SOLIDIFIED

Audit Responses on New Alchemy Report. June 2018.

Summary

Solidified Ltd. engaged New Alchemy for an external security audit of our Solidified Token Sale contracts which was delivered June 15th, 2018. The following are our responses to each issue they reported.

Issues Found

Moderate

1. Sale Stage May Stall

updateStage was intended to mitigate this scenario, but we agree that the workaround was not clean (and had another issue, see issue #1 of the Solidified internal audit) and a fix was implemented. This issue is no longer present in commit `9ca5e0bb28efba1693ce3d506c0df46866f11a25`

2. Inconsistent/Confusing Critical Constant Declaration and Behavior

This issue is no longer present in commit `9ca5e0bb28efba1693ce3d506c0df46866f11a25`

Minor

3. Two Functions Missing onlyOwner

Generally our philosophy is that if there are no security/functionality concerns with allowing a function to be called by anyone, then that function should not have the onlyOwner modifier. We disagree with the assertion “changing the contract internal state is usually only done by the contract owner”, as this is not true for the vast majority of contracts (purchases in our token sale, for example, necessarily update contract state).

4. Unnecessary Loss of Precision

While generally true, as stated in the report it does not apply to the particular instance cited; nevertheless the case in question was removed as a part of the fix to issue #2.

This issue is no longer present in commit `9ca5e0bb28efba1693ce3d506c0df46866f11a25`

5. Pause Functionality Incorporated but not Used

This issue is no longer present in commit `9ca5e0bb28efba1693ce3d506c0df46866f11a25`

6. Lack of Short-Address Attack Protections

As stated in the report, this is a contentious matter. We, like OpenZeppelin, are of the opinion that this mitigation does not belong at the smart contract layer.

7. Lack of two-phase ownership transfer

This is a good suggestion. However, given that transfership of ownership is very unlikely in our particular case, implementing it would require more modifications to OpenZeppelin code than we feel comfortable with in the time we have available.

8. ERC20 double-spend attack

SolidToken already extends OpenZeppelin's StandardToken, which as stated in the report contains an ERC20 compatible mitigation for this issue.

Closing Summary

All issues reported by New Alchemy have been addressed.