ETHANOL SMART CONTRACT REVIEW

DONE IN NOVEMBER 2020, BY WANDSE.IO

1. INTRODUCTION

This is a review of the Smart Contract `EthanolVault.sol` of Ethanol. The Smart Contract is aimed to be used in distributing cash back rewards on gas spent via the Ethanol Platform. Furthermore, there is a reward system integrated that allows token holders to lock their funds in exchange to a percentage reward.

Previously there have already been audits conducted on past iterations of this contract. **The Ethanol-Team is now confident to have fixed all the issues addressed**. Due to the shorter timeframe and close launch they requested a last qualitative review of the smart contract.

No quantitative tests have been conducted for this security review. As a result, no guaranteed claims can be made about its security. However, **the contract has been qualitatively reviewed for security vulnerabilities.** Below we share our findings.

The reviewed contract's source code evaluates to following sha-512 hash:
`1106CC48A0A5F0488ED3E7C31A748597FD909A7EF7E3D82FB5560BC43F266E0354998BC847B80AEEAD7709E027A969A23492CE2330A1EAB9D6DE0A6F4073BA17`

2. DISCLAIMER

This review makes no statements or warrants about utility of the code, safety of the code, suitability of the business model, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. This review documentation is for discussion purposes only.

3. SECURITY RISKS

The conducted review finds no breaking bugs or security vulnerabilities. The smart contract largely relies on secure OpenZeppelin source code.

The contract features two basic roles. Namely an `admin` and an `owner`. The `admin`-address is the deploying address of this contract and cannot be changed while the `owner`-address derives from the `Ownable.sol` OpenZeppelin standard and can thus be changed and renounced any time.

Only the `admin`-address can successfully call the `shareReward()`-function in order to add rewards to the internal balances of any address and increase the `totalSharedRewards`. While it being noteworthy, that if this address gets lost or an attacker gets access to its private keys this might result in a loss of funds, this is a valid design choice.

Only the `owner`-address can successfully call the `seedRewardPool()`-function transferring Ethanol-tokens to the contract and increasing the `rewardPool`-integer.

Neither the admin-address, nor the owner-address can directly withdraw any tokens. Furthermore, the presented contract does not feature upgradeability-logic or self-destruction, which means that **the owner or admin cannot easily withdraw all user funds with a single transaction call.** It is possible though for the admin-address to increase rewards and withdraw them. However, we do not classify this as a high-risk security vulnerability due to it being part of the platform's design.

4. ADDITIONAL ISSUES

One smaller issue in the Smart Contract is not security related.

- **Naming** some of the variable and parameter naming-scheme does not accurately reflect the underlying data or follows the Solidity programming standards. This does not have any effects on security vulnerability and does not require immediate attention.

5. SUMMARY OF THE AUDIT

This review has not found any high-risk security vulnerabilities in the presented contract. Additionally, there are no built-in possibilities for the owner or admin to withdraw user funds without their permission, in a single transaction.

We always recommend additional quantitative testing. Additionally, specifically for the Ethanol-token, we also recommend additional care when it comes to security and safeguarding of the `admin`- and `owner`-addresses.

WANDSE