HACKEN

# SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

**Customer**: Oneworldplan
**Date**: November 1st, 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 Oneworldplan. |
| **Approved by** | Andrew Matiukhin \| CTO Hacken OU |
| **Type** | ERC20 token; Staking |
| **Platform** | Ethereum / Solidity |
| **Methods** | Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review |
| **Files** | audit_care_updated.sol<br>audit_owp_updated,sol |
| **MD5 hashes** | bf9a68422f2ec67dfc0364e880f3dfa7<br>f56411bbb7f5bcfbea34f3d773c735b0 |
| **Technical Documentation** | NO |
| **JS tests** | NO |
| **Website** | oneworldplan.com |
| **Timeline** | 13 OCTOBER 2021 - 20 OCTOBER 2021 |
| **Changelog** | 20 OCTOBER 2021 - INITIAL AUDIT<br>01 NOVEMBER 2021 - 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 Oneworldplan (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 October 13th, 2021 - October 20th, 2021.

Second review conducted on November 1st, 2021.

# Scope

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

**Files:**
audit_care_updated.sol
audit_owp_updated.sol
**Commit:**
bf9a68422f2ec67dfc0364e880f3dfa7
f56411bbb7f5bcfbea34f3d773c735b0
**Technical Documentation:** No
**JS tests:** No
**Contracts:**
audit_care_updated.sol
audit_owp_updated.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<br>▪ Ownership Takeover<br>▪ Timestamp Dependence<br>▪ Gas Limit and Loops<br>▪ DoS with (Unexpected) Throw<br>▪ DoS with Block Gas Limit<br>▪ Transaction-Ordering Dependence<br>▪ Style guide violation<br>▪ Costly Loop<br>▪ ERC20 API violation<br>▪ Unchecked external call<br>▪ Unchecked math<br>▪ Unsafe type inference<br>▪ Implicit visibility level<br>▪ Deployment Consistency<br>▪ Repository Consistency<br>▪ Data Consistency |

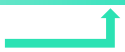
Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

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

## Executive Summary

According to the assessment, the Customer's smart contracts are well-secured.

| Insecure | Poor secured | Secured | Well-secured |
|---|---|---|---|

You are here 

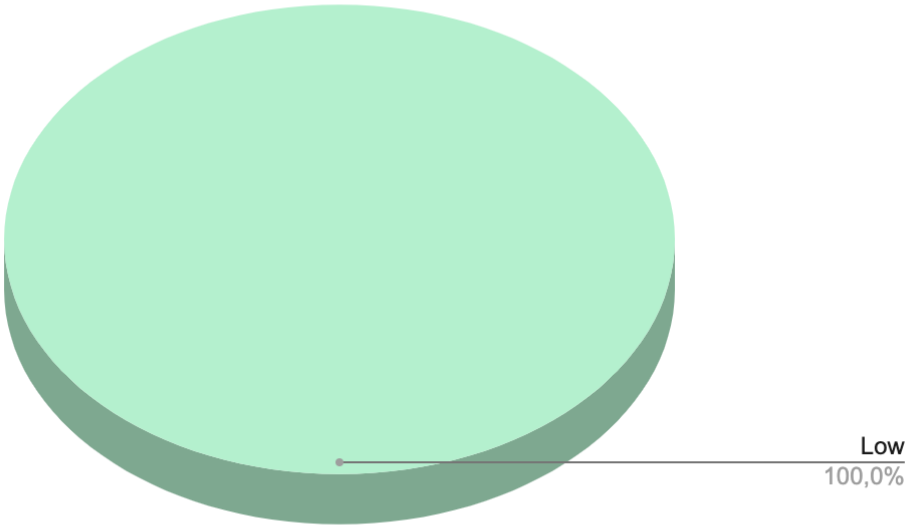
Our team performed an analysis of code functionality, manual audit, and automated checks with Mythril and Slither. All issues found during automated analysis were manually reviewed, and important vulnerabilities are presented in the Audit overview section. All found issues can be found in the Audit overview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

As a result of the audit, security engineers found **8** low severity issues.

After the second review security engineers found that “Pausable” functionality was added to both contracts and **4** issues were fixed and unresolved **4** low severity issues.

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 issues were found.

## High

No high severity issues were found.

## Medium

No medium severity issues were found.

## Low

1. A public function that could be declared external

**public** functions that are never called by the contract should be declared **external** to save gas.

**Contracts**: audit_owp.sol

**Recommendation**: Use the **external** attribute for functions never called from the contract.

2. A public function that could be declared external

**public** functions that are never called by the contract should be declared **external** to save gas.

**Contracts**: audit_care.sol

**Recommendation**: Use the **external** attribute for functions never called from the contract.

3. State variables that could be declared immutable

State variable which never change its value and initialized in the constructor should be declared immutable to save gas.

**Contracts**: audit_owp.sol

**Recommendation**: Use the **immutable** attribute for state variable which are initialized in the constructor and never change their value.

4. State variables that could be declared immutable

State variable which never change its value and initialized in the constructor should be declared immutable to save gas.

**Contracts**: audit_care.sol

**Recommendation**: Use the **immutable** attribute for state variable which are initialized in the constructor and never change their value.

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

5. No events on critical values changes

When changing critical values of the contract it is hardly recommended to emit events so it could be tracked off-chain.

**Contracts**: audit_owp.sol

**Recommendation**: Please emit events on changing critical values.

**Status**: Fixed.

6. No events on critical values changes

When changing critical values of the contract it is hardly recommended to emit events so it could be tracked off-chain.

**Contracts**: audit_care.sol

**Recommendation**: Please emit events on changing critical values.

**Status**: Fixed.

7. Conformance to Solidity naming conventions

Solidity defines a naming convention that should be followed.

**Contracts**: audit_owp.sol

**Contract Name**: OWP_Staketodonate

**Recommendation**: Follow the Solidity naming convention.

**Status**: Fixed.

8. Conformance to Solidity naming conventions

Solidity defines a naming convention that should be followed.

**Contracts**: audit_owp.sol

**Contract Name**: care_contract

**Recommendation**: Follow the Solidity naming convention.

**Status**: Fixed.

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 **8** low severity issues.

After the second review security engineers found that “Pausable” functionality was added to both contracts and **4** issues were fixed and unresolved **4** low severity issues.

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.