SOLIDIFIED

Audit Report for Opiria. April 13, 2018.

Summary

Audit Report prepared by Solidified for Opiria covering the token and crowdsale contracts.

Process and Delivery

Three (3) independent Solidified experts performed an unbiased and isolated audit of the below token sale. The debrief took place on April 13, 2018 and the final results are presented here.

Audited Files

The following files were covered during the audit:

- OpiriaToken.sol
- OpiriaCrowdsale.sol
- TimedPresaleCrowdsale.sol
- TokenCappedCrowdsale.sol

Notes

The audit was conducted on commit `1f032fe733ba9f43028cd9fc932819239bfc1218`
The audit was based on the solidity compiler `0.4.21+commit.c4cbbb05`

Intended Behavior

The purpose of these contracts is to create the Opiria token and distribute it to the public, in a crowdsale process.

Issues Found

1. Owner can deny distributing bonus tokens

The process of distributing bonus tokens is not trustless and relies on the owner calling the `distributeBonus` function, thus potentially having a chance of never occurring.

Recommendation
Consider adding the a withdraw pattern, so users can withdraw their own share of bonuses tokens.

AMENDED [17.04.2018]
Issue was fixed by Opiria in commit `ab18d6b5bae6e76b4e98cceee6f7d39eed8c8de8`

2. Owner can issue/mint tokens indefinitely

The function `sendTokensTo` allows for the owner to distribute unaccounted tokens at will. This happens, because the tokens distributed via the function `sendTokensTo` are not counted in the `tokensSold` variable, distorting the real amount sold. It can also be called after the sale has ended if the owner does not trigger the finalize function.

This also implies that the hard cap defined can be bypassed by calls to this function.

Recommendation
It is recommended to remove the method `sendTokensTo`. If the requirement still needs such method, include all the parameters that apply to regular token purchase.

AMENDED [17.04.2018]
Issue was fixed by Opiria in commit `ab18d6b5bae6e76b4e98cceee6f7d39eed8c8de8`

3. Ownership of tokens is locked in the crowdsale contract

In the deployment process, the token ownership is transferred to the crowdsale contract, so minting tokens is possible, but is never transferred back. That way, some of the token functions become unreachable, such as `finishMinting`.

Recommendation
After all sale steps have been made, transfer the ownership back to the original owner.

AMENDED [17.04.2018]
Issue was fixed by Opiria in commit `ab18d6b5bae6e76b4e98cceee6f7d39eed8c8de8`

Note: FinishMinting is triggered automatically, but the ownership still remains in the crowdsale contract. There is no security issue if Opiria team understands the implications of an ownerless token.

4. USD rate can be changed any time

This ability can greatly affect buyers, since they can't know for sure at what price they are buying. Owner can alter the prices right before any given transaction to manipulate purchases.

Recommendation
Consider implementing a fixed rate or utilizing a trustless oracle.

5. Reserved tokens unlocked early

As per the requirement document, reserved tokens are unlocked for distribution after 6, 12 and 24 months respectively. But as per the contract both 2nd and 3rd stage can be unlocked after 12 months. This violates the token unlock policy given in the requirement document.

Recommendation
It is recommended to change the token lock duration based on the spec.

AMENDED [17.04.2018]
Issue was fixed by Opiria in commit `ab18d6b5bae6e76b4e98cceee6f7d39eed8c8de8`

6. Bonus distribution can start early

As per the requirement document, bonus distribution should begin 30 days after unlocking the token transfers. Current implementation counts from token close time, which may not always be 30 days from token transfer.

Recommendation
It is recommended to calculate the bonus distribution lock time from transfer unlock time.

AMENDED [17.04.2018]
Issue was fixed by Opiria in commit `ab18d6b5bae6e76b4e98cceee6f7d39eed8c8de8`

7. Deviations from White paper

We noticed some deviations from the expected behavior according to the Opiria white paper, as follows:
- Tokens of "Increased Cap" (50mm), if not sold, are sent to the team wallet, effectively distorting the specified distribution percentages.

- Presales are limited to purchases over 5000 dollars, while the spec states that only purchases over 5000 dollars will receive the bonus, but mentions no limit. The limit is mentioned in the readme.md file from the contracts.

Recommendation
It is recommended to either update the whitepaper or conform the contracts to the defined spec

8. The hidden cap and state changes

The white paper states that there is a hidden presale cap, which if achieved will trigger the sale in 24 hours. There is no logic in the contract to deal with this event. The only function available is the `changeTimes` function, although it will require changing all four dates to dates in the future where (`presaleStart` < `presaleEnd` < `saleStart` < `saleEnd`). The `changeTimes` function also allows reopening the sale after it has been closed and finalized (see issue 9).

Recommendation
Although we understand the need not to publish the hidden cap from the start, a change of state after an external trigger (like the one used in `IncreaseCap`) would be a better solution, avoiding the need to change all dates. The trigger should only be available until the 5th day of presale, as stated in the white paper and it should be designed in a way not to allow parameters other than the ones defined in the white paper.

AMENDED [17.04.2018]
Issue was fixed by Opiria in commit `ab18d6b5bae6e76b4e98cceee6f7d39eed8c8de8`

9. The changeTimes function allows reopening the sale after it has been closed and finalized

The `changeTimes` function allows the owner of the contract to reopen the sale after it has been closed and finalized. If used when the contract if finalized, the state of the contract becomes incoherent, with `finalized = true` and `isPresale()` or `isSale()` returning true, affecting some of the modifiers used.

Recommendation
We recommend checking the state before allowing the change of dates. Dates in the past should not be allowed to change.

AMENDED [17.04.2018]
Issue was fixed by Opiria in commit `ab18d6b5bae6e76b4e98cceee6f7d39eed8c8de8`

10. Compiler Version

OpiriaCrowdsale, TimedPresaleCrowdsale and OpiriaToken have pragma version out of date `0.4.0`. TokenCappedCrowdsale is version `0.4.18`, while the current stable version is `0.4.21`. All of them are not locked to a compiler version. We recommend that the last stable version of the solidity compiler is used. It is recommended to lock the pragma version, as future compiler versions may handle certain language constructions in a way the developer did not foresee.

AMENDED [17.04.2018]
Issue was fixed by Opiria in commit `ab18d6b5bae6e76b4e98cceee6f7d39eed8c8de8`

11. Private presale results are not considered by contract

The white paper mentions that there will be a round of private presale, but the tokens sold during that period are not included in the distribution considered and are not accounted for when calculating the cap.

12. Consider triggering the finalization logic automatically

As it currently is, the owner of the contract needs to call the finalize function, but the finalization conditions are well defined in the contract and could be called automatically.

13. Low risk recommendations

1. OpiriaToken.sol
  - remove comment `///TODO: restrict tokens sent to the address of the token`

2. TimedPresaleCrowdsale
  - Constructor should have visibility specified (public)
  - No input validation in constructor
  - Importing Ownable not necessary

3. OpiriaCrowdsale

- remove comment `/// TODO: team tokens claimance`
- `distributeBonus` has no visibility specified
- unused variable `presaleBonusPercent`

4.TokenCappedCrowdsale
- Consider using `view` instead of `constant` (constant has been deprecated)

5.General
- Using events without `emit` is deprecated

AMENDED [17.04.2018]
Issue was fixed by Opiria in commit `ab18d6b5bae6e76b4e98cceee6f7d39eed8c8de8`

Closing Summary

Several major and minor issues were found during the audit which can break the desired behaviour. It is strongly advised that these issues are corrected before proceeding with the crowdsale. It is furthermore recommended to post the contracts on Solidified public bounty afterwards.

OpiriaToken.sol has been verified as fully ERC20 compliant.

Beyond the issues mentioned, the contracts were also checked for overflow/underflow issues, DoS, and re-entrancy vulnerabilities. None were discovered.

OpenZeppelin contracts such as Ownable/SafeMath/Crowdsale/etc. have been widely audited and secured, as such, they were not prioritized for auditing.

Disclaimer

Solidified audit is not a security warranty, investment advice, or an endorsement of the Opiria platform or its products. This audit does not provide a security or correctness guarantee of the audited smart contracts. Securing smart contracts is a multistep

process, therefore running a bug bounty program as a complement to this audit is strongly recommended.

The individual audit reports are anonymized and combined during a debrief process, in order to provide an unbiased delivery and protect the auditors of Solidified platform from legal and financial liability.

Solidified Technologies Inc.