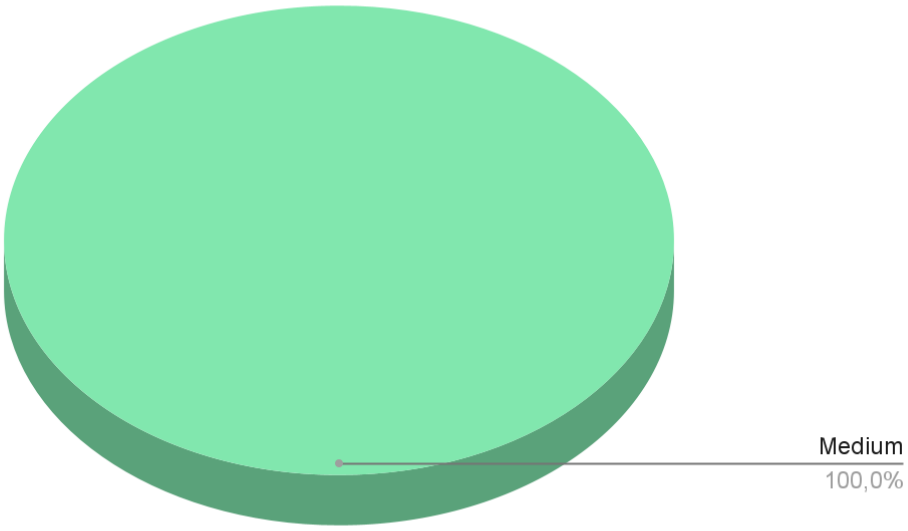
As a result of the audit, security engineers found **no issues**.

After the second review for updated contracts, security engineers found that the contracts were updated and found **1** medium and **2** low severity issues.

After the third review, security engineers found **1** medium severity issue.

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

Audit overview

Critical

No critical severity issues were found.

High

No high severity issues were found.

Medium

The owner could withdraw rewards token

The owner account could withdraw any tokens from the IDO contract, including the reward tokens before users will receive their reward allocations.

Recommendation: Please add the "**require**" statement to check that emergency withdrawal is requested not for the rewards token.

Lines: BaseClaim.sol#101-106

```
function emergencyWithdrawToken(ERC20 tokenAddress) external onlyOwner {
    tokenAddress.safeTransfer(
        msg.sender,
        tokenAddress.balanceOf(address(this))
    );
}
```

Low

1. Boolean equality

   Boolean constants can be used directly and do not need to be compared to true or false.

   Recommendation: Remove the equality to the boolean constant

   Fixed before the second review

2. Dust amounts could be left on percentage

   Because of finding percentage for before 90 days and after (20 and 80) there could be dust amount of tokens left. It's better to just subtract the unlockedOnClaim amount from the total to get the rest 80% instead of doing calculations

   Recommendation: Please consider subtracting the unlockedOnClaim value from the total

   Fixed before the second review

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

Conclusion

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools.

The audit report contains all found security vulnerabilities and other issues in the reviewed code.

As a result of the audit, security engineers found **no issues**.

After the second review for updated contracts, security engineers found that the contracts were updated and found **1** medium and **2** low severity issues.

After the third review, security engineers found **1** medium severity issue.

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.