SOLIDIFIED

Audit Report for BullToken. March 14, 2018.

Summary

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

Process and Delivery

Two (2) independent Solidified experts performed an unbiased and isolated audit of the below token sale. The debrief took place on March 10, 2018 and the final results are presented here. The review of the fixes and amends took place on March 14, 2018.

Audited Files

The following files were covered during the audit:

- BullToken.sol
- BullTokenMainSale.sol
- Curatable.sol
- Whitelist.sol
- BurnableCrowdsale.sol
- CappedCrowdsale.sol
- FinalizableCrowdsale.sol

Notes

The audit was conducted on commit `f0aefd9a74444d40cd0a562ec45fc4d8557000a1`
The audit was based on the solidity compiler `0.4.20+commit.3155dd80`

Intended Behavior

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

Issues Found

1. Reaching the soft cap will stop the sale prematurely

If the sale reaches the goal before the `initialEndTime`, no further purchase is possible until the extra sale period starts. This happens because the function `hasEnded` will return true when

both the `goalReach` and `initialEndTimeNotReached` returns true, making any further transaction fail.

Recommendation

Make a function to check whether the `initialEndTime` was reached, return false if not.

AMENDED [2018-03-14]
This issue has been fixed on commit `82b4a3de2fdc212d4673f002d185475e4f8cf31a`

2. Token burning mechanism is not present

As per the document shared, unsold tokens should be burned as soon as the token sale is finished. The current implementation has no such method to burn the remaining tokens during the finalization process and therefore, there's no guarantee that such action will be taken.

Recommendation

Consider implementing this in the finalization function rather than burning it manually.

AMENDED [2018-03-14]
BullToken team states that there's no way to insert this mechanism, since part of the contracts are already deployed.

3. There is no reliable way to stop the main sale

The intended behaviour states that there must be a way to stop the main sale if a bug is found, but such mechanism is not present in the contracts. The only way to achieve that is for the owner to burn all of his tokens, but that might not be possible (if owner is another contract, for example).

Recommendation

Implement a direct trigger that blocks purchases when activated.

AMENDED [2018-03-14]
BullToken team states that they can pause the token contract and, as a side effect, pause the sale, which is true. They also stated that they are aware of the implications of utilizing the token contract and decided to proceed. This decision does not affect the security of the contracts.

4. Owner can still transfer tokens during the sale

During the sale, or before, the owner is still allowed to transfer tokens as he wishes. That might be contrary to the statement in the intended behaviour that says *"No tokens should be handed out “for free” to anyone"*. Also, the sale contract relies on the owner’s balance for some calculation and therefore, transferring tokens outside can alter some sale parameters.

Recommendation

Consider reviewing the power of owner transferring tokens at any time.

AMENDED [2018-03-14]
BullToken team is aware of the implications of transferring tokens during the sale and assures that they won't take any actions that can possibly have an impact. This decision does not affect the security of the contracts.

5. Consider changing the refund calculation

For different combination of rate, hardcap and other parameters, value of `amountToBeRefunded` might get overwritten due to multiple assignments. This will not occur with the current values given in the document, but may occur with other values.

Recommendation

Consider rethinking the calculation method for refund to resolve this issue.

AMENDED [2018-03-14]
BullToken team is aware of this possibility and assures that they will use the parameters provided in the intended behaviour, which won't cause this issue to happen.

6. BullToken keeps information of non-holders

After transferring all the tokens, a user might still be present in the holders array and in `isHolder` mapping, in the BullToken contract. Therefore, those arrays wouldn't really reflect the true holders.

Recommendation

Review the need to keep this list on the contract, as its rather expensive and don't necessarily reflects the reality. If this information is needed, consider implementing an off-chain solution.

7. Consider removing duplicate validations

Some validations such as checking for the cap during the token buying process are performed multiple times. It is not a bad practice to have multiple checks, but avoiding it can save a bit of gas amount during the transaction.

8. Consider removing empty functions

In the contracts there are a few empty functions, which serve no purpose at all, so consider removing them. An example is the whitelisted constructor and the finalization function, but there are others.

9. Consider following the solidity style guide

Consider following the Solidity guidelines on formatting the code and commenting. It can improve the overall code quality and readability.

10. Update Compiler Version

The compiler version is outdated. It's recommended to use the latest compiler version (0.4.20)

Closing Summary

Some major and minor issues were found during the audit which can break the desired behaviour. It is strongly advised that these issues are correct before proceeding with the

crowdsale. It is furthermore recommended to post the contracts on public bounty afterwards.

Disclaimer

Solidified audit is not a security warranty, investment advice, or an endorsement of the BullToken platform. 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.