Decentraland MANA Token Audit

OPENZEPPELIN SECURITY | JULY 22, 2017

Security Audits

The Decentraland team asked us to review and audit their new MANA Token contract code. We looked at their contracts and now publish our results.

The audited contracts can be found in their mana repo. The version used for this report is commit `a13905356378cc0153dd3c2153c8ceae6400970d`.

Good work writing modular code and reusing existing contracts, as well as having ~ 96% automated test coverage.

Here's our assessment and recommendations, in order of importance.

EDIT: most of the issues were fixed in the latest version of the code.

Severe

Owner can finalize crowdsale before it has ended

`finalize` function from `FinalizableCrowdsale` is overriden by `MANACrowdsale` in line 115. The modified version removes the precondition requiring that the crowdsale `hasEnded`. This is not required by the smart contract specification, and makes the crowdsale vulnerable to premature closing by the owner.

It also introduces a race condition in which investors can still purchase tokens after the crowdsale is finalized but before the continuous sale is started, as the validation for processing a purchase in line 33 of `ContinuousCrowdsale` does not check the `isFinalized` flag. This causes an

Remove the overriding of `finalize` in `MANACrowdsale`, and also the overriding of `hasEnded` in line 111.

EDIT: fixed in commit `9f07555e51df352aff7bc58b6830904215a7dd14`.

Code for processPurchase is duplicated

The function `processPurchase` is defined in `ContinuousCrowdsale` and then redefined in `MANACrowdsale`. The latter consists of nearly the same code, except it calls `getRate` instead of reading the `rate` state variable. It's unclear whether such duplication was intentional or not. To avoid the issues duplication can cause, define the function in `ContinuousCrowdsale` in its most generic version calling `getRate`, with a simple implementation that returns the value of the `rate` state variable. Redefine `getRate` with the desired logic in `MANACrowdsale`.

EDIT: will be fixed along with issue #9.

Warnings

RateChange event and variable not related but have same name

In `MANACrowdsale` there is a `RateChange` event and a state variable. Even though they have the same name, they represent completely different things: the event means the base rate was chan `rateChange` ged, and the variable represents how much the rate decreases at each block. Consider renaming the state variable to `rateDecreaseStep`.

EDIT: fixed in commit `451ee4d49329b419ffc9d69fea17c1c56913e2a6`.

No on-chain validation that rate will not reach zero

The rate decreases at each block according to the `rateChange` parameter. The parameter could cause the rate calculation to fail if it accidentally reaches zero. Consider validating on construction that `rate &gt; rateChange * (endBlock - startBlock)`.

EDIT: fixed in commit `fd0c0a7574b44adafeffbe6033ef2bb6450921c2`.

Use safe math

~~doing a sum many times~~), or performing pre-condition checks on any math operation.

EDIT: fixed in commit `66cd12ef7d135a3cad17a35e48e974bb65a46cbf`.

Update to latest OpenZeppelin version

The project uses version 1.1.0 of OpenZeppelin. Version 1.2.0 was released a few days ago, which fixes ERC20 compliance. We recommend updating OpenZeppelin to 1.2.0, by changing the required version in *package.json*.

EDIT: fixed in commit `03ab3fe4567227569d778955f8e36b12b3d46898`.

Perform sanity checks on arguments

The following preconditions and sanity checks are missing.

1. `rateChange > 0` and `preferentialRate > 0` in `constructor` MANACrowdsale.
2. `rate > 0` in `setRate`.
3. `value > 0` in `burn`.
4. `investor != 0` in `addToWhitelist`.

EDIT: fixed in commits `fd0c0a7574b44adafeffbe6033ef2bb6450921c2`, `b78a711a0e4adf0bf68f972d4961cb90a1ec5ca0`, and `e34c411700fc0869d308249ff9aba08fe8a12eca`.

Burning tokens before crowdsale ends affects total token supply

MANA tokens can be burned invoking the `burn` function in `BurnableToken` line 17, which reduces the token's `totalSupply`. This reduces the number of tokens calculated as `finalSupply` in line 126 of `MANACrowdsale`, thus reducing the number of tokens emitted for the Decentraland Foundation, which are calculated as 60% of the `finalSupply` in line 129 of the same file. Consider preventing tokens from being burnt by adding a `burnable` flag in `BurnableToken`, disabled initially and only enabled once `MANACrowdsale` finalizes.

Whitelisted investors can make purchases with zero value

part of the `validPurchase` method. Consider adding the check by changing line 30 of `WhitelistedCrowdsale` to `super.validPurchase() || (!hasEnded() &amp;&amp; isWhitelisted(msg.sender) &amp;&amp; msg.value != 0)`.

Notes and Additional Information

- Good job using OpenZeppelin!
- Check `!hasEnded()` in line 67 of MANACrowdsale.sol is redundant.
- The name `checkContinuousPurchase` doesn't indicate clearly that the function changes contract state.
- In line 21 of BurnableToken.sol, the variable `burner` could be used instead of `msg.sender`.
- The public functions and events in `MANACrowdsale` are missing documentation. Even though the names are mostly self-explanatory, it's a good idea to clearly document the public API.
- If the `getRate` function in `MANACrowdsale` is not moved up to `ContinuousCrowdsale` as suggested previously, consider marking it as `constant` to ensure it has no side effects.
- Test coverage is very good (96.92% as measured with solidity-coverage). There are three failing tests in `MANACrowdsale` (reported in issue #4), which exercise the lines missing from the coverage.

Conclusions

Two severe security issues were found. Some small changes were proposed to follow best practices and reduce potential attack surface.

Good work writing modular code and reusing existing contracts, as well as having ~96% automated test coverage.

Note that as of the date of publishing, the above review reflects the current understanding of known security patterns as they relate to the MANA Token contract. We have not reviewed the related Decentraland project. The above should not be construed as investment advice or an

Related Posts

Beefy Zap Audit

BeefyZapRouter serves as a versatile intermediary designed to execute users' orders through routes...

Security Audits

OpenBrush Contracts Library Security Review

OpenBrush is an open-source smart contract library written in the Rust programming language and the...

Security Audits

Linea Bridge Audit

Linea is a ZK-rollup deployed on top of Ethereum. It is designed to be EVM-compatible and aims to...

Security Audits

Threat Monitoring
Incident Response
Operation and Automation

Zero Knowledge Proof Practice

Blog

Company

About us
Jobs
Blog

Contracts Library

Docs