SOLIDIFIED

Audit Report for MyBit on December 7th, 2018.

Summary

Audit Report prepared by Solidified for MyBit covering the MyBit token-sale smart contract.

Process and Delivery

Three (3) independent Solidified experts performed an unbiased and isolated audit of the code below. The debrief took place on December 7th, 2018, and the final results are presented here.

Audited Files

The following contracts were covered during the audit:

- TokenSale.sol

Notes

The audit was based on the repository `https://github.com/MyBitFoundation/MyBit-Tokensale.tech` commit hash `a5b8ea3e2bcb724be7bd1ebc20af43a4e412e013`, solidity compiler `0.4.25+commit.59dbf8f1`

Intended Behavior

The token sale contract is an ERC20 token sale in which a specific number of tokens are issued daily and are distributed based upon percentage of overall contribution once the sale closes. The following hold true during the sale:

1. Begin on Jan. 1, 2019 at 12:00 UTC and have 365, 24 hour periods.
2. Each period 100,000 MYB are distributed (total 36.5 million)
3. MYB is distributed proportionate to amount of ETH contributed
4. Users can withdraw MYB once a period finishes (either individually or batch withdraw)
5. Users can contribute to future periods

Issues Found

1. getTokensOwed may return an incorrect result if the given day has not finished yet (minor)

Since other investors are still free to contribute to the sale on any given unfinished day, `getTokensOwed()` might return a value to the user that does not correctly represent the amount of tokens they will be able to claim.

Recommendation
Require that the day passed to `getTokensOwed()` be a finished day in order to be able to return the true amount of owed tokens that an investor can withdraw.

Amended [13-12-2018]
The issue was fixed and is no longer present in commit `311951e51b5683d7feba402bc4b7f8edc89764df`.

2. Arbitrary constants limiting array length (note)

Relevant functions (`batchWithdraw()`, `batchFund()`, and `getTotalTokensOwned()`) include a `require` statement limiting array lengths to 50 and 100, respectively.

After calculating gas costs on these functions it appears the limitations are in place to prevent exceeding the gas block limit.

Recommendation
Add notation describing this limit.

Amended [13-12-2018]
The issue was fixed and is no longer present in commit `311951e51b5683d7feba402bc4b7f8edc89764df`.

3. Constant utilized deviates from specification (note)

The contract’s implementation of the `fund` function states that “ // @dev only accepts contributions between days 0 - 365”. This deviates from the intended specification which would allow for contributions between days 0 and 364, totalling one year’s worth of contributions, not a year and an additional day.

Recommendation

Change 365 to 364 within the comment block.

Amended [13-12-2018]

The issue was fixed and is no longer present in commit `311951e51b5683d7feba402bc4b7f8edc89764df`.

4. dayFor() returns all past values as day 0 (note)

Any timestamp passed into `dayFor()` that is prior to the `start` timestamp will be returned as 0, due to `dayFor()` only checking `if (_timestamp < start)`.
As 0 is a valid day during the sale this should be separated into its own check.

Recommendation

Make `dayFor()` an internal function to prevent accidental or incorrect past time values from being passed in. Additionally, one can add a separate check that will revert with past arguments, such as `require(_timestamp > start, 'Day has already passed.')`

Amended [13-12-2018]

The issue was fixed and is no longer present in commit `311951e51b5683d7feba402bc4b7f8edc89764df`.

5. TEST contract can be removed before publication (note)

The contract possesses an additional “TEST” contract which inherits from `TokenSale` that is included in a comment block.

Recommendation

As it is not utilized, this code can be removed.

Amended [13-12-2018]
The issue was fixed and is no longer present in commit `311951e51b5683d7feba402bc4b7f8edc89764df`.

6. getTotalTokensOwed does not check for duplicate days (note)

An array of `_days` can be passed into `getTotalTokensOwed()` that has duplicated day values (0-364). These will be processed as individual days through the for loop and continue adding their value to the `amount` variable that is returned.

Although `batchWithdraw()` and `getTotalTokensOwed()` both call `getTokensOwed()` to check the token amount owed to an individual contributor, `batchWithdraw()` deletes the contributions entry for a given day before moving on to the next in the loop. Because of this, it will not utilize the incorrect `amount` variable from `getTotalTokensOwed()`.

Closing Summary

MyBit’s contracts contain only one minor issue, along with several areas of note. All issues with the exception of the note #6 (this function is us) were amended.

We recommend the minor issue is amended, while the notes are up to MyBit's discretion, as they mainly refer to improving the operation of the smart contract.

Disclaimer

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