HACKEN

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Customer: DWeb
Date: October 18th, 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 DWeb.
Approved by	Andrew Matiukhin | CTO Hacken OU
Type	ERC20 token; ERC721 token
Platform	Ethereum / Solidity
Methods	Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review
Zip archive	dweb-contracts-audit-fixes.zip (md5: 7afc36464ba182c6a08d99949477487b)
Technical Documentation	YES
JS tests	NO
Timeline	30 SEPTEMBER 2021 - 18 OCTOBER 2021
Changelog	05 OCTOBER 2021 - Initial Audit 18 OCTOBER 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 DWeb (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 September 30th, 2021 - October 5th, 2021.

Second review conducted on October 18th, 2021.

Scope

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

Archive:
dweb-contracts-audit-fixes.zip
md5:
7afc36464ba182c6a08d99949477487b
Technical Documentation: Yes
JS tests: No
Contracts:
decentraname/IERC20Extended.sol
decentraname/IDecentraNameController.sol
decentraname/DecentraWebToken.sol
decentraname/AbstractDecentraName.sol
decentraname/DecentraName.sol
decentraname/DecentraNameController.sol
dwebregistrar/SafeMath.sol
dwebregistrar/RootRegistrarController.sol
dwebregistrar/IPriceEstimator.sol
dwebregistrar/PriceOracle.sol
dwebregistrar/StablePriceOracle.sol
dwebregistrar/StringUtils.sol
dwebregistrar/PriceEstimator.sol
registry/DWEBRegistry.sol
registry/DWEB.sol
root/Controllable.sol
root/Root.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

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

Executive Summary

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

Insecure	Poor secured	Secured	Well-secured

You are here 

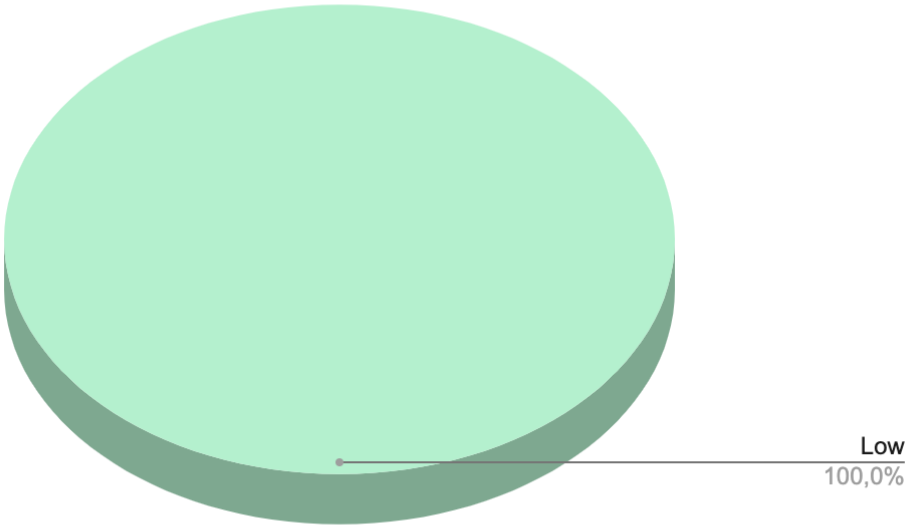
Our team performed an analysis of code functionality, manual audit, and automated checks with Mythril and Slither. All issues found during automated analysis were manually reviewed, and important vulnerabilities are presented in the Audit overview section. All found issues can be found in the Audit overview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

As a result of the audit, security engineers found **4** low severity issues.

After the second review security engineers found that there was one new function added. Therefore found **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. Missing event for “setDecentraName”

   Changing critical values should be followed by the event emitting for better tracking off-chain.

   Recommendation: Please emit events on the critical values changing.

Lines: DecentraNameController.sol#111

```
function setDecentraName(address _decentraName) external onlyOwner {
   decentraName = AbstractDecentraName(_decentraName);
}
```

2. No License header

   Each solidity source file must consist SPDX-License-Identifier header.

   Recommendation: Please add SPDX-License-Identifier to all solidity files.

   Fixed before the second review

3. **Issue**: Incompatible versions of Solidity

   The solidity version used for DecentraWebToken is too old (^0.6.2) and doesn’t correlate with the one used for the rest of the project (^0.8.4).

   Recommendation: Please consider using the recommended version of the solidity compiler and also using the one version for all contracts.

   Recommended versions are:

   0.6.11 - 0.6.12
   0.7.5 - 0.7.6
   0.8.7; 0.8.9

4. **Vulnerability**: Block timestamp

   Dangerous usage of block.timestamp. block.timestamp can be manipulated by miners. Some contracts are fully related on the block.timestamp.

   Recommendation: Please consider relying on the block.number instead.

5. A public function that could be declared external

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

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

Lines: DecentraWebToken.sol#338

```
function name() public view returns (string memory) {
```

Lines: DecentraWebToken.sol#342

```
function symbol() public view returns (string memory) {
```

Lines: DecentraWebToken.sol#346

```
function decimals() public view returns (uint8) {
```

Lines: DecentraWebToken.sol#350

```
function totalSupply() public view override returns (uint256) {
```

Lines: DecentraWebToken.sol#359

```
function transfer(address recipient, uint256 amount) public override
returns (bool) {
```

Lines: DecentraWebToken.sol#364

```
function allowance(address owner, address spender) public view override
returns (uint256) {
```

Lines: DecentraWebToken.sol#368

```
function approve(address spender, uint256 amount) public override returns
(bool) {
```

Lines: DecentraWebToken.sol#373

```
function transferFrom(address sender, address recipient, uint256 amount)
public override returns (bool) {
```

Lines: DecentraWebToken.sol#379

```
function increaseAllowance(address spender, uint256 addedValue) public
virtual returns (bool) {
```

Lines: DecentraWebToken.sol#384

```
function decreaseAllowance(address spender, uint256 subtractedValue)
public virtual returns (bool) {
```

Lines: DecentraWebToken.sol#389

```
function isExcludedFromReward(address account) public view returns (bool)
{
```

Lines: DecentraWebToken.sol#393

```
function totalFees() public view returns (uint256) {
```

Lines: DecentraWebToken.sol#397

```
function deliver(uint256 tAmount) public {
```

Lines: DecentraWebToken.sol#406

```
function reflectionFromToken(uint256 tAmount, bool deductTransferFee)
public view returns(uint256) {
```

Lines: DecentraWebToken.sol#482

```
function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
```

Lines: DecentraWebToken.sol#587

```
function isExcludedFromFee(address account) public view returns(bool) {
```

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 **4** low severity issues.

After the second review security engineers found that there was one new function added. Therefore found **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.