CERTIK

Wemade Tree

Audit Report

Produced by CertiK, LLC

Executive Summary

We identified one Minor vulnerability and two info-level points that we think developers should be aware of.

Lead Auditor Dominik Teiml

Date June 24, 2020

Contents

Section 1

Introduction

We have been invited by Wemade Tree to audit their Wemix Token. The exact `git` commit of the audited version was `d8715ba21d169d167233d73167b889816f5100d5`. The file in question is located at `/contracts/WemixToken.sol` of the respective repository.

Since we have already audited this project in the past, the audit took 1 day. This report contains our findings during this audit.

The findings are labeled with:

- discsussion
- info
- minor
- major
- critical,

in increasing significance.

Section 2

Collated Findings

2.1 Payer risk MINOR

`stakeDelegated` allows a malicous attacker Alice to stake tokens for a whitelisted partner. The parner is unwhitelisted after the stake, and Alice can choose parameters of the stake, giving her some power to act conceptually on behalf of the partner.

For example, she can set the `_blockWaitingWithdrawal` parameter to the lowest allowed, and `withdraw` as soon as that is up. Even if the parner would want to stake for longer / withdraw later (to enjoy more minted tokens), they do not have that possibility.

Fixed (/ Client's response)

2.2 Owner risk INFO

The `owner` also has the power to dilute everyone's balances, by setting the `mintTo*` variables appropriately. It would be possible to bound the `mintToPartner` function from below and the other two from above.

Fixed (/ Client's response)

2.3 Possible front-running INFO

`removeAllowedPartner` can be front-run with `stake`. We don't see any fixes that would be simple enough to warrant a change.

Fixed (/ Client's response)

Section 3

Methodology

At CertiK we adopt formal reasoning for auditing smart contracts. The full process consists of *three* stages.

The *first* step of the process is to understand the full specification of the code. The *second* phase is the initial review of the code. At this stage, it is important to identify invariants that the system must satisfy. Often things that seem out of the ordinary are pointed out by our team at this stage.

The *third* stage is the actual audit. We check whether the code at hand satisifies all the necessary invariants identified in Step 2. In this way, the correctness of the code is established and any incosistencies found. In other words, the "negative" path of the program is checked: we assume a fully Byzantine environment and confirm the safeness and security of the system. In particular, we check whether a contrarian agent can combine functions and inputs in such a way to put the contract in an unexpected state, and check the full implications of each of these vulnerabilities.

We also employ state-of-the-art static analysis tools to find potential bugs. All results are manually verified, and if they constitute vulnerability, are included in the Report.

Finally, we will compile our findings and produce an Audit Report. The Client has the opportunity to correct the vulnerabilities listed. We will do another review after, and check whether the points have been addressed.

The result of this process is high guarantess about the security of the codebase, from both the side of CertiK, as well as from the Client.

Section 4

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Verification Services Agreement between CertiK and Wemade Tree (the "Company"), or the scope of services/verification, and terms and conditions provided to the Company in connection with the verification (collectively, the "Agreement"). This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without CertiK's prior written consent.

As there have been numerous interactions with the Company throughout the entire duration of this audit, and as the codebase target for the audit has evolved over said duration, not all of CertiK's opinions or comments have necessarily made it into this final culmination.